Editor **DANA SIMIAN**

MODELLING AND DEVELOPMENT OF INTELLIGENT SYSTEMS

Proceedings of the First International Conference on
MODELLING AND DEVELOPMENT OF INTELLIGENT SYSTEMS

SIBIU, ROMANIA, October 22-25, 2009

Lucian Blaga University Press, 2009

Editor DANA SIMIAN

MODELLING AND DEVELOPMENT OF INTELLIGENT SYSTEMS

Proceeding of the First International Conference
"Modelling and Development of Intelligent Systems"
October 22-25, 2009, Sibiu, ROMANIA

Lucian Blaga University Press, Sibiu

2009

Lucian Blaga University Press, Sibiu, 2009

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN 2067-3965

Assistent editor Laura Cacovean
Cover design Ralf Fabian

Proceedings of the First International Conference
Modelling and Development of Intelligent Systems
October 22-25, 2009, Sibiu, ROMANIA

Preface

This volume contains refereed papers which were presented at the First International Conference *Modelling and Development of Intelligent Systems*. The conference was held between October 22-25, 2009, at Sibiu, Romania. It was organized by the Department of Informatics at the Faculty of Sciences, from " Lucian Blaga" University of Sibiu.

The aim of the conference was to bring together scientists, researchers, students, interested and working in fields which can be connected with modeling and development of intelligent systems. Specific topics of interest included but were not restricted to: evolutionary algorithms, evolutionary computing, genetic algorithms and their applications, grid computing and clustering, data mining, ontology engineering, intelligent systems for decision support, knowledge based systems, pattern recognition and model checking, motion recognition, e-learning, hybrid computation for artificial vision, knowledge reasoning for artificial vision, geometric modelling and spatial reasoning, modelling and optimization of dynamic systems, large scale optimization techniques, adaptive systems, multiagent systems, swarm intelligence, metaheuristics and applications, machine learning, self learning algorithms, mathematical models for development of intelligent systems. The talks were delivered by universities' members, researchers and students from 12 countries. During the conference a wide range of theoretical and practical problems related to the conference topics were discussed. Four plenary lectures were presented:

- Nikos E. Mastorakis - *Genetic Algorithms for the Solution of Partial Differential Equations*
- Ioana Moisil - *The Digital Culture Phenomenon. Can we model it?*
- Dana Simian - *Models for Intelligent Agents Based on Swarm intelligence*
- Milan Tuba - *Guided Maximum Entropy Method*

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference and to the quality of this proceedings volume.

December 2009

Dana Simian
Conference chairman

Scientific Committee

Octavian Agratini, Babes-Bolyai University of Cluj Napoca, Romania

Florian Boian, Babes-Bolyai University of Cluj Napoca, Romania

Ioana Chiorean, Univ. Babes Bolyai, Cluj-Napoca, Romania

Camelia Chira, Babes-Bolyai University of Cluj Napoca, Romania

Ioan Dzitac, Aurel Vlaicu University of Arad, Romania

Dan Dumitrescu, Babes-Bolyai University of Cluj Napoca, Romania

Stefka Fidanova, Institute for Parallel Processing, Bulgarian Academy of Sciences, Bulgaria

Vasile Georgescu, University of Craiova, Romania

Dejan Gjorgjevik, Faculty of Electrical Engineering and Information Technologies in Skopje, Macedonia

Heiner Gonska, Duissburg – Essen University, Germany

Paul Corneliu Iacob, "Transilvania" University, Brasov, Romania

Julian Ławrynowicz, University of Lodz, Polish Academy of Sciences, Poland

Suzana Loskovska, Faculty of Electrical Engineering and Information Technologies in Skopje, Macedonia

Daniela Marinescu, Transilvania University of Brasov, Romania

Nikos Mastorakis, Hellenic Naval Academy, Greece

Olga Miljkovic, Megatrend University of Belgrade, Serbia

Paulina Mitrea, Technical University, Cluj Napoca, Romania

Ioana Moisil, Lucian Blaga University of Sibiu, Romania

Cornelia Novac, "Dunarea de Jos" University,Galati, Romania

Ioana Cristina Plajer, "Transilvania" University, Brasov, Romania

Ioan Pop, "Lucian Blaga" University, Sibiu, Romania

Livia Sangeorzan, Transilvania University, Braşov, Romania

Dana Simian, Lucian Blaga University of Sibiu, Romania

Florin Stoica, "Lucian Blaga" University, Sibiu, Romania

Arpad Takaci, University of Novi Sad, Serbia

Milan Tuba, Megatrend University of Belgrade, Serbia

Cornelia Tudorie, "Dunarea de Jos" University,Galati, Romania

Dan Eugen Ulmet, University of Applied Sciences Esslingen, Germania

Anca Vasilescu, Transilvania University, Braşov, Romania

Mirela Cristina Vinerean Bernhoff, Karlstad University, Karlstad, Sweden

Lubin Vulkov, University “Angel Kunchev” of Rousse, Bulgaria

Table of Contents

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

On some linear positive operators: statistical approximation and q-generalizations

Octavian Agratini, Saddika Tarabie

Abstract

This paper is focused on sequences of linear positive operators, the starting point being represented by Popoviciu-Bohman-Korovkin criterion. Our first aim is to sum up recent investigation on statistical convergence of this type of approximation processes. The second aim is to construct a bivariate extension of Stancu discrete operators. This generalization is based on q-integers and on the tensor product method.

1 Introduction

Let $(L_n)_{n\geq 1}$ be a sequence of linear positive operators acting on the space $C([a,b])$ of all real-valued and continuous functions defined on the interval $[a,b]$, equipped with the norm $\|\cdot\|$ of the uniform convergence. Popoviciu-Bohman-Korovkin's theorem asserts: if the operators L_n, $n\in\mathbb{N}$, map $C([a,b])$ into itself such that $\lim_n \|L_n e_j - e_j\| = 0$ for $j = 0,1,2$, then one has $\lim_n \|L_n f - f\| = 0$ for every $f \in C([a,b])$. Here e_j represents the monomial of j-th degree, $j \in \{0,1,2\}$. A special extension of this criterion consists in replacing the uniform convergence by statistical convergence. This approach models and improves the technique of signals approximation in different function spaces. This is useful both in various areas of functional analysis and in obtaining numerical solutions of some differential and integral equations. Following this direction, recent results regarding statistical convergence criterions are collected in Section 2. Further on, we bring into light a bi-dimensional linear positive operator of Bernstein-Stancu type based on Quantum Calculus. The construction technique takes one's stand on the tensor product.

2 Statistical convergence theorems of Korovkin type

First of all, we briefly recall some basic facts with regard to the notion of statistical convergence. This concept, originally appeared in Steinhaus [22] and Fast [8] papers, is based on the notion of the density of subsets of $\mathbb{N} = \{1,2,3,\dots\}$ and it can be viewed as a regular method of summability of sequences.

The density of a set $K \subset \mathbb{N}$ is defined by

$$\delta(K) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \chi_K(k),$$

provided the limit exists, where χ_K is the characteristic function of K. Clearly, the sum of the right hand side represents the cardinality of the set $\{k \leq n :\ k \in K\}$. A sequence $x = (x_k)_{k\geq 1}$ is statistically convergent to a real number L, denoted $st - \lim_k x_k = L$, if, for every $\varepsilon > 0$,

$$\delta(\{k \in \mathbb{N} :\ |x_k - L| \geq \varepsilon\}) = 0$$

holds. Closely related to this notion is A-statistical convergence, where $A = (a_{n,k})$ is an infinite summability matrix. For a given sequence $x = (x_k)_{k\geq 1}$ the A-transform of x, denoted by $Ax = ((Ax)_n)$, is defined as follows

$$(Ax)_n = \sum_{k=1}^{\infty} a_{n,k}x_k, \quad n \in \mathbb{N},$$

provided the series converges for each n.

Suppose that A is non-negative regular summability matrix. Recall: A-summation method is regular if, whenever the sequence x converges to L, $\lim_n (Ax)_n = L$. The sequence x is A-statistically convergent to the real number L if, for every $\varepsilon > 0$, one has

$$\lim_{n\to\infty} \sum_{k\in I(\varepsilon)} a_{n,k} = 0,$$

where $I(\varepsilon) = \{k \in \mathbb{N} : \ |x_k - L| \geq \varepsilon\}$. We denote this limit by $st_A - \lim_k x_k = L$.

Taking in view both the definition of A-statistical convergence and the three regularity conditions claimed by Silverman-Toeplitz theorem, we notice that every convergent sequence is A-statistically convergent.

In the particular case $A = C_1$, the Cesàro matrix of first order, A-statistical convergence reduces to statistical convergence, see, e.g., [8], [9]. Also, if A is the identity matrix, then A-statistical convergence coincides with the ordinary convergence.

In Approximation Theory by linear positive operators, the statistical convergence has been examined for the first time by A.D. Gadjiev and C. Orhan. Popoviciu-Bohman-Korovkin criterion via statistical convergence will be read as follows [12; Theorem 1].

Theorem 1 *If the sequence of positive linear operators $L_n : C([a,b]) \to B([a,b])$ satisfies the conditions*

$$st - \lim_n \|L_n e_j - e_j\| = 0, \quad j \in \{0,1,2\}, \tag{1}$$

then, for any function $f \in C([a,b])$, we have

$$st - \lim_n \|L_n f - f\| = 0.$$

As usual, $B([a,b])$ stands for the space of all real valued bounded functions defined on $[a,b]$, endowed with the sup-norm.

Remark 1 *Examining the proof of Theorem 1 given by the authors, we notice that this statement is also true for A-statistical convergence, where A is a non-negative regular summability matrix.*

The quoted paper was to mean the beginning of an intensive study developed by many researches for obtaining criterions in order to decide the statistical convergence of a given sequence of linear positive operators in various function spaces.

In what follows, we exhibit noteworthy results which certify the achievements in this research field.

Using the concept of the rate of statistical convergence, a general class of linear positive operators defined on the space $C([0,b])$, $0 < b < 1$, has been investigated by Doğru, Duman and Özarslan [18].

Further on, set $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$ and let $e_{ij}(x,y) = x^i y^j$, $i \in \mathbb{N}_0$, $j \in \mathbb{N}_0$, $i + j \leq 2$, stand for the test functions corresponding to the bi-dimensional case of the classical criterion. The following A-statistical analogue of Korovkin-type approximation theorem in bi-dimensional case has been obtained by Erkuş and Duman [7; Theorem 2.1].

Theorem 2 *Let A be a non-negative regular summability matrix and let $(L_n)_n$ be a sequence of positive linear operators from $C(I \times J)$ into $C(I \times J)$, where $I = [a,b]$ and $J = [c,d]$. Then, for all $f \in C(I \times J)$,*

$$st_A - \lim_n \|L_n f - f\| = 0,$$

if and only if the following identities hold

$$\begin{cases} st_A - \lim\limits_n \|L_n e_{i,j} - e_{i,j}\| = 0, \ (i,j) \in \{(0,0),(0,1),(1,0)\}, \\ st_A - \lim\limits_n \|L_n(e_{2,0}+e_{0,2}) - (e_{2,0}+e_{0,2})\| = 0. \end{cases}$$

In the above, $\|\cdot\|$ indicates the sup-norm of the space $C(I \times J)$.

Remark 2 *In the Banach space $C(I \times J)$, if A-statistical convergence is replaced by the uniform convergence, then Theorem 2 turns into a result due to Volkov [23].*

Now, we make a halt in weighted spaces. A function $\rho \in \mathbb{R}^{\mathbb{R}}$ is usually called a *weight function* if it is continuous on the domain satisfying the conditions $\rho \geq e_0$ and $\lim\limits_{|x|\to\infty} \rho(x) = \infty$. Let consider the spaces

$$B_\rho(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} \mid \text{ a constant } M_f \text{ depending on } f \text{ exists such that } |f| \leq M_f \rho\},$$
$$C_\rho(\mathbb{R}) = \{f \in B_\rho(\mathbb{R}) \mid f \text{ continuous on } \mathbb{R}\},$$

endowed with the common norm $\|\cdot\|_\rho$, where $\|f\|_\rho = \sup\limits_{x\in\mathbb{R}} \dfrac{|f(x)|}{\rho(x)}$. Duman and Orhan [6; Theorem 3] proved the following weighted Korovkin type theorem via A-statistical convergence.

Theorem 3 *Let A be a non-negative regular summability matrix and let ρ_1, ρ_2 be weight functions such that*

$$\lim_{|x|\to\infty} \frac{\rho_1(x)}{\rho_2(x)} = 0.$$

Assume that $(L_n)_{n\geq 1}$ is a sequence of positive linear operators acting from $C_{\rho_1}(\mathbb{R})$ into $B_{\rho_2}(\mathbb{R})$. Then

$$st_A - \lim_n \|L_n f - f\|_{\rho_2} = 0 \text{ for all } f \in C_{\rho_1}(\mathbb{R}),$$

if and only if

$$st_A - \lim_n \|L_n F_j - F_j\|_{\rho_1} = 0, \quad j \in \{0,1,2\},$$

where $F_j(x) = x^j(1+x^2)^{-1}\rho_1(x)$, $j = 0,1,2$.

Remark 3 *As the authors of the above theorem specify, substituting $st_A - \lim\limits_n$ by ordinary limit one reobtains a result established by Gadjiev [11].*

Aiming at the same approach, we present an abstract version of the Korovkin type approximation criterion. Let (X,d) be an arbitrary compact metric space. We denote by $C(X)$ the space of all real-valued functions continuous on X. In the sequel $\mathbf{1}$ denotes the constant function 1. On the basis of [1; Theorems 1, 2] we state.

Theorem 4 *Let L_n, $n \in \mathbb{N}$, be positive linear operators acting on $C(X)$, where (X,d) is a compact metric space. Let α_n, β_n, $n \in \mathbb{N}$, be defined as follows*

$$\alpha_n(x) = L_n(d(\cdot,x),x), \quad \beta_n(x) = L_n(d^2(\cdot,x),x), \quad x \in X.$$

a) If

$$st - \lim_n \|L_n(\mathbf{1}) - \mathbf{1}\| = 0 \quad \text{and} \quad st - \lim_n \|\alpha_n\| = 0, \tag{2}$$

then the following identity holds

$$st - \lim_n \|L_n f - f\| = 0 \text{ for all } f \in C(X). \tag{3}$$

b) If $L_n(\mathbf{1}) = \mathbf{1}$ and $st - \lim\limits_n \|\beta_n\| = 0$, then (3) also holds.

We point out the following particular case. Let X be a real vector space endowed with an inner product $(\cdot,\cdot)$ and let Y be an arbitrary compact subset of X. We define the functions e, e_x, a_n, b_n ($n \in \mathbb{N}$) belonging to $\mathbb{R}^Y$ as follows

$$\begin{cases} e(x) = (x,x), & a_n(x) = (L_n e)(x) - e(x), \quad x \in Y, \\ e_x(t) = (x,t), & b_n(x) = (L_n e_x)(x) - e(x), \quad t \in Y, x \in Y. \end{cases}$$

Since $\beta_n(x) = a_n(x) - 2b_n(x) + e(x)((L_n\mathbf{1})(x) - 1)$, assuming

$$L_n(\mathbf{1}) = 1 \quad \text{and} \quad st - \lim_n \|a_n\| = st - \lim_n \|b_n\| = 0,$$

we deduce that identity (3) holds for every $f \in \mathbb{R}^Y$.

Remark 4 *Let $X = \prod_{i=1}^{p}[a_i, b_i] \subset \mathbb{R}^p$ be the p-dimensional parallelepiped. It's known that $\left\{\mathbf{1}, pr_1, \ldots, pr_p, \sum_{j=1}^{p} pr_j^2\right\}$ represents a Korovkin subset in $C(X)$, where pr_j, $1 \le j \le p$, are the canonical projections on X. Based on a classical result, see, e.g., the monograph [2, p. 245] for deciding if a sequence of linear positive operators $(L_n)_n$, $L_n : C(X) \to C(X)$, $n \in \mathbb{N}$, forms a classical approximation process, we need to evaluate L_n on these $p+2$ test-functions. For establishing the statistical convergence, we need to prove nothing else but identities (2).*

3 Modified Stancu operators in q-calculus

Due to the intensive development of q-Calculus, various generalizations of many classical approximation processes of positive type have emerged. The first researches have been achieved by A. Lupaş [15] in 1987 and by G.M. Phillips [19] in 1997 who proposed q-variants of the original Bernstein operators. While Lupaş' operators are given by rational functions, Phillips' operators are composed of polynomials, named at present time q-Bernstein polynomials. During the last decade, acquiring popularity, these new polynomials have been studied by many authors who obtained a great number of results related to various properties of these operators. Reviews of the results on the q-Bernstein polynomials along with an extensive bibliography on this matter is given in [17]. At the same time, integral extensions in q-Calculus of Bernstein operators have been investigated. We refer to the q-analogues of the Bernstein-Durrmeyer operators [5], [13] and of the Bernstein-Kantorovich operators [20].

A generalization of Bernstein operators with great potential in Approximation Theory was achieved by D.D. Stancu [21]. Recently, G. Nowak [16] introduced a q-analogue of Stancu's operators. The goal of this section is to investigate statistical approximation property of q-Stancu operators. We also introduce an extension of this class acting on the space of real valued functions defined on a rectangular domain.

At this stage we require some preliminary results concerning q-integers, see, e.g., [14]. In the sequel, for our purposes, we just assume that $q \in (0,1)$. It should be mentioned that many results relating to q-Bernstein polynomials aim at the case $0 < q < 1$. The reason is simple: under this assumption, q-Bernstein polynomials generate positive linear operators and this aspect is significantly used in investigation.

For any $n \in \mathbb{N}_0$, the q-integer $[n]_q$ and the q-factorial $[n]_q!$ are respectively defined by

$$[n]_q = \sum_{j=0}^{n-1} q^j, \quad [n]_q! = \prod_{j=1}^{n} [j]_q, \quad n \in \mathbb{N},$$

and $[0]_q = 0$, $[0]_q! = 1$. The q-binomial coefficients or Gaussian coefficients denoted by $\begin{bmatrix} n \\ k \end{bmatrix}_q$ are defined as follows

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \frac{[n]_q!}{[k]_q![n-k]_q!}, \quad k = 0, 1, \ldots, n.$$

Obviously, for $q=1$ one has $[n]_1!=n!$ and $\begin{bmatrix} n \\ k \end{bmatrix}_q = \binom{n}{k}$, the ordinary binomial coefficients.

For $f\in C([0,1])$, $\alpha\geq 0$ and each $n\in\mathbb{N}$, in [16] have been defined the operators

$$(B_n^{q,\alpha}f)(x)=\sum_{k=0}^{n}p_{n,k}^{q,\alpha}(x)f\left(\frac{[k]_q}{[n]_q}\right),\quad x\in[0,1], \tag{4}$$

where

$$p_{n,k}^{q,\alpha}(x)=\begin{bmatrix} n \\ k \end{bmatrix}_q \frac{\prod_{i=0}^{k-1}(x+\alpha[i]_q)\prod_{s=0}^{n-1-k}(1-q^s x+\alpha[s]_q)}{\prod_{i=0}^{n-1}(1+\alpha[i]_q)}, \tag{5}$$

$0\leq k\leq n$. Note, an empty product is taken to be equal to 1. This class contains as special cases the following well-known sequences.

i) For $\alpha=0$, $B_n^{q,0}\equiv B_n^q$ represents q-Bernstein operator introduced by Phillips [19].

ii) For $\alpha=0$ and $q=1$, $B_n^{1,0}\equiv B_n$ is the classical Bernstein polynomial.

iii) For $q=1$, $B_n^{1,\alpha}\equiv B_n^{\langle\alpha\rangle}$ turns into Stancu operator [21] defined as follows

$$(B_n^{\langle\alpha\rangle}f)(x)=\frac{1}{1^{[n,-\alpha]}}\sum_{k=0}^{n}\binom{n}{k}x^{[k,-\alpha]}(1-x)^{[n-k,-\alpha]}f\left(\frac{k}{n}\right),$$

$x\in[0,1]$. Here $t^{[m,a]}=\prod_{j=0}^{m-1}(t-ja)$ represents the generalized factorial power with the step a, $a\in\mathbb{R}$, $m\in\mathbb{N}$. The following identities hold [16; Theorem 2.5]

$$\begin{cases} (B_n^{q,\alpha}e_0)(x)=1, \quad (B_n^{q,\alpha}e_1)(x)=x, \\ \\ (B_n^{q,\alpha}e_2)(x)=\dfrac{1}{1+\alpha}\left(x(x+\alpha)+\dfrac{x(1-x)}{[n]_q}\right),\ n\in\mathbb{N},\ x\in[0,1]. \end{cases} \tag{6}$$

From the above, we easily deduce

$$\|B_n^{q,\alpha}e_2-e_2\|=\frac{1}{4(1+\alpha)}\left(\alpha+\frac{1}{[n]_q}\right)\leq\alpha+\frac{1}{[n]_q}. \tag{7}$$

Also, we observe that the operators interpolate the approximated function f at the endpoints of the interval.

In order to transform the sequence $(B_n^{q,\alpha})_n$ into an approximation process, for each $n\in\mathbb{N}$, we replace the constants q and α by the numbers $q_n\in(0,1)$ and $\alpha_n\geq 0$, respectively.

Theorem 5 *Let the sequences $(q_n)_n$, $(\alpha_n)_n$ be given such that $0<q_n<1$ and $\alpha_n\geq 0$, $n\in\mathbb{N}$. Let the operators $B_n^{q_n,\alpha_n}$, $n\in\mathbb{N}$, be defined as in (4). If*

$$st-\lim_n q_n=1 \quad \textit{and} \quad st-\lim_n \alpha_n=0, \tag{8}$$

then, for each $f\in C([0,1])$, one has

$$st-\lim_n\|B_n^{q_n,\alpha_n}f-f\|=0. \tag{9}$$

Proof. Our assertion is implied by Theorem 1 Indeed, taking in view the identities (6), relation (7) and our hypothesis (8), all three requirements from (1) are fulfilled. The conclusion follows. □

Remark 5 *Set $S=\{10^k:\ k\in\mathbb{N}_0\}$ and consider the sequences*

$$q_n=\begin{cases} \dfrac{1}{n}, & n\in S, \\ \\ 1-\dfrac{1}{n}, & n\in\mathbb{N}\setminus S, \end{cases} \qquad \alpha_n=\begin{cases} \log n, & n\in S, \\ \\ \dfrac{1}{n}, & n\in\mathbb{N}\setminus S. \end{cases}$$

Since relations (8) take place, statement (9) follows. On the other hand, $\lim_{n\to\infty}\|B_n^{q_n,\alpha_n}e_2-e_2\|$ does not exist, see (7). In this way we exhibited an example for which the Popoviciu-Bohman-Korovkin theorem does not work but the statistical convergence works.

A sequence $(u_n)_n$ of real numbers satisfies the so-called *one-sided Tauberian condition* if there exists a constant $C > 0$ such that for every $n \in \mathbb{N}$

$$n\Delta u_n \geq -C, \tag{10}$$

where Δu_n denotes the *backward difference* $u_n - u_{n-1}$, with $u_{-1} = 0$.

Clearly, this condition is trivially satisfied if the involved sequence is non-decreasing. Fridy and Khan [10; Theorem 2.2] proved the following.

If condition (10) takes place, then

$$st - \lim_n u_n = L \qquad implies \qquad \lim_n u_n = L.$$

On the basis of this result, we can state.

Theorem 6 *Let the sequences $(q_n)_n$, $(\alpha_n)_n$ be given such that $0 < q_n < 1$, $\alpha_n \geq 0$, $n \in \mathbb{N}$, and they satisfy condition (10). Let the operators $B_n^{q_n,\alpha_n}$, $n \in \mathbb{N}$, be defined as in (4). If (8) holds, then $(B_n^{q_n,\alpha_n})_n$ converges uniformly on $C([0,1])$ towards the identity operator.*

Now we are going to present the bivariate extension of q-Stancu operators. To achieve it, we use the method of parametric extensions of mentioned univariate operators. Both classical and new results concerning the tensor product operators can be found, e.g., in the well accomplished survey due to L. Beutel and H. Gonska [4].

Set $K = [0,1] \times [0,1]$ the unit square, and let $q(q_1, q_2)$ belong to the interior of K. We consider the parameter $\alpha(\alpha_1, \alpha_2) \in \mathbb{R}_+ \times \mathbb{R}_+$. For each $(n_1, n_2) \in \mathbb{N} \times \mathbb{N}$, we define the operator involving a cartesian product grid and acting on $C(K)$ as follows

$$(B_{n_1,n_2}^{\langle q,\alpha\rangle} f)(x_1, x_2) = \sum_{k_1=0}^{n_1} \sum_{k_2=0}^{n_2} f(\lambda_{n_1,k_1,q_1}, \lambda_{n_2,k_2,q_2}) p_{n_1,k_1}^{q_1,\alpha_1}(x_1) p_{n_2,k_2}^{q_2,\alpha_2}(x_2), \tag{11}$$

$(x_1, x_2) \in K$, where $\lambda_{n_\nu,k_\nu,q_\nu=[k_\nu]_{q_\nu}/[n_\nu]_{q_\nu}}$ and $p_{n_\nu,k_\nu}^{q_\nu,\alpha_\nu}$, $0 \leq k_\nu \leq n_\nu$, are defined by (5), $\nu = 0, 1$.

For the particular case $\alpha = (0,0)$, $B_{n_1,n_2}^{\langle q,0\rangle}$ turns into bivariate q-Bernstein operators studied by D. Bărbosu [3].

Remark 6 *It's known that the tensor product of two univariate operators inherits many properties of its factors. For instance, the constructed operators are linear positive and interpolate the function f on the vertices of the unit square, this means*

$$(B_{n_1,n_2}^{\langle q,\alpha\rangle})(\tau_1, \tau_2) = f(\tau_1, \tau_2), \quad \tau_1, \tau_2 \in \{0, 1\}.$$

Moreover, taking into account (6), by a straightforward calculation, we deduce the images of the test functions $e_{i,j}$, $0 \leq i, j \leq 2$, under these operators.

Theorem 7 *Let $q(q_1, q_2) \in (0,1) \times (0,1)$ and $\alpha(\alpha_1, \alpha_2) \in \mathbb{R}_+ \times \mathbb{R}_+$. The operators $B_{n_1,n_2}^{\langle q,\alpha\rangle}$, $(n_1, n_2) \in \mathbb{N} \times \mathbb{N}$, defined by (11) verify the following identities*

$$B_{n_1,n_2}^{\langle q,\alpha\rangle} e_{i,j} = e_{i,j}, \quad (i,j) \in \{(0,0), (0,1), (1,0)\},$$

$$(B_{n_1,n_2}^{\langle q,\alpha\rangle} e_{2,0})(x_1, x_2) = (B_{n_1}^{q_1,\alpha_1} e_2)(x_1), \quad (B_{n_1,n_2}^{\langle q,\alpha\rangle} e_{0,2})(x_1, x_2) = (B_{n_2}^{q_2,\alpha_2} e_2)(x_2),$$

for each $(x_1, x_2) \in K$.

At this moment, following a similar course as in the one-dimensional case, we can establish conditions for ensuring the uniform or the statistical convergence of the sequence of operators to the identity operator. For each $n(n_1, n_2) \in \mathbb{N} \times \mathbb{N}$, substituting α by $\alpha_n(\alpha_{1,n_1}, \alpha_{2,n_2})$, $\alpha_{\nu,n_\nu} \geq 0$, and q by $q_n(q_{1,n_1}, q_{2,n_2})$, $0 < q_{\nu,n_\nu} < 1$, where $\nu = 0, 1$, we get the following predictable result.

If $(\alpha_n)_n$ and $(q_n)_n$ are convergent (statistically convergent) to $(0,0)$ and $(1,1)$ respectively, then the sequence of operators is uniform convergent (statistically convergent) to the identity operator on the space $C(K)$.

References

[1] O. Agratini, On statistical approximation in spaces of continuous functions. *Positivity*, 13(4), 735-743, 2009.

[2] F. Altomare, M. Campiti, *Korovkin-type Approximation Theory and its Applications*, de Gruyter Studies in Mathematics, Vol. 17, Walter de Gruyter and Co., Berlin, 1994.

[3] D. Bărbosu, Some generalized bivariate Bernstein operators. *Mathematical Notes, Miskolc*, 1, 3-10, 2000.

[4] L. Beutel, H. Gonska, Quantitative inheritance properties for simultaneous approximation by tensor product operators II: applications. *Mathematics and its Applications, Proc. 17th Scientific Session, Braşov, April 2003, Ed. G.V. Orman*, pp. 1-28, Transilvania University Press, 2003.

[5] M.-M. Derriennic, Modified Bernstein polynomials and Jacobi polynomials in q-calculus. *Rendiconti del Circolo Matematico di Palermo, Serie II, Suppl.*, 76, 269-290, 2005.

[6] O. Duman, C. Orhan, Statistical approximation by positive linear operators. *Studia Math.*, 161(2), 187-197, 2004.

[7] E. Erkuş, O. Duman, A Korovkin type approximation theorem in statistical sense. *Studia Sci. Math. Hungar.*, 43(3), 285-294, 2006.

[8] H. Fast, Sur la convergence statistique. *Colloq. Math.*, 2, 241-244, 1951.

[9] J.A. Fridy, On statistical convergence. *Analysis*, 5(4), 301-313, 1985.

[10] J.A. Fridy, M.K. Khan, Statistical extensions of some classical Tauberian theorems. *Proc. Amer. Math. Soc.*, 128(8), 2347-2355, 2000.

[11] A.D. Gadjiev, On P.P. Korovkin type theorems. *Mat. Zametki*, 20, 781-786, 1976 (in Russian).

[12] A.D. Gadjiev, C. Orhan, Some approximation theorems via statistical convergence. *Rocky Mountain J. Math.*, 32(1), 129-138, 2002.

[13] V. Gupta, Some approximation properties of q-Durrmeyer operators. *Applied Mathematics and Computation*, 197, 172-178, 2008.

[14] V. Kac, P. Cheung, *Quantum Calculus*, Universitext, Springer-Verlag, New York, 2002.

[15] A. Lupaş, A q-analogue of the Bernstein operator. *University of Cluj-Napoca, Seminar on Numerical and Statistical Calculus*, Preprint, 9, 85-92, 1987.

[16] G. Nowak, Approximation properties for generalized q-Bernstein polynomials. *J. Math. Anal. Appl.*, 350, 50-55, 2009.

[17] S. Ostrovska, The first decade of the q-Bernstein polynomials: results and perspectives. *Journal of Mathematical Analysis and Approximation Theory*, 2(1), 35-51, 2007.

[18] M.A. Özarslan, O. Duman, O. Doğru, Rates of A-statistical convergence of approximating operators. *Calcolo*, 42, 93-104, 2005.

[19] G.M. Phillips, Bernstein polynomials based on the q-integers. *Ann. Numer. Math.*, 4, 511-518, 1997.

[20] C. Radu, Statistical approximation properties of Kantorovich operators based on q-integers. *Creative Math. & Inf.*, 17(2), 75-84, 2008.

[21] D.D. Stancu, Approximation of functions by a new class of linear polynomial operators. *Rev. Roumaine Math. Pures Appl.*, 8, 1173-1194, 1968.

[22] H. Steinhaus, Sur la convergence ordinaire et la convergence asymptotique. *Colloq. Math.*, 2, 73-74, 1951.

[23] V.I. Volkov, On the convergence of sequences of linear positive operators in the space of continuous functions of two variables. *Dokl. Akad. Nauk SSSR (N.S.)*, 115, 17-19, 1957 (in Russian).

Octavian Agratini
Babeş-Bolyai University
Faculty of Mathematics and Computer Science
400084 Cluj-Napoca
ROMANIA
E-mail: *agratini@math.ubbcluj.ro*

Saddika Tarabie
Tishrin University
Faculty of Sciences
1267 Latakia
SYRIA
E-mail: *sadikatorbey@yahoo.com*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Stable 3D Scene Restoration Using One Active PTZ Camera

Kiril Alexiev, Iva Nikolova, Georgi Zapryanov

Abstract

The active PTZ (Pan Tilt Zoom) camera is a key element of an intelligent surveillance system. The opportunity to control camera parameters significantly increases the abilities of these cameras as information sources. It is commonly regarded that a camera gives 2D presentation of 3D scene. The depth of the scene is irrevocably lost and only some image features may indirectly reveal the position of the objects in third dimension. The active camera can partially overcome this loss of information. The suitable control of camera parameters may be used for estimation of the depth of the observed objects. The paper discuses one of the methods for 3D scene restoration called "depth from defocus" and its inherited characteristics. All key points of the approach realization are described and commented. Experimental studies, using test patterns and real objects are presented to test its applicability.

1 Introduction

Many scientific and engineering applications require characterization of objects and phenomena occurring in our three-dimensional world. It is commonly regarded that the widespread digital cameras with CCD (Charge-Coupled Device) and CMOS (Complimentary Metal-Oxide Semiconductor) image sensors produce 2D presentation of the 3D environment. In that registration the location of objects such as angular coordinates in horizontal and vertical direction remains the same, but information about the distance to the objects is irrevocably lost. The restoration of the third dimension is, however, critically important for determining the actual spatial arrangement of objects, object tracking, understanding the spatial-temporal relationships between objects, evaluation of their behavior, and predicting future events. Scientists have long attempted to develop hardware and software tools for 3D recovery. Nowadays, there are professional CMOS video cameras, specially designed to capture video with depth information [1, 2], but unfortunately, they are too expensive and their resolution is a long way away from the quality of the usual CCD and CMOS cameras used today. Therefore, more research efforts are put into a software solution to the problem with standard video sensors.

Most of the currently available techniques on visual 3D recovering have focused on multisensor approach (stereo vision) and other algorithms that require multiple images, such as structure from motion, shape from shading, range from focus and depth from defocused images [4]. Depth estimation using frames from single camera is a difficult task, and it requires some prior knowledge about the scene and the global structure of the image. In this

paper, the problem of estimating distances to the objects in indoor scenes is discussed on the base of the well-known depth from defocus approach. This technique is the most attractive one with little hardware requirements, the small number of processed image frames and the absence of content-based image analysis. However, adequate spectral content and accurate information of the lens parameters of camera system must be ensured to get good estimates of the depth.

The automatic depth estimation requires several challenging problems to be successfully resolved during the camera frames analysis [9]: (i) image texture analysis; (ii) noise estimation for the processed part of the image frame; (iii) blur spot diameter determination; (iv) outlier detection and elimination; (v) distance/depth estimation.

The remainder of the paper comprises four sections. The second part gives the necessary mathematical information about depth from defocus approach. The third section deals with proposed in the paper solutions of the problems, cited above. Some experimental studies, results and concluding remarks are discussed in the last two sections.

2 Mathematical background of depth from defocus approach

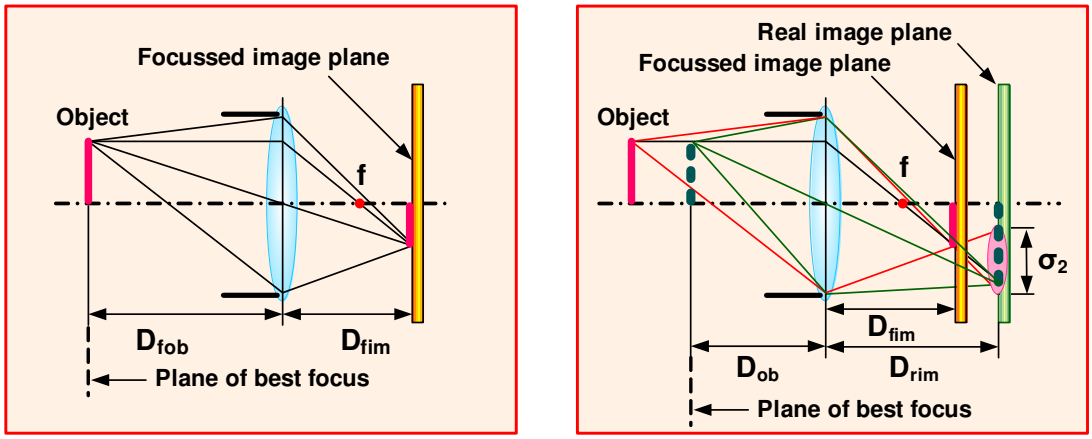
The defocus information in the image of an object formed by a camera system can be used to determine the distance (i.e. depth) to this object from the camera. The general principle of the methods for depth estimation by defocus exploits the physical effect produced by the modification of the focus length or the lens aperture, and the distance to an object on a received image. When a camera is focused on an object at a certain distance a clear (sharp) image is produced but other objects, both closer and farther than the focus distance, form spots more or less blurred according to their distance to the image plane (Figure 1). In case that the sensor is nearer or farther away from the lens than the corresponding lens focus length, the image becomes blurred due to the intersection of light rays either in front of, or behind, the sensor (image) plane. Another factor affecting the blur is lens aperture (iris). Decreasing a lens opening narrows the light rays passing through the lens and reduces defocus spot diameter. Practically, this means that the smallest lens opening will give the sharpest image for a scene of several objects at varying distances. When the aperture is relatively larger (i.e. the lens opening increases), the blur spot diameter becomes larger.

The methods proposed in the literature for depth estimation from blur [3]-[8] use different optical properties of the camera model. The most frequently used model with an intermediate level of complexity is thin lens model. It replaces the multi-lenses camera optic with a thin lens and the geometrical optics is used to derive some basic characteristics of focusing (Figure 1(a)). The Gaussian lens law postulates that:

$$\frac{1}{f} = \frac{1}{D_{fob}} + \frac{1}{D_{fim}} \tag{1}$$

where f is the focal length of the lens, D_{fob} is the distance from the object point to the lens center, and D_{fim} is the distance from the lens center to the plane on which the image of the observed object is in perfect focus. From Eq. 1 it follows that for a chosen focal length there is an infinite numbers of pairs (D_{fob}, D_{fim}), satisfying the equation. The pointed ambiguity shows that some restrictions have to be introduced to the camera model. These constrains stem from the realization of optical sensors. Choosing a suitable zoom setting, the user defines indirectly the scale parameter M – the ratio between the size L_{im} of the image of an object on the sensor matrix

and the actual size L_{ob} of the object. The scale uniquely defines the relation $M = \frac{L_{im}}{L_{ob}} = \frac{D_{fim}}{D_{fob}} = \frac{d_{ccd\ sensor}}{d_{FOV}}$. In the last expression the effective diagonal of the matrix is denoted with $d_{ccd\ sensor}$ and with d_{FOV} - the diagonal of field of view at distance D_{ob}. Furthermore, an additional valid constrain is: $D_{fob} + D_{fim} = D = const$.



(a) Focused object (b) Defocused object

FIGURE 1. Image formation process using thin lens camera model

Let σ_2 denote blur spot diameter (Figure 1 (b)), D_{rim} is the distance from the lens center to the plane of the taken image, B_2 is the diameter of the lens aperture, D_{fob} and D_{fim} are previously defined distances from the lens center to the object and to the plane of the focused image. All these parameters are related by the following equation:

$$\sigma_2 = \frac{B_2}{D_{fim}} abs(D_{rim} - D_{fim}) \tag{2}$$

The distance D_{fim} is expressed from the lens law:

$$D_{fim} = \frac{fD_{fob}}{D_{fob} - f} \tag{3}$$

Let consider that the real image plane is a focused image plane for an object, placed at distance D_{ob} (Figure 1(b)). Thus, Eq. 1 can be used again to express D_{rim}:

$$D_{rim} = \frac{fD_{ob}}{D_{ob} - f} \tag{4}$$

Substituting Eq. (3) and (4) into Eq. (2) gives:

$$\sigma_2 = \frac{B_2}{D_{fim}} abs\left(\frac{fD_{ob}}{D_{ob} - f} - \frac{fD_{fob}}{D_{fob} - f} \right) \tag{5}$$

According to Eq. 5, the diameter of the blur spot physically depends on the lens parameters (B_2 and f) and the depth D_{fob} of a scene point. Thus, focusing camera on different distances,

i.e. varying the focused distance D_{ob}, we obtain the functional dependency of blur spot diameter on D_{ob}, as it is shown on Figure 2, and therefore the distance D_{fob} can be calculated.

The ambiguity in determining the distance D_{fob} for a particular object point is due to the lack of function monotony (Figure 2). The object can be located at two different distances for one and the same value of the blur diameter. An example of the above statement is presented in Figure 2 where Point 1 corresponds to the distance of 2200 mm and Point 2 corresponds to the distance of 4700 mm - in this case, an enormous inaccuracy in the distance estimation for one and the same blur diameter can be detected. This uncertainty could be resolved by combining blur measurements from more than two images, obtained for different focal length settings of the camera.

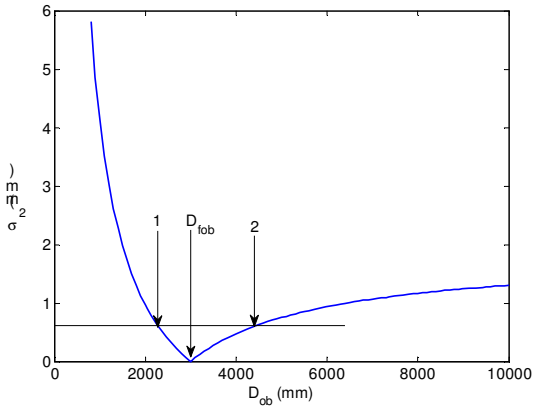


FIGURE 2. Dependency between blur spot diameter and distance to the object in case of a camera focused on distance of 3000 mm

3 Automatic depth estimation using "depth from defocus" approach

The generous algorithm of depth from defocus approach consists of the following steps: (1) Acquiring several image frames of the observed scenario with static camera under different focused distances (the camera settings remain the same, except for the focus setting). (2) Noise estimation. (3) Selection of the features/fields, depth of which will be evaluated. (4) Blur spot diameter determination. (5) Depth estimation of each features/fields in the image. These steps will be described in the following subsections.

3.1 Perceiving image frames

Two types of frame samples are perceived. The first sample consists of two or more frames, acquired under equal camera setting to estimate the noise in the images. The second sample consists of three or more frames taken at different focus settings of the camera. The minimal number of frames necessary for depth estimation is four (2+3=5 frames, but one of the frames in the first sample may used in the second sample too). The time interval for receiving such a quantity of frames usually does not exceed several hundreds of milliseconds. It is considered that this interval is small enough to regard the scene static. The assumption of static scene is not valid for the case of fast moving objects and an additional step for feature registration is required to be inserted into the general algorithm of depth recovery.

3.2 Noise estimation

Noise is the most important parameter, characterizing the quality of received images. The images with low level of the noise further better blur spot diameter estimation and thus enhance the accuracy of depth estimation. The International Standard IEC1146-1 regularizes the procedure of signal to noise ratio (SNR) for analogue cameras in laboratory conditions, but this approach is inappropriate for real time application. The SNR in image sensors can be determined by the ration of generated charge carriers (signal electrons) to the number of unwanted charge carriers (noise electrons). Let assume that the noise signal in an image pixel *(i,j)* is independent, identically distributed (iid) additive and stationary Gaussian with zero mean:

$$I_{i,j}(n) = S_{i,j}(n) + N_{i,j}(n), \tag{6}$$

where $S_{i,j}(n)$ is the useful signal amplitude from the *n*-th image frame, $N_{i,j}(n)$ is the corresponding noise signal and $I_{i,j}(n)$ is the received noisy signal. The intensity level of received signal is known and it is easy to be measured. The main problem is to evaluate the amplitude of the noise. The noise level may fluctuate given different conditions of work and have to be estimated without usage of calibrated sources of light. We propose to use the difference signal of two consecutive image frames with the same camera settings to estimate the noise level [13]:

$$\begin{aligned} &I_{i,j}(n) - I_{i,j}(n-1) = S_{i,j}(n) + N_{i,j}(n) - S_{i,j}(n-1) - N_{i,j}(n-1) = \\ &= S_{i,j}(n) - S_{i,j}(n-1) + N_{i,j}(n) - N_{i,j}(n-1) \end{aligned} \tag{7}$$

In the case of static scenes (static illumination and static objects), the difference signal will be mainly generated from noise. Even of the case of slight changes the difference of useful signal will be greatly depressed. The remaining image signal may be additionally removed by high pass filtering. At the same time (in the case of difference signal analysis) the noise variance will be doubled. Thus, the estimated level of noise variance will be:

$$\sigma_N^2 = \frac{\sigma_D^2}{2}, \tag{8}$$

where σ_D^2 is the estimated noise variance of the differential image.

If the noise is position dependent, the noise evaluation is performed for the pixels of the feature/field of interest.

3.3 Selection of features/fields

The selection of features/fields for depth evaluation is very important task, unresolved until now. It is clear that every pixel in the image plane corresponds to a point (area) from the scene with unique depth (at a given distance). The ambition to work with particular points can not be realized due to the lack of methods for blur estimation on a point.

All other suggested methods are based on multipoint analysis. These approaches have a serious drawback – there is not guarantee that all points correspond on one and the same depth.

Lines (edges, contours) in the image are the most commonly preferred features to be processed. Usually the choice of lines is validated by the fact that they determine the plane borders, and the scenes contain many lines and particularly straight lines. There are many well-developed relatively simple algorithms for line determination – Canny, Sobel, and etc. The very strong benefit of the line exploration is that the intensity change on them is assured.

Often the rectangular fields of different sizes of the image are analyzed. The study of intensity deviation in the field is mandatory for robust depth estimation. If the intensity deviation for all processed image frames doesn't exceed the noise variance for the same field, that field will not be useful in depth estimation.

3.4 Blur spot diameter estimation

The method used to estimate the diameter of the blur spot belongs to the so-called "early methods" of blur estimation. It relies on the analysis of lines detected in a camera image and it is well established with known pros and cons.

In this investigation the gradient analysis of image intensity in direction, orthogonal to the edge line, is used to estimate blur spot diameter. In many cases the gradient analysis is applied to a part of a line. The integration reduces the influence of additive Gaussian noise and improves the accuracy of the result. The brightness profile for different focus values – 1.3m, 1.4m, 1.5m, 1.6m, and 1.7m is depicted on Figure 3. The blur spot diameter estimate is received from the width of the brightness profile. In the case when the local template in the processed field disturbs the brightness profile the results are far from the true estimate.

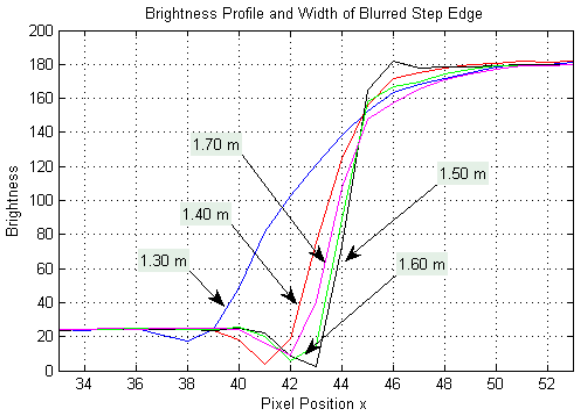


FIGURE 3. Brightness profiles on different focused distances

3.5 Automatic depth estimation using "depth from defocus" approach

It is proposed the object-to-camera distance evaluation to be performed by applying an optimization procedure. The nonlinear curve fitting task for depth estimation from defocus blur is determined in least squares sense: find the vector P of estimated parameters that minimizes the criterion $\min \sum_{i=1}^{m} \left(F(P, D_{ob}(i)) - \sigma_2(i)\right)^2$, where m is the number of the processed image frames, received on different camera focus settings. Here $F(P, D_{ob}(i))$ is the function from Eq. 5, calculating blur spot diameter for the used camera focus settings (distances $D_{ob}(i), \quad i = 1,...,m$) and focal length f. The blur spot diameters $\sigma_2(i)$ are measured for one and the same observed object in the processed image frames.

The parameter vector P consists of three elements: the real distance to the object D_{fob}, the iris diameter B_2 and the scaling coefficient M. The objective function is subject to constraints in the form of parameter bounds. The set of lower and upper bounds of the estimated parameters is determined by the admissible ranges of the camera parameters and the distance to the object. As can be seen from Figure 2 and Eq. 5, the objective function is nonlinear and its solution requires an iterative procedure to establish a direction of search the optimal value of the estimated distance to the object. This is achieved by the Levenberg–Marquardt algorithm [10-12], which interpolates between the Gauss–Newton algorithm and the method of gradient descent. The iterative minimization procedure starts, using an initial guess for the for the parameter vector P. The convergence of the algorithm to final solution - the global minimum, depends on the initial values

of these parameters, as well of the data obtained from measurements of the blur spot diameter. Usually the starting point for the estimated depth is chosen to be equal to the focused length of the camera, corresponding to the minimal blur in the received image frames.

Sometimes, the blur spot diameter cannot be properly estimated in the real scene images. This requires some additional blur estimates (a larger sample) to be taken and the outliers to be removed. Additional data are obtained, analyzing more images of the same object, taken at different focused distances. The outlier rejection is performed applying a simple procedure that detects the outliers by their relatively larger residuals. Then the optimization procedure restarts from the last "quasi-optimal" point.

4 Experimental results

Our experimental work has two goals: (i) to verify the applicability of the mathematical model to the practical camera system we use and to explore the dependency between the camera parameters and the scene characteristics and (ii) to test the evaluation accuracy of the recovered depths in a real scene. Two sets of experiments are conducted using Axis214 PTZ IP video surveillance camera.

(a) Test templates – scene 1 (b) Real scene 2 (c) Real scene 3

FIGURE 4. Experimental scenario

In the first group of experiments seven planar patterns, having two types of vertical edges – "inside" edges (belongs to the same plane of the pattern) and "outside" edges (formed on the transition from one pattern to another) with high contrast are placed at different, a priory known distances from the camera (Figure 4(a)). Three identical experiments were conducted: for shorter distances (1-4 m), for the middle distances (3-6 m) and for longer distances (4-7 m). The patterns (templates) are positioned at intervals of 50 cm. The camera is focused consecutively on each template under different zooms in the range of 6x-9x. The width of blur is calculated for the different camera parameter settings. The difference in pixel intensity is used in order to reduce the influence of the changes in illumination. The influence of the additive Gaussian noise is lowered by integrating up to a hundred points per line.

The second group of experiments concerned real partially structured scenes with many vertical lines (Figure 4(b,c)). The real distances to the object edges were measured in advance by laser distance meter Leica DISTO D3, with measurement accuracy of ±1 mm. The camera is focused consequently on different distances – from an initial selected position through 50 cm and under different zoom settings.

The received image frames were sequentially processed by several procedures: (i) utilization of a Canny algorithm for edge detection and localization; (ii) estimation of the blur spot diameter of the discovered edges; (iii) utilization of the Levenberg–Marquardt optimization procedure, using the blur estimates of the same edge in several frames as input data; (iv) object points distance calculation. Some of the results obtained during the experiments are shown in Table 1.

TABLE 1. Multiple depth recovery: accuracy evaluation.

Test templates – scene 1 Zoom 9x			**Real Scene 2 Zoom 6x**		**Real Scene 3 Zoom 6x**	
	Inside Edges	**Outside Edges**				
Real distance [m]	Estimated distance [m]	Estimated distance [m]	Real distance [m]	Estimated distance [m]	Real distance [m]	Estimated distance [m]
3.0	3.16	2.67	1.37	1.30	3.78	1.64
3.5	3.20	3.27	1.77	1.68	3.22	1.20
4.0	3.65	3.72	2.43	1.49	2.81	2.50
4.5	3.91	4.02	2.04	1.99	2.47	1.75
5.0	4.54	4.68	2.04	1.62	1.69	1.65
5.5	4.83	5.04	1.91	1.92	1.65	1.49
6.0	5.27	4.91	2.06	1.86		

* Focused distances: 1 – 3 m (real scene); 3 - 6 m (test templates)

5 Analysis of results and concluding remarks

In this paper a realization of an approach for computing distance to scene objects when multiple, defocused images are captured from active camera is proposed. The depth recovery task is presented as non-linear line fitting optimisation problem. The received at this early stage of evaluation results show that the proposed technique for estimating the distance to the object points is effective for the purposes of automatic depth perception. In some cases, independently of its easy implementation, it can yield to inaccurate results (see Table 1). The main sources of errors are: (i) improper calibration of camera parameters; (ii) lack of noise level estimation; (iii) failures in edge detection and localisation; (iv) inaccurate blur spot diameter estimate for an edge point. Furthermore, it should be noted that the experimental evaluations were conducted with conventional video surveillance PTZ camera, which is not specially designed for depth estimation purposes.

The thorough analysis of the main sources of errors and careful tuning of parameters of the used algorithms may limit the errors in the distance evaluation to a few percent. Unfortunately, we did not find a testbed for evaluation of depth recovery algorithms for a single PTZ video surveillance camera.

- Based on the performed experimental work with test patterns and real scene targets, the following conclusions and recommendations can be drawn:
- Estimating the scene depth from defocus using Levenberg–Marquardt algorithm requires at least three (better 5 or more) image frames, captured at different focused distances due to the number of the estimated parameters and the presence of outliers.
- The parameter of crucial importance on depth estimation procedure is the blur measure in defocused image frames. In most cases, the distance estimation errors for the 'inside' and the 'outside' edges of the test patterns are approximately equal (Table 1). However, the analysis of the edge intensity profile did not proved itself as reliable algorithm in real scenes, where the edges may have different local structure. The standard gradient operators fail to detect and localize edges when the blur scale, contrast and image noise level exceed some admissible threshold, and therefore the wrong results are received (Table 1, Real scene 3, Real distances 3.78 m and 3.22 m).
- It is necessary to recommend situating camera focus around or in the front of the object, rather than behind it, because the errors in blur spot diameter estimation on the steepest part of the function (Figure 2) lead to smaller errors in distance estimation.

Acknowledgement: This paper is supported by the Bulgarian Ministry of Education and Science under grants VU-MI-204/06.

References

[1] http://www.swissranger.ch
[2] http://www.3dvsystems.com
[3] Schechner, Y., Nahum K., "Depth from Defocus vs. Stereo: How Different Really Are They?," International Journal of Computer Vision, v.39 n.2, p.141-162, Sept. 2000.
[4] Namboodiri, V.P., Chaudhuri, S., "On defocus, diffusion and depth estimation," Pattern Recognition Letters, Volume 28, Issue 3, pp. 311-319, Feb. 2007.
[5] McCloskey, S. Langer, M. Siddiqi, K., "The Reverse Projection Correlation Principle for Depth from Defocus," In Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), pp. 607-614, 2006.
[6] Asada, N., Baba, M., Oda, A., "Depth from blur by zooming," Proc. Vision Interface, pp.165-172, 2001.
[7] Surya, G., Subbarao, M., "Depth from defocus by changing camera aperture: A spatial domain approach," Proc. CVPR, pp.61-67, 1993.
[8] Favaro, P, Mennucci, A., Soatto, S., Observing Shape from Defocused Images, International Journal of Computer Vision, v.52 n.1, p.25-43, April 2003.
[9] Nikolova, I., Zapryanov, G., Alexiev, K., "Detecting of Unique Image Features by Using Camera with Controllable Parameters," In Proceedings of the Fourth International Bulgarian-Greek Conference Computer Science'2008, Kavala, Greece, Volume 3, pp. 920-925, 2008.
[10] Levenberg, K., "A Method for the Solution of Certain Problems in Least-Squares," Quarterly Applied Math. 2, pp. 164-168, 1944.
[11] Marquardt, D., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM Journal Applied Math., Vol. 11, pp. 431-441, 1963.
[12] More, J. J., "The Levenberg-Marquardt Algorithm: Implementation and Theory, "Numerical Analysis, ed. G. A. Watson, Lecture Notes in Mathematics 630, Springer Verlag, pp. 105-116, 1977.
[13] Lingfeng Chen, Xusheng Zhang, Jiaming Lin,Dingguo Sha, "Signal-to-noise ratio evaluation of a CCD camera", Optics & Laser Technology Journal, 41 (2009) 574–579.

KIRIL ALEXIEV
Institute for Parallel Processing – Bulgarian Academy of Sciences
Department of Mathematical Methods for Sensor Information Processing
25A Acad.G.Bonchev Str., Sofia, 1113
BULGARIA
E-mail: alexiev@bas.bg

IVA NIKOLOVA
Technical University of Sofia
Department of Computer Systems, Faculty of Computer Systems and Control
8 Kl. Ohridski Blvd. bl 1, Sofia 1000
BULGARIA
E-mail:inni@tu-sofia.bg

GEORGI ZAPRYANOV
Technical University of Sofia
Department of Computer Systems, Faculty of Computer Systems and Control
8 Kl. Ohridski Blvd. bl 1, Sofia 1000
BULGARIA
E-mail:gszap@tu-sofia.bg

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Automated contour determination in ultrasound medical images

Vera Behar, Pavlina Konstantinova, Milen Nikolov

Abstract

Contour determination in images is often the main goal in image utilization. There is no universal segmentation algorithm for images acquired by different technologies in different practical applications. The developed contour tracking algorithm, which is a combination of probabilistic data association approach and Interacting Multiple Model approach, shows attractive results for successful contour determination. This algorithm is based on the assumption that the lesion has convex form, which is often true in medical images. The developed algorithm is forced by equally spaced radii from a pre-defined seed point inside the lesion area. A disadvantage of this method is that this point must be selected manually by the user. In this paper an algorithm for automated selection of the internal seed point is proposed. It is supposed that the image intensity can be divided into two classes of intensities: one - for the background and the other - for the lesion. The main idea is to convert the image in a binary form using appropriate thresholding. Then the pixels of the lesion class are processed to find the center of gravity of the lesion area. This center is assumed to be the required seed point.

1 Introduction

In recent years ultrasound imaging is widely used in different practical applications. It is non-invasive, suitable for real time applications and has comparatively low cost. Contour determination in images is often the main goal in image utilization. There is no universal segmentation algorithm for images acquired by different technologies in different practical areas of application. Usually some prior information is used to make possible an effective solution of the segmentation problem. In [9], an extensive research of the ultrasound medical image segmentation methods is presented. The authors classify the considered methods by their clinical applications. This is motivated by the fact that some specific image characteristics like the convex form of the medical structures or other peculiarities are taken into account. The final paper of the survey in the ten selected influential works is that of Abolmaesumi and Sirospour [1]. The authors developed cavity boundary extraction algorithm based on the known tracking algorithm intended for tracking maneuvering targets in clutter. The algorithm combines the advantages of two powerful approaches- probabilistic data association (PDA) approach and Interacting Multiple Model approach. The algorithm is forced by the equally spaced radii from a

pre-defined seed point inside the lesion area [1]. It incorporates the edge magnitude in probabilistic data association to find a more appropriate point of the contour.

In [2], a new efficient multiple model Monte Carlo algorithm is developed for progressive contour tracking, which takes into account convex, non-circular forms of the lesion areas for segmentation. Another realization of idea proposed in [1] is developed in [6] and [8]. In [8], a new feature, based on the statistical mean intensity of the region around the predicted contour point (pixel) is used to modify the weights of the candidate pixels and to improve the reliability of the tracking process.

The contour determination algorithms based on the tracking idea, show attractive results for successful contour determination. Their common property is the necessity of an internal seed point, which is usually selected manually by the user.

The goal of this paper is to fully automate the segmentation process by automatically selecting the internal seed point. It is supposed that the image contains two classes of intensities: one for the background and the other - for the lesion. The main idea is to convert the image in a binary form using appropriate thresholding. Then the pixels of the lesion class in the binary image are processed to find the center of gravity of the lesion area. This center is assumed to be the required seed point.

The paper is organized as follows. After the introduction the problem of contour determination by using the tracking algorithm is formulated. The proposed algorithm for determination of the internal seed point is presented in section 3. In section 4 the results of using the proposed algorithm are given on the base of both modeled and real ultrasound medical images. In section 5 conclusions are given.

2 Problem Formulation

In medical images the cysts and other lesions more often have a convex form. This property allows the following assumption - the contour of interest is star-shaped i.e. all contour points can be seen from an appropriately selected seed point inside the assumed contour. The contour determination algorithm developed in [1, 6, 8] uses PDA and IMM approaches for tracking of low observable maneuvering target in the presence of false alarms [12]. The contour is treated as a target trajectory and the contour points are defined on the equally spaced radii from the selected seed point inside the assumed contour.

The system state vector x, describing the target dynamic evolves in time according to

$x(k+1) = F(k)x(k) + \nu(k),$ with the measurement vector z, given by:

$z(k+1) = H(k)x(k) + \omega(k)$, where F is the system transition matrix, H is the measurement matrix, $\nu(k)$ and $\omega(k)$ are the zero-mean mutually independent white Gaussian noise sequences with the known covariance matrices $Q(k)$ and $R(k)$, respectively, and k is the time sample.

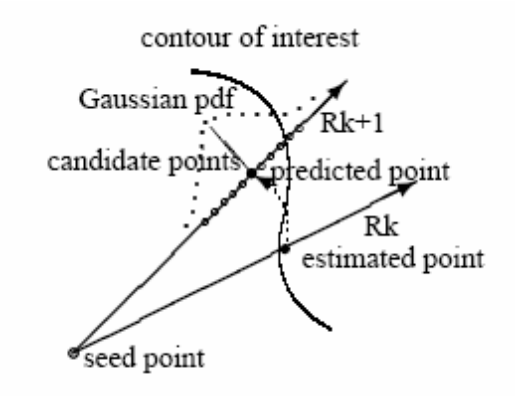


Fig .1. Scheme of the algorithm using seed point

The system dynamic is forced by the equally spaced radii. The schematic presentation of this formulation is shown in figure 1, where two consecutive radii R_k and R_{k+1} are presented. The estimated point is located on the radius R_k and the corresponding predicted point is located on the radius R_{k+1}. The candidate points around the predicted point with their Gaussian pdf associated with the assignment of the corresponding point to the trajectory of the contour are also presented.

The state vector is $x = \left[D \quad \dot{D}\right]'$, where D is the distance from the seed point to the current point and $\dot{D} = \frac{dD}{d\theta}$. The increment of the angle $\Delta\theta = 2\pi / \mathrm{N_r}$, N_r is the number of radii that corresponds to the number of the evaluated contour points. For more details see [3, 12, 6].

3 Automated determination of the internal seed point

The algorithm for automatic determination of the seed point is developed under the following assumptions:

- The image intensity consists of two main classes of intensities – for the object and for the background;
- Only one object is located in the processed region of interest;

The algorithm includes the following processing steps:

1. Select the appropriate threshold with the Otsu algorithm;
2. Form the binary image using the determined threshold;
3. Find the center of gravity of the pixels, corresponding to the class of intensities that characterizes the object. This center is the desired internal point.

The first important step in this algorithm is to select the appropriate threshold. The algorithm developed by Otsu in [7] can be used for this purpose. It seems to be suitable for the case of two classes of pixels. The Otsu algorithm minimizes the weighted sum of within-class variances of the object and background pixels, which is equivalent to maximization of between-class variance. This fact proves the optimality of the determined threshold.

Otsu algorithm description according to [7]

Lets represent the intensities of the image by L gray levels {0,1,2,…L-1} The number of pixels of the level i is denoted by h_i and the total number of pixels is denoted by N. The gray level histogram is normalized and is regarded as a probability distribution function:

$$p_i = h_i / N, \qquad p_i >= 0, \qquad \sum_{i=0}^{L-1} p_i = 1 \ .$$

We assume that the pixels can be divided into two classes C_0 and C_1 by a threshold of the intensity k. Class C_0 includes pixels with the intensity levels {0,1,…,k} and C_1 includes pixels with the intensity levels {k+1,k+2,…,L-1}.

The definition of these classes depends on the lesion types:

- For lesions of type "Cyst" – the objects are darker than the background i.e. the objects are of class C_0 and the background – of class C_1;
- For lesions of type "Tumor" – the objects are lighter than the background i.e. the objects are of class C_1 and the background – of class C_0;

The probabilities for class occurrences ω and the class mean levels μ for both classes are given for each k by:

$$\omega_0(k) = \sum_{i=0}^{k} p_i \ , \qquad \omega_1(k) = \sum_{i=k+1}^{L-1} p_i = 1 - \omega_0(k),$$

$$\mu_0(k) = \sum_{i=0}^{k} i \times p_i / \omega_0(k), \qquad \mu_1(k) = \sum_{i=k+1}^{L-1} i \times p_i / \omega_1(k) = \frac{\mu_{tot} - \mu_k(k)}{1 - \omega_0(k)},$$

where $\mu_k(k) = \sum_{i=0}^{k} i \times p_i$ and the total mean $\mu_{tot} = \sum_{i=0}^{L-1} i \times p_i$.

The class variances are:

$$\sigma_0^2(k) = \sum_{i=0}^{k} (i - \mu_0(k))^2 \, p_i \quad , \sigma_1^2(k) = \sum_{i=k+1}^{L-1} (i - \mu_1(k))^2 \, p_i = \sigma_{tot}^2 - \sigma_0^2(k) \quad ,$$

where the total standard deviation is $\sigma_{tot}^2 = \sum_{i=0}^{L-1} (i - \mu_{tot})^2 \, p_i$.

Otsu suggested the minimization of the weighted sum of within-class variances of both the object and the background pixels to establish the optimal threshold. Recall that the minimization of within-class variances is equivalent to the maximization of between class variance. For each intensity k, the between class variance can be evaluated by:

$\sigma_B^2(k) = \omega_0(k)(\mu_0(k) - \mu_{tot})^2 + \omega_1(k)(\mu_1(k) - \mu_{tot})^2$.

The optimal threshold is defined as $k_{opt} = \max_k \{\sigma_B^2\}$.

Auxiliary binary image

The auxiliary binary image is obtained by using the array of intensities of the original image $Org\,\mathrm{Im}$ and the optimal threshold k_{opt} . Then the binary image array, which is previously cleaned by zeros, is formed as follows:

if $Org\,\mathrm{Im}(i,j) > k_{opt}$ $Bin\,\mathrm{Im}(i,j) = 1$; for i=1...N ; j=1...M , where NxM is the image size.

The binary $Bin\,\mathrm{Im}$ array is used to find the center of gravity of the pixels of the object class. The center of gravity (x_o, y_o) (or center of mass with uniformly distributed masses of the pixels) is obtained by:

$$x_o = \frac{\sum_{y=1}^{N}\sum_{x=1}^{M} x \quad for\ I(y,x)\in C_{obj}}{K} \quad \text{and} \quad y_o = \frac{\sum_{y=1}^{N}\sum_{x=1}^{M} y \quad for\ I(y,x)\in C_{obj}}{K},$$

where K is the number of the pixels $I(x,y)$ belonging to the object class C_{obj} .

This center is used as the internal seed point in the tracking algorithm for contour determination.

Image denoising

In some cases, prior the contour determination algorithm it is useful to perform the preprocessing - denoising procedure in order to reduce speckle noise. In our case we use a wavelet technique that provides an appropriate basis for noisy signal processing. The justification is that the wavelets transform the signals in the *time - frequency* domain [11, 13]. It is stated that in the frequency domain a small number of coefficients represent the significant information and a large number of coefficients with small values is supposed to represent the noise. Because of the multiplicative nature of the speckle noise a logarithmic transform is applied to transform multiplicative speckle noise to additive noise [10]. In [10] and also in our recent works we find out that dual-tree complex wavelet algorithm, developed and described by Kingsbury and Selesnick in [11, 13], gives the best results for contour detection. After wavelet coefficients thresholding the inverse logarithmic transform gives the smoothed image array.

4 Illustrative examples for automated determination of the seed point

Reliable tests of the described algorithm for automated determination of the seed point can be received by using simulated ultrasound images obtained by the Simulation program Field II [4, 5]. In

order to make our test more thoroughly we apply the algorithm to different forms of the objects. In addition we put together the results obtained for the original image and that after denoising.

One simulated image containing an elliptical tumor is presented on fig. 2a. A binary image produced from the simulated image by using the thresholding technique is shown on fig. 2b. On the same fig. 2b, the center of gravity of the pixels from the lesion object class is represented by little circle (in the center of the tumor). The estimated contour around the tumor (white line) is presented in fig. 2c. The true contour of the modeled object is presented by the black line. On fig. 3, the histogram of the intensities of the simulated image is presented and the determined threshold is marked on the x-axes with the symbol 'x'. In this case the threshold is 148, which means that the pixels with intensity greater than the threshold (148) are classified as pixels of the object (lighter than the background).

On fig. 4, the errors of the modeled contour are presented. The errors along axes x and y are presented by the thin lines, and the summarized distance error is presented by the bold line. It can be seen that only a few part of the contour points have error greater than 1 [mm].

The simulated image (from fig. 2a) is firstly denoised, and then the same processing algorithm is applied to the resultant image. The successful determination of the internal seed point is shown in fig. 5b and the outlined contour (white line) can be seen on fig. 5c. The errors shown on fig. 7 are decreased in comparison with the ones on fig. 4. The histogram on fig. 6 shows the increased difference between the centers of intensity of the two classes in comparison with the non-denoising case on fig. 3.

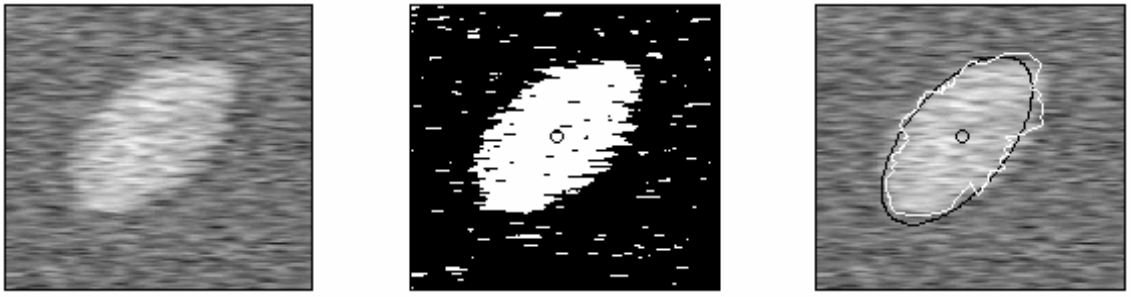
The more complicated form of the Oval Cassini is used to model another object. On fig. 8a, a modeled image with a tumor in form of the Oval Cassini is presented. The auxiliary binary image with the found center of gravity is shown on fig. 8b. In fig. 8c, the estimated contour around the tumor (white line) is presented. For comparison, the true contour of the modeled object is presented by the black line. On fig. 10, the errors of the modeled contour are presented. It can be seen that there are errors greater than 2.5 [mm]. On fig. 9, the histogram of the intensities of the image is presented and the determined threshold is marked on the x-axes with the symbol 'x'.

The results for the denoised image are presented on figures fig. 11a, fig. 11b and fig. 11c correspondingly. The automatically determined seed point (fig. 11b) is used for the tracking algorithm for tumor segmentation. It can be seen that the errors are decreased to the level under 1.5 [mm].

Two real images are processed by the proposed algorithm. The results obtained are presented on fig. 14 and fig. 15.

On fig. 14, a real ultrasound image of a heart is shown. The region of interest (ROI) is selected by using MATLAB Graphical User Interface (GUI). For this region, an auxiliary binary image is obtained and the computed center of gravity for the pixels of the region is marked on fig. 14 with 'o'. The received contour is drawn with white line.

On fig. 15, a real ultrasound (US) image of a breast cyst [14] and the obtained results, achieved by analogical operations, are shown.



a) Image with lesion area b) Auxiliary binary image c) Contour around lesion area

Fig. 2 Results for image with lesion – ellipse

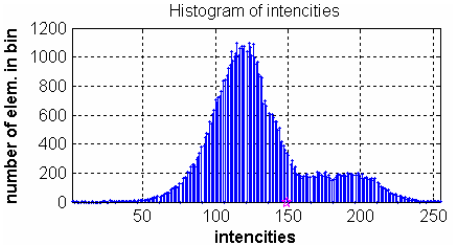


Fig.3 Histogram of image with ellipse

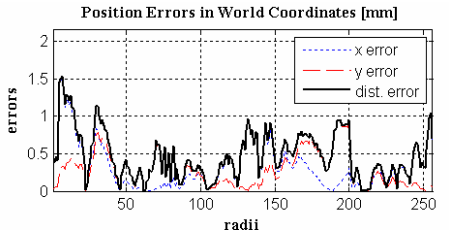
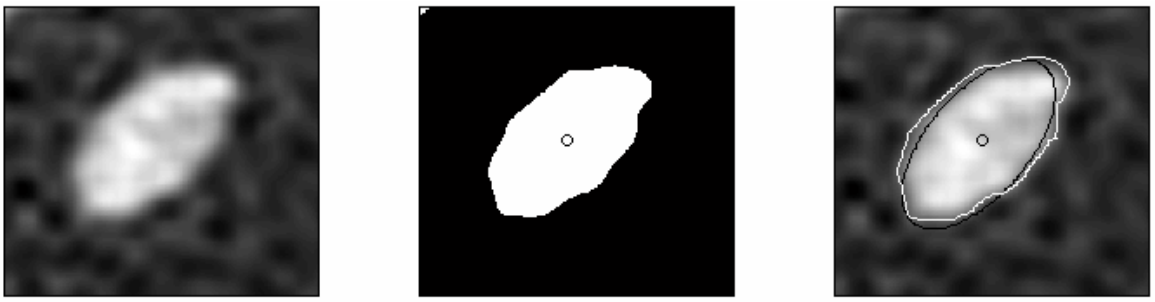


Fig.4. Contour position errors



a) Image with lesion area b) Auxiliary binary image c) Contour around lesion area

Fig.5 Results for denoised image with lesion – ellipse

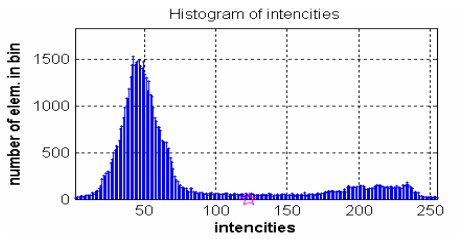


Fig.6 Histogram of denoised image (ellipse)

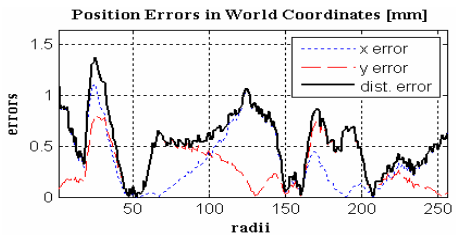
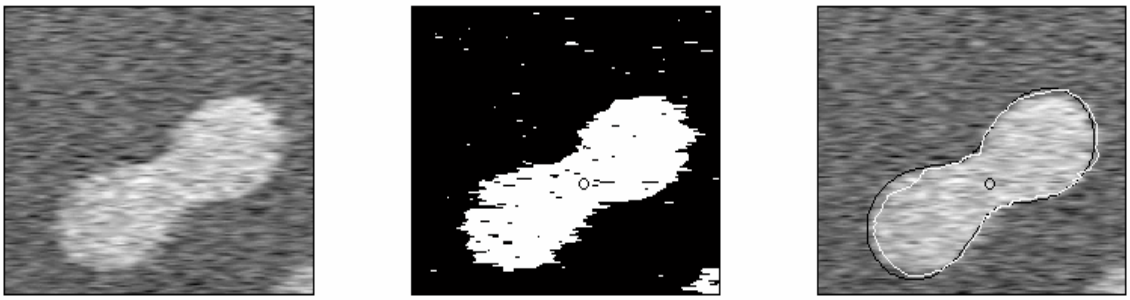


Fig.7. Contour position errors of ellipse



a) Image with lesion area b) Auxiliary binary image c) Contour around lesion area

Fig.8 Results for image with lesion - oval Cassini

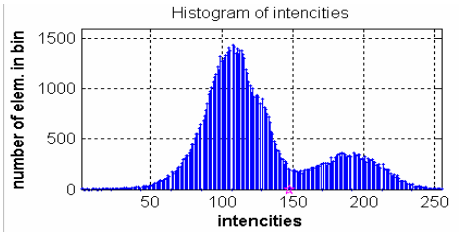


Fig.9 Histogram of image (oval Cassini)

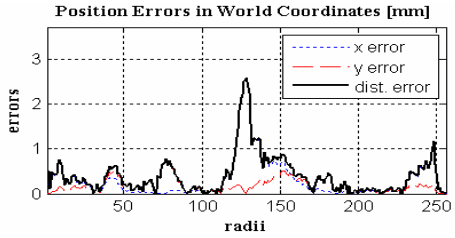


Fig.10. Contour position errors

a) Image with lesion area b) Auxiliary binary image c) Contour around lesion area

Fig.11. Results for denoised image with lesion - oval Cassini

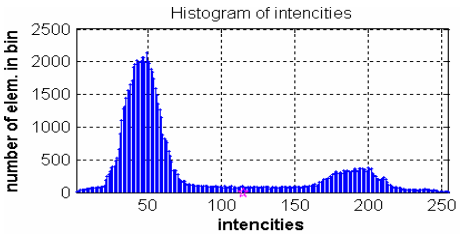


Fig.12 Histogram of denoised image (oval Cassini)

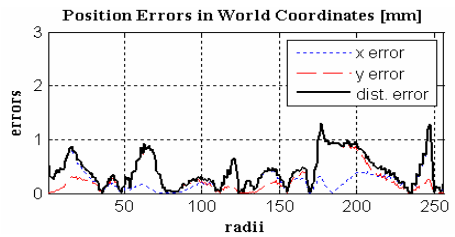


Fig.13. Contour position errors of denoised image (oval Cassini)

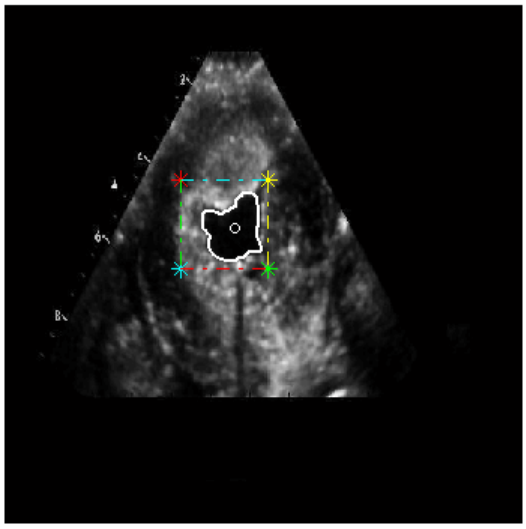


Fig.14 Real image of a heart with specified ROI

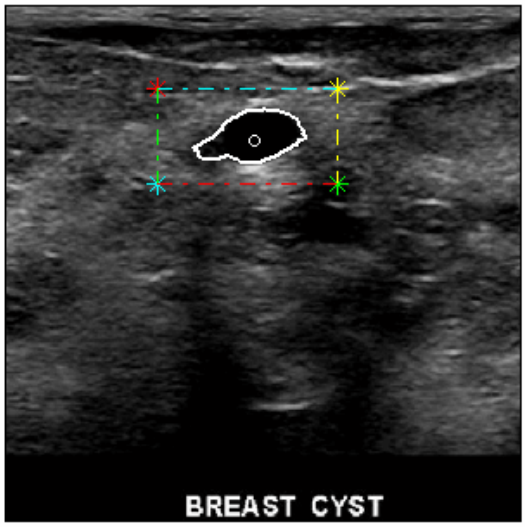


Fig.15 Real image of a breast cyst with specified ROI

5 Conclusion

A new method for automated selection of the internal seed point for convex object segmentation in ultrasound images is proposed. The method uses an auxiliary binary image produced by thresholding of the original image. The assumptions are that the objects have convex form and that the image contains two main classes of intensities: one - for the object and the other - for the background. The threshold is determined by using the Otsu algorithm for threshold determination. The center of gravity of the pixels of the object class is calculated and used as an internal seed point. The method is illustrated by simulating ultrasound images with known true contours of the objects. The estimated contour errors are shown to be mainly less than 1 [mm]. Two real medical images are also processed by the proposed algorithm and the corresponding results are shown.

Acknowledgement: This work is partially supported by the Bulgarian Foundation for Scientific Investigations: MI-1506/05 and VU-MI-204/06.

References:

[1] Abolmaesumi, P., Sirouspour, M.R.: *An Interacting Multiple Model Probabilistic Data Association Filter for Cavity Boundary Extraction from Ultrasound Images*, IEEE trans. on Medical Imaging, vol.23, No 6, June (2004) 772-784

[2] Angelova D., Mihaylova L., *Contour Extraction from Ultrasound Images Viewed As a Tracking Problem*, 12th International Conference on Information Fusion, Seattle, WA, USA, July 6-9, 2009, pp. 284-291 , 978-0-9824438-0-4 ©2009 ISIF.

[3] Bar-Shalom Y., Li X.R., Kirubarajan T. (2001). *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Wiley, New York.

[4] Jensen J.A.: Field: *A Program for Simulating Ultrasound Systems*, Paper presented at the 10th Nordic-Baltic Conference on Biomedical Imaging Published in Medical & Biological Engineering &

Computing, pp. 351-353, Vol. 34, Supplement 1, Part 1, 1996.
http://server.oersted.dtu.dk/personal/jaj/field/

[5] Jensen J. A. and N. B. Svendsen: *Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducer*s, IEEE Trans. Ultrason., Ferroelec., Freq. Contr., 39, pp. 262-267, 1992. (visited 19.05.2008)

[6] Konstantinova P., D. Adam, D. Angelova and V. Behar, *Contour Determination in Ultrasound Medical Images Using Interacting Multiple Mode Probabilistic Data Association Filter*, Springer-Verlag,2007, LNCS 4310, pp.628-636.

[7] Nacereddine N., L. Hamami, M. Tridi, and N. Oucief, Non-Parametric Histogram-Based *Thresholding Methods for Weld Defect Detection in Radiography*, World Academy of Science, Engineering and Technology ,9, 2005, pp. 213-217.

[8] Nikolov M., P. Konstantinova and V. Behar, *An Improvement of a Probabilistic Based Image Segmentation Algorithm for Ultrasound Medical Images*, International Conference, Automatics and Informatics, 3-6 Oct. 2007, Sofia, Bulgaria VII-9 to VII-12

[9] Noble J. A., Boukerroui D., "*Ultrasound Image Segmentation: A Survey*", IEEE Trans. Medical Imaging,Vol. 25, No. 8, pp. 987–1010, 2006.

[10] Konstantinova P., Nikolov M., Behar V., "*Ultrasound Image Enhancement via Wavelet Thresholding*", International Conference "Automatics and Informatics'08" , Sofia, October 1-4 2008, pp. IV-5 до IV-8, ISSN 1313-1850

[11] Kingsbury N. G., "*A dual-tree complex wavelet transform with improved orthogonality and symmetry properties*", Proceedings of the IEEE Int. Conf. on Image Proc. (ICIP), 2000, http://taco.poly.edu/WaveletSoftware (visited 19.11.2008)

[12] Kirubarajan T., Bar-Shalom Y., (2004). *Probabilistic Data Association Techniques for Target Tracking in Clutter*, Proc. of the IEEE, vol.92, No 3, march, pp. 536-557.

[13] Selesnick I., R. Baraniuk, and N. Kingsbury, *The Dual-Tree Complex Wavelet Transform*, IEEE Signal Processing Magazine, Nov. 2005, pp.123-151.

[14] http://smiswi.sasktelwebhosting.com/ultrasound.htm#BreUS#BreUS (US breast image) (visited 18.08.09)

VERA BEHAR
Institute for Parallel Processing (IPP) - Bulgarian Academy of Sciences (BAS), Department of Mathematical Methods for Sensor Information Processing.
Address: Institute for Parallel Processing, BAS, "Acad. G. Bonchev", Str., bl.25-A, 1113 Sofia, BULGARIA.
E-mail: behar@bas.bg

PAVLINA KONSTANTINOVA
Institute for Parallel Processing (IPP) - Bulgarian Academy of Sciences (BAS), Department of Mathematical Methods for Sensor Information Processing.
Address: Institute for Parallel Processing, BAS, "Acad. G. Bonchev", Str., bl.25-A, 1113 Sofia, BULGARIA.
E-mail: pavlina@bas.bg

MILEN NIKOLOV
Institute for Parallel Processing (IPP) - Bulgarian Academy of Sciences (BAS), Department of Mathematical Methods for Sensor Information Processing.
Address: Institute for Parallel Processing, BAS, "Acad. G. Bonchev", Str., bl.25-A, 1113 Sofia, BULGARIA.
E-mail: milenik@bas.bg

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Some approximations in the bar deflection analysis, II

Nicolae Boja, Georgeta Brăiloiu, Remus Ene

Abstract

The aim of this paper is to approximate the strain energy of deformation and to determine the work of a charge P acting **oblique** on an elastic slender prismatic bar, by using the method of bar discretization. The discrete distribution of the deformation state is described by means of recurrence formulas. Also, in the case of a curved bar (cross-section is a curvilinear trapezium), a dependence on the principal curvatures of the middle surface of the bar is put in evidence.

The study is made for a bar viewed as a simple uniform and isotropic material body ([12]), assumed as a trivial manifold without boundary endowed with a single coordinate chart.

Finally, a numerical example for a bar with known physical and geometrical characteristics is given.

Keywords: strain energy, intrinsic representation, deflection map and angle of deflection, principal curvatures.

1. Introduction.

Consider an elastic slender bar of length L with a fixed endpoint A and a free endpoint B loaded with a force P; $\overline{\mathrm{AB}}$ denotes its initial centroidal axis, as a vertical straight line segment.

Usually the physical properties of the bar by the elasticity modulus E and by the inertial moment I are given, and the geometrical ones by dimensions, by the cross-section shape, and by the centroidal axis aspect (especially by the curvature $1/R$) are described. The magnitude of deformation at each point of the bar also depends on the charge value P as well on the direction of the force action, that is of the load vector $\mathbf{P}$, ($P = \| \mathbf{P} \|$).

Especially two directions of the load vector action are of interest for the technicians:

I. The action of $\mathbf{P}$ is vertically,

II. The action of **P** is oblique, but its direction is passing through a fixed point of the plane of deformation.

Because the first case was treated in our paper [6], in this paper will be approached only the case II.

In the first case the bar is subjected (at time t = 0) to an axial compression which will be constant in time (t > 0). In the second one the compression is not axial.

Many studies about the elastic bar deflections by a long sequence of authors were made (see [1], [2], [3], [4], etc.). They analyzed especially the deflections of prismatic bars with deformed centroidal axis relative to a 'Cartesian coordinate system', associated to a reference configuration, i.e. to an embedding

$$\varphi_0 : \overline{AB}\ \alpha\ \mathbf{E}^2,$$

where $\mathbf{E^2}$ is the Euclidean 2- space, called *plane of deformation*.

Here we have to mention the fact that because a bar is usually viewed as a simple uniform and isotropic material body, it can be assumed as a trivial manifold without boundary for which is enough to consider a single coordinate chart φ_0 that cover the wholly body. So, $\mathbf{E^2}$ can be initially endowed with an orthogonal frame as (Axy), with the origin at fixed end A and with $\overline{AB}$ along to the axis (Ay).

Different from the former papers in this paper we approach the study of the bar deflections having in view a deformed centroidal axis with respect to an 'intrinsic representation'. Until a point this study can be made in a similar way both for bars with rectangular and with curved (but symmetrical) cross-section, that is independently of the cross-section shape.

This last aspect permit us to correct the differential equation of the deformed centroidal axis that follows from the expression $M = EI \cdot 1/R$ of the bending moment.

Thus, the geometric aspect of the deformation will be described with respect to a new coordinate frame of the Euclidean affine plane $\mathbf{E^2}$. So, during deformation we consider a mobile orthogonal frame (OXY) with the origin O placed (at each time t > 0) at the free end of the centroidal axis of bar and with axes (OX) and (OY) parallel with the initial axes (Ay) and (Ax), respectively, but having opposite orientation.

With respect to this frame the deformed centroidal axis $\overline{\overline{AO}}$ can be looked as an image of the *deflection map*

$$\phi : \overline{AB}\ \alpha\ \overline{\overline{AO}},$$

defined by means of an intrinsic equation

$$\psi = \psi(s), \quad (s \in [0, L]), \tag{1}$$

where $s = \lambda(\mathrm{O}\widetilde{\mathrm{M}})$ is the arc length from the origin point $\mathrm{O} = \phi(\mathbf{B})$ to an arbitrary point $\widetilde{\mathrm{M}} = \phi(\mathrm{M})$, $(\mathrm{M} \in \overline{\mathrm{AB}})$, and ψ denotes the angle measure of the tangent straight line at $\widetilde{\mathrm{M}}$ to the centroidal axis of the deformed bar with the positive orientation of the axis (OX), called *angle of deflection* at M. We mention that here ϕ can be seen as a composed map, $\phi = \varphi \circ \varphi_0$, where φ is another configuration in $\mathbf{E}^2$ of the centroidal axis $\overline{\mathrm{AB}}$ which becomes at time t (> 0) a curvilinear arc $\overline{\overline{\mathrm{AO}}}$; (1) is called *intrinsic equation* of the deformed bar.

Further, we use the exact expression of the curvature

$$\frac{1}{R} = \frac{d\psi}{ds} \tag{2}$$

instead of the approximate one $\mathrm{d}^2\mathrm{y} / \mathrm{dx}^2$, which is frequently used by many researchers in order to obtain a second order differential equation that can be easily integrated. In such a case the general method adopted to reduce he nonlinearity degree of the differential equation describing the bar deformation makes use of the Cebishev's polynomials.

The deformations of a slender bar such as is considered above by an equation of Bessel type are described:

$$\frac{d^2\psi}{ds^2} + \frac{w}{EI} s \cdot \cos(\beta - \psi) = 0, \tag{3}$$

where w is the specific weight per unit length and β is the angle measure of undeformed bar with the horizontal plane.

Exact solutions of (3) are known for the cases of horizontal ($\beta = 0$) or vertical ($\beta = \pi / 2$) initial positions of the bar.

For instance, in the case of a vertical bar the deflection ϕ by a linear equation is described (see Denman and Schmidt, [1970]) :

$$\frac{d^2\psi}{ds^2} + \mathbf{k}_1^{\,2}\, s \cdot \psi = 0, \tag{4$_\mathrm{I}$}$$

where

$$\mathbf{k}_1^{\,2} = 2wJ_1(\alpha) / \alpha \cdot EI \tag{5}$$

is a constant depending on the Bessel function

$$J_1(\alpha) = \frac{1}{\pi\alpha} \int_{-\alpha}^{\alpha} \psi \sin\psi \cdot (\alpha^2 - \psi^2)^{-1/2}\, d\psi \,.$$

We note, this equation is obtained from (3) taking $\beta = \pi/2$ and $\sin\psi = \frac{2J_1(\alpha)}{\alpha}\cdot\psi$ on the symmetric real interval $[-\alpha, \alpha]$.

In this case the known general solution of (4_I) is

$$\psi = As^{1/2}J_{1/3}(\frac{2}{3}ks^{3/2}) + Bs^{1/2}J_{-1/3}(\frac{2}{3}ks^{3/2}),$$

where A, B and k are constants.

One can see that this exact solution is not quite a simple one because of more difficulties of the integrals computing. This is the reason we tray to find an alternative solution by means of the bar discretization; the solution so obtained is consisting in a recursive approximation.

Finally, we end the introduction with the following remark.

In this study the bar is considered to be vertical. However, the differential equation (4_I) will be used only in the case when the load **P** is vertically too, while when direction of **P** is oblique it is better to make use of another differential equation of the deformed centroidal axis obtained from the bending moment equation. So, expressing the moment M in two different ways, we have

$$EI\cdot\frac{d\psi}{ds} = -P_X\cdot Y + P_Y\cdot X, \qquad (4_{II})$$

where P_X, P_Y are the projections of **P** on the axes (OX), (OY), respectively, and X, Y are the coordinates of the point $\widetilde{M}$ with respect to these axes.

In this case, another positive constant will be useful in our study:

$$\mathbf{k}_0^2 = P/EI. \qquad (6)$$

Let C be the intersection point of the straight line which defines the direction of **P** with the (Ay) axis of the initial frame and denote by c its distance to the free end B of the bar. If the measure of the angle of the previous two straight lines is accepted to be small, we can approximate by P the projection P_X, i.e. $P_X \cong P$. Thus, if δ is the horizontal displacement during the bar deflection of the free end B, i.e. $\delta = d(O,(Ay))$ (see Fig.1), by a geometrical reason, it follows $P_Y/\delta = P_X/c$, which implies that $P_Y \cong P\cdot\frac{\delta}{c}$.

On the other hand we know that at each point $\widetilde{M}$ (X, Y) the relations

$$\frac{dX}{ds} = \cos\psi, \quad \frac{dY}{ds} = \sin\psi$$

are specific for a plane curve given in an intrinsic representation, such as (1).

Thus, by derivation of the equation (4_{II}) with respect to *s*, this one can be

$$\frac{d^2\psi}{ds^2} = -\mathbf{k}_0^2(\sin\psi - \frac{\delta}{c}\cos\psi)$$

or, still,

$$\frac{1}{2}\frac{d}{ds}(\frac{d\psi}{ds})^2 = -\mathbf{k}_0^2(\sin\psi - \frac{\delta}{c}\cos\psi)\frac{d\psi}{ds}$$

Multiplying this equation by ds and integrating it one obtains

$$\frac{1}{2}(\frac{d\psi}{ds})^2 = \mathbf{k}_0^2(\cos\psi + \frac{\delta}{c}\sin\psi) + C\,; \tag{7}$$

the integration constant *C* can be found with the help of boundary conditions (8), given bellow.

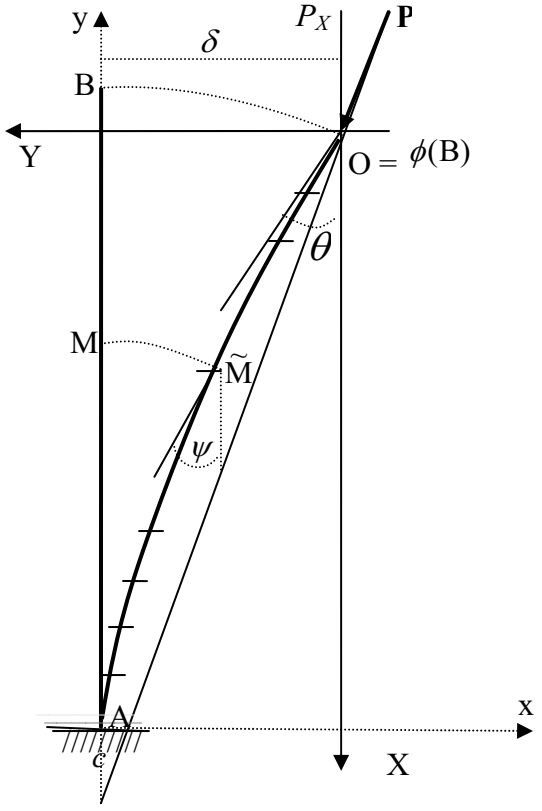


Fig.1. Bar deflection under the action of an oblique load **P**

2. Boundary value problems and the bar discretization.

Consider a deformed bar and assume the deformation ϕ consists in a deflection only, which means the cross-sections remain undeformed.

The plane of minimal bending rigidity, which is at the same time the plane of symmetry, determines the longitudinal section in the bar and contains the points A, M, and $\widetilde{\mathrm{M}}$. This plane will be denoted by $\prod^{L}_{(\mathrm{A},\mathrm{M},\widetilde{\mathrm{M}})}$, called *plane of deformation*, being spanned by the vectors $\widetilde{\mathbf{e}}_2$ and $\widetilde{\mathbf{e}}_3$ of the orthogonal Darboux frame $\{\widetilde{\mathrm{M}};\ \widetilde{\mathbf{e}}_1,\ \widetilde{\mathbf{e}}_2,\ \widetilde{\mathbf{e}}_3\}$ associated to the middle surface $\widetilde{\mathrm{S}} = \phi\,(\mathrm{S})$ of the deformed bar, where S denotes the *middle surface* (see a mathematical definition in our book [11], p.35) of bar at the initial state. Here we have to mention that S is unique only in the case of prismatic (initially, curved or not) bars, while the round bars have before deformation an infinite number of middle surfaces consisting in all plane sections containing centroidal axis. But, during the deformation, $\widetilde{S}$ can be defined as a surface containing the deformed centroidal axis, symmetric with respect to it, of which tangent plane at each point $\widetilde{\mathrm{M}} = \phi(\mathrm{M})$ is spanned by the pair $\{\widetilde{\mathbf{e}}_1, \widetilde{\mathbf{e}}_2\}$ of orthogonal vectors, while $\widetilde{\mathbf{e}}_3$ is a unit normal vector to $\widetilde{S}$, tangent to the trajectory $\widetilde{\Gamma}$ of the point $\widetilde{\mathrm{M}} = \phi(\mathrm{M})$ during the deflection, that is to the *orbit* of M by the deformation ϕ. This curve is not contained by $\widetilde{S}$. But, the pair $\{\widetilde{\Gamma}_1, \widetilde{\Gamma}_2\}$ of curvature lines of $\widetilde{S}$ at each point $\widetilde{\mathrm{M}} = \phi(\mathrm{M})$ are assumed to be the plane curves: intersection of $\widetilde{S}$ with cross-section and the deformed centroidal axis, respectively. These curves are tangent to $\{\widetilde{\mathbf{e}}_1, \widetilde{\mathbf{e}}_2\}$, correspondently.

Similarly, $\prod^{T}_{\widetilde{\mathrm{M}}}$ designates the plane of cross-section of the bar through the point $\widetilde{\mathrm{M}}$; it is spanned by the vectors $\widetilde{\mathbf{e}}_1$ and $\widetilde{\mathbf{e}}_3$ of the previous frame and is tangent to the orbit $\widetilde{\Gamma}$.

The cross-sections at different points are assumed to remain undeformed during the deflection.

Let θ be the measure (in radians) of the angle $\angle(\widetilde{\mathbf{e}}_2, \mathbf{e}_2)$ at the deformed position O of the free endpoint B and ψ the similar angle at current position $\widetilde{\mathrm{M}}$.

Taking $s\,(= \lambda(\mathrm{O}\widetilde{\mathrm{M}}))$ as a parameter relative to its interval of variation, [0, L], the following initial conditions must be satisfied:

$$\psi(0) = \theta,\ \psi'(0) = 0; \quad \psi(L) = 0,\ \psi'(L) = 0 \tag{8}$$

which by a technical point of view are requested.

To render the problem amenable to a numerical treatment we achieve a "discretization" of the bar by means of a sequence of cross-section planes $\{\prod^{T}_{\widetilde{\mathrm{M}}_i}, (i \in \overline{0,n})\}$ trough the points $\widetilde{\mathrm{M}}_i, \left(i \in \overline{0,n}\right)$, such that $\widetilde{\mathrm{M}}_0 \equiv \mathrm{O}$ and $\widetilde{\mathrm{M}}_n \equiv \mathrm{A}$. We observe the considered above points correspond to the values

$$s_i = i \cdot \frac{L}{n}, \ (n \in N^*), \tag{9}$$

which define a division of the interval [0, L]; the norm of division is equal to $\nu = L/n$.

The exact equations of the different positions in the Euclidean affine space $\mathbf{E^3} \equiv$(OXYZ) occupied by the deformed centroidal axis during the deflection cannot be generally known.

So, the deformed state of the bar caused by the action of bending load P can be described geometrically with the help of the system of values $\{\psi_i, \psi'_i\}_n$ at a sufficient number of points of centroidal axis, which also permit to compute the axis curvatures $(1/R)_i$ at each point $\tilde{\mathrm{M}}_\mathrm{i}$, $\left(i \in \overline{0,n}\right)$. The problem is to express them with the help of some known elements.

This is the reason of our following theorem:

Theorem 1. *The system of values* $\{\psi_i, \psi'_i\}_n$ *defining the "discrete distribution" of the deformation state of a vertical bar with a free end, discretized by a sequence of cross-section planes* $\{\prod^T_{\tilde{\mathrm{M}}_\mathrm{i}}, (i \in \overline{0,n})\}$, *can be estimated by recurrence formulas function only of the following positive numbers* θ, $\mathbf{k}_0^2$, ν, *that represent the measure of deflection angle of the bar free endpoint, the constant of physical properties, and the norm of division of the discretization, respectively. In a similar manner can be expressed the curvatures corresponding to all the points of deformed centroidal axis with respect to that discretization.*

Note. Here we have to mention that θ is a constant for a given load **P** (the charge intensity and its direction are known), in other words it depends of this last one.

Proof. Putting $\psi(s_i) = \psi_i$, $(i \in \overline{0,n})$, and using the well known approximation formulas

$$\psi'_j = \frac{\psi_{j+1} - \psi_{j-1}}{2\nu}, \quad \psi''_j = \frac{\psi_{j+1} - 2\psi_j + \psi_{j-1}}{\nu^2} \tag{10}$$

for $j \in \overline{1, n-1}$, and

$$\psi'_0 = \frac{\psi_1 - \psi_0}{\nu}, \quad \psi'_n = \frac{\psi_n - \psi_{n-1}}{\nu}, \tag{10'}$$

the initial conditions (8) lead also to the approximate values of ψ at the neighboring points of the endpoints:

$$\psi_1 = \theta, \quad \psi_{n-1} = 0, \tag{11}$$

which implies that the first and the last parts of the discretization have a similar behavior as that of the endpoints of the bar.

Also one may consider the following usual approximations:

$$\cos\,\psi_i = 1-\frac{1}{2}\psi_i^{\,2},\ \sin\psi_i = \psi_i - \frac{1}{6}\psi_i^{\,3},\ \ \left(i\in\overline{1,n}\right). \tag{12}$$

Case II. The action of **P** is oblique.

Taking in view the first two initial conditions (8), corresponding to the point $\mathrm{O} = \phi(\mathbf{B})$, we can determine the constant C for the equation (7), as

$$C = -\mathbf{k}_0^2\,(\cos\theta+\frac{\delta}{c}\sin\theta)\,,$$

such that this one becomes

$$\left(\frac{d\psi}{ds}\right)^2 = 2\mathbf{k}_0^2\,[\cos\psi-\cos\theta+\frac{\delta}{c}(\sin\psi-\sin\theta)]\,. \tag{13}$$

From here we can express the values of the derivatives along to the deformed bar with respect to the values of deflection angles at corresponding points. If we have in attention the points of chosen discretization, these values will be

$$\psi_i'(s) = -\sqrt{2}\,\mathbf{k}_0[\cos\psi_i - \cos\theta+\frac{\delta}{c}(\sin\psi_i - \sin\theta)]^{1/2} \tag{14}$$

at all points $\widetilde{\mathrm{M}}_{\mathrm{i}}$, $\left(i\in\overline{0,n}\right)$, where ψ_i have to be computed for this case.

On the other hand, we also can compute the coordinates (X, Y) of an arbitrary point $\widetilde{\mathrm{M}}$ of the centroidal axis of deformed bar by evaluating the curvilinear integrals

$$X = \int_0^s \cos\psi\,ds,\quad Y = \int_0^s \sin\psi\,ds\,,$$

if these integrals are transformed into some simple integrals with the help of arc element expressed from (13) as

$$ds = -\frac{\sqrt{2}}{2}\mathbf{k_0^{-1}}[\cos\psi - \cos\theta+\frac{\delta}{c}(\sin\psi - \sin\theta)]^{-1/2}\,d\psi\,.$$

But here we have to mention that such kind of integrals are not simple to be computed because of the complicated expression of ds. So, it remains to compute only the coordinates (X_i, Y_i) of the points $\widetilde{\mathrm{M}}_{\mathrm{i}}$, $\left(i\in\overline{0,n}\right)$, of chosen discretization by using the values ψ_i given bellow and the values (9) of s_{i}.

The initial conditions (8) are verified (for $s=0$, and $s=L$) if and only if the following condition holds

$$tg\frac{\theta}{2}=\frac{\delta}{c}. \tag{15}$$

Moreover we also recall the validity of (11) for this case too, such that the remaining values of $\psi_k, k\in\overline{2,n-2,}$ can be computed by means of some recurrence formulae as follows. First we need to express the value ψ_2 by integrating $\psi_2'(s)$ with de help of (12). So, we obtain

$$\psi_2 = \theta[1+\mathbf{k}_0^2\nu^2(\frac{\theta^2}{6}-1)]+\mathbf{k}_0^2\nu^2 tg\frac{\theta}{2}(1-\frac{\theta^2}{2}), \tag{16}$$

where $\nu=L/n$ is the norm of division. The value ψ_2 can be consider as an element of reference in order to compute the intermediate values of the function ψ at other points of the division of the real interval [0, *L*]. The first part of its expression,

$$A = \theta[1+\mathbf{k}_0^2\nu^2(\frac{\theta^2}{6}-1)],$$

represents the value of ψ_2 when the action of charge **P** is vertical.

In a similar way we also obtain the other values of the deflection angle at the points of discretization as

$$\psi_k=\psi_{k-2}[\frac{1}{6}\mathbf{k_0^2}\nu^2(\psi_{k-1}^2-\theta^2)+\frac{1}{\theta}A+1]-\psi_{k-2}+\psi_2-A+\frac{1}{2}\mathbf{k_0^2}\nu^2 tg\frac{\theta}{2}\cdot(\theta^2-\psi_{k-1}^2), \tag{17}$$

where the first part

$$B_k=\psi_{k-2}[\frac{1}{6}\mathbf{k_0^2}\nu^2(\psi_{k-1}^2-\theta^2)+\frac{1}{\theta}A+1]-\psi_{k-2}$$

represents the values of ψ at the nodes of division for $k\in\overline{2,n-2,}$ as well when the action of charge **P** is vertical.

These end the proof of the first part of theorem.

Besides, the values (14) may be used to compute the curvatures of the deformed centroidal axis at the points $\tilde{M}_j$, $\left(j\in\overline{1,n-1}\right)$:

$$(\frac{1}{R})_j=\frac{d\psi_j}{ds},\ (j\in\overline{1,n-1}) \quad . \tag{18}$$

The curvatures corresponding to endpoints of the bar are obtained using (8) and (18) as

$$(\frac{1}{R})_0=\psi'_0=\psi'(0)=0,\quad (\frac{1}{R})_n=\psi'_n=\psi'(L)=0. \tag{19}$$

But the curvatures of centroidal axis $\widetilde{\Gamma}_2$ at the points $\widetilde{\mathrm{M}}_{\mathrm{j}}$, $\left(j \in \overline{1,n-1}\right)$, also depend on the principal curvatures of the middle surface $\widetilde{S}$ of a curved prismatic bar. $\widetilde{S}$ is assumed undeformed, excepting a simple bending along $\widetilde{\Gamma}_2$ such that the first curvature line Γ_1 of $\widetilde{S}$ remains undeformed during bar deflection. More precisely, these values depend on the principal curvature $\widetilde{k}_2$ according to the fact that $\widetilde{\Gamma}_2$ is also a curvature line of $\widetilde{S}$. So, we may use the formula **(**see Boja, Ivan, Brailoiu, [1987] **)**

$$\widetilde{k}_2 = \frac{1}{\sqrt{\widetilde{g}_2}} \cdot (\frac{1}{R})_2, \quad (\widetilde{g}_2 = \langle \widetilde{\mathbf{e}}_2 ; \widetilde{\mathbf{e}}_2 \rangle) \tag{20}$$

in order to compute the curvatures $(\widetilde{k}_2)_{\mathrm{i}}$ of $\widetilde{S}$ at all points $\widetilde{\mathrm{M}}_{\mathrm{i}}$, $\left(i \in \overline{0,n}\right)$.

This ends the proof. #

3. Approximate and exact formulas of the bar bending energy

In two papers published before (see Boja, Ivan, Brailoiu, [1993] and Brailoiu, Boja, [1993]) the following results were established:

Theorem 2. *The strain energy of deformation (under a pure bending moment M) of a slender bar with a fixed endpoint, loaded with a charge P and acting oblique at the free endpoint, is given by the exact formula*

$$U_M = P \cdot \int_0^L [\cos\psi - \cos\theta + tg\frac{\theta}{2}(\sin\psi - \sin\theta)]\cos\psi \cdot ds, \tag{21}$$

where L *denotes the length of the bar,* ψ *is defined by the centroidal axis equation (1), and* $\theta = \psi(0)$ *by an initial condition is given.* ##

Starting with a discretization of the bar with the help of the family of planes $\{\prod_{\widetilde{\mathrm{M}}_{\mathrm{i}}}^{T}, (i \in \overline{0,n})\}$ orthogonal to the tangent straight lines to the centroidal axis at points that correspond to the sequence of arc values (9) one obtain the following formulae for the energy of deformation

$$\Delta U = P\frac{L}{n} \cdot [\sum_{i=1}^{n} (\cos\ \psi_i - \cos\ \theta)\cos\ \psi_i + tg\frac{\theta}{2}\sum_{i=1}^{n}\ (\sin\psi_i - \sin\theta)\cos\ \psi_i], \tag{22}$$

where also can be used the approximations (12).

This corresponds to the case I, when **P** acts vertically. So, only its measure P is involved.

The values (21) and (22) can be evaluated when in (12) all the values $\psi(s_i) = \psi_i$, $(i \in \overline{1, n})$, are known. Thus, if we make use of the constant $\mathbf{k}_0{}^2$ introduced by (6), one can obtains by recurrence the asked before values. We observe the values (11) do not depend on $\mathbf{k}_0$ because of the assumption that at the neighbor points to endpoints of the centroidal axis
the approximation can be considered alike that at the bar extremities.

The vertical displacement of the free end

$$\lambda = \frac{1}{2}\int_0^L tg^2\psi \cdot cos\ \psi\, ds \tag{23}$$

can be used to compute the work of the charge P, $\Delta T = P \cdot \lambda$. Thus, for a discretization as that considered before, we have

$$\Delta T = \frac{P \cdot L}{2n} \cdot \sum_{i=1}^{n} tg^2\psi_i \cdot cos\,\psi_i\ . \tag{24}$$

We remark the expression (24) can not be used to determine critical value of P.

4. Numerical example

Consider a bar with the physical and geometrical characteristics given below.

To solve the problem exposed in the Sections 2 and 3 we used a FreeFem++ soft.

```
//*** Physical and geometrical constants of the material : Iron ***
real L=1.60;                    //m; length of the bar
real E=210*1e+009;              //  N*m^(-2)
real a=2*1e-002;                //  m ; length of an edge of the rectangular cross section
real b=5*1e-002;                //  m ; length of another edge of the rectangular cross section
real vol=a*b*L;                 //  volume of the bar
real ro=7874;                   //  kg/m^3
real mass =vol*ro;              //  mass of the bar
int n=24;                       //  nr of the nodal points
real niu=L/n;                   // norm of the division
real P=6000;                     // N; load of the bar
real I=masa*(a^2+b^2)/12;  // inertial moment
real k0=(P/(E*I))^(1/2);
real theta0=1;                   // radians
//************* Finite elements' space ***************
mesh Th=square(5,2);
fespace Vh(Th,P2);
real[int] psi(n);
real[int] xx(n), yy(n);
psi[0]=theta0;
psi[1]=theta0;
psi[2]=theta0*(1+(k0^2)*(niu^2)*((theta0^2)/6-1));
psi[n-1]=0;
int i;
for (i=3;i<n-1;i++)
{psi[i]=psi[i-1] * (1+(1/6)*(k0^2)*(niu^2)*((psi[i-1])^2-theta0^2)+(1/theta0)*psi[2]) - psi[i-2];};
```

```
for (i=0;i<n;i++)
{xx[i]=i*niu;  yy[i]=psi[i];  cout << i << " " << yy[i] << "\n";};
plot([xx,yy], cmm =" grafic: theta= "+theta0+ " lungimea barei = "+L, wait=1, nbiso=20, fill=1, value=1,
ps="lucrare2.eps");
```

4.1. Table of the values so obtained in the case of the bar loaded oblique:

Node „i”	s_i	ψ_i	Node „i”	s_i	ψ_i
0	0	1.3	12	0.8	1.06881
1	0.0666667	1.3	13	0.866667	1.02853
2	0.133333	1.29637	14	0.933333	0.985484
3	0.2	1.28913	15	1	0.93982
4	0.266667	1.2783	16	1.06667	0.891679
5	0.333333	1.26391	17	1.13333	0.841214
6	0.4	1.246	18	1.2	0.788588
7	0.466667	1.22465	19	1.26667	0.733968
8	0.533333	1.1999	20	1.33333	0.677528
9	0.6	1.17185	21	1.4	0.619449
10	0.666667	1.14059	22	1.46667	0.559916
11	0.733333	1.1062	23	1.53333	0

References

[1]. H. H. Denman, , R Schmidt. An approximate method of analysis of large deflections; *J. of Appl. Math and Phys.*, vol.21, **3**, 412 - 421, Basel , 1970.

[2]. R.,Schmidt, D. A. Dadeppo. Large deflections of heavy cantilever beams and columns; *Quart. Appl. Math.*, 28, **3**, 441- 444, 1970.

[3]. M. C. Dokmeci . A general theory of elastic beams*; Int. J. Solids Struct.*, **8**, 1205 - 1222, 1972.

[4]. A. Pisanty. Geometrically nonlinear behavior of initially curved plane slender bars and frames; *Ing.-Archiv.*, **46**, 233 - 244, 1977.

[5]. N. Boja., M. Ivan., G. Brailoiu. Sur une expression geometrique de l’etat deformé pour les barres a la section courbe; *Proceed. of the 2nd Symp. Math.and its Appl.*, Timisoara, 282 - 286, 1987.

[6]. N. Boja., G. Brailoiu, R. Ene. Some approximations in the bar deflection analysis,I.*; Bull.for Appl.and Comp.Mathematics*, BAM- CXIII, #2382, Budapest, 001 - 008, 2008.

[7]. I. Bors, P. Alexa. Elastica of a beam taking into account the axial and shear effects; *Stability of Steel Structures*, Akad. Kiadó, Budapest, vol.**2**, 1291 - 1296, 1990.

[8]. N. Boja., M. Ivan., G. Brailoiu. Sur l’énergie de déformation des barres a section courbe discretisées; *Proceed. of the 5nd Symp. Math.and its Appl.*, Timisoara, 32 - 39, 1993.

[9]. L. Trabucho, I. M. Viaño. *Mathematical Modelling of Rods*, Handbook of Numerical Analysis; Elsevier

Sci. Publ. B.V., North-Holland, 1993.

[10]. G. Brailoiu, N. Boja. Sur l'énergie de déformation des barres discretisées sous l'action des forces passant par un point fix*; Bul. St. Teh., Univ.Tech.Timisoara, Ser. Mat.-Fiz*., Tom 38(52), 45-50, (1993).

[11]. N. Boja., G. Brailoiu. *Geometric Approach to the Continuum Deformation Analysis*; Monograph. Book. Series , MB-**3** / PAMM, Techn. Univ. Budapest, 1997.

[12]. M. Epstein, M. Elzanowski. *Material Inhomogeneities and their Evolution - A Geometric Approach*; Springer- Verlag, Berlin – Heidelberg - New York, 2007.

[13]. D. Lanc, G. Turkalj, G. Vizentin . An algorithm for numerical creep buckling analysis of beam-type structures*; Bull. For Appl. and Comp. Math*., vol. BAM – CX , #2302, Beograd-Budapest, 55 - 60, 2007.

NICOLAE BOJA
"Politehnica" University Timisoara
Department of Mathematics
Victoriei Sq.2
ROMANIA
E-mail: nicolae.boja@yahoo.com

GEORGETA BRAILOIU
"Politehnica" University Timisoara
Department of Mathematics
Victoriei Sq.2
ROMANIA
E-mail: -

REMUS ENE
"Politehnica" University Timisoara
Department of Mathematics
Victoriei Sq.2
ROMANIA
E-mail: eneremus@gmail.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

An Algebraic Specification for CTL with Time Constraints

Laura Florentina Cacovean

Abstract

In this paper we present the algebraic methodology and its utilization for CTL model checker specifications. This allows the possibility of automatic generation of the model checking algorithms for temporal logics into sets of algebraic specifications. We use ANTLRWorks for implement the all macro-operations of CTL model checker. In this paper we extend the CTL model checker and we give the implementation of time constraints of algebraic model checker specification. Next we give an algebraic specification of time of CTL model checker and a case study which proof that specification and our proposed model is correct construct.

1 Introduction

The model checkers are tools which can be used to verify that a given system satisfies a given temporal logic formula. The model is a directed graph where the nodes represent the states of the system and the edges represents the state transitions. The nodes and the edges can be labelled with atomic propositions what describe the states and the transitions of the system. In order to be verified by a given model, a property is written as a temporal logic formula across the labelled propositions from the model. A model checker is an algorithm that determines the states of a model that satisfy a temporal logic formula.

The algebraic methodology and its utilization in developing instruments for model checker specifications as maps in form $C:L_s \rightarrow L_t$ [4], where L_s is the source language of temporal logic, L_t is the target language representing sets of states of the model M and $C(f \in L_s)=\{s \in M | s\ f\}$, where $\models$ is satisfaction relation. This allows the possibility of automatic generation of the model checking algorithms for temporal logics into algebraic specifications sets. Extensibility and flexibility of algebraic methodology show how the model checkers for various temporal logics can be generated from algebraic specification. In paper [1] we showed how this algebraic context can be used to the specification of *CTL* (Computation Tree Logic) model checker.

2 CTL model checker

CTL model checker is branching-time logic, meaning that its formulas are interpreted over all paths beginning in a given state of the Kripke structure. A Kripke model M over AP is a triple $M=(S, Rel,$

$P:AP \rightarrow 2^S$) where S is a finite set of states, $Rel \subseteq S \times S$ is a transition relation, $P:S \rightarrow 2^{AP}$ is a function that assigns each state with a set of atomic propositions.

A CTL formula is evaluated on a Kripke model M. A path in M from a state s is an infinite sequence of states from S, denoted in the following with $\pi = [s_0, s_1, \ldots, s_{i-1}, s_i, s_{i+1}, \ldots]$ such that $s_0 = s$ and $(s_i, s_{i+1}) \in Rel$ holds for all $i \geq 0$. We write $(s_i, s_{i+1}) \subseteq \pi$ and $s_i \in \pi$. If we express a *path* as $\pi = [s_0, s_1, \ldots, s_i, \ldots, s_j, \ldots]$, for $i < j$, we say that s_i is a state earlier than s_j in π as $s_i < s_j$.

Definition 1 *(Syntax Definition of CTL model checker [3]) A CTL has the following syntax given in Backus-Naur form:*

$f :: \top | \bot | p | (\neg f_1) | f_1 \wedge f_2 | f_1 \vee f_2 | f_1 \subset f_2 | AX f_1 | EX f_1 | AG f_1 | EG f_1 | AF f_1 | EF f_1 | A[f_1 U f_2] | E[f_1 U f_2]$ (1)

where $\forall p \in AP$.

Semantic definition of CTL model checker is provided in [3]. Let M=(S, Rel, P:AP$\rightarrow 2^S$) be a Kripke model for CTL. Given any s in S, in [3] is defined whether a CTL formula f holds in state s. This is denoted this by (M, s) $\models$ f. The satisfaction relation $\models$ is defined by structural induction on fourteen CTL formulas [3].

Many model checking algorithms were developed for different temporal logics [12], thus in this paper is presented a simple universal algorithm based on the algorithm of homeomorphism computation which is used in an algebraic compiler [4]. The generic homeomorphism algorithm is customized by a set of specifications to construct a model checker, implemented as an algebraic compiler $C:L_s \rightarrow L_t$. The specifications consist from a finite set of rules in which each of the rules defines the syntax of some classes constructed in the source language and also the semantic values of these constructs as expressions in the syntax of the target language.

3 Algebraic Structure of CTL

In an algebraic compiler $C:L_s \rightarrow L_t$ the source and the target language used are defined using heterogeneous Σ-algebras and Σ-homeomorphisms [4,5]. The operator scheme of a *Σ-algebras* is a tuple $\Sigma = \langle S, O, \sigma \rangle$ where S is a finite set of states, O is it a finite set of operator names, and $\sigma: O \rightarrow S^* \times S$ is a function which defines the signature of the operators. These signatures are denoted as $\sigma(o) = s_1 \times s_2 \times \ldots \ldots \times s_n \rightarrow s$, where $s, s_i \in S$, $1 \leq i \leq n$. A Σ-algebra is a tuple $A_\Sigma = \langle \{As\}_{s \in S}, Op(O) \rangle$, where $\{A_s\}_{s \in S}$ is a family of non-empty sets indexed by the states S of Σ, called the carrier sets of the algebra, and $Op(O)$ is a set of operations across the sets in $\{A_s\}_{s \in S}$ such that for each $o \in O$ with signature $\sigma(o) = s_1 \times s_2 \times \ldots \times s_n \rightarrow s$, $Op(o)$ there is a function $Op(o): A_{s1} \times A_{s2} \times \ldots \times A_{sn} \rightarrow A_s$. In the following $Op(o)$ is identified with O. A Σ- algebra is a tuple defined through $L = \langle Sem, Syn, \mathcal{L}: Sem \rightarrow Syn \rangle$ [4], where Sem is a Σ- algebra called the *semantic language*, Syn is a Σ- word or a term algebra called the *syntax language* and $\mathcal{L}$ is a partial mapping called the language learning function.

In order to define a *CTL* model as a Σ-language, shall define an operator scheme Σ_{ctl} as the tuple $\langle S_{ctl}, O_{ctl}, \sigma_{ctl} \rangle$ where the states $S_{ctl} = \{F\}$ are mapping the formulas f [4] and $O_{ctl} = \{true, false, \neg, \wedge, \vee, \rightarrow, AX, EX, AU, EU, EF, AF, EG, AG\}$ [1]. In this paper it imports us to know just until operation, AU and EU. Therefore we show the description in Sem_{ctl} algebra of σ_{ctl} just for AU formulas.

Operator	*Description in Sem_{ctl} algebra*
$\sigma_{ctl}(U): F \times F \rightarrow F$	*if f1,f2* $\in Sem_F$ *then A [f1 U f2]* $\in Sem_F$

Table 1 Operator scheme Σ_{ctl} in Sem_{ctl} algebra

CTL can be defined as the Σ_{ctl}-language given in the form $L_{ctl}=\langle Sem_{ctl}, Syn_{ctl}, \mathcal{L}_{ctl}: Sem_{ctl} \rightarrow Syn_{ctl}\rangle$. Syn_{ctl} is the word algebra of the operator scheme Σ_{ctl} generated by the operations from O_{ctl} and a finite sets of variables, denoting atomic propositions, *AP*. Sem_{ctl} represents *CTL* semantic algebra defined across the satisfied sets of *CTL* formulas for a given model *M*. $\mathcal{L}_{ctl}$ is a mapping which associates satisfiability sets in Sem_{ctl} from the *CTL* expressions in Syn_{ctl} that they satisfy and ε_{ctl} is a homeomorphism that evaluates *CTL* expressions in Syn_{ctl} to their satisfied sets in Sem_{ctl}. Although the rules for forming *CTL* formulas are independent of any model, the signification of the resulting formulas is dependent upon a given model. Thus in the algebraic definition of *CTL*, Syn_{ctl} is independent from any model while Sem_{ctl} is dependent on the given model *M*. The word algebra Syn_{ctl} is unique into homeomorphism in the class of algebras with operator scheme Σ_{ctl}. This has as the carrier set Syn_F, the collection of *CTL* formulas. This is called terms or words, created by the juxtaposition of variables from *AP* and the operator symbols from O_{ctl} according to the rules shown in Table 1.

3.1 Algebraic structure of model

The CTL model checker algorithm maps the CTL formulas in the syntax algebra Syn_{ctl}. In order to understand these mappings in [4] is structured the model $M=\langle S,Rel,P:AP\rightarrow 2^S\rangle$ as a Σ-language whose syntax algebraic contains the sets of the expressions and whose semantics algebra contains the sets of 2^S. The operator scheme for this language is $\Sigma_{sets}=\langle S_{sets}, O_{sets}, \sigma_{sets}\rangle$ where $S_{sets}=\{S,B\}$, *S* is the sort for sets, *B* is the sort for the boolean values, $O_{sets}=\{S, \emptyset, C, \cap, \cup, Imply, Urm_{all}, Urm_{some}, LFP_{all}, LFP_{some}, All_{global}, All_{future}, Exist_{global}, Exist_{future}\}$ and σ_{sets} is shown in Table 1. In [1] we used different symbols to denote the operations, because the operators in Sem_{ctl} operate on sets and the operators from Syn_{ctl} operate on terms. The operators from Sem_{ctl} corresponding to the names $\{true, false, \neg, \wedge, \vee, \rightarrow, AX, EX, AU, EU, EF, AF, EG, AG\}$ in O_{ctl} of Σ_{ctl} are respectively named $\{S, \emptyset, C, \cap, \cup, Imply, Urm_{all}, Urm_{some}, LFP_{all}, LFP_{some}, All_{global}, All_{future}, Exist_{global}, Exist_{future}\}$. The model *M* defined as Σ_{sets}-language $L_M=\langle Sem_{sets}, Syn_{sets}, L_{sets}: Sem_{sets} \rightarrow Syn_{sets}\rangle$, where ε_{sets}: $Syn_{sets} \rightarrow Sem_{sets}$ evaluates set expressions to the sets they represent. In this language, Sem_{sets} is the semantic algebra with the carrier sets $Sem_S = 2^S$ and $Sem_B = \{true, false\}$. The operators in the algebra and their signatures as defined by σ_{sets} are shown in [1]. We retain that semantic Sem_{sets} and Sem_{ctl} have the carrier sets in the relation $Sem_F \subseteq Sem_S$. This allows due to similarity to show in the scheme all elements of carrier sets of Sem_{ctl} through their occurrences in the carrier sets Sem_{sets}.

3.2 Algebraic description of CTL model checker

A *CTL* model checker defined as an algebraic compiler $C:L_{ctl}\rightarrow L_M$ by pair of embedding morphisms $\langle T_C, H_C\rangle$. T_C: $Syn_{ctl} \rightarrow Syn_{sets}$ maps *CTL* formulas from word algebra Syn_{ctl} to set expressions in Syn_{sets}, which evaluate to the satisfiability of sets of the *CTL* formulas, $H_C : Sem_{ctl} \rightarrow Sem_{sets}$, maps sets in Sem_{ctl} by the identity mapping to sets in Sem_{sets} and thus is constructed using the following approach [4,10]:

1. Associate each operation o_{ctl} from algebra Sem_{ctl} with a set expression $d(o_{ctl})$ from algebra Syn_{sets} with the property $H_C(o_{ctl}(s_1, \ldots, s_n))=\varepsilon_{sets}(d(o_{ctl})(d_{ctl}(s_1), \ldots, d_{ctl}(s_n)))$.

$o \in Sem_{ctl}$	$D_{ctl}(o) \in Syn_{ctl}$
$LFP_{all}(t_1,t_2)$	$Z:=\emptyset$; $Z':= d_{ctl}(t_2)$; $Z'':= d_{ctl}(t_1)$; while $(Z \neq Z')$ do $Z:=Z'$; $Z':=Z' \cup (Z'' \cap \{s \in S \mid succ(s) \subseteq Z'\})$; end while $d_{ctl}(LFP_{all}(t_1,t_2)):=Z'$;

Table 2 The construction of *d* over the generators and operations Sem_{ctl} and Syn_{ctl}

2. The set of expressions from the second column of table 2 defines the operations of an algebra Syn'_{sets} which is similar to the operations from Sem_{ctl} and from Syn_{ctl}.
3. Syn'_{sets} is a sub-algebra of Syn_{sets}, the embedding $T_C : Syn_{ctl} \rightarrow Syn'_{sets}$ is constructed by the composition of T'_C and injection function $I : Syn'_{sets} \rightarrow Syn_{sets}$, given by $T_C = T'_C \circ I$.

The morphisms T_C and H_C thus constructed make the diagram commutative. Commutatively assures the fact that T_C keeps the meaning of the formulas from Syn_{ctl} when mapping them to set expressions in Syn_{sets}. The diagonal mapping $Dctl:Sem_{ctl} \rightarrow Syn_{sets}$ is generated by d_{ctl} defined in table 2 and shows the translation process performed by T_C using *derived operations* [4].

4 Algebraic implementation of CTL

Construction of T_C in the Subsection 3.2 can be entered into an algorithm which implements the *CTL* model checker. This algorithm is universal in the sense that being given operator scheme Σ_{ctl} and a model M, the model checker L_{ctl} is automatically generate from the specifications of $\langle \Sigma_{ctl}, D_{ctl} \rangle$. This specification is obtained by associating each operation $o \in O_{ctl}$ with an derived operation $d_{ctl}(o) \in D_{ctl}$. To define derived operations that implement the operations of Σ_{ctl}, from the Syn_{sets} algebra, we use meta-variables that take as values the set expressions of the carrier sets of Syn_{sets}. For each operation $o \in O_{ctl}$ such that $\sigma_{ctl}(o) = s_1 \times \ldots \times s_n \rightarrow s$, $d_{ctl}(o)$ takes as the formal parameters the meta-variables denoted by $@_i$, $1 \leq i \leq 2$, where $@_i$ denotes the set expression associated with i-th argument of $d_{ctl}(o)$; the meta-variable $@_0$ is used to denote the resulting set expression, as example $@_0 = d_{ctl}(o)(@_1, \ldots, @_n)$ [1,4].

The *CTL* rules set which are directly specified in the algebra Sin_{ctl} can be ambiguous, therefore it is necessary that the set F from Sin_{ctl} to be divided in the non-terminal symbols denote with *Ctlformula, Formula, Factor, Termen* and *Expresie*. Thus, the defined rules deliver non-ambiguous specification in algebra Sin_{ctl}.

In following we show the specification of *AU* formula

```
Ctlformula → "(" Ctlformula "au" Ctlformula ")" ;
Macro: sets Z,Z1,Z2;
        Z:=empty_set;Z1:=@3; Z2:= @1;
        while(Z not_equiv Z1) do
          Z:=Z1; Z1:=Z1 union {Z2 intersect {s in all_setS | (succ(s)subset Z1)}};
        endwhile
        @0:=Z1;
```

The *ANTLR 3* [6] tools are used for the construction of software instruments as translators, compilers, recognition and parser of static/dynamic programs. The ANTLR is a generator of compilers; it receives as input a grammar, with a precise description of a language, and generates the source code and other auxiliary files for lexer and parser. The source code of our *ANTLR* grammar presented in paper [1] must contain the specification of all macro-operations presented in the section 3. We "decorate" the grammar for formula language with actions. *ANTLR* inserts our actions in the generated code for parser, and parser must execute embedded actions after matching the preceding grammar element. This is the mechanism of formula evaluation for a given model. In *ANTLRWorks* was implemented all macro-operations of CTL model checker. The program receives as input the model M where are defined the sets S, Rel and P.

The detailed specification of *AU* operation defined in table 2 as *actions* in *ANTLR* grammar are presented in the following:

```
ctlFormula returns [HashSet set]
```

```
@init {System.out.println("Incepe..."); init();}
      :  '(' c1=ctlFormula 'au' c2=ctlFormula ')'
                {   HashSet rez = new HashSet(); //Z:=∅;
                    HashSet rez1 = new HashSet($c2.set); // Z':= dctl(t2);
                    HashSet rez2 = new HashSet($c1.set); // Z'':= dctl(t1);
                    while (!rez.equals(rez1)) {
                     rez.clear();rez.addAll(rez1);
                     HashSet tmp = new HashSet();
                     boolean include;
                     for (int i=0; i<MAX_STARI; i++) {
                      include = true;
                      for (int j=0; j<MAX_STARI; j++)
                       if (rel[i][j]==1)
                        if (!(rez1.contains(new Integer(j))))
                         include = false;
                      if (include) tmp.add(new Integer(i));
                     }
                     tmp.retainAll(rez2); rez1.addAll(tmp);
                    }
                    trace("ctlFormula",1);
                    printSet(" ( " + $c1.text + " au " + $c2.text + " ) ",rez1);
                    $set = rez1;
                }
```

The behavior of the model checker algorithm demonstrated in [1] consists of identifying the sets of states of a model *M* which satisfy each sub formula of a given *CTL* formula *f* and constructing the set of states, from these sets, that satisfy the formula *f*. This is certainly the behavior of the algorithm for the homeomorphism computation performed by an algebraic compiler. Thus is evaluated an expression by repeatedly identifying its sub expressions and replacing them with their images in the target algebra. In the case of the model-checking algorithm, sub expressions are *CTL* sub formulas and their images are the sets of states in the model satisfy the sub formulas.

5 Algebraic specification of Real-Time CTL extension of CTL model checker

Many times we need to specify when an event is necessary to be happened. For this we need a clock can measure the time. The main idea is to add the feasible constraint clock to states and transition.
Formal verification methods have been developed to reason about the correctness of a system with respect to a given specification. In particular, model checking [7] of temporal logics has become one of the most successful verification techniques. Using this technique requires to adequately model a system by a finite state transition system so that specifications given in temporal logics can be checked for that model.

Real-time systems must perform certain actions within limited time bounds or should start actions only after some point of time. It is therefore natural to label the transitions of the abstract transition system by numbers that denote the time required to move from one state to another one. In general, a transition from state s_1 to state s_2 with label $k \in N$ means that at any time x, where we are in state s_1, we can perform an atomic action that requires k units of time. The action terminates at time $x+k$, where we are in state s_2.

The development of discrete real-time extensions of *CTL* has been initiated in [8], where the temporal operators have been extended by time bounds to limit the number of fixpoint iterations

required to evaluate the considered temporal expression. The models used in [8] were still traditional finite-state transition systems where each transition requires a single unit of time. In order to represent real-time systems in a more compact way, [9] introduced timed transition systems, where transitions are labelled by natural numbers that denote the time consumption of the action associated with the transition.

For come to an *CTL* model checking which generate a real time logic is necessary to extend the *CTL* model with an until bounded operator. This operator contains an interval with a lower bound, denoted with *mi*, and an upper bound, denoted with *ms*, of time step number who allow the transition from one *mi* time to an *ms* time until an event to must happen. For this extension of *CTL* model checking is necessary to extend and modify the target language of sets to handle the various temporal logics.

A real time *CTL* model checker, denoted CTL_T is extending via adds a single specification rule. In this case the until operator *U* have attached an interval of time denoted by *[mi,ms]* which represent when the clock begin to measure and when is stop for that event happened.

Be f_1 and f_2 two CTL_T formula holds on a path $\pi = [s_0, s_1, ..., s_i, ... , s_j, ...]$ if f_2 holds on some states s_i, $mi \le i \le ms$ and f_1 holds on all states s_j, $0 \le j \le i$.

Definition 2 *(Syntax Definition of CTL_T model checker) If f_1 and f_2 are CTL_T formula and $mi,ms \in N$ then the syntax of CTL_T can be given in Backus-Naur form:*

$$f :: \top | \bot | p | (\neg f_1) | f_1 \wedge f_2 | f_1 \vee f_2 | f_1 \subset f_2 | AX_{[mi,ms]} f_1 | EX_{[mi,ms]} f_1 | AG_{[mi,ms]} f_1 | EG_{[mi,ms]} f_1 | AF_{[mi,ms]} f_1 | EF_{[mi,ms]} f_1 | A[f_1 U_{[mi,ms]} f_2] | E[f_1 U_{[mi,ms]} f_2] \quad (2)$$

where $\forall p \in AP$.

Definition 3 *(Semantic Definition of CTL_T model checker) Let $M=(S, Rel, P:AP \to 2^S)$ be a Kripke model for CTL_T, and s in S, then the semantics of the logic is recursively defined as follows:*

- $(M, s) \models \top$ and $M, s \not\models \bot$ for all $s \in S$.
- $(M, s) \models p$ iff $p \in P(s)$.
- $(M, s) \models \neg f$ iff $(M, s) \not\models f$.
- $(M, s) \models f_1 \wedge f_2$ iff $(M, s) \models f_1$ and $(M, s) \models f_2$.
- $(M, s) \models EX_{[mi,ms]} f$ iff for some s_1 such *that* $s \to s_1$, $(M, s_1) \models f$.
- $(M, s) \models E[f_1 U_{[mi,ms]} f_2]$ holds iff some a *path* $[s_0, s_1, s_2, ...]$, where $s_0 = s$, and some i with $(mi<i<ms)$ and $s_i \models f_2$ *and all* $j\ [0 \le j < i$ *then* $s_j \models f_1]$.
- $(M, s) \models A[f_1 U_{[mi,ms]} f_2]$ holds iff all *paths* $[s_0, s_1, s_2, ...]$, where $s_0 = s$, and some i with $(mi<i<ms)$ and $s_i \models f_2$ *and all* $j\ [0 \le j < i$ *then* $s_j \models f_1]$.

Algebraically the *CTL* algebra Syn_{CTL} is extend to Syn_{CTL_T} which is a heterogeneous algebra with carrier sets *F* and *N* where *F* is the carrier set of Syn_{CTL_T} formulas and *N* is the carrier set of positive integer constants used in the until bounded operator. The until operation is defined by $U:F \times F \times N \times N \to F$. The construct formulas use the bounded until operator and is represented in the algebraic specification of the model checker as *BNF* rules. The algebraic specifications used to generate model checking implementation for CTL_T are the same with *CTL* with the addition of bound.

In following we show the specification of *AU* formula in CTL_T model checker

```
CtlFormula → "(" Ctlformula "au" "[" mi "," ms "]" Ctlformula ")" ;
Macro: sets Z,Z1; unsigned integer ms,mi,count;
        Z= empty_set; Z_new:= empty_set; Z1:= @2; Z2=@9;
```

```
mi=@5;ms=@7;count=mi;
while ((Z1 not_equiv empty_set) and (count≤ms)) do
  Z:=Z1;
  Z_new:= Z2 intersect {s in all_setS | (succ(s) subset Z1)};
  if(count≥mi and count≤ms)
     Z1:=Z1 union Z_new;
  endif
    count:=count+1;
endwhile
@0:=Z1;
```

Correctness of a system depends in some cases on the exact timing of events. As a consequence, the models must include the time at which events occur. A usually used formalism to model and reason about timed systems is timed automata [11]. In next section we show an example where we used a timed automaton which is an extension of finite state automata that define a set of real-valued clock variables.

5.1. Train Gate Controller Example

In this subsection we construct two figures which represent the description of Train Gate Controller.

Problem description for Train Gate Controller is: Consider a trackage crossing whose physical layout is represented in Fig.1. There have a road crossing a trackage. Trains and cars cross the passageway area in turns. The crossing involves a gate who keeps the barrier up while the train not coming. The trackage have four sensors detecting when a train enters respectively exits the crossing. When the train approach and cross the first sensor the gate begin to close and also the clock begin to measure. The system consists of three main components, the trackage, the road, the controller, and its behaviour.

Based on these sensor signals, a controller should signal the gate to open/close. In following we prove the properties for the system: *When a train is in the crossing, the gate is closed.*

In Fig. 1 we show the trackage who was split in three regions. The first region, denoted by I, represent the process when the closing gate because the train approach. The second region, denoted by II, represents the process when the gate is close and the car waiting the train. The third region, denoted by III, represents the process when the opening gate because train move away. All the three regions contain a *clock* that is start when the train rive over the first red point. In enounce problem we named this red point with sensor. Red Points, represent by  , means that *clock* start for counting. When the clock is start the gate moving down. In our example we choose the time for closing/opening gate to 6 time units and 8 time units for close gate. For all three regions the time is a parameter who can be modified in terms of various situation, e.g. *when the train go fast or slower*. Return to our example the action for first region happens in 6 time units until the train rive over the second red point. The *clock* is bounded with a lower bound, denoted by *mi*, and the upper bound, denoted by *ms*. Because the first region has 6 time units, the clock can measure time $\in R^{\geq 0}$. A clock constraint of form $mi \leq x \leq ms$. That is $mi:=0, count:=0, 0 \leq x \leq 6$. When the count is 6 means the time is up and the gate is close. In the right side of Fig 1 we show how we split the time in three regions.

Definition 4 *The clock are usually written by x,y,..., sets of clock are* $Clock_1$, $Clock_2$, *... A clock-constraint, denoted with Clock_Con(Clock), over clocks Clock is* $g \in Clock\_Con(Clock)$ *where* $g::=x<c|x\leq c|x\geq c|x>c|g\wedge g$ *where* $g \in Clock\_Con(Clock)$ *and* $c \in N$.

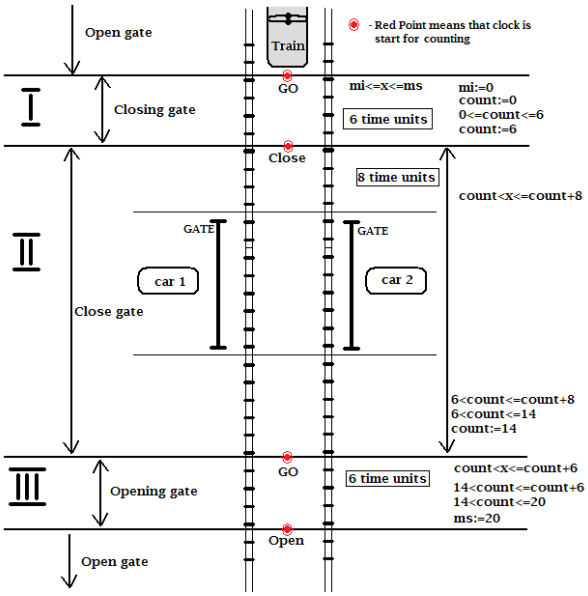


Fig 1. Description time for Train Gate Controller

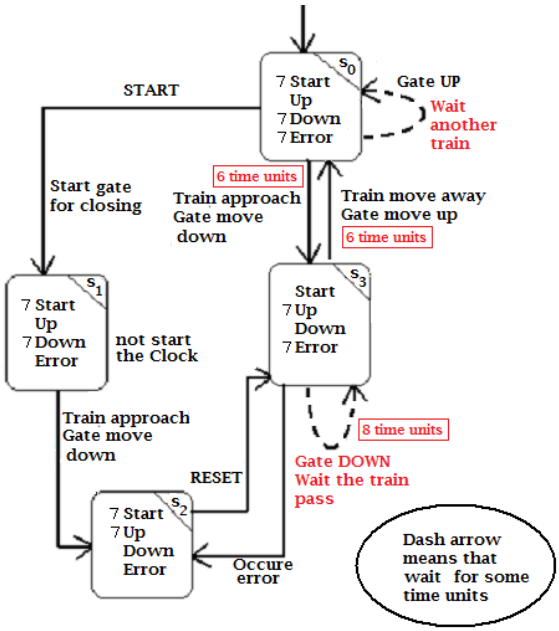


Fig 2. The CTL structure of Train-gate Controller

As example we present in Fig. 2 a scenario for a train gate control system. The state-transition diagram showed in Fig. 2 has four locked-state events. These locked-state events occur because the Gate Train Controller, in most instances, takes one action and then awaits a response before moving for a new state. In fact, only three event flows, *Move Up Request*, *Move Down Request* and *Start the clock* (when we denote with *Up* and *Down* the *move* state when the request exist) do not qualify as locked-state events, because each of them can arrive any time a train and the clock start. The remaining events can occur while the train on-coming to the gate.

Suppose that we have a train gate control which includes in first case a process for normal moving of gate (like, $\{s_0,s_3\}$) and in second case a faulty process (like, $\{s_0,s_1,s_2\}$). In first case for the normal moving of gate process, doesn't shall appear the errors, so the gate is closing and opening normal. The *Clock* is start to counting and cars are shall be stop when the gate is moving or is down. The second process is the faulty process, when the gate doesn't moving when the *Clock* is start to counting. We construct this form of model, to find where the faulty process is, because the objective of model is to correct the event which contains the faulty process. *CTL* structure for the train-gate control is presented in the Fig. 2 and states of the system are denote with $s_0, s_1, \ldots, s_3$.

The Kripke model has four states and the propositional variables are from the set {*Start*, *Up*, *Down*, *Error*}. *Start* represented the *start Clock* when start moving up or down the gate train, *Up* represent the *Up gate*, *Down* is the *Down gate* and *Error* means occur some *error*.

The formal definition of the Kripke structure of the train-gate control is given by: $M = (S, Rel, P)$, where $S=\{s_0,s_1,s_2,s_3\}$, $Rel=\{(s_0,s_0),(s_0,s_1), (s_0,s_3), (s_1,s_2), (s_2,s_3), (s_3,s_0), (s_3,s_2), (s_3,s_3)\}$, $AP=\{Start,Up, Down, Error\}$, P assigns state s_0 in M with *not Start, Up*, *not Down* and *not Error*, that is set $\{\neg Start, Up, \neg Down, \neg Error\}$. P assigns state s_1 in M with $\{\neg Start, Up, \neg Down, Error\}$, the state s_2 in M with $\{\neg Start, \neg Up, Down, Error\}$, the state s_3 in M with $\{Start, \neg Up, Down, \neg Error\}$.

If the path $\pi = s_0 \rightarrow_{\tau 1} s_1 \rightarrow_{\tau 2} s_2 \rightarrow_{\tau 3} s_3 \ldots \rightarrow_{\tau n} s_n$ is a time-divergent compressed path then $\pi \vDash f_1 U_{[mi,ms]} f_2$ if and only if there is some i such that $s_i + d \vDash f_2$ for some $d \in [0,d_i]$ with $d+\sum_{k=0,\ldots,i-1} d_k \in [mi,ms]$ and for all j≤I and all $d' \in [0,d_j]$ such that $d'+\sum_{k=0,\ldots,j-1} d_k \leq d+\sum_{k=0,\ldots,i-1} d_k$ the relation $sj + d' \vDash f_1 \vee f_2$ is valid. This represents the semantics of CTL_T and is inspired from [2]. For our example the path can be write like $\pi = s_0 \rightarrow_{\tau 1} s_3 \rightarrow_{\tau 2} s_3 \rightarrow_{\tau 3} s_0$ where we define the execution like $ExecutionTime(\pi) = \sum_{\tau i \in R}^{\geq 0}$. That is $\pi = s_0 \rightarrow_6 s_3 \rightarrow_8 s_3 \rightarrow_6 s_0 \equiv s_0 \rightarrow_6 s_0 + 6 \rightarrow_a s_3 \rightarrow_8 s_3 + 8 \rightarrow_b s_0 \rightarrow_6 s_0 + 6$.

In our example we have three regions. Here we have three clock constraints consists of atoms x < | ≤ | ≥ | > | c for some $c \in N$ by *definition 4*. Consider clocks x,y,z and regions $Reg=" x \in (0,6] \wedge y \in (6,14] \wedge z \in (14,20]"$ like in Fig 3.

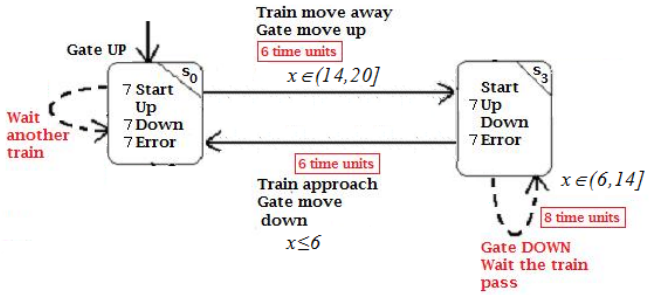


Fig 3. Transition in Region Transition System

Remarks on Region Transition System [2] are $Reg \vDash g$ if and only if for all $\alpha \in Reg$ with $\alpha \vDash g$ if and only if there exists $\alpha \in Reg$ with $\alpha \vDash g$ then there is no ambiguity in the labelling. *Clock* constraints of f CTL_T formula become atomic propositions in Region Transition System, RTS(TA,f), where TA is the *timed automata* defined in [2] and $TA \vDash f$ represent the semantics CTL_T.

Return to our model we check up the following properties: *Is the gate closed for less then 8 minute?* The formula is, $TA \vDash A((Up \wedge \neg Error)\ U_{[7,14]}\ (Down \wedge \neg Error))$.

We showed at first of this section AU formula specification in CTL_T model checker. Beginning of this the $A((Up \wedge \neg Error)\ U_{[7,14]}\ (Down \wedge \neg Error))$ formula executing like bellow:

We initialize all sets with $Z := empty\_set;\ Z_{new} := empty\_set;\ Z1 := @2;\ Z2 = @9;$ where $Z1$ is set with state $Up \wedge \neg Error$. This state is $\{s_0\}$. $Z2$ is set with state $Down \wedge \neg Error$. This state is $\{s_3\}$. The Z_{new} set is constructing with all state s from all_setS which have the successor in $Z1$ and intersect with $Z2$ set. We initialize all positive integers with $mi=@5, ms=@7, count=mi$. That is $mi=7; ms=14; count=mi;$

```
while (({s0} ≠ ∅) and (7≤14)) do
  Z:=Z1={s0};
  Znew:= {s3}∩{s0,s3}={s3};
  if(7≥7 and 7≤14)
     Z1:={s0}∪{s3}={ s0,s3};
  endif
   count:=7+1=8;
  Return to while in next step
       while (({s0,s3} ≠ ∅) and (8≤14)) do
       Z:=Z1={s0,s3};
       Znew:= {s3}∩{s0,s2,s3}={s3};
       if(8≥7 and 8≤14)
          Z1:= {s0,s3}∪{s3}={ s0,s3};
       endif
```

count:=8+1=9;
Return to while in next step until count:=14

Fig 4. Interpretation step by step for AU formula in CTL_T model checker

If count is between *mi* and *ms* bound of until operator these nodes are added to the set of states satisfying formula. The loop terminates when no new nodes are added or the number of steps exceeds the upper limit. The correctness of implementation is given in Fig. 4 for our proposed model.

6 Conclusion

The behaviour of the real time model checker algorithm demonstrated in the section 5 consists of identifying the sets of states of a model M which satisfy each sub formula of a given CTL_T formula f and constructing the set of states, from these sets, that satisfy the formula f over the bound. This is certainly the behaviour of the algorithm for the homeomorphism computation performed by an algebraic compiler. Thus is evaluated an expression by repeatedly identifying its sub expressions and replacing them with their images in the target algebra. In the case of the real time model-checking algorithm, sub expressions are CTL_T sub formulas and their images are the sets of states in the model satisfy the sub formulas.

References

[1] Laura Cacovean, Florin Stoica, *Algebraic Specification Implementation for CTL Model Checker Using ANTLR Tools*, 2008 WSEAS International Conferences, Computers and Simulation in Modern Science - Volume II, Bucharest, Romania, Nov. 2008, ISSN: 1790-5117, ISBN: 978-960-474-032-1

[2] http://www-i2.informatik.rwth-aachen.de/i2/fileadmin/user_upload/documents/AMC09/amc_lec15.pdf

[3] M. Huth and M. Ryan, Logic in Computer Science: *Modelling and Reasoning about Systems*, Cambridge University Press, 2000.

[4] E. Van Wyk, Specification Languages in Algebraic Compilers, Theoretical Computer Science, 231(3):351--385, 2003

[5] P.J. Higgins. *Algebras with scheme of operators*. Mathematische Nachrichten, No.27, 1963/64

[6] Terence Parr , *The Definitive ANTLR Reference, Building Domain-Specific Languages*, version: 2007

[7] E. Clarke and E. Emerson. Design and Synthesis of Synchronization Skeletons using Branching Time Temporal Logic. In D. Kozen, editor, Workshop on Logics of Programs, volume 131 of Lecture Notes in Computer Science, 1981.

[8] E. Emerson, A. Mok, A. Sistla, and J. Srinivasan. Quantitative Temporal Reasoning. Journal of Real-Time Systems, 1992.

[9] S. Campos and E. Clarke. *Real-Time Symbolic Model Checking for Discrete Time Models*. Theories and Experiences for Real-Time System Development, AMAST Series in Computing.World Scientific Press, AMAST Series in Computing, 1994.

[10] E.M. Clarke, E.A. Emerson, and A.P. Sistla. *Automatic verifications of finite-state concurrent systems using temporal logic specifications*. ACM Transactions on Programming Languages and Systems, Vol. 8, No. 2, April 1986

[11] R. Alur and D. L. Dill. *A theory of timed automata*. Theoretical Computer Science, 126(2):183–235, 25 April 1994.

LAURA F. CACOVEAN
Lucian Blaga University of Sibiu, Faculty of Sciences
Department of Computer Science
Str. Dr. Ion Ratiu 5-7, 550012, Sibiu
ROMANIA
laura.cacovean@ulbsibiu.ro

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

SGIA – Self Growing Intelligent Agent

Laurențiu Ciovică, Liviu Ciovică

Abstract

We live in a world where the use of agents is getting bigger by day. Each intelligent agent has a list of basic characteristics that must be followed and respected. The aim of this article is to study and create new techniques for an agent to have one more characteristic: that to create, by itself, new services and new functionalities in order to automatically adapt to new challenges or environment changes. In other words, we want to give to an agent the power to create code that will helps it in his purpose – the power of course is restricted by well defined rules.

1. Introduction

Nowadays, the computerized machines / tools are used almost everywhere: from our digital clock to airplanes and cars; they are helping us in doing our job faster, better and cheaper and in same cases tasks that humans are not yet capable in doing them.

After these things that were achieved, the mankind has a bigger idea, a bigger ambition, to make machines (computers) to think for us – an idea that raises many ethical and philosophical questions.

"Artificial Intelligence is a combination of cognitive science, linguistics, ontology, physiology, psychology, philosophy, operations research, economics, control theory, neuroscience, computer science, probability, optimization and logic. AI is a very large subject-matter. It consists of many different fields, from machine vision to expert systems.

The aim of all the fields is the creation of machines that can "think". Researches hope that AI machines will be capable of reasoning, knowledge, learning, communication, planning, perception and the ability to move and manipulate objects."[1]

Every human / machine process fallows a standard or a non-standard flow. We can design for each process a well defined, standard, flow chart.

Our aim, within this article, is to create this new agent characteristic – that through which the agent can grow by auto generating source code and compile it in new modules – based on flowcharts (logical schemes).

Continued on article you will find sections, in the given order, describing: an intelligent agent, the SGIA concept, the Logical Schema Builder (identified also through LSB), the proposed learning, growing, forgetting processes, some security constrains and the conclusions taken after this first stage of research in artificial intelligence field.

2. State of the art

The field of intelligent agents is a vast one and a lot of research was done. Still, what we intend to research and develop is not yet, as far we now, published. This new characteristics is to be build upon the core of a previous developed tool called Logical Schema Builder. This tool is original through idea, design and implementation. The tool core is responsible in translating flowcharts into source code.

So this approach, based on flowcharts, is new in the field.

3. Intelligent Agent

As the science and world expands the amount of information is bigger for humans to handle. A solution for this is to build intelligent systems that can find, filter and process information and take care of a certain tasks for us.

For example, you want to keep track with all the article that appear in different web magazines but you do not have the time necessary to navigate on web to grab the news and read them. Instead of you searching them you can use an agent that search on the web for you favourite magazines and grabs the latest news / articles and present them to you in one place faster than you do.

There some key characteristics of an agent: autonomy, persistence, the ability to interact with its environment, cooperation and learning, communication, pro-activeness and reactivity. Others researchers in the field would disagree these characteristics and/or can prioritize others.

"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. A human agent has eyes, ears, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors. A robotic agent substitutes cameras and infrared range finders for the sensors and various motors for the effectors. A software agent has encoded bit strings as its percepts and actions."[2]

4. SGIA – Self Growing Intelligent Agent Concept

SGIA – Self Growing Intelligent Agent – wants to be an intelligent agent that besides its fundamental characteristics has another important one: the capacity to grow and evolve form a small to complex online entity.

The process of growing wants to be more human like, meaning that the agent will have, besides the common agent's functionalities, the next functionalities from which some of them are common to all agents and other not:

- capability to learn by observing, learning and experiencing;
- capability to adapt to different environments;
- autonomy and persistence over time and space;
- sociability – will be able to interact with other agents and communicate with them even if they spoke a different "language"
- capability to literally grow in size and knowledge, meaning: capability to develop new skills and ways to lean and interact with different agents and environment.

5. The Logical Schema Builder

The Logical Schema Builder is a software tool through which flowcharts (logical schemes) can be designed, analyzed and translated to source code (for now only to Java or C# is supported).

Through this tool you can perform the next operations:
- to design flowcharts through the specialized graphical editor
- to transform from flowcharts into intermediary code and validate it
- to transform from intermediary code into virtual machine scripting language
- to compile the generated virtual machine scripting code
- to generate source code – Java, C# - from scripting code

The LSB tool is original by design, idea and implementation. Out there on the market are only translators from source code to flowcharts but not vice versa.

These designed flowcharts can be stored either as Xml files or into specialized tables.

By using the current LSB version, you can define all the statements defined within a method. For the next version the tool is to be extended in order to support also the definition of classes (a UML like approach).

Also the LSB scripting language is to be extended in order to create a standard common growing language.

6. The growing and learning process

The growing process is a time-consuming and slower process. A person can grow in knowledge and in size. We saw, in the previous section, that also a machine can grow in knowledge through different, most human-like, ways of learning; but are a little process in growing in size.

Definition 1 *The growing process is a process through which an intelligent agent can develop new services, new functionalities by itself, for it to have a better reactivity to the environment.*

For example: an agent is first developed to do a certain functionality, let say to read newspapers for you and bring you only those articles in which you are interested. But in time you need more from you agent, to read also blog entries. So, for this new functionality you have to define and implement new logic, new processes into that agent.

We know that an agent can learn and take decisions using different techniques and algorithms. My study is to research and find the better technique to be applied to an intelligent agent for it to learn to create – design, implement – different modules in order to support a wide area of functionalities.

By adding also this capability, we have then an intelligent agent that can think and take some decisions, growing both in knowledge and in "size" – in functionalities.

We can design any process into a flowchart. (i.e. the common coffee maker flowchart).

The SGIA will have a specialized module through which any process is to be defined, step by step, into flowcharts. These flowcharts are to be stored into a huge knowledge repository.

The SGIA will also use the common learning approaches but will try to translate the acquired knowledge, if are processes or actions, into flowcharts.

After the processes flowcharts are defined, the LSB (Logical Schema Builder) core is to be used in order to generate the source code in order to materialize that process and to create support for it.

The optimized LSB core will read, interpret and process any defined knowledge synapses. These synapses are to be created between two or many flowcharts that are used to describe a complex process.

Besides this module, the SGIA will have also templates which will facilitate the support generation and creation of new functionalities.

7. The forgetting process

As we described in the previous sections, the growing and learning process will form a huge knowledge repository.

In order to optimize the memory consummation we introduce the forgetting process.

As human forget the information that is no longer used for a period of time also the SGIA will "forget" the new gained functionalities if they are no longer used more than X% during Y days.

The forgetting process is defined as:

- An SGIA will forget the gained functionalities by disabling the functionality support module, remove it from memory, archive it and creating a reference to it.

Those references shall be used, by the agent if, and only if, an old functionality is needed and the module must be recalled.

This "recall" mechanism will reduce the time necessary for loading the new functionality.

8. Security constrains

Security constrains are to be taken in order to limit the growing process and knowledge repository access.

The knowledge repository access is to be made upon a SGIA-P (SGIA Passport). The passport shall be formed from a unique identifier, IP address of the agent host plus others attributes.

The growing security constrains are to be defined first by the creator and secondly by the end-user through a specialize rules.

The search about these rules is only at an incipient phase so for now we can not specify well-defined rules.

9. Conclusions

The SGIA – Self Growing Intelligent Agent – is a new approach on how intelligent agents can be more adaptable to environment changes. Through more adaptable we understand the fact that besides well defined learn techniques the agent should have also other characteristics: that to write (develop, implement) itself services / functionalities for a more reactivity and for growing not only in knowledge and also in skills.

The learning and growing process is based upon the LSB (Logical Schema Builder) core which will be extended in order to support the future defined common growing language.

The services and new functionalities are to be generated by the LSB core respecting well-defined templates. If there are no templates defined the agent shall try to generate support for the new functionality based on it's previous experience and after given a high level of trust it can standardize that approach and create a template and publish it to the knowledge repository for others to use.

The next steps of research is to standardize the common growing language, to extend the LSB core for support the language and standardize also the growing, learning and forgetting processes.

References

[1] P.V. Raja Shekar & CH. Madhuri – *Intelligent Agent*

[2] Artificial Intelligence - A Modern Approach - Stuart J. Russell and Peter Norvig

[3] Adrian A. Hopgood, *Intelligent Systems for Engineers and Scientists,* CRC Press, Boca Raton London New York Washington, D.C.

[4] Langley, P., *Elements of Machine Learning*, Morgan Kaufmann, 1995.

[5] Watson, I. D., *Applying Case-Based Reasoning: techniques for enterprise systems*, Morgan Kaufmann, 1997.

[6] Article "Logical Schema Builder" - Laurentiu Ciovica, Liviu Ciovica, Gabriel Sofonea - published at the National Scientific Research Student's Competition, Alba Iulia, 2009 with **ISSN 1583-6088** by Aeternitas Publisher.

Laurentiu Ciovica
University Lucian Blaga of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
E-mail: ciovica_laurentiu@yahoo.com

Liviu Ciovica
University Lucian Blaga of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
E-mail: ciovica_liviu@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

iAsistent in Differential Diagnostic

Liviu Ciovica, Laurentiu Ciovica

Abstract

Nowadays, when the world population is growing and when the number of diseases is increasing the need for good doctors is bigger than the last two-three years. Time! The time factor is the most important thing when dealing with human life and their chance to survival. Knowing these needs the author of this paper imagined and starts creating a software tool capable in decreasing the time consumed in differential-diagnostic process, in finding the patient medical history and in searching medical relevant articles and information. Within this system a virtual entity, an intelligent virtual assistant, is to be used in order to help the physician in differential diagnostic process.

Introduction

Medical field – a field in which things fallow an exponential growth. New diseases or mutations of existing ones are discovered every day. In order to keep track with this growth and mutations we have to be up to date with all. In order to facilitate this process and to save time, the iMDD (Intelligent Medical Differential Diagnostic – identified, in article, also through iMDD) system was designed and implemented.

A short description of the iMDD system will be: iMDD is a collaborative platform through which the physicians can easily interact one with each other, without any spatial frontiers. The platform is like a central point of medical knowledge and a tool that facilitates the medical information access and the differential diagnostic process through collaborative tools (mails, chat, whiteboard, etc).

As part of iMDD design is a virtual, intelligent, entity – an assistant – that will help the physician in the diagnostic process. The functionality and the design of this assistant (identified in article also through i.A – intelligent Assistant) is to be presented further on article.

Continued on the article you will find information about the medical context that helped in designing and implementing the system, a summary description about the system itself, the role of artificial intelligence in medicine, the iAssistant structure and functionalities.

Medical Context

In common modern usage, diagnosis is the identification of a disease by investigation of its signs, symptoms and other manifestations. In medical terminology, however, the precise meaning of the word is obscured by the many ways in which it is used: clinical diagnosis, laboratory diagnosis, physical diagnosis, anatomical diagnosis, bacteriological diagnosis, x-ray diagnosis, electrocardiography diagnosis. In each of these cases the word diagnosis connotes an appreciation of

the meaning of observations made by particular methods, but it does not necessarily signify the identification of a disease by these means.

Procedures involved in diagnosis

Diagnosis involves two procedures: 1) collecting the facts and 2) analyzing the facts. Errors in diagnosis may be due to imperfect performance of either of these procedures. When the factual data are inadequate or incorrect, or when they have been misinterpreted, the analysis, though faultless in itself, will lead to an erroneous conclusion. On the other hand, even when the collections of facts has been complete and accurate, and the data have been correctly interpreted, the conclusion may be in error because of faulty analysis.

The diagnostic process is frequently very complex. The number of facts that can be collected in a detailed medical history and in a thorough physical examination is almost limitless. The laboratory tests, x-ray studies and specialized technical procedures that can be employed are numerous and costly. Therefore every question that physician asks in obtaining the medical history, every maneuver that he carries out in the physical examination, and every laboratory test or other procedure that he orders should be thoughtfully selected with a view to eliminating some of the possible diagnoses and guiding the search expeditiously to ward a specific disease. Unselective ordering of tests – the blunderbuss approach – is often confusing, costly, and redundant and is an inadequate substitute for sequential diagnostic analysis; it is to be condemned unless the patient is so ill that time will not permit orderly diagnostic progress.

Steps in Diagnosis

In brief, the successive steps leading toward the diagnosis are the following:

1. Collecting the facts:
   - Clinical history
   - Physical examination
   - Ancillary examinations
   - Observations of the course of the illness

2. Analyzing the facts:
   - Critically evaluate the collected data
   - List reliable findings in order of apparent importance
   - Select one or preferably two or three central features
   - List diseases in which these central features are encountered
   - Reach final diagnosis by selecting from the listed diseases either the single disease that best explains all the facts or if this is not possible, several diseases, each of which best explains some facts.
   - Review all the evidence – both positive and negative – with the final diagnosis in mind.

Through the iMDD system components and tools, some of these steps are sustained and supported in an easy, intuitive and friendly user interface.

In the next article section, the steps that are supported and assisted by the system are to be described.

Collecting the facts

The facts used in differential diagnosis come from four sources:

1. Clinical history
   - supported by the iMDD system with the help of an integrated EMR tool (Electronic Medical Record) for patients and with the help of medical knowledge database for diseases.
2. Physical examination
   - the examination is to be assisted, in real time, by the intelligent virtual entity, iA.
3. Ancillary examinations involving special techniques and laboratory methods
   - these methods will be also sustained and assisted by the iA and by the system itself. The system will be providing all the necessary information regarding the methods and techniques.
4. Observations of the course of the illness
   - the iA will keep track of the course of the illness by reading, interpreting and storing history of the information provided by the live sustain devices.

Interpreting and Analyzing the Facts

The physician who is called upon to analyze the facts may find himself in a position of a drama critic who is asked to render an opinion about a play after having been permitted to see only the third of four facts. He may be able to learn something about the first two acts from other relatively unobservant and technically inexperience members of the audience who actually saw those acts. The fourth act however has yet to be performed. The wise critic would of course, withhold his opinion unless he was permitted to see play through from beginning to end. The physician, however, despite his limited view, may not arbitrarily withhold his opinion. He is dealing with matters of life and death, and he has been trained to render his opinion at any stage of the play.

The physician must therefore begin his analysis of the facts not only by determining their reliability and intrinsic significance but by considering their relations to the patient's total illness. For example, a negative agglutination test for a specific organism may be of no significance early in the illness but assumes great significance at a later stage. Critical evaluation of various physical findings and laboratory tests requires knowledge of the natural history of diseases as well as of the manifestations that are to be evaluated. From the evidence that he has observed, the physician selects the important manifestations and records them as the symptoms and signs of the diseases. These are the raw data to be used in the subsequent clinical analysis. Before the physician begins his analysis, he must decide which facts to consider, how to designate them, how to verify them, and in what form to record them. These are the steps that convert bedside observations into useful clinical data. Objectivity and consistency must be exercised in each of these steps.

iMDD – the system

The iMDD is:

- an accurate and time saving differential diagnosis tool that reminds you instantly of diagnostic possibilities and minimizes medical errors;
- designed to make your practice safer, enhance the quality of care, reduce misdiagnosis and billing errors, and save time - especially in diagnosing complicated cases;
- the ultimate differential diagnosis tool for a busy physician.

iMDD will provide

- a real time assistance in differential diagnostic process
- full and complete patient medical history
- a mature medical customized search engine
- virtual whitetable for easy, real time, brainstorming processes
- disease history, symptoms and possible treatments
- capability in self patient monitoring and condition notifications

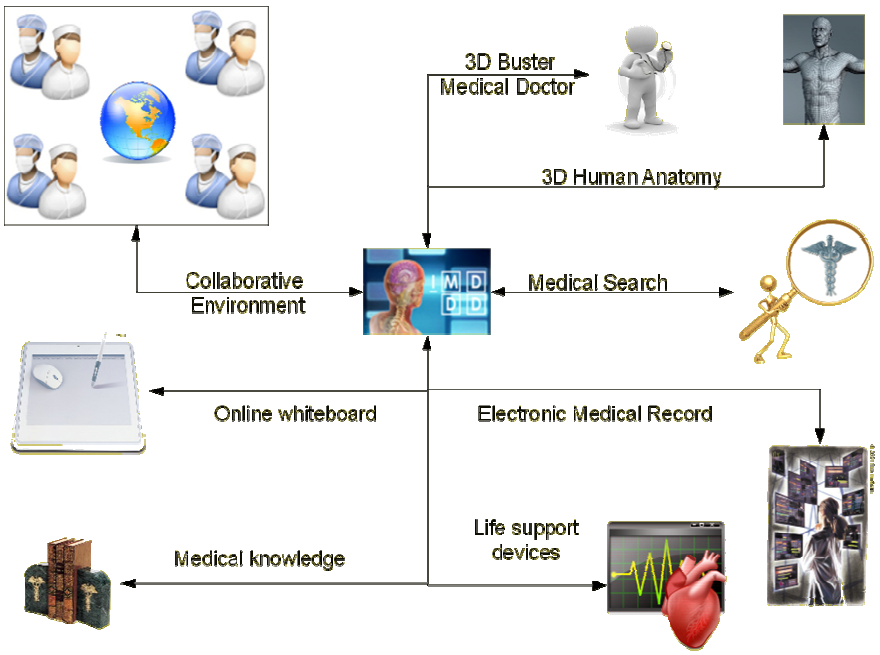


Fig. 1 – iMDD System - Components

Artificial intelligence in medicine

Diagnostic run against the clock is the most important thing to remember in the diagnostically process and e very well knows fact by the physician. The AI technology give a hand of help to the modern physician who is confronting with a growing amount of work and patients and the only think he doesn't have is the precious time.

Artificial intelligence in medicine is a new research area that combines sophisticated representational and computing techniques with the insights of expert physicians to produce tools for improving health care.

Medicine is a field in which technology is much needed. Increasing expectations of the highest quality health care and the rapid growth of ever more detailed medical knowledge leave the physician without adequate time to devote to each case and struggling to keep up with the newest developments in his field.

Due to lack of time, most medical decisions must be based on rapid judgments of the case relying on the physician's unaided memory. Only in rare situations can a literature search or other extended investigation be undertaken to assure the doctor (and the patient) that the latest knowledge is brought to bear on any particular case.

We view computers as an intellectual, deductive instrument, which can be integrated into the structure of the medical care system. The idea that these machines can replace the many traditional activities of the physician is probably. Advocators for artificial intelligence research envisions that physicians and the computer will engage in frequent dialogue, the computer continuously taking note of history, physical findings, laboratory data, and the like, alerting the physician to the most probable diagnoses and suggesting the appropriate, safest course of action

Machine learning systems can be used to develop the knowledge bases used by expert systems or intelligent agents. Given a set of clinical cases that act as examples, a machine learning system can produce a systematic description of those clinical features that uniquely characterize the clinical conditions. This knowledge can be expressed in the form of simple rules, or often as a decision tree.

Some systems require the existence of an electronic medical record system to supply their data, and most institutions and practices do not yet have all their working data available electronically. Others suffer from poor human interface design and so do not get used even if they are of benefit. Much of the reluctance to use systems simply arose because expert systems did not fit naturally into the process of care, and as a result using them required additional effort from already busy individuals. Computer illiteracy of healthcare workers is also a problem with artificial intelligent systems. If a system is perceived as beneficial to those using it, then it will be used. If not, it will probably be rejected.

Expert systems can apply to different clinical tasks, such as:

Generating alerts and reminders. In so-called real-time situations, an expert system attached to a monitor can warn of changes in a patient's condition. In less acute circumstances, it might scan laboratory test results or drug orders and send reminders or warnings through an e-mail system.

Diagnostic assistance. When a patient's case is complex, rare or the person making the diagnosis is simply inexperienced, an expert system can help come up with likely diagnoses based on patient data.

Therapy critiquing and planning. Systems can either look for inconsistencies, errors and omissions in an existing treatment plan, or can be used to formulate a treatment based upon a patient's specific condition and accepted treatment guidelines.

Agents for information retrieval. Software 'agents' can be sent to search for and retrieve information, for example on the Internet that is considered relevant to a particular problem. The agent contains knowledge about its user's preferences and needs, and may also need to have medical knowledge to be able to assess the importance and utility of what it finds.

Image recognition and interpretation. Many medical images can now be automatically interpreted, from plane X-rays through to more complex images like angiograms, CT and MRI scans. This is of value in mass-screenings, for example, when the system can flag potentially abnormal images for detailed human attention.

iAssistant – Intelligent Assistant

iA is an rational agent, an virtual entity – materialized with a 3D human body, that will assist, in real time, the physician in differential diagnostic processes. Being a part of the iMDD system, the assistant will also interact and access all the tools, knowledge repositories from it.

In order to fulfill the decision assistance the iA will use a decision analysis expert system.

The decision analysis process “involves a careful study of the possible actions and outcomes as well as the preferences placed on each outcome**.** It is traditional in decision analysis to talk about two roles: the decision maker states preferences between outcomes, and the decision analyst enumerates the possible actions and outcomes and elicits preferences from the decision maker to determine the best course of action.” [4] (page 509)

The decision analysis expert system must:
- "Determine the scope of the problem"
- "Lay out the topology"
- "Assign probabilities"
- "Assign utilities"
- "Enter available evidence"
- "Evaluate the diagram"
- "Obtain new evidence"
- "Perform sensitivity analysis" [4] (page 510).

The assistant, which is a rational agent, will have as environment the hospital, the patient and the web (for medical information searching) and as goal the patient health and to minimize the costs and time consumed in diagnostic process.

The iA will learn how to read and analyze any medical relevant images, like X-Rays or MRI, and the condition given by the life keeping devices. Also it will percepts the patient symptoms, answers and findings.

By accessing the specialize medical search engine, the iA will know to search for any relevant information regarding a given case.

Being sustained by the decision analysis expert system, the assistant will generate questions, indicate tests and treatments. Besides this, the system will generate alerts for patient critical conditions.

The assistant will interact with the physician through speech recognition and narration characteristics. So, in other words, any produced information will be narrated while displayed on the screen. The speech recognition module it is used for vocal commands; this characteristic is to be added in order to eliminate the time spend for typewriting and the pc dependencies.

Conclusions

The iA – intelligent Assistant – is a rational agent that will help and assist the physician during the differential diagnostic process. Being an agent, the iA will have all the common agent functionalities and characteristics.

The agent will be materialized with a 3D human body, with speech recognition and narration capabilities, and integrated with the iMDD system.

The iMDD system is a collaborative platform which provides a central point of medical knowledge and supports the differential diagnostic process.

The next steps are to design and build the 3D model, establish the agent learning techniques, to implement a decision analysis system – or to integrate an existing one, and to implement the agent itself.

References :

1. Tim Morris (2004). *Computer Vision and Image Processing*. Palgrave Macmillan.
2. Wilhelm Burger and Mark J. Burge (2007). *Digital Image Processing: An Algorithmic Approach Using Java*. Springer.
3. Ethem Alpaydın (2004) *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*, MIT Press.
4. Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, NJ: Prentice Hall.
5. Luger, George; Stubblefield, William (2004). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (5th Ed.). The Benjamin/Cummings Publishing Company, Inc.
6. Robb, RA (1999). *Biomedical Imaging, Visualization, and Analysis*. John Wiley & Sons, Inc.

Liviu Ciovica
University Lucian Blaga of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
E-mail: ciovica_liviu@yahoo.com

Laurentiu Ciovica
University Lucian Blaga of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
E-mail: ciovica_laurentiu@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

AGILE ENTERPRISE THAT SENSE THE MARKET WITH OPINION MINING ALGORITHM

Domenico Consoli

Abstract

Nowadays, enterprise is open and flexible in the use of technological tools to "sense" customers and market. Acquiring information in real-time allows the company to be agile and to develop "Sense and Response" capabilities. An agile enterprise respond immediately to any internal or external event as customer demand or customer opinions. Knowing what the customer thinks of a given product/service helps top management to introduce improvements in processes and products. Customer opinions are very important for enterprise and represent a potential of knowledge so high that it can be considered a true strategic asset for the acquisition of competitive advantages. In our paper we present an original opinion mining algorithm to polarize customer opinions.This algorithm that we have developed focuses on emotions expressed in opinion text.

1 Introduction

Nowadays, in the global market, an enterprise, for the acquisition of competitive advantages, needs of an agile structure to adapt continually to market changes. Agility [1] means rapid response, high flexibility, thin structure, high speed transmission of information. The company must know how to react quickly to situations that are not repetitive. The agile enterprise transmits useful information to decision and business makers so they can quickly identify and resolve problems.

Agile enterprise hears customers to understand the level of satisfaction and behaves accordingly. The dysfunctions about product/service can be sent to competent offices for the necessary improvements [2]. Nowadays, the enterprise can collects and analyses customer opinions by web 2.0 tools and instruments of social networks (forum, chat, blog, wiki). There are various web sites that collect and make free available customer reviews [3]: epinions.com, cnet.com, complaints. com, planetfeedback.com, ecomplaints.com, ciao.it, dooyoo.it . The coming of web 2.0 promoted the birth of a sharing business philosophy and stimulated conversations and exchange among people [4]. The enterprises often encourage exchange of opinions, by making available virtual communities, e.g. Italian Nikon's Camera forum, where people review Nikon products (http://www.nital.it/forum/), the blog on Benetton products (http://benettontalk.com), and so on.

For obtaining an enterprise add value it is important to process and polarize automatically the opinion as positive, negative or neutral. The original algorithm and software that we have developed to polarize customer opinions mainly focuses on six Ekman emotional indexes [5]. We use these indexes because, according to us, each textual opinion expresses sentiments and emotions.

In both processes, consumption and purchase, the satisfaction of emotional needs is searched. The consumer shift its purchase behavior from needs to emotions/experiences. Many times the emotional/symbolic traits are highly representative of the specific identity and brand. For example, in luxury goods, the emotional aspects as brand, uniqueness and prestige for purchasing decisions, are more important than rational aspects such as technical, functional or price.

Another factor that influence purchasing customer is, for example, the disgust that plays a key role in the inverse relationship between attitude and intention to purchase. Customer don't buy products disgusting. The disgust is a repugnance toward any object, action or person. The disgusting is an index of variation of the intention to purchase.

In our algorithm we measure words and statements affectivity and polarity. The polarity(p) depends from affectivity (a) of words: $p = f(a)$. For the polarity we take in consideration the sign and the amplitude. With sign we observe if the customer is satisfied (+) or unsatisfied (-). If the customer is very dissatisfied (amplitude) the enterprise must intervene with urgency and priority. It is interesting for enterprise to know also the statement affectivity for following strategic moves: *Fear*(special promotional campaigns for closing the customer), *Anger*(reassure and accompany customer in post-sale paths), *Sadness*(gladden customer with unique gadgets), *Disgust*(improve immediately the product design).

This paper is organized as follows: in the next section we present related works for opinion mining. The sections, from third to seventh, are devoted to present our methodology and approach to customers opinions polarization. In the eighth section we give a description of results in case study. Finally some conclusions are drawn.

2 Related works

In the literature, for polarizing opinions, various methods of opinion mining and sentiment analysis [6] have been proposed.

In the Keyword Spotting [7] approach, text is classified into affective categories based on the presence of fairly unambiguous affective words like distressed, happiness and anger. All terms that describe emotional states represent the most direct way to communicate emotion by text. The simplest and most used analysis is based on the search for keywords (like happy, sad, angry, etc). This is at the basis of the Elliot's Affective Reasoner [8] that watches for almost 200 affect keywords and intensity modifiers (e.g. extremely, somewhat, mildly).

The Lexical Affinity [9] method assigns to words a probabilistic affinity (trained from linguistica corpora) for a particular emotion. For example, "accident" might be assigned a 80% probability of being indicating a negative affect, as in "car accident", "hurt by accident". In this case there are two types of problems. First, lexical affinity, operating solely on the word-level, can easily be tricked by negative sentences like "He avoided an accident" and others word senses like "She met her boyfriend by accident". Second, lexical affinity probabilities are often biased toward a text of a particular genre, derived by the specific source of the linguistic corpora.

In the Statistical Method [10] by feeding a machine learning algorithm with a large affective training corpus, it is possible for the system to learn the affective valence of keywords (like keyword spotting) and take into account the valence of other arbitrary keywords (like lexical affinity) and co-occurrence frequency of words. We can do an analysis on lists containing so many emotional adjectives and after, with appropriate statistical techniques, reduce these lists on a shortlist of latent variables or factors. Statistical methods such as latent semantic analysis (LSA) have been more used for affective text classification. This method has an acceptable accuracy with a sufficiently large text in input.

Esuli and Sebastiani [11] have created SentiWordNet, a lexical resource for opinion mining, where they assign to each synset (set of synonyms) of WordNet a sentiment scores: positivity, negativity and objectivity (i.e. neutral). The opinion is positive if the positivity of its terms is higher than negative and objective scores.

WordNet Affect [12] is a linguistic resource for a lexical representation of affective knowledge. In WordNet Affect each synset of WordNet is labeled by one or more affective-labels, representing the affective meaning of the synset. Examples of affective-labels are emotion, mood, trait, cognitive state, physical state, etc...

Paul Ekman [5], in the study of facial expressions and emotions, proposed six universal emotions: happiness, disgust, sadness, anger, surprise, fear. All human beings respond with the same facial movements to the same emotional states. We perceive and interpret certain facial movements as expressing distinct emotions.

In the same way, in our algorithm, we thing that there is a connection with sentiments, emotions, affectivity present in the text (words and sentences) and a polarity of customer opinion.

3 The methodology to polarize customer opinions

The goal of our approach is to polarize customer opinions about a topic, that is a characteristic of a (part of) product/service.

The proposed methodology is formed by the following steps:

- Preprocessing
- affective annotations
- Affective vectors computing
- Polarization estimation

4 Preprocessing

Since opinions are written in Natural Language, to process them, we need specific pre-processing techniques and in particular to this end we have used the General Architecture for Text Engineering (GATE) library. The goal of this phase is to obtain statements and significant words for each opinion. The preprocessing consists of the following steps:

4.0.1 Sentence extraction.

From every post we extract minimum sentences. In this step we eliminate all interrogative clauses. These clauses don't carry affective information.

4.0.2 Statement extraction.

The goal of this case is to divide the sentence in statements. A statement is an elementary sub-sentence that expresses a single positive, neutral or negative polarity. A single sentence can express more than one opinion. For example the sentence "The restaurant is beautiful but the waiters are ungracious" may be splits into 2 statements with different polarity: "The restaurant is beautiful" (positive polarity), "but the waiters are ungracious "(negative polarity). To divide sentences in statements it is necessary to separate the words in the proximity of those conjunctions that link two propositions with opposite polarity; for example "but" (coordinative conjunction) or "although, even, thus, whereas, while" (subordinate conjunctions).

4.0.3 Tokenization.

In this stage various statements are divided into units called tokens where each token is a word or a number, a punctuation mark, a date, etc... The token boundary is represented by a whitespace (space, tab or beginning of line).

4.0.4 Stemming.

This stage extracts the root of a word removing affixes and endings. For example *inhibits*, *inhibition*, *inhibited* have as common root *inhibit*. Stemming operate on single words.

4.0.5 Elimination of stopwords.

In this phase the software identifies and removes the words with low discriminating capacity, such as articles, prepositions and conjunctions. These words are too common to be useful for our analysis and they don't add any affective information.

4.0.6 Selection of index terms.

In our case the index terms are all those words that could carry, either alone or in group, affective content.

After pre-processing phase we obtain a statements-words matrix W (n x m)

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{pmatrix}$$

where the element w_{ij} represents the frequency or number of occurrence of a word j in a statement i, with $i = 1, \dots, n$ and $j = 1, \dots, m$.

5 Affective Annotations

We classify words of statements in: Direct Affective Word(DAW) and Indirect Affective Word(IAW).
The DAW group is formed by words expressing an emotional state in the specific domain. For example, the word *happiness* and *delight* carry a positive emotional state. For our goal DAW are the more important words. These words, contain minor errors, are independent from the context and convey always the same emotional state.
Words that directly don't express any emotional state belong to IAW group. For example, the word *ice-cream* by itself doesn't convey any emotional state. If we insert in the statement *I like ice-cream* the word acquires a positive emotional state clearly different from *I hate ice-cream*. Therefore, this group includes words whose affectivity depends on the context in which they are included.
Respect to statements we can make the same distinction: Direct Affective Statement (DAS) and Indirect Affective Statement (IAS).
For every directly affective word and statement we associate a vector of six Ekman emotional indexes: Happy, Surprise, Fear, Sad, Angry, Disgust.

6 Affective vectors computing

The goal of this phase is to calculate affectivity of IAW and IAS from the manual assignment of DAW and DAS. Then from the affective vectors of IAS the polarity of statements can be estimated.

6.1 Word affectivity

The value of new affective vectors of IAW depends on the affective vector of the most similar DAW in the statements-words space. The idea is that similar words transport also similar affective state.

Therefore the computation bases on similarity concept evaluated by the normalized scalar product k

$$k = \frac{w_p . w_q}{\|w_p\| . \|w_q\|}$$

where w_p, w_q are affective vector of word p and q, $w_p \in IAW$ and $w_q \in DAW$ and $w_p \cdot w_q$ is the scalar product of two vectors.

The similarity measures the degree of correlation between words.

For calculating new affective vector of IAW [13] we use two methods:

$$w_{ap} = k \cdot w_{aq} \quad k > s \quad \forall w_p \in IAW \; \exists \, w_q \in DAW \qquad (WA1)$$

$$w_{ap} = w_{aq} \quad k > s \quad \forall w_p \in IAW \; \exists \, w_q \in DAW \qquad (WA2)$$

In the last case we consider that affective vectors of IAW are egual to similar vectors of DAW.

In both cases s represents the threshold that plays an important role on the error control. For low threshold values also affective vectors of non-similar words are calculated. Increasing this threshold value, affective vectors are calculated for similar DAW. In this case the estimated error decreases. This threshold s avoids to calculate the affectivity of words of the training set too dissimilar. An high error in word affectivity influences the statement affectivity estimation. Words, with their affective vectors, whose k is below a threshold s don't are considered.

6.2 Statement affectivity

For the calculation of statement affectivity of IAS we consider DAW and IAW. In the IAW, the context plays a key role. Obviously, to make IAW more reliable, we consider an average value between different emotional contexts in which they are inserted. If the word is used for 90% in affective positive statements (high value of happy) and the remaining 10% in negative statement (high values of disgust, angry, sad, fear) statistically it will have a positive weight in the calculation of affective vector and polarity. If that word is used for 50% of statements with positive emotions and the remaining 50% with negative emotions, it should have minimal affect in the affectivity and polarity.

The estimated error in the calculation may be reduced if we consider a corpus large and statistically independent with various examples of the contexts. The method that we use to calculate the affective vector for statement i (s_{ai}) is the following:

$$s_{ai} = \frac{\sum DAW}{n_{DAW}} \cdot \alpha + \frac{\sum IAW}{n_{IAW}} \cdot (1-\alpha) \qquad \forall s_i \in IAS \quad \alpha \in [0,1] \qquad (SA)$$

The $\sum$ is the sum of possible DAW e IAW that are present on the statement. The parameter α indicate the weight of DAW on estimation of statement affectivity. In this estimation (1- α) is the weight of IAW.

n_{DAW}, n_{IAW} are respectively number of words directly and indirectly affective.

7 Polarity estimation

We propose three methods to estimate the polarity p of a statement.

In the first method the polarity is given as the weighted difference between positive and negative affective indexes. Since the "surprise" sometimes may assume negative meaning, in the second method we assign the value positive or negative depending on predominant sentiment in the statement (difference between "happy" and negative indexes). The third method uses a linear regression for polarity estimation.

7.0.1 Method SP1.

Surprise index is considered a positive element.

$$p = \frac{1}{2}\sum_{i=1}^{2} e_i \cdot \alpha_p - \frac{1}{4}\sum_{i=3}^{6} e_i \cdot (1-\alpha_p) \quad \alpha_p \in [0,1]$$

where $(e_1, e_2, ...e_6)$ are indexes of affective vector ("happy", "surprise", "fear", ..."disgust"), α_p is the weight assigned to positive indexes, $(1-\alpha_p)$ correspond to α_n (weight assigned to negative indexes).

7.0.2 Method SP2.

Surprise index follows prevalent sentiment of statement.

$$p = \frac{1}{2}\sum_{i=1}^{2} e_i \cdot \alpha_p - \frac{1}{4}\sum_{i=3}^{6} e_i \cdot (1-\alpha_p)$$

if happy $\geq$ (sad+angry+fear+disgust), otherwise

$$p = e_1 \cdot \alpha_p - \frac{1}{5} \sum_{i=2}^{6} e_i \cdot (1 - \alpha_p)$$

7.0.3 Method SP3.

We estimate statement polarity by linear regression.

$$p = a + \sum_{j=1}^{6} a_j \cdot s_{aj} + e$$

where a is intercept of straight line, a_j are regression coefficients of six affective indexes, s_{aj} the six elements of statement affective vector and e the statistical error.

There is a linear relationship between the values of statement affective vectors and its polarity. Regression allows us to calculate the coefficients a_j.

The parameter vectors based on the training set (DAS) are estimated. Using the criterion of optimality of least squares, the parameters a_j can be obtained by minimizing the square of euclidean distance. For estimating regression coefficients and error, this method needs of a labeled training set of statements.

8 Case study

In order to test the validity of our methodology we have gathered 800 posts from web forums on customer opinions about a resort in Sharm el-Sheikh and in particular we selected opinions about services: Kitchen, Restaurant, Room Service, and Administration. Opinions were collected from various Internet sites, like alpharooms.com and realholidayreports.com.

8.1 Preprocessing.

After gathering web posts and running the pre-processing phase of our software we obtain the statements-words matrix W. In the assignment of weights to various elements $w_{i,j}$ of this matrix, we considered $w_{ij} = freq_{ij}$. We considered also the case of a double weight to DAW but the result were worse and therefore we have discarded it. After the pre-processing phase, the software saved into database 1303 statements and 2300 words.

8.2 Affective annotations.

In this phase we manually label and categorize directly affective words and statements . We labeled manually 900 DAS and 374 DAW. The remaining words are IAW and the remaining statements are IAS.

For DAS we manually assign also a polarity value. We varied polarity p between -10 and +10. The amplitude is useful for agile enterprise to process high negative opinions with a certain priority.

For each DAW and DAS we assign six dimensional affective vectors with the support of the interface of software that we have developed (Fig. 1).

The value of these index can vary between 0 and 10 (values controlled by the software). Zero means no affectivity while the value 10 expresses an highest amplitude of affectivity. For example, we can associate to the word "stench" the affective vector (0, 0, 0, 2, 2, 6). It means that in the affective meaning of the word stench, the elements sad and angry contribute with a small value, the element disgust with a high value and other elements don't produce any contribution. Statements with high values of contrasting affective indexes are ambiguos. For example: Happy = Disgusted = 10 and Sad = Angry = Surprise = Fear = 0.

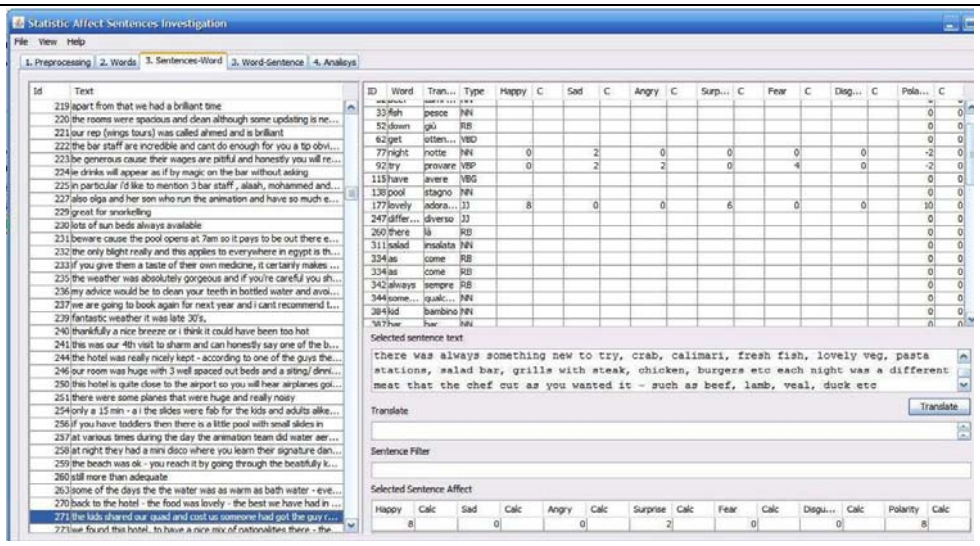


Figure 1: Software interface for manually assignment.

8.3 Experimental planning.

In our experiments we consider a training set and a test set for words (TrainW and TestW) and for statements (TrainS and TestS). In both case we consider as training set the 50% of DAW and DAS manually assigned. Remaining 50% is used as test set. In this way we can compare values manual assigned with values calculated by software.

As error in the affectivity estimation we consider the mean squared error (MSE) or Euclidean distance and the standard deviation (σ) of it.

$$e = MSE(\widehat{\theta}) = E((\widehat{\theta} - \theta)^2)$$

where $\widehat{\theta}$ is the *estimator* and θ the estimated value.

For polarity estimation we consider the amplitude and the sign. The amplitude error is given from the normalized difference between assigned and estimation polarity:

$$e = |p_a - p_e|$$

where p_a is the assigned polarity and p_e is the estimated polarity.

The percent sign error (misclassification) will be given by the sum of false positives and false negatives divided by the total statements of the test set.

In our case study we take in consideration the following experimental planning:

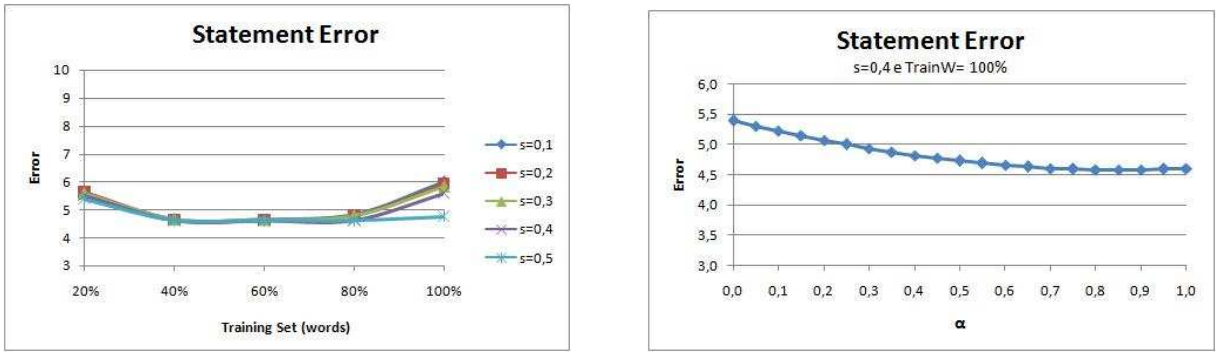
Word affectivity estimation. We vary the TestW from 20% to 100% and the threshold s from 0,1 to 0,5. We consider both methods WA1 and WA2 in the words affectivity estimation. Figure 2 shows the results of experiments. The values enclosed in round brackets represent σ error. Applying WA2, increasing the training set size and decreasing the threshold s, the error decreases. This is mainly due to the fact that decreasing s there are many words for comparisons. In the method WA1 increasing the training set the error increases, because there is the negative influence of the similarity coefficient k with an intrinsic error. The estimation error becomes larger when increasing the size of TrainW. With a reduced TrainW the error is acceptable. In our case the better method is WA2.

Sentence affectivity estimation. We vary the TestW from 20% to 100%, the threshold s from 0,1 to 0,5 and α from 0 to 1. The parameter α represents the weight of DAW set in the statement affectivity (SA). Since SA depends on word affectivity, we considered also, in this case, both methods WA1 and WA2. Figure 3(a) refers to WA1. Initially as the training set (TrainW) increases the error decreases. For

s	Training Set (TrainW)									
	20%		40%		60%		80%		100%	
	WA1	WA2	WA1	WA2	WA1	WA2	WA1	WA2	WA1	WA2
0,1	6,16 (2,22)	8,96 (1,98)	5,94 (2,70)	6,57 (4,41)	6,43 (2,67)	5,84 (3,74)	5,81 (3,32)	6,70 (4,73)	6,44 (2,83)	5,85 (3,31)
0,2	6,20 (2,30)	8,76 (2,48)	5,50 (3,06)	6,41 (4,29)	6,24 (2,79)	5,73 (3,76)	5,73 (3,34)	6,33 (4,70)	6,26 (2,85)	6,00 (3,79)
0,3	6,17 (2,69)	8,33 (2,73)	5,46 (3,41)	6,15 (4,20)	6,25 (3,28)	5,85 (4,04)	5,78 (3,52)	6,29 (4,58)	6,05 (3,14)	6,40 (4,01)
0,4	5,09 (3,15)	6,76 (3,56)	5,52 (3,67)	6,33 (3,81)	6,02 (3,42)	6,23 (3,82)	5,85 (3,62)	6,55 (4,40)	6,19 (3,27)	7,03 (3,97)
0,5	5,31 (3,53)	6,39 (3,51)	5,31 (3,53)	6,39 (3,51)	5,76 (3,71)	6,53 (3,88)	5,89 (3,83)	6,38 (4,25)	6,58 (4,14)	6,08 (3,59)

Figure 2: Error (MSE $\pm$ (σ)) of word affectivity in methods WA1 and WA2 varying TrainW and s.

high values of TrainW, the error introduced by the coefficient k is high and therefore the statement error increases. To this point the threshold s is important. When the value of s is 0.5, the number of words contributing in the estimation decreases reducing MSE . Figure 3(b) refers to method WA2. A low value of α means a more influence of IAW and then the error is high. Increasing the value of α, the contribution of DAW is greater than IAW and error decreases. For α value near 1 the error may lightly increase in presence of ambiguity. For example "The spaghetti are a beautiful disgusting". In the statement there are only two direct affective contrasting words (beautiful and disgusting). In this case the meaning of statement depends of the context and then the error increases. The error increasing is more evident in the method WA1.



(a) (b)

Figure 3: Error of statement affectivity varying TrainW, s and α_p.

Polarity estimation. For polarity estimation (Fig. 4) we consider three methods SP1, SP2, SP3. The polarity error for methods SP1 and SP2 is shown in Fig.4(a) and for method SP3 in Fig.4(c). Fig. 4(b) represents the misclassification error for the first two method while Fig.4 (d) refers to last method SP3.

In the first two methods we vary s from 0,1 to 0,5 and α_p from 0 to 1. The parameter α_p represents the incidence of words with positive affectivity in the statement polarity.

In the last method (SP3) we vary the TrainS from 20% to 100%.

In all three methods, in average, the polarity error decreases with the increasing of TrainS and parameter α_p.

The accuracy of word affectivity improves with low values of threshold s but this worsens the statement affectivity because in the estimation many words are involved. Since the goal is to obtain a good estimate of affectivity and polarity, we must consider an intermediate threshold value. Analysing data output of our software we have observed that the better experimental method is the polarity estimation with linear regression (SP3). It is important to comment the Figure 4(b) and in particular the low value of error in

correspondence of α_p equal to 1. This occurs because in this case we don't take into account negative indexes but only the positive one. Negative indexes (fear, anger, sadness,disgust) are more numerous than positive; with their lack, error decreases.

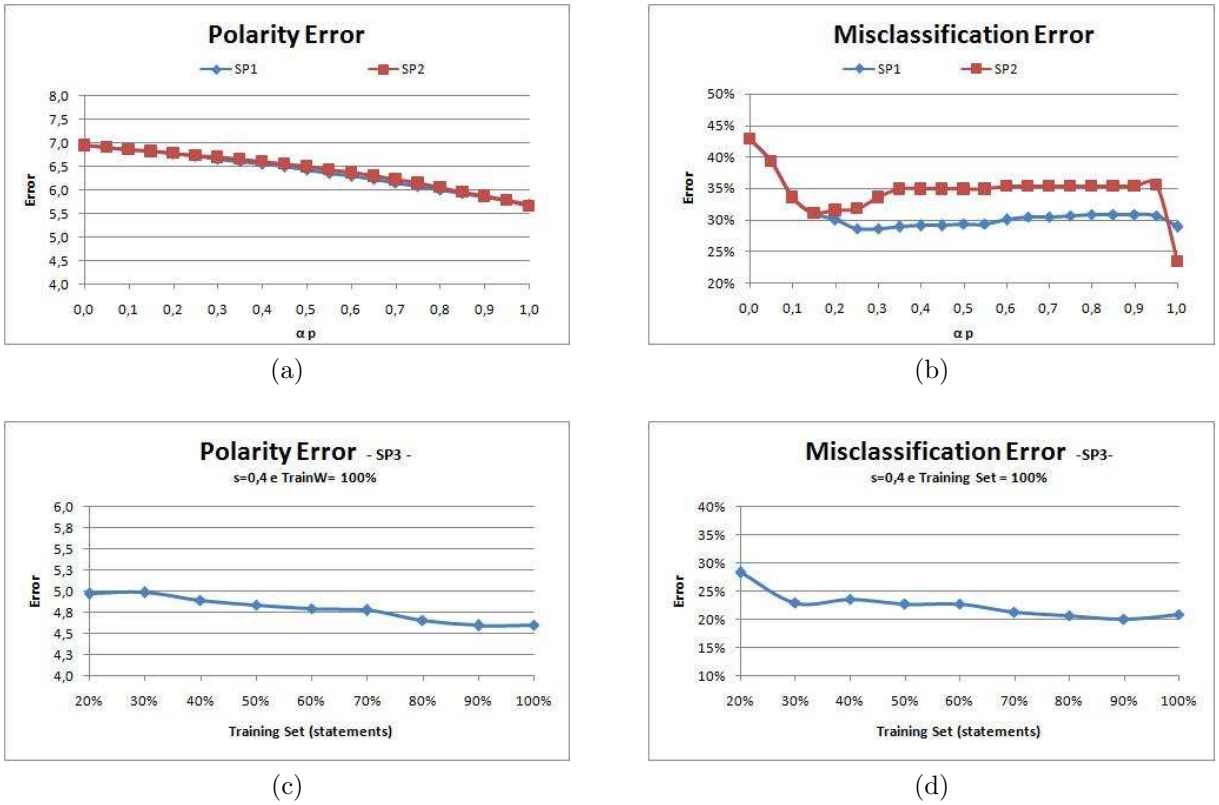


Figure 4: Polarity error (amplitude and sign) for three methods varying all parameters.

Concluding, with these experiments we can say that the better methods for word and sentence affectivity is WA2 method (word affectivity estimation without similarity coefficient k) and SP3 method (regression linear) for polarity estimation.
To improve the algorithm accuracy we must make manual labeling more rigid, with strict rules, and take in consideration only the affective index dominant without intermediate values. For example if, in a statement, both terms disgust and fear appear, we must consider only the dominant sentiment. It is important also consider a better weighting of parameters (α, s, k) for words and sentences affectivity. In the polarity estimation we can refine the linear regression with polynomial regression or neural networks.

9 Conclusions

Nowadays, for the enterprise, it is important gather a large amount of customer opinions on product/service. In this paper we illustrate an original approach to polarize opinions based on affective Ekman indexes. In our opinion, these indexes, allows to better capture the emotional state of customers about purchasing. The approach, based on the affective value of each single statement, produces good results on documents of medium-large dimensions. In this paper we have shown different schemes to calculate affectivity of single words or statements and mainly to calculate statement polarity. From single statement it is possible understand the positive or negative opinion with its amplitude and therefore if the customer is, more or less, satisfied. The polarity of opinion on (part of) product allows to agile enterprise to respond immediately to changes requests of the market.

References

[1] D. Plummer and D. McCoy, "Achieving agility: The view through a conceptual framework," *Garter Research*, 2006.

[2] D. Consoli, C. Diamantini, and D. Potena, "Improving the customer intelligence with customer enterprise customer model," in *Proceedings of the 10th International Conference on Enterprise Information Systems*, Barcelona, Spain, 2008a, pp. 323–326.

[3] Y. Cho, I.Im, R. Hiltz, and J. Fjermestad, "An analysis of online customer complaints: implications for web complaint management," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. IEEE, 7-10 Jan. 2002, pp. 2308–2317.

[4] E. Wenger and W. M. Snyder, "Communities of practice: The organizational frontier," *Harvard Business Review*, pp. 139–145, 2000.

[5] P. Ekman, *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life*. NY: OWL Books, 2007.

[6] H. Liu, H. Lieberman, and T. Selker, "A model of textual affect sensing using real-world knowledge," in *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2003, pp. 125–132.

[7] A. Boucouvalas and X. Zhe, "Text-to emotion engine for real time internet communication," in *Proceedings of International Symposium on CSNDSP*. UK: Staffordshire University, 2002, pp. 164–168.

[8] C. Elliott, "Knowledge acquisition within the affective reasoner," in *Notes for the AAAI Spring Symposium: Cognitive Aspects of Knoweledge Acquisition*, 1992, pp. 64–72.

[9] A. Valitutti, C. Strapparava, and O. Stock, "Developing affective lexical resources," vol. 2, no. 1. WIT Press, 2004, pp. 61–83.

[10] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc," in *Computer Software*, Mahwah,NJ, 2001.

[11] A. Esuli and F. Sebastiani, "SentiWordNet: A publicly available lexical resource for opinion mining," in *Proceedings of LREC-06, 5th Conference on Language Resources and Evaluation*, Genova, IT, 2006, pp. 417–422.

[12] C. Strapparava and A. Valitutti, "Wordnet-affect: An affective extension of wordnet," in *The 4th International Conference On Language Resources and Evaluation*, 2004, pp. 1083–1086.

[13] D. Consoli, C. Diamantini, and D. Potena, "Improving the customer intelligence with customer enterprise customer model," in *Proceedings of the 10th International Conference on Enterprise Information Systems*, Barcelona, Spain, 2008, pp. 323–326.

Domenico Consoli
Universitá Politecnica delle Marche
Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione
Via Brecce Bianche 12, 60131 Ancona
Italy
E-mail: *consoli@diiga.univpm.it*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

The Musical Mode from Mathematical Point of View

Georgi Dimkov

Abstract

The history of the development of the concept of *mode* and *modality* in Music theory is more than 3000 years long. From the point of view of Combinatorics the musical mode is connected with the problem of decomposition of a positive natural number n in a sum of k numbers $(1 \le k \le n)$. The aim of the present paper is to establish a systematization of the *mode type* structures appearing in the so called *chromatic scale*. For that purpose we introduce a classification system and use Diophantine equations.

1 Introduction

The musical tones are produced by the vibration of different objects: strings, air columns in a pipe, bells, metallic or wooden plates, membranes etc. Their vibrations have different frequencies and we say that they produce different tones. The following fact is of major importance in the theory: Two tones with the ratio of their corresponding frequencies 2:1 have the same names in the usual system of appellation. In fact, they are different but they are similar from some aesthetic point of view. Theoretically as well practically it is said that the tone with the double frequency is higher than the other one by one *octave*.

It is necessary to make the following remark: Some notions from Music theory will appear below. To simplify the wording and spare long expressions and explications we shall not follow their exact definitions.

2 Scales

For the purposes of the *music aesthetics* it is necessary to determine the number and the frequency ratios of tones placed between two tones forming an octave. The first strict and complete investigations on this topic were done by Pythagoras. The concept of frequency appeared in the 17th century in a work of Isaac Beeckman (1588-1637). But the results of Pythagoras are calculated in ratios between the lengths of different parts of a string. The exact values of these ratios are not important for our investigations. We shall mention only that between the two end-tones of the octave there are six intermediate tones placed and two different ratios appear between the lengths of the respective parts of the string producing the new tones. The

bigger ratio is called a *whole tone – wt* and the smaller one – *semitone – st*. The wt's are five and the st's are two. A tone sequence with the following ratios between the consecutive tones

$$wt - wt - st - wt - wt - wt - st$$

is called (*Pythagorean*) *diatonic scale*.

Many centuries later the development of the music practice and aesthetics introduced five more tones dividing the wt's into two st's. So, a new object appeared a sequence of 12 tones with equal consecutive ratios of the corresponding parts of string, namely 1 st. (In fact these st's were not equal.) This new object is called a *chromatic scale*.

3 Mode and Combinatorics

Looking on the Pythagorean scale we can say that it is constructed by seven objects – st's and wt's. Two of them are of one kind and five of another one. The number of variations of these objects is given by the well-known formula

$$\binom{7}{2} = \binom{7}{5} = \frac{7!}{5!2!} = 21.$$

Let us think that the above sequence of wt's and st's are points on a circle. An illustration of the music practice in the last 4000 years or longer is the following: Choose arbitrarily one of the points and tour the circle. So we obtain seven sequences of wt's and st's. In each of them we shall never meet two neighbor st's. Moreover, the st's are always separated either by two or by three wt's. Roughly speaking, these seven sequences are called *modes*.

This last example is one reason to talk about the modes from the point of view of Combinatorics. The maximum possible combinations are 21 but only 7 are used in the practice. How to use or is it reasonable to use some other variations is a problem to be discussed by the musicians.

Since the wt consists of two st's, give the st the value 1 and the wt the value 2. Say that the value of the chromatic scale, calculated in st's, is 12. Following the same idea, the value of the diatonic scale is also 12. Comparing both cases we see that we have two examples of decomposition of the number 12 in a sum of natural numbers. That's one more reason to use Combinatorics to give some light on the abstract possibilities to compose tone sequences using the chromatic scale as a source.

The East Asian music uses the so called *pentatonic* scale. Inside the octave are placed four intermediate tones separated by

$$wt - wt{+}st - wt - wt - wt{+}st.$$

Or, the pentatonic is connected with following representation

$$2+3+2+2+3=12.$$

4 Subsets of the chromatic scale

Compare once again the chromatic and the diatonic scales. The chromatic consists of two tones forming an octave and 11 intermediate tones and we say that this is a 12-tone scale. The diatonic scale consists of the same two tones forming one octave and 6 intermediate tones; hence this is a 7-tone scale. Consequently the pentatonic is a 5-tone scale. Then the formula

$$\binom{11}{6} = 462$$

shows how much diatonic scales, in a larger sense of this notion, we can form, using the 12 tones of the chromatic scale. For the pentatonic this number is

$$\binom{11}{4} = 330$$

.

If we accept the idea that scales with another number of tones are also possible, the number of all scales would be

$$\sum_{i=0}^{11}\binom{11}{k} = 2^{11} = 2048$$

.

5 Intervals

In Music theory a pair of two different tones appearing together or consecutively is called *interval*. We need to distinguish the intervals numerically. For this purpose we introduce the notion *length* of an interval. The length is counted as the number of st's we can place between two different tones. So the diatonic scale contains intervals of lengths 1 and 2; the pentatonic contains intervals of lengths 2 and 3. And the chromatic scale contains only intervals of length 1.

The discussion of these three scales leads to the following ascertainment:

a) there are scales that contain different number of intervals;

b) there are scales that contain the same number of intervals but the intervals are not of the same lengths.

Then we can ask: *is it possible to compose scales by means of more than two intervals?*

And one more question: *a combination of intervals of which length could be used for the composition of different scales?*

6 The problem

Let us have an ordered string of tones, the first and the last tone forming an interval of length 12 (one octave). The intervals between the consecutive tones could be equal or different. (Here we avoid the notion scale.) Denote by ${}_kS(i_1,i_2,\ldots,i_k)$ the set of all such strings containing k different intervals with the lengths $i_1 < i_2 < \cdots < i_k < 12$. So, the chromatic scale as a tone string is an element of ${}_1S(1)$, the diatonic scale is an element of ${}_2S(1,2)$ and the pentatonic is an element of ${}_2S(2,3)$.

Our aim will be to list and investigate all sets ${}_kS(i_1,i_2,\ldots,i_k)$.

7 The Diophantine equation

Suppose we wish to know if it is possible to compose a tone string, determined by the intervals 2, 4 and 5. For this purpose we form the equation

$$2.x+4.y+5.z=12.$$

When such an equation allows only integer variables it is called *Diophantine equation*. In our case we are interested only in positive natural solutions. If the equation has a solution, x will be the number of intervals with length 2, y the number intervals with length 4 and z the number intervals with length 5. The quantity $x+y+z$ is the number of different tones in the string we are looking for.

8 Systematization and solution

The introduction of the sets ${}_kS(i_1, i_2, \ldots, i_k)$ gives the possibility to describe systematically all strings of the above type, changing consecutively k and the length of the intervals. Then forming the respective Diophantine equation we shall look for its natural solutions.

8.1 Case *k*=1

We deal with the relatively simple set ${}_1S(i_1)$. To determine how much intervals with lenght i_1 we need, we have to solve in natural numbers the equation

$$i_1.x=12,\ i_1=1, 2, 3, \ldots, 12.$$

It is clear that solution exists only for i_1=1, 2, 3, 4, 6, 12 and x=12, 6, 4, 3, 2, 1 respectively.

So $\bigcup {}_1S(i_1)$ for all admissible values of i_1 contains **6** elements.

8.2 Case *k*=2

We have to solve the equation

$$i_1.x + i_2.y = 12$$

for all admissible values of i_1, i_2. It is clear that $i_1 + i_2 \le 12$. To check the admissibility of a pair we shall use the following conditions. A pair is admissible if either

$12-(i_1+i_2)=0$

or

$12-(i_1+i_2)>0$ and $12-(i_1+i_2)$ is divisible either by i_1 or by i_2.

8.2.1 Let $i_1 = 1, i_2 \in [2,11]$.

Since each number is divisible by 1, according to the above conditions, all pairs $(1, i_2), i_2 \in [2,11]$ are admissible. Then we have to solve the equations

$$x + i_2.y = 12,\ 2 \le i_2 \le 11.$$

- $i_2 = 2 \rightarrow {}_2S(1,2)$. The respective equation is x+2.y=12.

Then x=2.(6-y). And according to the conditions we have the possibilities y=1, 2, 3, 4, 5. So we obtain the following solutions:

(2, 5), (4, 4), (6, 3), (8, 2), (10, 1).

The strings corresponding to these solutions contain 7, 8, 9, 10 and 11 different tones respectively. The number of the different variations in each string is as follows

$$\binom{7}{2} = 21,\ \binom{8}{4} = 70,\ \binom{9}{3} = 84,\ \binom{10}{2} = 45,\ \binom{11}{1} = 11.$$

The number of elements of ${}_2S(1,2)$ is 21+70+84+56+11=**231**.

- $i_2 = 3 \rightarrow {}_2S(1,3)$. The respective equation is x+3.y=12.

Or x=3.(4-y) and we have necessarily y=1, 2, 3 and the solutions are

(3, 3), (6, 2), (9, 1).

So we have strings with 6, 8 and 10 tones respectively with the number of variations

$$\binom{6}{3} = 20,\ \binom{8}{2} = 28,\ \binom{10}{1} = 10.$$

The number of elements of ${}_2S(1,3)$ is 20+28+10=**58**.

- $i_2 = 4 \to {}_2S(1,4)$. The respective equation is $x+4.y=12$.

Now $x=4.(3-y)$ and $y=1, 2$ and the solutions are

(4, 2) and (8, 1)

determining strings with 6 and 9 tones respectively and variations

$$\binom{6}{2} = 15,\ \binom{9}{1} = 9.$$

The number of elements of ${}_2S(1,4)$ is 15+9=**24**.

- $i_2 = 5 \to {}_2S(1,5)$. The respective equation is $x+5.y=12$.

Write the equation in the form $y = 2 + \frac{2-x}{5}$. It is equivalent to $\frac{2-x}{5} = \begin{cases} 0 \\ 5 \end{cases}$. We obtain $x=2, 7$, two solutions

(2, 2), (7, 1)

and two strings with 4 and 8 tones and corresponding variations

$$\binom{4}{2} = 6,\ \binom{8}{1} = 8.$$

The number of elements of ${}_2S(1,5)$ is 6+8=**14**.

- $i_2 \in [6,11]$. Now $i_1 + i_2 > 6$ and the only solution is $(12 - i_2, 1)$. The corresponding string, one per case, consists of $12 - i_2 + 1 = 13 - i_2$ tones with number of variations

$$\binom{13-i_2}{1} = 13 - i_2.$$

Then we have $\sum_{i_2=6}^{11} 13 - i_2 = \mathbf{27}$ variations in this group of cases.

Now we can summarize the variations of all cases when $i_1 = 1$ and the result is

231+58+24+14+27=**354**.

8.2.2 Let $i_1 = 2, i_2 \in [3,10]$.

There are two non-admissible pairs, namely (2, 7) and (2, 9).

- $i_2 = 3 \to {}_2S(2,3)$.

The equation $2.x+3.y=12$ written in the form $y = 4 + \frac{2.x}{3}$ has the only solution (3, 2). Consequently we have only one string with 5 tones and number of variations of ${}_2S(2,3)$

$$\binom{5}{2} = \mathbf{10}.$$

- $i_2 = 4 \to {}_2S(2,4)$.

After a simple transformation of the equation $2.x+4.y=12$ we obtain $x=2.(3-y)$. The values $y=1, 2$ lead to the solutions (2, 2), (4, 1) and two strings with 4 and 6 tones and corresponding numbers of variations

$$\binom{4}{2}=6 \text{ and } \binom{5}{1}=5.$$

The number of elements of ${}_2S(2,4)$ is 6+5=**11**.

- $i_2=5\rightarrow {}_2S(2,5)$; $12-(i_1+i_2)=12-(2+5)=5$.

This calculation shows that we have only one solution – (1, 2). The string has three tones and the number of elements of ${}_2S(2,5)$ is **3**.

- $i_2=6\rightarrow {}_2S(2,6)$.

After transformation of the standard equation we obtain $y=1-\frac{x}{3}$. The solution is (3, 1) – four tones string.

The number of elements of ${}_2S(2,6)$ is **4**.

- (2, 7) and (2, 9) are not admissible.
- $i_2=8\rightarrow {}_2S(2,8)$ and $i_2=10\rightarrow {}_2S(2,10)$.

Both cases are quite clear and we have

${}_2S(2,8)$ - **3** elements, ${}_2S(2,10)$ - **2** elements.

The variations of all cases when $i_1=2$ are

10+11+3+4+3+2=**33**.

8.2.3 Let $i_1=3, i_2\in[4,9]$.

- $(3,i_2), i_2=4,5,7,8$ are not-admissible pairs.
- $i_2=6\rightarrow {}_2S(3,6)$ and $i_2=9\rightarrow {}_2S(3,9)$.

The variations of all cases when $i_1=3$ are

2+3=**5**.

8.2.4 Let $i_1=4, i_2\in[5,8]$.

- $i_2=8\rightarrow {}_2S(4,8)$ is the only admissible case.

The variations of all cases when $i_1=4$ are **2**.

8.2.5 Let $i_1=5, i_2\in[6,7]$.

- $i_2=7\rightarrow {}_2S(5,7)$ is the only admissible case.

The variations of all cases when $i_1=5$ are **2**.

8.2.6 Let $i_1=6, i_2=6$.

${}_2S(6,6)\equiv {}_1S(2)$. It is already counted.

Then we can summarize.

The total of variations for the case *k*=2 is

354+33+5+2+2=**396**.

8.3 Case *k*=3

We have a similar condition for the admissibility of a triple (i_1, i_2, i_3). The general form of the Diophantine equation is

$$i_1.x + i_2.y + i_3.z = 12.$$

The solutions will give tone strings with $x+y+z$ tones. The number of variations in each string is given by the formula

$$\frac{(x+y+z)!}{x!\,y!\,z!}.$$

8.3.1 Let $i_1 = 1$.

- $i_2 = 2, i_3 \in [3, 9]$.
  - $i_3 = 3 \rightarrow {}_3S(1, 2, 3)$. We have to solve the equation $x+2.y+3.z=12$. Rewrite it in the form

$$x + 2.y = 3.(4 - z)$$

having in mind that $x + 2.y \geq 3$. Then we have $z=1, 2, 3$. Putting $z=1$ we obtain a new equation $x+2.y=9$. In item **2.** we solved many similar equations therefore we shall give directly the solutions.

(1, 4, 1), (3, 3, 1), (5, 2, 1), (7, 1, 1)

We obtained four strings with 6, 7, 8 and 9 tones respectively. The variations in each string are

$\frac{6!}{1!4!1!} = 30$, $\frac{7!}{3!3!1!} = 140$, $\frac{8!}{5!2!1!} = 168$, $\frac{9!}{7!7!1!} = 72$, or in total 30+140+168+72=410.

For $z=2, 3$ we obtain three mire solutions

(2, 2, 2), (4, 1, 2), (1, 1, 3)

with variations

$\frac{6!}{2!2!2!} = 90$, $\frac{7!}{4!1!2!} = 105$, $\frac{5!}{1!1!3!} = 20$ with the total 90+105+20=215.

The number of variations in ${}_3S(1, 2, 3)$ is 410+215=**625**.

  - $i_3 = 4 \rightarrow {}_3S(1, 2, 4)$. We have to solve the equation $x+2.y+4.z=12$. Rewrite it in the form

$$x + 2.y = 4.(3 - z).$$

We repeat the above procedure and putting $z=1, 2$ we obtain the solutions

(2, 1, 2), (2, 3, 1), (4, 2, 1), (6, 1, 1)

with variations

$$\frac{5!}{2!\,1!\,2!} = 30,\ \frac{6!}{2!\,3!\,1!} = 60,\ \frac{7!}{4!\,2!\,1!} = 105,\ \frac{8!}{6!\,1!\,1!} = 56.$$

The number of variations in ${}_3S(1, 2, 4)$ is 30+60+105+56=**251**.

  - $i_3 = 5 \rightarrow {}_3S(1, 2, 5)$. Since 1+2+5=8 it is clear that $z=1$. The new equation $x+2.y=7$ gives three solutions

(1, 3, 1), (3, 2, 1), (5, 1, 1)

with variations

$$\frac{5!}{1!\,3!\,1!} = 20,\ \frac{6!}{3!\,2!\,1!} = 60,\ \frac{7!}{5!\,1!\,1!} = 42.$$

The number of variations in ${}_3S(1, 2, 5)$ is 20+60+42=122.

- $i_3 = 6,7,8,9 \rightarrow {}_3S(1, 2, i_3)$. Since $1+2+i_3 \geq 9$ z=1. The solutions are
  - $i_3 = 6 \rightarrow (2, 2, 1), (4, 1, 1);$
  - $i_3 = 7 \rightarrow (1, 2, 1), (3, 1, 1);$
  - $i_3 = 8 \rightarrow (2, 1, 1);$
  - $i_3 = 9 \rightarrow (1, 1, 1).$

Here are the corresponding variations

$$\frac{5!}{2!\,2!\,1!} = 30,\ \frac{6!}{4!\,1!\,1!} = 30,\quad \frac{4!}{1!\,2!\,1!} = 12,\ \frac{5!}{3!\,1!\,1!} = 20,\ \frac{4!}{2!\,1!\,1!} = 12,\ \frac{3!}{1!\,1!\,1!} = 6.$$

The number of variations in $\bigcup_{i_3=3}^{9} {}_3S(1, 2, i_3)$ is 625+251+122+60+32+12+6=**1108**.

- $i_2 = 3, i_3 \in [4,8].$
  - $i_3 = 4 \rightarrow {}_3S(1, 3, 4)$. The transformed equation is $x+3.y = 4.(3-z) \geq 4$. Hence z=1,2. We have three solutions

$$(2, 2, 1), (5, 1, 1), (1, 1, 2).$$

The variations:

$$\frac{5!}{2!\,2!\,1!} = 30,\ \frac{7!}{5!1!1!} = 42,\ \frac{4!}{1!1!2!} = 12$$ Total in ${}_3S(1, 3, 4)$ - 30+42+12=84.

  - $i_3 = 5 \rightarrow {}_3S(1, 3, 5)$. Clearly z=1 and the equation x+3.y=7 gives the solutions

$$(1, 2, 1), (4, 1, 1).$$

The variations:

$$\frac{4!}{1!2!1!} = 12,\ \frac{6!}{4!1!1!} = 30.$$ Total in ${}_3S(1, 3, 5)$ – 12+30=42.

  - $i_3 = 6, 7, 8 \rightarrow {}_3S(1,3,i_3)$.

$$1+3+i_3 \geq 10 \Rightarrow z = 1$$

$$x+3.y = 12-i_3 \geq 6 \Rightarrow y = 1$$

The solutions are

  - $i_3 = 6 \rightarrow (3, 1, 1),$
  - $i_3 = 7 \rightarrow (2, 1, 1),$
  - $i_3 = 8 \rightarrow (1, 1, 1)$

with variations

$$\frac{5!}{3!\,1!\,1!}=20,\ \frac{4!}{2!\,1!\,1!}=12,\ \frac{3!}{1!1!1!}=6.$$

The number of variations in $\bigcup_{i_3=4}^{8}{}_3S(1,3,i_3)$ is 84+42+20+12+6=**164**.

- $i_2=4, i_3=5,6,7$. As earlier we conclude that *y*=*z*=1 and we have one solution per case

(3, 1, 1), (2, 1, 1), (1, 1, 1).

The variations

$$\frac{5!}{3!\,1!\,1!}=20,\ \frac{4!}{2!\,1!\,1!}=12,\ \frac{3!}{1!1!1!}=6$$ have the total 20+12+6=**38**.

- $i_2=5, i_3=6\rightarrow{}_3S(1,5,6)$. There is only one solution (1, 1, 1) with **6** variations.

The number of variations in $\bigcup{}_3S(1,i_2,i_3)$ for all admissible triples $(1,i_2\,i_3)$ is 1108+164+38+6=**1316**.

8.3.2 Let $i_1=2$.

- $i_2=3$. The triple (2, 3, 6) is not-admissible.
  - $i_3=4$. Solution: (1, 2, 1). Variations: $\frac{4!}{1!\,2!\,1!}=12$.
  - $i_3=5$. Solution: (2, 1, 1). Variations: $\frac{4!}{2!\,1!\,1!}=12$.
  - $i_3=7$. Solution: (1, 1, 1). Variations: $\frac{3!}{1!\,1!\,1!}=6$.
- $i_2=4$. The triple (2, 4, 5) is not-admissible.
  - $i_3=6$. Solution: (1, 1, 1). Variations: $\frac{3!}{1!\,1!\,1!}=6$.

The number of variations in $\bigcup{}_3S(2,i_2,i_3)$ for all admissible triples $(2,i_2\,i_3)$ is 12+12+6+6=**36**.

8.3.3 Let $i_1=3$.

There is only one admissible triple (3, 4, 5). Solution: (1, 1, 1). Variations: $\frac{3!}{1!\,1!\,1!}=6$. And the total is **6**.

The total of variations for the case *k*=3 is

1216+36+6=**1258**.

8.4 Case *k*=4.

The corresponding equation is $i_1.x+i_2.y+i_3.u+i_4.v=12.$ From 1+2+3+4=10 we see that $u=v=1$.

- $i_1=1, i_2=2, i_3=3, i_4\in[4,5,6]$.

- $i_4 = 4 \to {}_4S(1,2,3,4)$. The equation $x+2.y=5$ gives the solutions

$$(1, 2, 1, 1), (3, 1, 1, 1)$$

with variations

$$\frac{5!}{1!\,2!\,1!\,1!} = 60\,,\ \frac{6!}{3!\,1!\,1!\,1!} = 120\,.$$

- $i_4 = 5,6 \to {}_4S(1,2,3,5),\ {}_4S(1,2,3,6)$. Since $1+2+3+i_4 \geq 11$, we have $y=u=v=1$.

Solutions: (2, 1, 1, 1), (1, 1, 1, 1). Variations:

$$\frac{5!}{2!\,1!\,1!\,1!} = 60\,,\ \frac{4!}{1!\,1!\,1!\,1!} = 24\,.$$

- $i_1 = 1, i_2 = 2, i_3 = 4, i_4 = 5 \to {}_4S(1,2,4,5)$. There is only one solution (1, 1, 1, 1) with

$$\frac{4!}{1!\,1!\,1!\,1!} = 24$$

variations.

The total of variations for the case $k=4$ is

60+120+60+24+24=**288**.

At the end of this investigation we can summarize the results for the four cases:

6+396+1358+288=2048.

So we enumerated all tone strings.

9 Examples.

Some examples was given the introduction part. We shall repeat them we shall add some other practical cases.

1. The well-known natural major and minor scales are contained in ${}_2S(1,2)$ and especially in the solution (2, 5).

2. The so called *melodic* major and minor are also in ${}_2S(1,2)$ by the same solution. But all the four scales are different variations of the solution (2, 5).

3. Another variety of the major/minor scale is the so called *harmonic* scale. It is contained in ${}_3S(1,2,3)$ and is variation of the solution (3, 3, 1).

4. The pentatonic is contained in ${}_2S(2,3)$ and given by the solution (3, 2).

5. ${}_1S(1)$ is the chromatic scale.

6. ${}_1S(6)$ is the so called *whole tone* scale. It was used in the first half of the 20th century by Rimski-Korsakov, Debussy, Ravel and other composers.

7. The music 0f the 20th and 21st uses *octatonic*. This is variation of the solution (4, 4) in ${}_2S(1,2)$.

8. In the 11th-12th century the composers used *hexatonic*. The names of the tones come from a very popular hymn dedicated to Saint John composed by Guido d'Arezzo. The hexatonic used comes from ${}_3S(1,2,3)$, solution (1, 4, 1).

After a careful study of different musical compositions one could find many other cases, contained among the 2084 tone strings described above.

References

[1] Kenneth R. Rumery, *Octatonic Scales*, http://jan.ucc.nau.edu/~krr2/ct_octindex.html, 1996.
[2] Klaus Lang,*Auf Wohlklangswellen durh der Toene Meer*, Graz, 1999.
[3] Kyle Gann, *An Introduction to Historical Tunings*, http://www.kylegann.com/histune.html, 1997.
[4] L.Krasinskaya, V.Utkin, *Elementary Music Theory* (in Russian), Moscow, 1983.
[5] Peter A. Frazer, *The Development of Musical Tuning Systems*, http://www.midicode.com/tunings/, 2001.
[6] Stefan Goeller, *Introduction to Musical Scales*, University of Warwick, 2001.
[7] V.Serpinskii, *On Entire Numbers Solutions of Equations* (in Russian), Moscow, 1961

Georgi Dimkov
Institute of Mathematics and Informatics
Department of Complex Analysis
8, Acad. G.Bonchev str.
1113 Sofia
BULGARIA
gdimkov@math.bas.bg

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

An Approach to Analog Circuit Synthesis using Genetic Algorithms

Vladislav Durev, Elissaveta Gadjeva

Abstract

An approach to analog circuit synthesis using Genetic Algorithms (GA) is proposed in the present paper. The Modified Nodal Analysis (MNA) is used for the construction of the admittance circuit matrices. A circuit of Operational Amplifiers (OA) based high-pass filter and a circuit of band-pass filter are used for the demonstration of the method. The algorithm is implemented in MATLAB environment.

1 Introduction

The automated synthesis of analog circuits is an important problem that calls for solution in the recent years and many papers are published with different approaches, including the use of evolutionary algorithms. There is not a common approach to analog circuit synthesis, but different approaches, applied to different circuit types and applications [1-4].

As the analog circuit synthesis needs a mathematical representation of the circuit, which has a number of variables with different ranges of variation, it is reasonable to use GA [5] to aid the synthesis procedure.

An approach to automated analog circuit synthesis, based on the application of Genetic Algorithms (GA) is presented in the present paper. The software is implemented in MATLAB using the GA Toolbox [6] functionality. The approach optimizes both the structure and the parameter values in the analog circuit. It constructs the mathematical model of the circuit using MNA [7] and the embryonic electrical circuit concept [1, 8]. The circuit elements without admittance description are defined by the corresponding component equations. As a result, the circuit matrix order increases, but MNA does not impose restrictions on the element types and is very applicable for circuit description with programming language. MNA fully describes the nature and the structure of the circuit using several matrices.

Two examples of synthesized circuits are presented – a third order unity-gain Bessel high-pass filter, based on OA and an asymmetric band-pass filter.

2 Application of GA in the synthesis of Operational Amplifier (OA) based circuits

The schematic of a third order unity-gain Bessel high-pass filter, based on OA, is shown in Fig. 1. The frequency response of this schematic is used as an input data for the GA, which is

programmed to synthesize a schematic, based on the input data, according to a given purpose function. The task is to design a third-order unity-gain Bessel high-pass filter with the corner frequency f_c = 1 kHz.

The *Input* matrix is introduced, which gives the connection between the components types and values and the structure of the schematic. The structure of the *Input* matrix is like the structure of *PSpice* netlist file:

$$[Input] = \begin{bmatrix} type\_1 & node\_11 & node\_12 & value\_1 \\ type\_2 & node\_21 & node\_22 & value\_2 \\ \dots & \dots & \dots & \dots \\ type\_i & node\_i1 & node\_i2 & value\_i \end{bmatrix} \quad (1)$$

Every row from this matrix represents one passive component in the schematic, which is of *type_i* (resistor – 1, inductor – 2 or capacitor – 3), between the nodes *node_i1* and *node_i2*, with value *value_i*. All the variables in the *Input* matrix can be optimized from the GA or have fixed values. It depends on the user and the application.

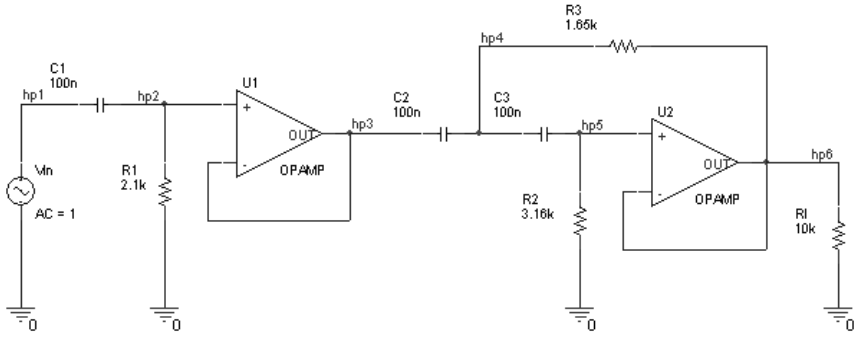


Fig. 1. Initial schematic of a third order unity-gain Bessel high-pass filter, based on OA

The *FieldD* matrix [6] determines the range of variation for every variable in the *Input* matrix:

$$[FieldD] = \begin{bmatrix} PRECI & \dots & PRECI \\ 0 & \dots & COMP\_VALUE\_\min \\ NODES\_NUMBER - 1 & \dots & COMP\_VALUE\_\max \\ 1 & \dots & 1 \\ 0 & \dots & 0 \\ 1 & \dots & 1 \\ 1 & \dots & 1 \end{bmatrix} \quad (2)$$

The value of *PRECI* determines the precision factor for representing the numbers in the GA, *NODES_NUMBER* contains the number of the nodes in the schematic. The number of the columns in the *FieldD* matrix corresponds to the number of variables, which are optimized by the GA. The first column in (2) represents a number of a component node, whose value is varied from 0 (*GND*) to (*NODES_NUMBER*-1). It is assumed that the number of the components is equal to the number of nodes. If the two terminals of a passive component are connected to one and the same node, this component is excluded from the generated schematic. The last column in (2) represents a component value, which will be varied by the GA in the range (*COMP_VALUE_min* ÷ *COMP_VALUE_max*). Thus, the ranges of variation of the variables in the *Input* matrix are fixed in the *FieldD* matrix [6].

The admittance *Y*-matrix of the schematic is built from the *Input* matrix, using MNA. The values, generated for the components types and the nodes numbers are rounded. The influence of

the OAs is defined in the *Y*-matrix and the influence of the source of the schematic is defined in the *Y*-matrix and the *I*-matrix. The nodes voltages are obtained in the *U*-matrix, using the equation (3) [7]:

$$[U] = [Y]^{-1}[I] \tag{3}$$

In the following example the input source *Vin* and the load resistor *Rl* are fixed using the embryonic circuit concept [1, 8]. Moreover, the components *C*1, *R*1, *U*1 and *U*2 are fixed and the rest of the components are optimized by the GA (Fig. 1). The following variation ranges are written in the *FieldD* matrix for the components *R*2, *R*3, *C*2, *C*3 correspondingly:

*N*5_1 = 0 ÷ 5 kΩ
*N*5_2 = 0 ÷ 5 kΩ
*N*6_1 = 0 ÷ 200 nF
*N*6_2 = 2 ÷ 200 nF

The variables *N*5_1 ÷ *N*6_2 are members of the *Input* matrix, together with the numbers of the nodes of the optimized components:

```
Input = [1  2                       0               2.1e3;
         1  round(N11(ix))          round(N12(ix))  N5_1(ix);
         1  round(N21(ix))          round(N22(ix))  N5_2(ix);
         3  1                       2               100e-9;
         3  round(N31(ix))          round(N32(ix))  N6_1(ix);
         3  round(N41(ix))          round(N42(ix))  N6_2(ix);
         1  (NODES_NUMBER-1)        0               10e3];
```

For example the second row of the *Input* matrix represents a resistor, connected between nodes with numbers round (*N*11(*ix*)) and round(*N*12(*ix*)) and resistance value *N*5_1(*ix*) Ω. Only the type of this component is fixed and the nodes numbers and the component value are optimized by the GA. The fourth row represents a capacitor between nodes 1 and 2 with value 100 nF. All the properties of this component are fixed. It depends on the user which values in the *Input* matrix are fixed or optimized. An internal cycle calculates the *Input* matrix for every individual *ix*, as the independent input variables are defined as vectors of generated values, and the number of these values is equal to the number of the individuals in the population [6].

As the OAs *U*1 and *U*2 (Fig. 1) are assumed ideal amplifiers, their impact in the *Y*-matrix, together with the impact of the input source *Vin* is given in (4):

$$[Y] = \begin{vmatrix} \dots & \dots & \dots & \dots & \dots & \dots & 1 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 1 \\ 1 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 1 & -1 & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 1 & -1 & \dots & \dots & \dots \end{vmatrix} \tag{4}$$

The corresponding MATLAB code is:

```
Y((NODES_NUMBER+1), 2) =  1; %U1 input +
Y((NODES_NUMBER+1), 3) = -1; %U1 input -
Y((NODES_NUMBER+2), 5) =  1; %U2 input +
Y((NODES_NUMBER+2), 6) = -1; %U2 input -
Y(3, (NODES_NUMBER+1)) =  1; %U1 output
Y(6, (NODES_NUMBER+2)) =  1; %U2 output
```

Similarly, the impact of the input source *Vin* in the *Y*- and the *I*-matrix can be represented, using the following code:

```
Y(1, NODES_NUMBER) = 1;
Y(NODES_NUMBER, 1) = 1;
I(NODES_NUMBER, 1) = Vin;
```

The GA is optimizing the circuit according to a given purpose function. The goal of the purpose function is to minimize the difference between the initial and the optimized circuit, i.e. between the initial frequency response of the circuit in Fig. 1 and the frequency response of the optimized circuit. The expression for the purpose function takes into account the values of the real and the imaginary parts of the initial (*Vout_initial*) and the current (*Vout_current*) output voltage (between the nodes *hp*6 and *GND*, Fig. 1) in the frequency domain, using the least squares values method:

$$G_{fun} = \sum_{i=1}^{n} \left[\Re\left(V_{out\_current}\right) - \Re\left(V_{out\_initial}\right)\right]^2 + \sum_{i=1}^{n} \left[\Im\left(V_{out\_current}\right) - \Im\left(V_{out\_initial}\right)\right]^2 \qquad (5)$$

The corresponding MATLAB code for (5) is:

```
g_fun = g_fun + ((real(Vout_current) - real(Vout_initial(i))).*(real(Vout_current) - real(Vout_initial(i)))) + ((imag(Vout_current) - imag(Vout_initial(i))).*(imag(Vout_current) - imag(Vout_initial(i)))); %Least squares approx method
```

The value of the purpose function (5) is optimized, taking into account the following parameters of the GA: number of individuals in the population *NIND* = 200; maximal number of iterations *MAXGEN* = 1000; number of input variables *NVAR* = 12, precision factor *PRECI* = 200, generation gap *GGAP* = 0.7 [6]. The last generated *Input* matrix is:

```
Input = [ 1.00000  2.00000  0.00000  2100
          1.00000  6.00000  4.00000  2069.21630055992
          1.00000  0.00000  5.00000  4435.65198208762
          3.00000  1.00000  2.00000  1e-7
          3.00000  4.00000  3.00000  5.37061759121239e-8
          3.00000  5.00000  4.00000  1.05775362735491e-7
          1.00000  6.00000  0.00000  10000];
```

The circuit, which corresponds to the given *Input* matrix is shown in Fig. 2. The frequency and phase responses of the initial (Fig. 1) and the synthesized (Fig. 2) circuits are given in Fig. 3 and marked with *Initial* and *GA* correspondingly. Both the circuits are simulated in *PSpice* for verification of the proposed approach. An excellent agreement was achieved without visual difference between the frequency and the phase responses.

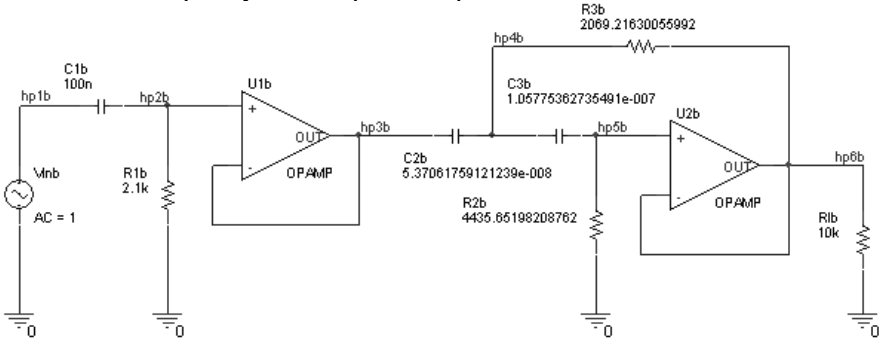


Fig. 2. GA synthesized third order unity-gain Bessel high-pass filter, based on OA

3 Application of GA in the synthesis of asymmetric band-pass filter

The approach used in section 2. can be applied to every passive analog schematic with independent voltage or current sources. Optimization of passive filter in respect to the number of the components is a problem, where the GA can be applied in the synthesis procedure.

The presented approach is applied to the Nielsen filter problem [9], which is widely used to test the effectiveness of the circuit synthesis procedures. Nielsen's band-pass filter is targeted for a modem application where one band of frequencies (31.2 to 45.6 kHz) must be isolated from another (69.6 to 84.0 kHz). This asymmetric filter is difficult to design and the standard design

procedure needs 10th order elliptic function. It is shown in [1] that the filter can be synthesized automatically, but the number of components for the best suited circuit is 38. Better result is shown in [4], where the filter is synthesized with only 9 components, but additional numerical optimization is applied together with the GA. The frequency response of this 9-component filter is used as an input data for our approach.

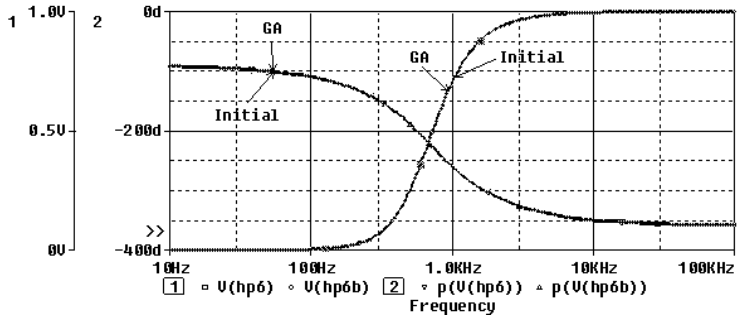


Fig. 3. Frequency and phase responses of the initial circuit from Fig. 1 (Initial) and the GA synthesized circuit from Fig. 2 (GA)

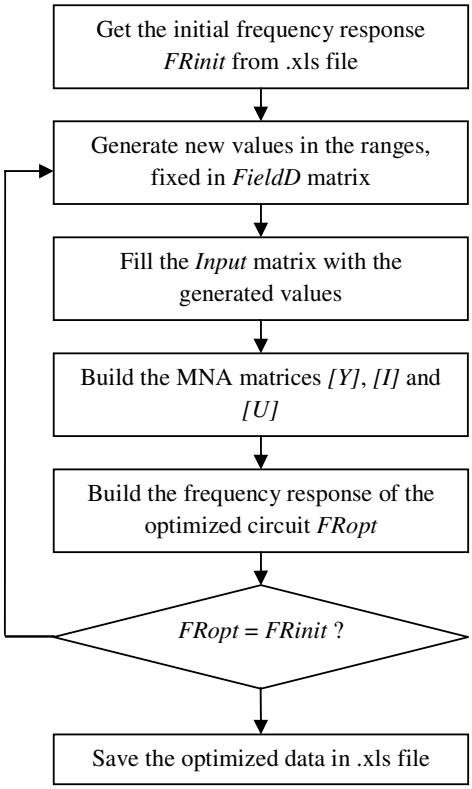


Fig. 4. The sequence of steps, used in the GA-based analog circuit synthesis approach

The expression of the purpose function in our case will optimize the schematic in respect to the frequency response:

$$G_{fun} = W.\sum_{i=1}^{n}\left|\left[M\left(V_{out\_current}\right) - M\left(V_{out\_initial}\right)\right]\right| \qquad (6)$$

The parameters of the GA are: *NIND* = 200, *MAXGEN* = 1000, *NVAR* = 27, *PRECI* = 200, *GGAP* = 0.7 for 9 optimized passive components and 7 nodes. The component values are normalized and the range of variation for every component value is (0 ÷ 12). The *Input* matrix is represented as:

```
Input = [ 1  1                        2                1;
          2  round(N11(ix))           round(N12(ix))   N10_1(ix);
          2  round(N21(ix))           round(N22(ix))   N10_2(ix);
          2  round(N31(ix))           round(N32(ix))   N11_1(ix);
          2  round(N41(ix))           round(N42(ix))   N11_2(ix);
          2  round(N51(ix))           round(N52(ix))   N12_1(ix);
          3  round(N61(ix))           round(N62(ix))   N12_2(ix);
          3  round(N71(ix))           round(N72(ix))   N13_1(ix);
          3  round(N81(ix))           round(N82(ix))   N13_2(ix);
          3  round(N91(ix))           round(N92(ix))   N14_1(ix);
          1  (NODES_NUMBER-1)  0                      1];
```

It includes 9 components (5 inductors, 4 capacitors, like the obtained circuit in [4]), whose values and connections are optimized and 2 resistors for termination. The synthesized circuit is shown in Fig. 5.

The frequency responses of the initial circuit from [4] and the synthesized circuit from Fig. 5 are simulated in *PSpice* for verification and the results are shown in Fig. 6.

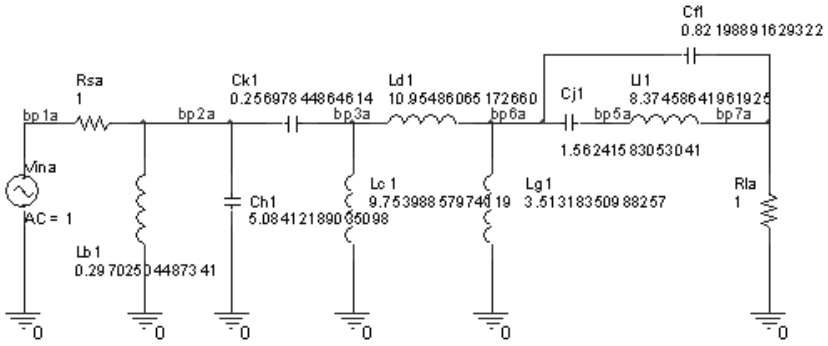


Fig. 5. The synthesized circuit of the band-pass filter

It is obvious from Fig. 6 that the presented approach finds the desired band and its width and it is compatible with the requirements, excluding the attenuation for the upper stop band. This is achieved only using the presented GA approach without any numerical optimization. The GA approach can give enough information though and the filter can fit the constraints if 3 additional poles are added to improve the upper stop band attenuation.

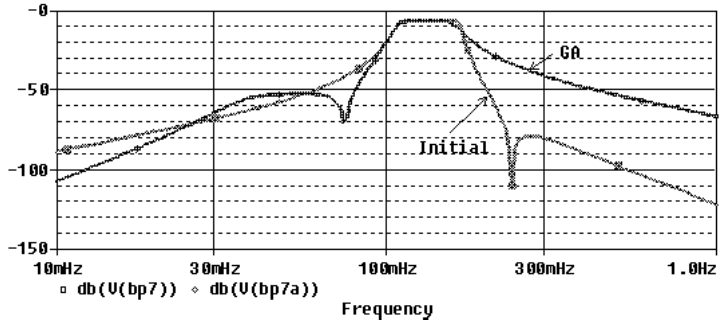


Fig. 6. The frequency responses of the band-pass filter in [4] (Initial) and the frequency response after applying the presented circuit synthesis procedure (GA)

The effectiveness of the presented approach depends mainly on the definition of the purpose function and the definition of the ranges for the input values. The structure of the schematic and the parameter values are optimized simultaneously, but in practice after some experiments is was proved that the structure of the schematic is optimized only during the first 50 iterations.

4 Conclusion

An approach to analog circuit synthesis using GA in the MATLAB environment is presented in the present paper, based on the use of embryonic circuit concept and MNA for building the circuit's equations. The presented procedure takes around 1 hour in order to synthesize a circuit with 10 components for 1000 iterations on Core 2 Duo architecture. It is shown that the obtained results are in agreement with the previous published data. The main advantage of the presented approach is the direct connection to the physical nature of the analog circuit using the MNA representation in the GA body.

The approach is successfully tested for circuits with up to 30 passive components.

References

[1] J. Koza, F. Bennet III, D. Andre, M. Keane, *Four Problems for which a Computer Program Evolved by Genetic Programming is Competitive with Human Performance*, Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, 20-22 May, 1996, Nagoya, Japan, pp. 1-10, ISBN: 0-7803-2902-3.

[2] S. Zhao, L. Jiao, J. Zhao, Y. Wang, *Evolutionary design of analog circuits with a uniform-design based multi-objective adaptive genetic algorithm*, Proc. NASA/DoD Conference on Evolvable Hardware, 29 June-1 July 2005, pp. 26 – 29, ISBN: 0-7695-2399-4.

[3] C. Reinhold, P. Kralicek, W. John , J. Teich, *Synthesizing Passive Networks by Applying Genetic Programming and Evolution Strategies*, The 2003 Congress on Evolutionary Computation, 8 – 12 Dec., 2003, vol. 3, pp. 1740 – 1747, ISBN: 0-7803-7804-0.

[4] J. Grimbleby, *Automatic analogue circuit synthesis using genetic algorithms*, IEE proceedings. Circuits, devices and systems, 2000, vol. 147, no. 6, pp. 319-323, ISSN 1350-2409.

[5] S. Sivanandam, S.Deepa, *Introduction to Genetic Algorithms*, Springer, 2008, ISBN 978-3-540-73189-4.

[6] A. Chipperfield, P. Fleming, H. Pohlheim, C. Fonseca, *Genetic Algorithm Toolbox for use with MATLAB*, User's Guide Version 1.2, Department of Automatic Control and Systems Engineering, University of Sheffield.

[7] J. Vlach, K. Singhal, *Computer Methods for Circuit Analysis and Design*, Kluwer Academic Publishers, 2003, ISBN 0-442-01194-6.

[8] J. Koza, F. Bennett, J. Lohn, F. Dunlap, M. Keane, D. Andre, *Automated Synthesis of Computational Circuits Using Genetic Programming*, IEEE International Conference on Evolutionary Computation, 13 – 16 April, 1997, Indianapolis, IN, USA, pp. 447-452, ISBN: 0-7803-3949-5.

[9] I. Nielsen, *A C-T Filter Compiler – From Specification to Layout*, Analog Integrated Circuits and Signal Processing, vol. 7, issue 1, pp. 21-33, 1995, Kluwer Academic Publishers, ISSN:0925-1030.

VLADISLAV DUREV
Technical University - Sofia
Department of Electronics and Electronic Technology
8 Kliment Ohridski Str., 1000 Sofia
BULGARIA
E-mail: v_p_durev@yahoo.com

ELISSAVETA GADJEVA
Technical University - Sofia
Department of Electronics and Electronic Technology
8 Kliment Ohridski Str., 1000 Sofia
BULGARIA
E-mail: egadjeva@tu-sofia.bg

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

A Constraint-Based Approach to the Timetabling Problem

Cristian Frăsinaru

Abstract

It is well known that timetabling problems are usually hard to solve and require a lot of computational effort. There are many theoretical models that address this type of problems and various algorithms have been developed in order to attempt to solve them efficiently. However, it is not easy at all to apply these models to real life situations. This paper presents a solution to create an universal constraint-based model for representing the timetabling problem that can be applied in universities, schools or any other similar domain. Once the model is created it can be effectively solved with any CSP solver. We have used our own CSP solver, called OmniCS[1] (Omni Constraint Solver), that allows an incremental, human-aided approach to the timetabling problem, which proved very useful in practical applications.

1 Introduction

A timetabling problem can be defined as the scheduling of some activities during a certain period of time. Each activity has a set of properties, like the participants who attend it or the resources it requires, and it is subject to certain restrictions regarding its possible planning. University timetables for instance must manage entities like courses, students, teachers and rooms in order to create a mapping between courses and the time-slots of the week. Usually, timetables cycle every week or every fortnight but this will not become a requisition of our model.

Traditionally, the problem is solved manually and it is a tedious job that requires days or even weeks. Automated building of the timetables is also very difficult because there are many types of restrictions that must be accounted for and it is not easy to express them in computational forms.

The timetabling problem has been studied intensively since the sixties ([11]) and different techniques for solving combinatorial problems have been used, such as graph coloring ([3]), integer programming ([13]), simulated annealing ([1]), tabu-search ([5]) or genetic algorithms ([2]). A survey can be found in ([16]).

Despite the fact that these methods have given good results, using them in real life applications was not easy and quite counterintuitive not only because the complexity of the restrictions could not be formalized properly but also because solving algorithms could not be modulated to follow human judgement, an aspect which is very important in the interactive creation of the timetble.

In recent years, many computationally difficult problems from areas like planning and scheduling have been proven to be easily modelled as constraint satisfaction problems (CSP) ([6], [18]) and a new programming paradigm emerged in the form of constraint programming, providing the opportunity of having declarative descriptions of CSP instances and also obtaining their solutions in reasonable computational time. As a result, constraint satisfaction techniques have been applied to the timetabling problem ([10], [15]).Because constraint programming received very much attention also from the industry, a lot of CSP

[1] *Omnics* is freely available at `http://omnics.sourceforge.net`

solvers emerged, i.e. applications that offer solutions to model a problem using constraints and also an engine able to solve it. To give only a few examples, we can mention Ilog ([12]), Minion ([14]), Choco ([4]) or our solver OmniCS ([7]).

This paper presents a solution to create an universal constraint-based model for representing the timetabling problem that can be applied in universities, schools or any other similar domain. Once the model is created it can be effectively solved with any CSP solver. We have used our solver OmniCS because it has distinctive features that make it appropriate for the timetabling problem, like the fact that it allows human interaction in the process of finding a solution ([8]).

2 The Timetabling Problem

We attempt to describe the timetabling problem in a very flexible way such that the model we create could be applied in an uniform manner in any domain that requires this specific type of planning.

The main entities that we deal with are: *events*, *actors*, *resources*, *restrictions* and a *temporal domain* that contains the available time-slots.

2.1 The Events

We call *events* the atomic activities that must be scheduled. The property of atomicity specifies the fact that we are developing a model in which the events are continuous, they cannot be interrupted and resumed later. This is the most common situation in timetabling problems. However, there may be situations, like planning the proceedings of a conference or workshop, when activities are not continuous but fragmented during one or several days. In these cases, because the length of the fragments are known, we will use multiple events to describe one complex activity and add appropriate constraints.

We have considered two categories of events:

- *Iterative* - the most common situation in schools or universities; here the temporal domain is usually the carthesian product of the available week's days and the possible starting hours, for instance:

$$\{Monday, \ldots, Friday\} \times \{08:00, \ldots, 19:00\}$$

  Most of the events will repeat by a specified number of times, usually the number of weeks in a semester, and the timetable will be complete once we know the starting date and the number of repetitions.

- *Unique* - used for representing events that will occur only once, like exams for instance; in this case, the temporal domain contains explicitly specified pairs of dates and hours.

Regardless of the nature of the timetabling problem, the main features of an event are:

- *the length* - the required time for its completion, measured in an abstract manner;
- *the number of repetitions*, only for the events that are iterative;
- *the frequency*, once a week or once at two weeks, only for the events that are iterative;
- *the participants* that will attend to this event;
- *the resources* that are required, described at generic level;
- *the constraints* - specifications that will restrict the possible solutions for scheduling this event.

Scheduling an event involves the following operations:

- setting a value from the temporal domain: "the class x will be held at the moment t";
- assigning the resources: "the class x will take place in the room s";

- satisfying the constraints that apply to this event.

We say that an event is *resolved* if it has been scheduled.

Let e an event and T the temporal domain of the problem. If e is resolved, we note $start(e) \in T$ the moment when it begins and $end(e) \in T$ the moment when it is completed, otherwise $start(e)$ and $end(e)$ are undefined. We also note $week(e) \in \{0, 1, 2\}$ the week in which the event will take place (0 - all weeks, 1 - odds, 2 - even).

The fact that two events e_1 and e_2 are *concurrent* can be written as a disjunction $concurrent(e_1, e_2) = C1 \vee C2$, where

$$(C_1)\ start(e_1) < end(e_2) \wedge start(e_2) < end(e_1)$$

is the condition that the events will not overlap inside a week and

$$(C_2)\ week(e_1) = 0 \vee week(e_2) = 0 \vee week(e_1) = week(e_2)$$

is the condition that checks that the events take place at the same parity. We say that two events are *independent* either if at least one of them is not resolved or they are not concurrent.

2.2 The Temporal Domain

We call *temporal domain* or *temporal space* a set of integers $T = \{t_1, ..., t_n\}$ representing in an abstract manner time-slots of a specified calendar. The events that must be scheduled will be assigned values from this domain in order to be considered resolved. In this representation, the integer 1 will signify the temporal unit of the domain and the events will have their lengths specified as a number of temporal units, as oppossed to specifying them in minutes. Each temporal domain will be assigned a mapping function responsible with transforming its elements into real dates.

Let us consider a common representation of a temporal domain in the case of a timetable whose activities are iterative over a specified number of weeks. In this situation the structure of the week is the same throughout the whole period, an event beeing scheduled on a certain day and at a certain hour (each day has a start hour and an end hour). Let us note n the number of temporal units available every day and m the number of days in a week. The domain would contain all the integers in the interval $[0, nm - 1]$, as suggested in the following graphical representation:

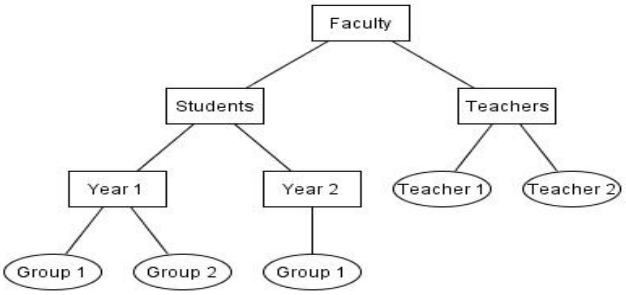
0	n	$\dots$	$n(m-1)$
1	$n+1$	$\dots$	$n(m-1)+1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	$2n-1$	$\dots$	$nm-1$

If we note h_0 the hour when the classes begin and d the length in minutes of a temporal unit, then for an element $t \in T$ the mapping function will be defined as $date(t) = (day(t), hrs(t), min(t))$, where $day(t) = 1 + [t/n]$, $hrs(t) = h_0 + (t - n[t/n])(d/60)$ and $min(t) = (t - n[t/n]) - 60(hrs(t) - h_0)$.

2.3 The Actors

Each event involves a number of entities, some of them beeing assigned as preordained properties of the event, while others are alloted dynamically depending on various conditions. We call *actor* or *participant* an entity that is assigned to some event *prior to* the process of creating the timetable, for instance a teacher or a group of students. Usually, actors are shared among events so they will be subject to constraints that prevent, for example, an actor beeing in two places at the same time. The set of actors may contain elements that are not independent regarding the inclusion, so representing actors in a unified structure is very important for determing the relationships between them. From the point of view of the *event-actor* association, we shall consider that an event e may have any number of actors, usually at least one, noted as $actors(e)$. In case of university timetables there are many situations when more than one group of students attend a lecture or more teachers have to participate to a certain event, such as a scientific meeting.

We define the *inclusion tree* as a structure that both enumerates all the actors that will participate to the timetable's events and surprises the relation of inclusion between them. The leaves of the tree will represent *atomic actors* that is persons or groups of persons that are regarded as a whole and the internal nodes will represent *composite actors*, precisely the union of all their leaves. An example of such a tree is given below:



An event may have actors assigned to it from any level of the inclusion tree, either atomic or composite. We say that two actors a_1, a_2 are *independent* if $a_1 \cap a_2 = \emptyset$. It is easy to see that if a_1 and a_2 are not independent then there exists a path from the root of the inclusion tree to a leaf, that contains both a_1 and a_2. Of course, when we schedule two events at the same time their actors must be independent.

2.4 The Resources

We call *resource* or *specific resource* an entity that is assigned to some event dynamically *during* the process of creating the timetable, for instance a room, a projector or some other didactic equipment. *The capacity* of a resource is the number of events that have this resource assigned to them and can be held at the same time. Usually, the capacity is 1 and this makes sense especially for rooms, where obviously only one event can be placed at a specific time. When it is not 1, the capacity of a resource represents a number of concrete items handled in an uniform manner; for instance the resource may be "Projector" with capacity 3, meaning that there are actally 3 different projectors available in the storehouse. If we place 4 courses on Monday at 08:00 and all require projectors, then we have a problem. We call a resource with capacity 1 *simple*, otherwise we say that it is *cumulative*.

A *generic* resource is either a single specific resource or a set of specific resources. For instance, the resource "Laboratories" may be the set of all available laboratories, say $\{L_1, L_2, L_3\}$.

An event may require some generic resources in different quantities and this is established at design time, before we begin the process of creating the timetable. We note $res(r)$ the resources required by an event e. The event e must be held in a laboratory, so it has assigned the generic resource "Laboratories". A resolved event must be assigned specific resources, in the same quantities as requested, so when we place the event e on the timetable we have to choose one of $\{L_1, L_2, L_3\}$ and assign it to e.

3 Constraint Satisfaction Problems

Once we have an informal description of the timetabling problem, we must represent it in a CSP specific manner.

A *constraint network* ([6]) is a triplet $\mathcal{R} = (X, \mathcal{D}, \mathcal{C})$ where:

- $X = \{x_1, ..., x_n\}$ is a finite set of variables;
- $\mathcal{D} = \{D_1, ..., D_n\}$ represents finite domains that are associated to variables of X;
- $\mathcal{C} = \{C_1, ..., C_t\}$ is a finite set of constraints.

A *constraint* C_i is a relation R_i on a subset of variables S_i, $S_i \subseteq X$, which denotes their simultaneous legal value assignments. In the above definition there is no restriction on the types of variables, their domains may be integers, strings or anything else. There are also no specifications on how constraints are defined. The purpose of a constraint is to restrict the possible values a variable x_i can be assigned from D_i.

The *instantiation* of a variable is the process of assigning it to a value from its domain.

If $\mathcal{R}$ has at least one solution, we say that it is *satisfiable* or *consistent*.

A partial instantiation of a set of variables S is *consistent* if and only if it satisfies all the constraints defined only over variables already instantiated.

In order to represent the timetablig problem as a network of constraints we have to identify the variables, the domains and the constraints. As in most cases, there are more than one formal models of the real life problem and, even if they are equivalent from a theoretical point of view, they may have an important impact in the process of obtaining the solution. Two main directions emerge:

a. the variables represent *events* and their values will be elements from the temporal domain;

b. the variables represent *time-slots* and their domains will be the set of events.

In our approach we have chosen the first representation, not only because it allows us to include in the model events that are not iterative but because we wanted to keep the size of the variable domains smaller than the number of variables. Empirically, this seems to help the solving process.

3.1 Variabiles and Domains

For each activity e we consider a single variable noted also with e. Normally, the domains of the variables would be sets of elements of the temporal domain of the problem. But, because each event requires a set of resources, such as rooms or didactic equipment, it proved to be better to include these specifications into the domain of the variables.

Let us consider a variable e, $T = \{t_1, t_2, ..., t_k\}$ the temporal space of the problem and $\{R_i | i = 1, p\}$ the required resources. Then, we represent the domain of e as the carthesian product:

$$D_e = T \times R_1 \times R_2 \times ...R_p$$

For example, let us consider the following partial description of an event: *"The class will be held in one of the laboratories* $\{L_1, L_2\}$*, it requires a projector* P *and one of the special equipments* $\{E_1, E_2\}$*"*. If the temporal space would be $T = \{t_1 = Monday(10-12), t_2 = Tuesday(12-14)\}$ then the domain of the event's variable will contain eight elements, respectively:
$\{(t_1, R_1, P, E_1), (t_1, R_1, P, E_2), (t_1, R_2, P, E_1), (t_1, R_2, P, E_2),$
$(t_2, R_1, P, E_1), (t_2, R_1, P, E_2), (t_2, R_2, P, E_1), (t_2, R_2, P, E_2), \}$

Most of the times, the number of resources required by an event is small and this prevents the domains of the variables to become oversized.

If the event takes place once at two weeks then its domain will have to reflect this also. In this case, the domain we consider for the event is:

$$D_e = W \times T \times R_1 \times R_2 \times ...R_p$$

where $W = \{1, 2\}$ represents the week (1-odd, 2-even).

3.2 The Constraints

The constraints must represent in the first place the restrictions of the timetable that should not be violated at any time, like the fact that an actor cannot participate at two events at the same time or a simple resource cannot be assigned simultaneously to different activities. These are the *hard constraints*.

In order to create a timetable that is to be accepted by all the participants it is not enough to satisfy only the hard constraints. Nobody will be happy if they have 12 hours of classes in a row or a day that is

very fragmented. Of course, teachers or special study groups might have preferences regarding the time-slots that are acceptable for them in order to attend their classes. However, these type of restrictions could be violated if they lead to the imposibility of creating the timetable (suppose for instance that two teachers require the same room and the same time-slot for their classes). Because of that, we call them *soft constraints* ([17]).

The classical model of constraint satisfaction is defined over the premises that identifying a solution means satisfying all the constraints of the problem. But there are many real life situations that cannot be solved this way, either because they are *over-constrained* ([9]) and thus not consistent or because their restrictions cannot be imposed in a *yes-or-no* manner. A good example of such a problem is the timetabling problem. In a "perfect" solution all these preferences would be satisfied but in most cases this is not feasible and we are concerned in creating a timetable that is "as good as possible", in other words minimizing somehow the number or the magnitude of the constraints that are not satisfied. Considerring the soft constraints, the timetabling problem is usually represented as an optimization problem that uses some *valuation structure* in order to determine when a solution is better than another. This structure must specify levels of preferences that are assigned to constraints, meaning how bad is it if a constraint is violated, and an operator that "combines" levels of preferences, specifying how good is an instantiation with respect to satisfying all the constraints of the network. The result is called the *degree of satisfaction* offered by an instantiation.

From the practical point of view we have designed an XML representation of the restrictions that apply to the timetabling problem. Each actor will have a *descriptor file* that formalizes its preferences and also the resources required by his events. A special parser will transform these descriptions into constraints specific to the OmniCS solver, which are actually Java classes. Each restriction might have a penalty assigned. The presence of the penalty produces a soft constraint, otherwise the constraint is considered to be hard.

Some of the constraints we have implemented are listed below.

Compatibility constraints

We say thet two resolved events e_1 and e_2 are *compatible* if they do not share actors or generic resources (decided at design time) or they are not concurrent (decided at runtime):

$$((actors(e_1) \cap actors(e_2) = \emptyset) \wedge (res(e_1) \cap res(e_2) = \emptyset)) \vee independent(e_1, e_2)$$

Thus, for every pair of events that share actors or resources, we must create such a hard constraint. Compatibility constraints represent the most numerous set of constraints so their implementation is critical for the performance of the solving process.

Resource Capacity Constraints

This type of constraint imposes that the number of concurrent events that share a cumulative resource does not exceed the capacity of that resource. Let r be a resource with capacity $c(r)$. Let us note $\{e_1, ..., e_k\}$ a set of concurrent events each using $p(e_i)$ units of resource r. The constraint states that:

$$\sum_{i=1,k} p(e_i) \leq c(r)$$

For every resource we have to create such a global constraint defined over the set of all variables representing events that require that resource.

Resource Disponibility Constraints

Another type of hard constraints is related to the fact that certain resources are not available throughout the whole temporal space of the problem. This is a situation common for faculties that have different timetables but share some rooms, each faculty having alloted only some days or time intervals to hold classes in that rooms.

For every restricted resource we have to create a global disponibility constraint over the events that require the resource, specifing also the *temporal subspace* that is available. This is a subset of the temporal space and can be represented as an array a having hd elements where h is the number of temporal units of a day and d is the number of days in a week, containing the following elements:

- 0 - allowed, with no penalty

- $+\infty$ - strictly forbidden
- $0 < c < \infty$ - allowed, with penalty c; in that case the constraint is soft, otherwise it is hard.

Actor Disponibility Constraints
The majority of participants have preferences regarding the time-slots that are to be assigned to their events. These types of preferences might be of common sense, like not having more than six hours in a day, or of personal nature, like not beeing able to hold classes too late in the evening. These restrictions can apply to all events of an actor or only to subsets of them. As in the case of resources, for each actor and each of his preferences we have to create a constraint that accepts as argument the temporal subspace that is available. We have designed a descriptive, easy to use method to specify this constraint:

```
<prefs actor="A">
    <event id="E1, E2">
        <res id="Laboratories"/>
        <include day="1,2" hour="8,10,12"/>
        <include day="5"/>
    </event>
    <event id="E3">
        <exclude day="1"/>
        <exclude day="2" penalty="10"/>
    </event>
</prefs>
```

If no penalties are specified, this unary constraints will have the immediate effect of reducing the domain of the variables and they will be removed from the network of constraints.

Sequence and Ordering Constraints
This type of constraints imposes that a set of events should take place in a compact block. We use the directive `linked` to specify that a set of events should form a sequence and we can add special directives like `order`, `first`, `last` to define a partial relation of ordering over these events. The exemple below states that all three events E_1, E_2, E_3 should take place in the same day, but E_1 should precede E_2.

```
<prefs actor="A">
    <event id="E1,E2,E3">
        <include day="1,2,3"/>
        <linked/>
    </event>
    <order events="E1,E2" first="E1" last="E2"/>
</prefs>
```

Limitation Constraints
These are constraints that limit the number of days or hours in which a set of events should take place. They are usually applied globally over the set of events of an actor, like in the following example that states that actor A should have classes at most two days of the week and the number of hours should not exceed six per day.

```
<prefs actor="A">
    <days max="2"/>
    <hours max="6"/>
</prefs>
```

The document type definition (DTD) of the XML model we have designed for representing the timetabling problem is described below:

```
<!ELEMENT prefs (event|days|hours|linked|order)*>
<!ATTLIST prefs actor CDATA #IMPLIED>
<!ELEMENT event (res|include|exclude|days|hours|linked|order)*>
```

```
<!ATTLIST event id CDATA #IMPLIED>
<!ELEMENT res EMPTY>
<!ATTLIST res id CDATA #IMPLIED>  <!-- Specifies the required resources -->
<!ELEMENT include EMPTY>
<!ATTLIST include                 <!-- Specifies a temporal subspace -->
    day CDATA #IMPLIED            <!-- One ore more days, comma separated -->
    hour CDATA #IMPLIED           <!-- One ore more hours, comma separated -->
    penalty CDATA #IMPLIED>       <!-- Penalty inflicted by not satisfying this -->
<!ELEMENT exclude EMPTY>
<!ATTLIST exclude                 <!-- Excludes a temporal subspace from domain -->
    day CDATA #IMPLIED            <!-- One ore more days, comma separated -->
    hour CDATA #IMPLIED           <!-- One ore more hours, comma separated -->
    penalty CDATA #IMPLIED>       <!-- Penalty inflicted by not satisfying this -->
<!ELEMENT linked EMPTY>
<!ATTLIST linked                  <!-- Specifies that some events form a sequence -->
    events CDATA #IMPLIED         <!-- One or more events, comma separated -->
    penalty CDATA #IMPLIED>       <!-- Penalty inflicted by not satisfying this -->
<!ELEMENT order EMPTY>
<!ATTLIST order                   <!-- Specifies that some events are ordered -->
    events CDATA #IMPLIED         <!-- One or more events, comma separated -->
    first CDATA #IMPLIED          <!-- The events that should take place before -->
    last CDATA #IMPLIED           <!-- The events that should take place after -->
    penalty CDATA #IMPLIED>       <!-- Penalty inflicted by not satisfying this -->
<!ELEMENT days EMPTY>
<!ATTLIST days                    <!-- Limitation of days within a week -->
    events CDATA #IMPLIED         <!-- One or more events, comma separated -->
    max CDATA #IMPLIED            <!-- Maximum number of days -->
    min CDATA #IMPLIED            <!-- Minimum number of days -->
    penalty CDATA #IMPLIED>       <!-- Penalty inflicted by not satisfying this -->
<!ELEMENT hours EMPTY>
<!ATTLIST hours                   <!-- Limitation of hours within a day -->
    events CDATA #IMPLIED         <!-- One or more events, comma separated -->
    max CDATA #IMPLIED            <!-- Maximum number of hours -->
    min CDATA #IMPLIED            <!-- Minimum number of hours -->
    penalty CDATA #IMPLIED>       <!-- Penalty inflicted by not satisfying this -->
```

4 The Solving Process

Once a problem is represented as a CSP instance we have to use a CSP solver in order to obtain a solution or to discover that it is inconsistent. We have used our own CSP Solver, called OmniCS ([7], [8]), which has some distinctive features that make it appropriate for approaching effectively the timetabling problem, like the abilities to solve both classical and soft constraint satisfaction problems in an uniform manner and to provide a mechanism for controlling the whole process of systematically searching the solution. The basic structure of the algorithm we have used to develop our solver OmniCS is *backtracking* that uses a flexible filter-and-propagate mechanism in order to obtain efficiency. In order to solve both classical and soft CSP instances the solver also uses a *branch-and-bound* algorithm.

The systematic search algorithm must make a series of decisions in order to explore the search space. A *forward strategy* is responsible with selection of the next variable that will be instantiated, thus defining a relation of ordering over the whole set of variables. However, this order is not static and can be specified during execution depending on specific conditions that can be evaluated only at runtime.

For this specific problem we have used a strategy that selects first variables with the smallest domains, thus reducing the width of the search space. We considered more important the temporal nature of the domain and for each event a *tightness* factor was computed, representing the degree of temporal

restrictions that apply to it, according to the preferences of its actors. A value close to 0 means very restrictive, while a value near 1 means very relaxed ($tightness \in [0, 1]$). The solver will instantiate variables in an order based on the tightness values of their corresponding events, starting with the lowest value. The second ordering criteria is *maximal adjacency* (we say that two events are *adjacent* if they share at least one actor). The solver selects the variable that has as many adjacent resolved events as possible; instantiating such a variable is likely to determine a better behavior of filter-and-propagate algorithms.

An *assignment strategy* is responsible with defining a relation of ordering over the values of a variable's domain. As in the case of the forward strategy, this relation can be defined dynamically during execution. We have used an assignment strategy that ensures a low fragmentation of the timetabling, from the actors point of view.

A *backward strategy* defines how the solver will select the variable from which it will resume the search process, after a failure was detected. Here, we have attempted to use an heuristic that attempts to identify the real variable whose current instantiation is responsible for the current failure and return to it rather than to return to the last chronologically variable instantiated before the one that provoked the failure. Unfortunately, this was time consuming and did not improve the solving process very much.

Using these strategies, for a problem with around 300 events and more than 20.000 constraints, that was not over-constraint, the solver OmniCS was able to find a solution in a couple of minutes with a very small number of backtracks.

However, creating a benchmark was not our interest here. In real life applications, there are a number of limitations concerning the process of creating the timetable as a monolith. Traditionally, the algorithms for solving this type of problems have been designed considering that the network created after the modelling phase is in its final form, that is assuming that it is static during the solving process. In a real life situation, it is not uncommon that in the middle of the solving process we receive additional information about the preferences of some participant that must be added to the existing ones. It would be very frustrating if we had to start the whole process again, wasting all the computational effort performed so far.

Fortunately, OmniCS offers the possibility to change the problem dynamically. Not only it permits to rewrite the problem at runtime, without having to restart the whole process again, but it allows human intervention in the search process, overwriting the default behavior of the solver. We have used this facility in order to stop/resume the solving process, to adjust the partial solution manually, to add new constraints or to remove deprecated ones.

Also, because the solver generates events as it searches for solutions and informs special objects called *observers* it was easy to integrate it into a graphical user interface, especially designed for the timetabling problem.

5 Conclusion

The timetabling problem is a typical planning problem that is very tedious to solve manually but also difficult to approach with an automated solving technique. This complexity comes mostly from the numerous types of restrictions that have to be formalized and taken into account in order to satisfy all the educational requirememts and the various preferences of the participants.

Our goal was to create a set of specifications that allows the modelling of the timetabling problem in a flexible, declarative manner that is suited not only for universities, but also for schools or other similar domains. Then we show how to transform this model into a CSP instance, more precisely into a network of constraints, and how to employ constraint programming techniques in order to obtain a solution or to find the inconsistency of the network. For that, we have used our own CSP solver, called OmniCS, that was invoked in a dynamical and interactive fashion. Eventually, the combination of automated solving and human judgement proved to be very effective from the practical point of view, shortening very much the process of creating the timetable.

The software system that has emerged from this study was used successfully at Faculty of Computer Science of Iaşi, România, solving timetabling problems with more than 300 variables and 20.000 constraints.

References

[1] D. Abramson. Constructing school timetables using simulated annealing: Sequential and parallel algorithms, 1991.

[2] Edmund Burke, David Elliman, and Rupert Weare. A genetic algorithm based university timetabling system. In *East-West Conference on Computer Technologies in Education, Crimea, Ukraine*, pages 35–40, 1994.

[3] E.K. Burke, D.G. Elliman, and R. Weare. A university timetabling system based on graph colouring and constraint manipulation, 1993.

[4] Choco. http://choco.sourceforge.net.

[5] Daniel Costa. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, 76(1):98–110, July 1994.

[6] Rina Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[7] Cristian Frasinaru. Omnics. http://omnics.sourceforge.net.

[8] Cristian Frasinaru. Basic techniques for creating an efficient csp solver. *Scientific Annals of Computer Science*, pages 83–112, 2007.

[9] Eugene C. Freuder. Partial constraint satisfaction. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89, Detroit, Michigan, USA*, pages 278–283, 1989.

[10] Hans-Joachim Goltz, Georg Kchler, and Dirk Matzke. Constraint-based timetabling for universities. In *In Proceedings INAP'98, 11th International Conference on Applications of Prolog*, pages 75–80, 1998.

[11] C.C Gotlieb. The construction of class-teacher timetables. In *In Proceedings of IFIP Congress, North-Holland Pub. Co., Amsterdam, 73-77*, 1962.

[12] Ilog. http://www.ilog.com.

[13] N.L. Lawrie. An integer programming model of a school timetabling problem. *The Computer Journal*, 12:307–316, 1969.

[14] Minion. http://minion.sourceforge.net/.

[15] Hana Rudova and Keith Murray. University course timetabling with soft constraints. In *Practice And Theory of Automated Timetabling IV. Springer-Verlag LNCS 2740*, pages 310–328. Springer, 2002.

[16] Andrea Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13:87–127, 1999.

[17] Thomas Schiex. Soft constraint processing. First International Summer School on Constraint Programming, July 2005.

[18] Edward Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.

Cristian Frăsinaru
University A.I.Cuza of Iaşi
Faculty of Computer Science
General Berthelot 16, 700483 Iaşi
ROMANIA
E-mail: *acf@info.uaic.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Improvements of Apriori Algorithm

Ioan Daniel Hunyadi

Abstract

In this paper, I describe the main techniques used to solve mining frequent itemset problems and give a comprehensive survey of the most influential algorithms that were proposed. The Apriori Algorithm was first algorithm proposed for mining frequent itemsets problem. I will present new improvements on Apriori algorithm which solve these mining problems more efficiently.

1 Introduction

Most of the established companies have accumulated masses of data from their customers for decades. With the e-commerce applications growing rapidly, the companies will have a significant amount of data in months not in years. The scope of Data Mining, also known as Knowledge Discovery in Databases (KDD), is to find trends, patterns, correlations, anomalies in these databases which can help us to make accurate future decisions.

Data mining is not magic. No one can guarantee that the decision will lead to good results. Data Mining only helps experts to understand the data and lead to good decisions. Data Mining is an intersection of the fields Databases, Artificial Intelligence and Machine Learning.

Since their introduction in 1993 by Argawal et al. [1], the frequent itemset and association rule mining problems have received a great deal of attention. Within the past decade, hundreds of research papers have been published presenting new algorithms or improvements on existing algorithms to solve these mining problems more efficiently.

2 Problem Descriptions

Let $I=(i_1,i_2,...,i_m)$ be a set of transactions. Each i is called an *item*. D is the set of transactions where each *transaction* T is a set of items (itemset) such that $T \subseteq I$. Every transaction has a unique identifier called the TID. An itemset having k items is called a *k-itemset*. Let X and Y be distinct itemsets. The *support* of an itemset X is the ratio of the itemsets containing X to the number of all itemsets. Let us define $|X|$ as the number of itemsets containing X and $|D|$ as the number of all items, $|X,Y|$ as the number of itemsets containing both X and Y. The support of itemset X is defined as follows:

$$support(X)=\frac{|X|}{|D|}$$

The rule $X \Rightarrow Y$ has *support s* if %*s* of the transactions in *D* contain *X* and *Y* together.

$$support(X \Rightarrow Y) = \frac{|X,Y|}{|D|}$$

Support measures how common the itemsets are in the database and *confidence* measures the strength of the rule. A rule is said to have *confidence c* if %*c* of the transactions that contains *X* also contains *Y*.

$$confidence(X \Rightarrow Y) = \frac{support(X,Y)}{support(X)}$$

Given a set of transactions *D* the task of association rule mining is to find rules $X \Rightarrow Y$ such that the support of the rule is greater than a user specified minimum support called *minsupp* and the confidence is greater than a user specified minimum called *minconf*. An itemset is called *frequend* if its support is greater than *minsupp*.

The collection of frequent itemsets in D which have their support grater then *minsupp* is denoted by *F*.

$$F=\{ X \subseteq I \mid support(X) \geq minsupp\}.$$

The task of association rule mining can be divided into two: In the first phase, the frequent itemsets are found using *minsupp*, and in the second phase, the rules are generated using *minconf*.

The collection of frequent and confident association rules with respect to *minsupp* and *minconf* is denoted by *R*.

$$R = \{X \Rightarrow Y \mid X,Y \subseteq I, X \cap Y = \{\}, X \cup Y \in F, confidence(X \Rightarrow Y) \geq minconf\}$$

The algorithms that implement association mining make multiple passes over the data. Most algorithms first find the frequent itemsets and then generate the rules accordingly. They find the large itemsets incrementally increasing itemset sizes and then counting the itemsets to see if they are large or not. Since finding the large itemsets is the hard part, research mostly focused on this topic.

3 Apriori Algorithm

The first algorithm to generate all frequent itemsets and confident association rules was the AIS algorithm by Agrawal et al. [1], which was given together with the introduction of this mining problem. Shortly after that, the algorithm was improved and renamed Apriori by Agrawal et al., by exploiting the monotonicity property of the support of itemsets and the confidence of association rules [3, 15].

For simplicity the items in transactions and itemsets are kept sorted in their lexicographic order unless stated otherwise. The itemset mining phase of the Apriori algorithm is given in Listing 1. I use the notation *X[i],* to represent the i^{th} item in *X*. The *k*-prefix of an itemset *X* is the *k*-itemset {*X[1], . . . ,X[k]*}.

Listing 1. Apriori algorithm – Itemset mining

```
Input: D, minsupp
Output: F
  C1={{i}|i ∈ I};
  k=1;
  while Ck≠{} do{
    //Compute the supports of all candidate itemsets
    forall transactions(tid,D) ∈ D
      forall candidate itemsets X ∈ Ck
        if ( X ⊆ I )
          X.support++;
    //Extract all frequent itemsets
    Fk = {X|X.support ≥ minsupp};
```

```
    //Generate new candidate itemsets
    forall X,Y ∈ F_k, X[i]=Y[i] for 1 ≤ i ≤ k-1, and X[k]<Y[k]{
        I = X ∪ {Y[k]};
        if (∀J ⊂ I,| J |= k, J ∈ F_k)
            C_{k+1}= C_{k+1} ∪ I;
    }
    k++;
}
```

The algorithm performs a breadth-first search through the search space of all itemsets by iteratively generating candidate itemsets C_{k+1} of size $k+1$, starting with $k = 0$. An itemset is a candidate if all of its subsets are known to be frequent. More specifically, C_1 consists of all items in I, and at a certain level k, all itemsets of size $k+1$ are generated. This is done in two steps. First, in the *join* step, F_k is joined with itself. The union $X \cup Y$ of itemsets $X,Y \in F_k$ is generated if they have the same $(k-1)$ - prefix. In the prune step, $X \cup Y$ is only inserted into C_{k+1} if all of its k-subsets occur in F_k.

To count the supports of all candidate k-itemsets, the database, which retains on secondary storage in the horizontal database layout, is scanned one transaction at a time, and the supports of all candidate itemsets that are included in that transaction are incremented . All itemsets that turn out to be frequent are inserted into F_k.

If the number of candidate $(k+1)$ - itemsets is too large to retain into main memory, the candidate generation procedure stops and the supports of all generated candidates is computed as if nothing happened. But then, in the next iteration, instead of generating candidate itemsets of size $k+2$, the remainder of all candidate $(k+1)$ - itemsets is generated and counted repeatedly until all frequent itemsets of size $k + 1$ are generated.

4. Data Structures

The candidate generation and the support counting processes require an efficient data structure in which all candidate itemsets are stored since it is important to efficiently find the itemsets that are contained in a transaction or in another itemset.

4.1. Hash-tree

In order to efficiently find all k-subsets of a potential candidate itemset, all frequent itemsets in F_k are stored in a hash table.

Candidate itemsets are stored in a hash-tree [2]. A node of the hash-tree either contains a list of itemsets (a leaf node) or a hash table (an interior node). In an interior node, each bucket of the hash table points to another node. The root of the hash-tree is defined to be at depth 1. An interior node at depth d points to nodes at depth $d+1$. Itemsets are stored in leaves.

When we add a k-itemset X during the candidate generation process, we start from the root and go down the tree until we reach a leaf. At an interior node at depth d, we decide which branch to follow by applying a hash function to the $X[d]$ item of the itemset, and following the pointer in the corresponding bucket. All nodes are initially created as leaf nodes. When the number of itemsets in a leaf node at depth d exceeds a specified threshold, the leaf node is converted into an interior node, only if $k>d$.

In order to find the candidate-itemsets that are contained in a transaction T, we start from the root node. If we are at a leaf, we find which of the itemsets in the leaf are contained in T and increment their support. If we are at an interior node and we have reached it by hashing the item i, we hash on each item that comes after i in T and recursively apply this procedure to the node in

the corresponding bucket. For the root node, we hash on every item in *T*. An example of hash-tree structure for five items is presented in figure 1.

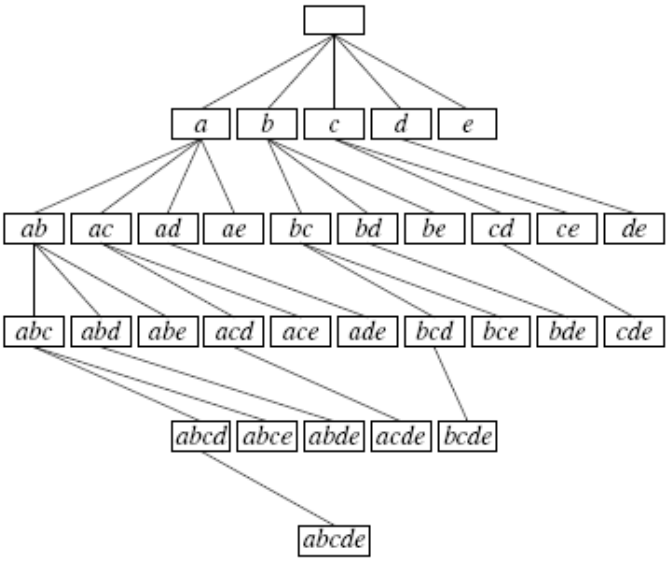


Figure 1. Hash-tree structure

4.2. Trie

Another data structure that is commonly used is a trie (or prefix-tree) [5,7, 8, 6]. In a trie, every *k*-itemset has a node associated with it, as does its (*k−1*) - prefix. The empty itemset is the root node. All the 1-itemsets are attached to the root node, and their branches are labeled by the item they represent. Every other *k*-itemset is attached to its (*k−1*) - prefix. Every node stores the last item in the itemset it represents, its support, and its branches. The branches of a node can be implemented using several data structures such as a hash table, a binary search tree or a vector. An example of prefix-tree structure for five items is presented in figure 2.

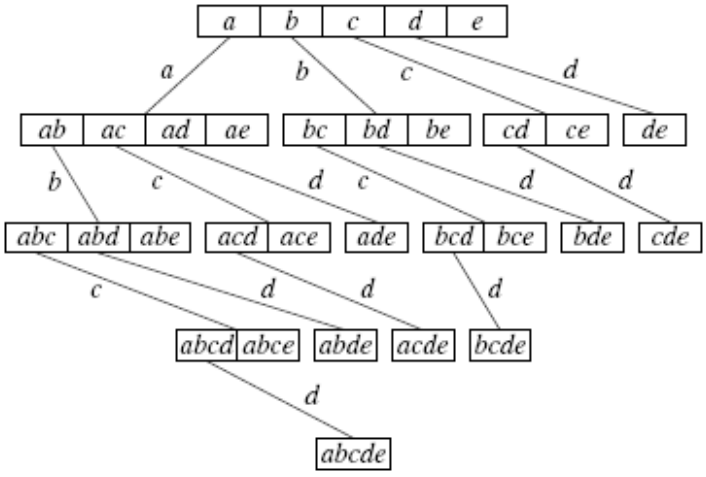


Figure 2. Prefix-tree structure

At a certain iteration *k*, all candidate *k*-itemsets are stored at depth *k* in the trie. In order to find the candidate-itemsets that are contained in a transaction *T*, we start at the root node. To process a transaction for a node of the trie, (1) follow the branch corresponding to the first item in the

transaction and process the remainder of the transaction recursively for that branch, and (2) discard the first item of the transaction and process it recursively for the node itself. This procedure can still be optimized, as is described in [7].

Also the join step of the candidate generation procedure becomes very simple using a trie, since all itemsets of size k with the same $(k-1)$ – prefix are represented by the branches of the same node (that node represents the $(k-1)$ - prefix). Indeed, to generate all candidate itemsets with $(k-1)$ – prefix X, we simply copy all siblings of the node that represents X as branches of that node. Moreover, we can try to minimize the number of such siblings by reordering the items in the database in support ascending order [7, 8, 6]. Using this heuristic, we reduce the number of itemsets that is generated during the join step, and hence, we implicitly reduce the number of times the prune step needs to be performed. Also, to find the node representing a specific k-itemset in the trie, we have to perform k searches within a set of branches. Obviously, the performance of such a search can be improved when these sets are kept as small as possible.

An in depth study on the implementation details of a trie for Apriori can be found in [7].

5. Optimizations

A lot of other algorithms proposed after the introduction of Apriori retain the same general structure, adding several techniques to optimize certain steps within the algorithm. Since the performance of the Apriori algorithm is almost completely dictated by its support counting procedure, most research has focused on that aspect of the Apriori algorithm. The performance of this procedure is mainly dependent on the number of candidate itemsets that occur in each transaction.

5.1. AprioriTid, AprioriHybrid

Together with the proposal of the Apriori algorithm, Agrawal et al. [3, 2] proposed two other algorithms, AprioriTid and AprioriHybrid. The AprioriTid algorithm reduces the time needed for the support counting procedure by replacing every transaction in the database by the set of candidate itemsets that occur in that transaction. This is done repeatedly at every iteration k. The adapted transaction database is denoted by $\overline{C_k}$. The algorithm is given in Listing 2.

More implementation details of this algorithm can be found in [4]. Although the AprioriTid algorithm is much faster in later iterations, it performs much slower than Apriori in early iterations. This is mainly due to the additional overhead that is created when $\overline{C_k}$ does not fit into main memory and has to be written to disk. If a transaction does not contain any candidate k-itemsets, then $\overline{C_k}$ will not have an entry for this transaction. Hence, the number of entries in $\overline{C_k}$ may be smaller than the number of transactions in the database, especially at later iterations of the algorithm. Additionally, at later iterations, each entry may be smaller than the corresponding transaction because very few candidates may be contained in the transaction. However, in early iterations, each entry may be larger than its corresponding transaction.

Therefore, another algorithm, AprioriHybrid, has been proposed [3,2] that combines the Apriori and AprioriTid algorithms into a single hybrid. This hybrid algorithm uses Apriori for the initial iterations and switches to AprioriTid when it is expected that the set $\overline{C_k}$ fits into main memory. Since the size of $\overline{C_k}$ is proportional with the number of candidate itemsets, a heuristic is used that estimates the size that $\overline{C_k}$ would have in the current iteration. If this size is small enough and there are fewer candidate patterns in the current iteration than in the previous iteration, the algorithm decides to switch to AprioriTid

Listing 2. AprioriTid algorithm

```
Input: D, minsupp
Output: F
  Compute F1 of all frequent items;
  C̄1 =D; (with all items not in F1 removed)
  k=2;
  while Fk-1≠{} do{
    Compute Ck of all candidate k-itemsets
    C̄k ={};
    // Compute the supports of all candidate itemsets
    forall transactions(tid, T) ∈ Ck {
      CT = {};
      forall X∈ Ck
        if ( {X[1],...,X[k−1]} ∈ T ∧ {X[1],...,X[k−2], X[k]} ∈ T ){
            C_T = C_T ∪ {X};
            X.support++;
        }
      if (CT≠{})
        C̄k = C̄k ∪ {(tid, C_T)}
    }
    Extract Fk of all frequent k-itemsets;
    k++;
}
```

5.2 Counting candidate 2-itemsets

Shortly after the proposal of the Apriori algorithms described before, Park et al. proposed another optimization, called DHP (Direct Hashing and Pruning) to reduce the number of candidate itemsets [13]. During the k^{th} iteration, when the supports of all candidate k-itemsets are counted by scanning the database, DHP already gathers information about candidate itemsets of size $k + 1$ in such a way that all $(k + 1)$-subsets of each transaction after some pruning are hashed to a hash table. Each bucket in the hash table consists of a counter to represent how many itemsets have been hashed to that bucket so far. Then, if a candidate itemset of size $k+1$ is generated, the hash function is applied on that itemset. If the counter of the corresponding bucket in the hash table is below the minimal support threshold, the generated itemset is not added to the set of candidate itemsets. Also, during the support counting phase of iteration k, every transaction trimmed in the following way. If a transaction contains a frequent itemset of size $k+1$, any item contained in that $k+1$ itemset will appear in at least k of the candidate k-itemsets in C_k. As a result, an item in transaction T can be trimmed if it does not appear in at least k of the candidate k-itemsets in C_k. These techniques result in a significant decrease in the number of candidate itemsets that need to be counted, especially in the second iteration. Nevertheless, creating the hash tables and writing the adapted database to disk at every iteration causes a significant overhead.

Although DHP was reported to have better performance than Apriori and AprioriHybrid, this claim was countered by Ramakrishnan if the following optimization is added to Apriori [14]. Instead of using the hash-tree to store and count all candidate 2-itemsets, a triangular array C is created, in which the support counter of a candidate 2-itemset $\{i, j\}$ is stored at location $C[i][j]$. Using this array, the support counting procedure reduces to a simple two level for-loop over each transaction. A similar technique was later used by Orlando et al. in their DCP and DCI algorithms [11, 12].

Since the number of candidate 2-itemsets is exactly $\binom{|F_1|}{2}$, it is still possible that this number is too large, such that only part of the structure can be generated and multiple scans over the

database need to be performed. A lot of candidate 2-itemsets do not even occur at all in the database, and hence, their support remains 0. Therefore, we propose the following optimization. When all single items are counted, resulting in the set of all frequent items F_1, we do not generate any candidate 2-itemset. Instead, we start scanning the database, and remove from each transaction all items that are not frequent. Then, for each trimmed transaction, we increase the support of all candidate 2-itemsets contained in that transaction. However, if the candidate 2-itemset does not yet exists, we generate the candidate itemset and initialize its support to 1. In this way, only those candidate 2-itemsets that occur at least once in the database are generated.

5.3. Support lower bounding

Apart from the monotonicity property, it is sometimes possible to derive information on the support of an itemset, given the support of all of its subsets. The first algorithm that uses such a technique was proposed by Bayardo in his MaxMiner and Apriori-LB algorithms [6].

In practice, this lower bound can be used in the following way. Every time a candidate $(k + 1)$ - itemset is generated by joining two of its subsets of size k, we can easily compute this lower bound for that candidate. Indeed, suppose the candidate itemset $X \cup \{i_1, i_2\}$ is generated by joining $X \cup \{i_1\}$ and $X \cup \{i_2\}$, we simply add up the supports of these two itemsets and subtract the support of X. If this lower bound is higher than the minimal support threshold, then we already know that it is frequent and hence, we can already generate candidate itemsets of larger sizes for which this lower bound can again be computed. Nevertheless, we still need to count the exact supports of all these itemsets, but this can be done all at once during the support counting procedure. Using the efficient support counting mechanism as I described before, this optimization could result in significant performance improvements.

Calders and Goethals presented a generalization of all these techniques resulting in a system of deduction rules that derive tight bounds on the support of candidate itemsets [9]. These deduction rules allow for constructing a minimal representation of all frequent itemsets, but can also be used to efficiently generate the set of all frequent itemsets. Unfortunately, for a given candidate itemset, an exponential number of rules in the length of the itemset need to be evaluated. The rules presented in this section, which are part of the complete set of derivation rules, are shown to result in significant performance improvements, while the other rules only show a marginal improvement.

5.4. Dynamic Itemset Counting

The DIC algorithm, proposed by Brin et al. tries to reduce the number of passes over the database by dividing the database into intervals of a specific size [8]. First, all candidate patterns of size 1 are generated. The supports of the candidate sets are then counted over the first interval of the database. Based on these supports, a new candidate pattern of size 2 is already generated if all of its subsets are already known to be frequent, and its support is counted over the database together with the patterns of size 1. In general, after every interval, candidate patterns are generated and counted. The algorithm stops if no more candidates can be generated and all candidates have been counted over the complete database. Although this method drastically reduces the number of scans through the database, its performance is also heavily dependent on the distribution of the data.

Although the authors claim that the performance improvement of reordering all items in support ascending order is negligible, this is not true for Apriori in general. Indeed, the reordering used in DIC was based on the supports of the 1-itemsets that were computed only in the first interval. Obviously, the success of this heuristic also becomes highly dependent on the distribution of the data.

The CARMA algorithm (Continuous Association Rule Mining Algorithm), proposed by Hidber [10] uses a similar technique, reducing the interval size to 1. More specifically, candidate itemsets are generated from every transaction. After reading a transaction, it increments the supports of all candidate itemsets contained in that transaction and it generates a new candidate itemset contained in that transaction, if all of its subsets are suspected to be relatively frequent with respect to the number of transactions that has already been processed. As a consequence, CARMA generates a lot more candidate itemsets than DIC or Apriori. Additionally, CARMA allows the user to change the minimal support threshold during the execution of the algorithm. After the database has been processed once, CARMA is guaranteed to have generated a superset of all frequent itemsets relative to some threshold which depends on how the user changed the minimal support threshold during its execution. However, when the minimal support threshold was kept fixed during the complete execution of the algorithm, at least all frequent itemsets have been generated. To determinate exact supports of all generated itemsets, a second scan of the database is required.

5.5. Sampling

The sampling algorithm, proposed by Toivonen [16], performs at most two scans through the database by picking a random sample from the database, then finding all relatively frequent patterns in that sample, and then verifying the results with the rest of the database. In the cases where the sampling method does not produce all frequent patterns, the missing patterns can be found by generating all remaining potentially frequent patterns and verifying their supports during a second pass through the database. The probability of such a failure can be kept small by decreasing the minimal support threshold. However, for a reasonably small probability of failure, the threshold must be drastically decreased, which can cause a combinatorial explosion of the number of candidate patterns.

6. Conclusions

A lot of people have implemented and compared several algorithms that try to solve the frequent itemset mining problem as efficiently as possible. Unfortunately, only a very small selection of researchers put the source codes of their algorithms publicly available such that fair empirical evaluations and comparisons of their algorithms become very difficult.

Different implementations of the same algorithms could still result in significantly different performance results. Different compilers and different machine architectures sometimes showed different behavior for the same algorithms. Also, different kinds of data sets on which the algorithms were tested showed remarkable differences in the performance of such algorithms.

In this paper I presented some algorithms which made a significant contribution to improve the efficiency of frequent itemset mining. Also, I propose two implementations for Apriori and AprioriTid algorithms.

References

[1] R. Agrawal, T. Imielinski, and A.N. Swami, Mining association rules between sets of items in large databases*, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, volume 22(2) of SIGMOD Record, pages 207–216. ACM Press, 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. MIT Press, 1996.

[3] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, *Proceedings 20th International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann, 1994.

[4] R. Agrawal and R. Srikant, *Fast algorithms for mining association rules*. IBM Research Report RJ9839, IBM Almaden Research Center, San Jose, California, June 1994.
[5] A. Amir, R. Feldman, and R. Kashi. *A new and versatile method for association generation. Information Systems*, 2:333–347, 1997.
[6] R.J. Bayardo, Efficiently mining long patterns from databases, *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, volume 27(2) of SIGMOD Record, pages 85–93. ACM Press, 1998.
[7] C. Borgelt and R. Kruse, Induction of association rules: Apriori implementation, *Proceedings of the 15th Conference on Computational Statistics*, pages 395–400,
[8] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, Dynamic itemset counting and implication rules for market basket data*, Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, volume 26(2) of SIGMOD Record, pages 255–264. ACM Press, 1997.
[9] T. Calders and B. Goethals, Mining all non-derivable frequent itemsets, Proceedings *of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, volume 2431 of Lecture Notes in Computer Science, pages 74–85. Springer, 2002.
[10] C. Hidber, Online association rule mining, Proceedings *of the ACM SIGMOD International Conference on Management of Data*, volume 28(2) of SIGMOD Record, pages 145–156. ACM Press, 1999.
[11] S. Orlando, P. Palmerini, and R. Perego, Enhancing the apriori algorithm for frequent set counting, *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery*, volume 2114 of Lecture Notes in Computer Science, pages 71–82. Springer, 2001.
[12] S. Orlando, P. Palmerini, R. Perego, and F. Silvestri, Adaptive and resource-aware mining of frequent sets, *Proceedings of the IEEE International Conference on Data Mining. IEEE Computer Society*,2002.
[13] J.S. Park, M.S. Chen, and P.S. Yu, An effective hash based algorithm for mining association rules, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, volume 24(2) of SIGMOD Record, pages 175–186. ACM Press, 1995.
[14] R. Srikant, *Fast algorithms for mining association rules and sequential patterns,* PhD thesis, University of Wisconsin, Madison, 1996.
[15] R. Srikant and R. Agrawal, *Mining generalized association rules*, 1999, pages 407–419.
[16] H. Toivonen, Sampling large databases for association rules, *Proceedings 22nd International Conference on Very Large Data Bases*, pages 134–145. Morgan Kaufmann, 1996.
[17] HAN Feng, ZHANG Shu-mao, DU Ying-shuang, The analysis and improvement of Apriori algorithm, *Journal of Communication and Computer*, ISSN1548-7709, USA, Sep. 2008, Volume 5, No.9 (Serial No.46), pages 12-18

Ioan Daniel Hunyadi
"Lucian Blaga" University of Sibiu
Informatics Departament
Dr. Ratiu No. 5-7, Sibiu
Romania
daniel.hunyadi@ulbsibiu.ro

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Departure model and its mathematical expression

Jan Krcal

Abstract

For control of the traffic on traffic lights controlled intersections, it is very important to understand the behavior of the traffic flow. Particularly today it can be very difficult to regulate traffic networks, which are very complicated. Movement of vehicles on traffic lights controlled intersections is determined by many characteristics, which are essential for regulating or modelling of the traffic flow. Departure model is just one of these characteristics. It is about finding the time the vehicles need to leave the space in front of the stop line after the appearance of the green light. The resulting time is affected by many factors that are more or less relevant. Our aim is to create a new mathematical model, which can take into account these factors, find dependence between them and thus be more accurate for the specific intersections.

1 Introduction

If we want to examine behavior of a traffic flow and thus dynamics of vehicles´ movement on the traffic lights controlled intersection, it is necessary to divide the system into several parts (sub-systems) and to examine them separately. Traffic flow is like an organism, which is composed of heterogeneous parts. To better understand it, it is useful to work with its smaller parts. Movement of vehicles on a traffic lights controlled intersection can thus be divided into the following parts:

- arrival of a vehicle on the intersection;
- departure of a vehicle from the "pre-shift" area;
- movement of a vehicle inside the intersection;
- departure of a vehicle from the intersection.

Particularly the departure model helps us characterize dynamics of vehicles´ movement upon their departure from the "pre-shift" area. In simple terms, the departure model is determined by departure times (the so-called entry times), which help us determine the number of vehicles able to move through the intersection during the green light phase.

The first departure model was developed by Greenshilds ([7]) already in 1947. Since then, several other departure models were developed, for example by Professor Medelská ([6]) or by Webster. Each of them works with their own departure times, which are determined empirically. It is therefore probably not necessarily exact to refer to them as departure models (we will stay with this term however), but rather as to calculated average arrival times for a specific sequence of vehicles in a line. Professor Medelská attempted a mathematical departure model as well ([6]). She subsequently presented the measured and averaged entry times with the help of a quadratic equation.

Since the last (revised) departure model was developed in 1979, these models are not capable to adequately reflect current circumstances of road transportation.

This was fully confirmed by a research proposal of Ministry of Education, Youth, and Sports of the Czech Republic (MSM 21260025 – Modeling of Traffic Processes). Its main aim was to create such a simulation program, which would be capable of modeling different types of controlling an intersection. This simulation program allows for choosing of one of the known departure models. The program was naturally confronted with the real data obtained at an observed intersection and its outcome was biased particularly because it made use of departure models no longer corresponding to a reality.

Solution of this problem lies in closer examination of patterns in a departure model and in attempting to define own mathematical apparatus, which would better reflect contemporary traffic needs.

2 Departure model

Departure model is often defined by the values of entry times of vehicles entering the area of traffic light controlled intersection during the green light interval. In other words, the entry time of the specific vehicle in the row is the time required for passage of this vehicle (of its front part) through the "stop" line related to the beginning of the green light interval. We can thus say that the departure model describes behavior of vehicles standing in a row in front of a light controlled intersection and their subsequent movement at the green light until they cross the "stop" line.

Entry time can be divided into these parts:

- reaction of a driver to signal "go";
- time necessary for acceleration;
- time necessary for clear-out of the "pre-shift" area until the point when the vehicle did not yet cross the "stop" line with its front bumper.

As already stated, several authors created their own departure model and these models bear their names.

According to ([6]), departure of vehicles can be characterized by the following patterns:

- The second vehicle in the row needs most of the time, because the driver reacts only to the movement of the first vehicle.
- Vehicles from the fifth position onwards move at equal time intervals as long as a row of waiting vehicles exists.
- If there is no longer a row of waiting vehicles and there is still signal "go", vehicles enter the intersection randomly.

The following table illustrates the departure model (entry times) by different authors:

Vehicle sequence	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Each additional**
Greenschields	3.8	6.9	9.6	12.0	14.2	16.3	18.4	+ 2.1
Fischer	3.1	5.4	7.5	9.5	11.4	13.3	15.2	+ 1.9
Medelská	2.3	5.5	8.3	10.8	13.1	15.2	17.2	+ 2.0
Medelská (revised)	1.2	3.9	6.6	9.1	11.6	14.0	16.3	+ 2.3
ÚSMD	3.6	6.5	8.9	11.2	13.4	15.4	17.4	+ 2.0
Webster	1	2	3	4	5	6	7	+ 1.0

Table 1: Departure models of different authors

This means that for example according to Fischer, vehicle, which is the fourth in the line, crosses the intersection after the signal "go" comes up in 9.5 sec.

3 Mathematical expression of our departure model

We assume the following conditions for our departure model. Departure of vehicles in a row is dependent on the departure of the previous vehicle (naturally with the exception of the first one), as well as on parameters, which influence its departure. We will consider the model in the following equation form:

$$y_t = \beta u_t + \kappa + \epsilon_t \tag{1}$$

where,

- $y_t = [y_{1;t}, y_{2;t}, \dots, y_{m;t}]'$ is the vector of modeled departure times of individual vehicles in the row where its length is expressed by m and the time index by t
- β is matrix of parameters
  - $\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \dots & \beta_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{m1} & \beta_{m2} & \dots & \beta_{mn} \end{bmatrix}$
  - n is the number of parameters
- $u = [u_{1;t}, u_{2;t,} \dots, u_{n;t}]'$ is the vector of values of variables influencing departures
- $\kappa = [\kappa_1, \kappa_2, \dots, \kappa_m]'$ is the model´s constant
- $\epsilon = [\epsilon_1, \epsilon_2, \dots, \epsilon_m]'$ is the noise with the median value of zero and covariance matrix R

Covariance matrix R is symmetric and positive definite and can be therefore unequivocally divided into form "LDL‘", where "L" is the lower rectangular matrix with numbers "1" on the main diagonal "D" and "D" is diagonal matrix.

$$R = LDL' = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \rho_{21} & 1 & 0 & \cdots & 0 \\ \rho_{31} & \rho_{32} & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \rho_{m1} & \rho_{m2} & \cdots & \rho_{mn-1} & 1 \end{bmatrix} \begin{bmatrix} \sigma_1^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3^2 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & \sigma_m^2 \end{bmatrix} \begin{bmatrix} 1 & \rho_{21} & \rho_{31} & \cdots & \rho_{m1} \\ 0 & 1 & \rho_{32} & \cdots & \rho_{m2} \\ 0 & 0 & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \rho_{mn-1} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Now, instead of the original noise ϵ_t, we will consider a new noise e_t, in such a way that $\epsilon_t = Le_t$ and noise has e_t a diagonal covariance matrix "D". It holds that:

$$E[\epsilon_t \epsilon_t'] = E[Le_t e_t' L'] = LE[e_t e_t']L' = R$$

Using an established substitution, it is possible to record the model of Equation 1 in this form

$$y_t = bu_t + k + Le_t$$

If we multiply the entire previous equation from the left by the matrix L^{-1}, which always exists and which is also lower triangular with numbers "1" on the main diagonal, we arrive at

$$L^{-1}y_t = L^{-1}\beta u_t + L^{-1}\kappa + e_t$$

If we create a substitution $a = -L^{-1} + I$, where

- I is a unit matrix or order m, then it is possible to record this model in the form

$$y_t = ay_t + bu_t + k + e_t \tag{2}$$

Whereas matrix a has on and above the main diagonal zeros, elements below the main diagonal are marked $a_{i,j}$. Noise e_t has elements, which are not mutually correlated. We can record the model as m of independent equations in the following manner.

$$y_{1;t} = \beta_{11}u_{1;t} + \beta_{12}u_{2;t} + \cdots + \beta_{1n}u_{n;t} + e_{1;t}$$
$$y_{2;t} = a_{21}y_{1;t} + \beta_{21}u_{1;t} + \beta_{22}u_{2;t} + \cdots + \beta_{2n}u_{n;t} + r_{21}e_{1;t} + e_{2;t}$$
$$y_{3;t} = a_{31}y_{1;t} + a_{32}y_{2;t} + \beta_{31}u_{1;t} + \beta_{32}u_{2;t} + \cdots + r_{31}e_{1;t} + r_{32}e_{2;t} + e_{3;t}$$

$$y_{m;t} = a_{m1}y_{1;t} + a_{m2}y_{2;t} + \cdots + a_{mm-1}y_{m-1;t} + \beta_{m1}u_{1;t} + \beta_{m2}u_{2;t} + \cdots + \beta_{mm-1}u_{m-1;t} + r_{m1}e_{1;t} + r_{m2}e_{2;t} + \cdots + r_{mm-1}e_{m-1;t} + e_{m;t}$$

where

- $e_{i;t}$ have zero expectations and dispersions σ_i^2.

This is the matrix expression of this model:

$$\begin{bmatrix} y_{1;t} \\ y_{2;t} \\ y_{3;t} \\ \vdots \\ y_{m;t} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ a_{m1} & a_{m2} & \cdots & a_{m(m-1)} & 0 \end{bmatrix} \begin{bmatrix} y_{1;t} \\ y_{2;t} \\ y_{3;t} \\ \vdots \\ y_{m;t} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1n} \\ b_{21} & b_{22} & \ldots & b_{2n} \\ b_{31} & b_{33} & \cdots & b_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \ldots & b_{mn} \end{bmatrix} \begin{bmatrix} u_{1;t} \\ u_{2;t} \\ u_{3;t} \\ \vdots \\ u_{n;t} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ r_{21} & 1 & 0 & \cdots & 0 \\ r_{31} & r_{32} & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ r_{m1} & r_{m2} & \cdots & r_{m(m-1)} & 1 \end{bmatrix} \begin{bmatrix} e_{1;t} \\ e_{2;t} \\ e_{3;t} \\ \vdots \\ e_{m;t} \end{bmatrix} \quad (3)$$

In equation 2 or 3, we arrive precisely at the mathematical application which we need. We model the departure of a vehicle, or rather its entry time, with dependence on departure of previous vehicles and other parameters which influence it.

4 Parameters influencing the departure model

It is apparent that entry times of vehicles are, besides the preceding vehicle, influenced by other factors, or parameters, which can vary for different vehicles. It is difficult, if not impossible, to define these parameters, because of their number and our capacity to measure them. We also need to consider whether or not the particular parameter indeed influences the entry time of the vehicle. This is very important, because it helps us to develop a suitable complete model. Among the measured parameters are, for example: type of the vehicle, weather, road surface, descent angle, geometrical order of the intersection, impact of vehicles in the opposite direction, or hypothetically whether the driver is male or female. We may also not forget those parameters, which are difficult to measure, such as health, age, psychological well-being, or alertness of the driver.

5 Real data and our departure model

We tested our model on a total of 2,254 vehicles. This examination is only a first one of many and data is still being collected. Nevertheless, even this preliminary evaluation reveals that the departure model, which we propose, differs from the other ones mentioned. Figure 1 displays comparison of our departure model with other models vis-à-vis the measured data. This comparison was undertaken using the method of least squares.

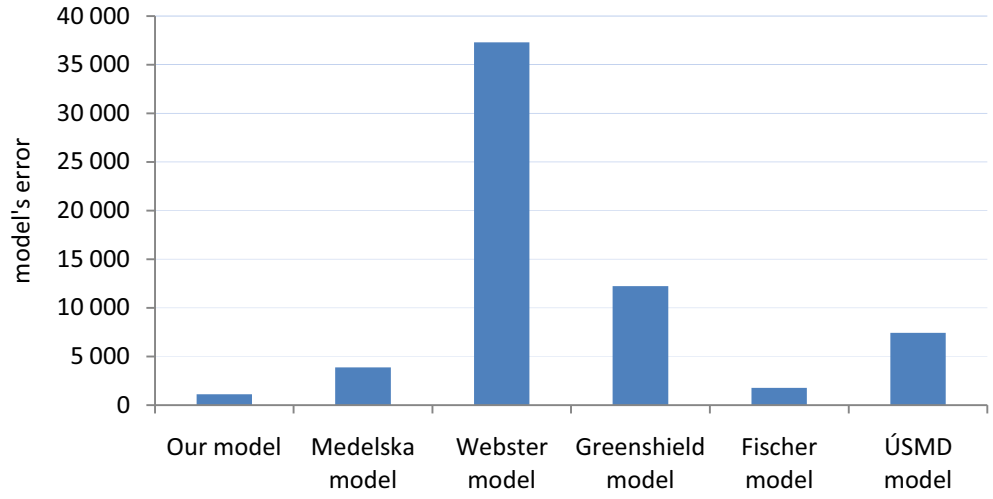


Fig. 1: Comparison of Departure models

If we compare our model with the departure model of Fischer, which shows the second smallest error value, we arrive at the following result: the model of Fischer is worse by approximately 62%, or rather the sum total of errors of the model of Fischer is worse than our departure model by 62%. If we compare our departure model with the one of Professor Medelská (revised), whose margin of error is the third smallest, we find that her departure model is worse than ours by 252%.

6 Conclusion

According to the preliminary results (see Figure), our departure model is significantly more accurate than other models. For example, the second most accurate model of Fischer, is worse than ours by 62%. The mathematical apparatus which we have created is adjusted as to incorporate into the departure model different parameters. It is thus very strong, because these parameters can render our departure model substantially more accurate. Based on different parameters we can then define departure model for different specific intersection, or, even better, for all intersections with identical parameters. Such variables must not be only physical (such as angle of the road), but also psychological. These are, however, difficult to measure. Queries, examination, and verification of these variables, which influence the departure of vehicles queuing in front of the light controlled intersection, is demanding and thus requires further research.

Acknowledgement: This paper is founded from the research project Czech Ministry of Education No. MSM 6840770043.

References

[1] Dlouhá E., Rojan J., Slabý P., *Městské komunikace, Návody k projektu*. ČVUT Praha, 1997, ISBN: 80-01-01604-8.
[2] Plíhal J., *Simulační model dopravního uzlu*. Disertační práce VUT Brno, 1997.
[3] Medelská V., *Príspevok k určovaniu hodnot vstupných časov vozidel vstupujících do svetelne riadenej križovatky*, Silniční obzor, 5, 1972.
[4] Liškutín I., *Navrhování SSZ pro řízení silničního provozu*, TP81, CDV Brno, 1996, ISBN 80-902141-2-6
[5] Anoškin J., *Podmienky návrhu plošné koordinácie dopravy v mestách strednej velikosti*, Dizertační práce, STU-Stavebná fakulta, Bratislava, 1994

[6] Medelská V., Príspevok k určovaniu hodnot vstupných časov vozidel vstupujících do svetelne riadenej križovatky, Silniční obzor, 5, 1972
[7] Grennshields B.D., Schapiro D., Traffic Performance at Urban Intersections, New Haven Ct, Burreau of Highway Traffic, Techn. Report Nr. 1, Yale University Press 1947

Jan Krcal
Czech Technical University in Prague
Faculty of Transportation Sciences
Konviktska 20, Prague, 11000
CZECH REPUBLIC
E-mail: krcal@fd.cvut.cz

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

The Subset Sum Problem

Petko Lalov, Stefan Dimitrov

Abstract

One of the main problems in **Complexity Theory** is the so called **Subset Sum Problem** (**SSP**). Two algorithms for solving this problem are presented in this paper. The first one is polynomial but with approximations. The second one is built on the basis of sequential application of „**Binary Search**", if solution is available, it finds this solution and it has polynomial behaviour.

1 Introduction

One of the major problems in **complexity theory** is the so called **Subset Sum Problem** (**SSP**). Its definition is:
Being given the integer positive number **c** and **n** integer positive numbers $w_1, w_2, \ldots, w_n$, $c \geq w_i$, find:

$$\max \sum_{i=1}^{n} w_i x_i , \tag{1}$$

satisfying the condition

$$\sum_{i=1}^{n} w_i x_i \leq c$$

where $\mathbf{x_i}$ are binary variables.

This problem can also be interpreted in the following equivalent form:

Provided a set of integer positive numbers $W=\{w_1, w_2, \ldots, w_n\}$ is given, does a subset $W' \subseteq W$ exist, such that

$$\sum_{w_i \in A'} w_i = \sum_{w_j \in W \setminus W'} w_j \tag{2}$$

It is evident that the first problem is equivalent to the second if $c = \frac{1}{2} \sum_{i=1}^{n} w_i$.

It is well known that problem (1) can be solved fully exhausting all the possible 2^n variants, represented by n-dimensional binary codes. It is evident that this approach is unacceptable and is applicable if and only if the value of **n** is small enough.
In this paper we suggest several algorithms, of heuristic and precision type, that are extremely effective. The common between them is the considerable reduction of the number of variants considered.

2 Denotations and Definitions

By F_n^2 we denote the set of n-dimensional binary codes. The value of the scalar product **W.x** is denoted by **s(x).** Let **S** is the set of all **s(x)**, where **x** is the binary representation of 1, 2, ...,2^n -1.
In stating the problem we need the function **pos(l,x)** that delivers the corresponding bit (0\1) to position l for code **x**.

Definition 1. If $x_1 \in F_n^2$ *and* $x_2 \in F_n^2$, we say that $x_1 < x_2$ if $x_1^{(10)} < x_2^{(10)}$, where $x^{(10)}$ is the decimal value of x . But from $x_1 < x_2$ it does not always follow that $s(x_1) \le s(x_2)$.

Definition 2. We divide F_n^2 to **n-1** disjunctive sets, called „classes" K_2, K_3, …..,K_n. According [2], $x \in K_l$, l=2, 3, ……,n-1, if the following conditions are satisfied:

- pos(l, x)=1 and pos(l+1, x)=0
- in the positions from 1 to l-1 at least one more bit 1 should exist
- if pos(m, x)=0 for $2 \le m \le l-1$ in positions m-1, m-2, ….1 no more than one bit 1 should be present.

The rest of the codes, that do not satisfy the above conditions, are of the class K_n.

Example: x={1 0 0 1 1 0 1 0} $x \in K_5$
x={0 0 1 0 0 0 1 1} $x \in K_2$
x={1 0 0 0 0 0 0 1} $x \in K_8$

Definition 3. The subset of the class K_l of codes for which $pos(l+2, x)$=pos(l +3,x)=…..=pos(n,x)=0 we call „pure class K_l" and we denote by K_l^p .

From now on we suppose that the set **W** is sorted: $w_n \ge w_{n-1} \ge ..... \ge w_1$.

3 Characteristics of set S

It is not difficult to assume that the solution of (1) is reduced to finding the maximum element of **S** for these **x**, for which $s(x) \le c$. If **S** is sorted, the solution can be found with time complexity of $O(n)$ by the procedure „**binary search**". Unfortunately, in the general case S is not sorted. The single case, for which this is true:

$$w_m \geq w_{m-1} + w_{m-2} + ... + w_1 \quad for\ m = 2,3,....,n$$

Then for each pair $x \in F_n^2$ *and* $y \in F_n^2$ from x<y, it follows that $s(x) \leq s(y)$.

In Figure 1 is drawn the discrete graphics of S at the following data:

W={18, 15, 12, 9, 6, 3, 1}

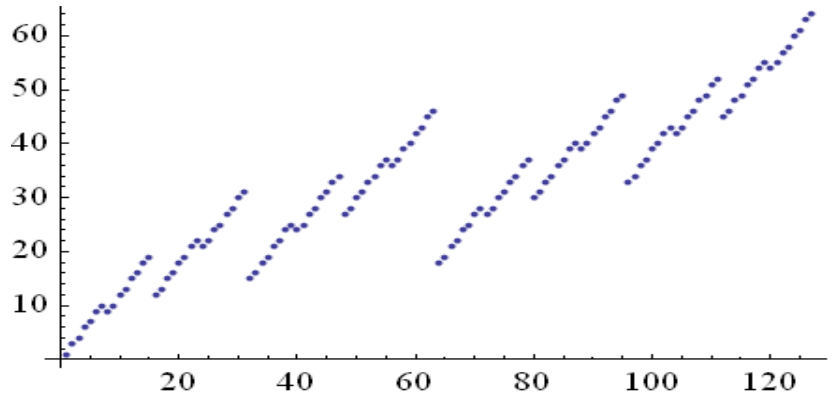


Figure 1

S consists of segments that are sorted. Let define the length and number of these segments. Let **m** is the greatest index for which is true

$$w_m \geq \sum_{i=1}^{m-1} w_i \, , w_{m-1} \geq \sum_{i=1}^{m-2} w_i \, , ..., w_3 \geq \sum_{i=1}^{2} w_i \qquad (3)$$

It is evident that **m** could be at least **2**. By **m** we denote „sorting period".

Theorem 1. The length of each segment of the set **S** at sorting period **m** is $\mathbf{2^m}$.

Proof: The first segment of S includes the values **s(k)**, where **k** are the binary codes of 0, 1, 2,.....,2^m-1, shown in the table that follows:

m	m-1	m-2	……..	3	2	1
0	0	0		0	0	0
0	0	0	…….	0	0	1
0	0	0	…….	0	1	0
0	0	0	…….	0	1	1
0	0	0	…….	1	0	0
……						
1	1	1	……..	1	1	1

It is evident that, considering (3), it is not difficult to assess that if $x^{(10)} < y^{(10)}$ *follows* $s(x) \leq s(y)$.

So **S** can be divided into $\mathbf{2^{n-m}}$ monotonically increasing segments with length 2^m - S_1, S_2,……,$S_{2^{n-m}}$. The limits of these segments are at:

$[s(0), s(2^m - 1)]; [s(2^m), s(2^{m+1} - 1)]; .....[s(2^{n-1}), s(2^n - 1)]$.

The elements of $\mathbf{S_2}$ are obtained from $\mathbf{S_1}$ by adding $\mathbf{w_{m+1}}$ to each element, the elements of $\mathbf{S_3}$ are obtained of $\mathbf{S_2}$ by adding $\mathbf{w_{m+2}}$ to each element, etc.

4 Algorithm A1

1) Q={[0, 2^n-1]}

2) We bisect the interval [0, 2^n-1] and scrutinize the intervals [0, 2^{n-1}-1] and [2^{n-1}, 2^n-1]; determine to which of the intervals [s(0), s(2^{n-1}-1)] and [s(2^{n-1}), s(2^n-1)] belongs **c**. The case **c** to belong to both intervals is not excluded. We form a new set **Q** of these intervals.

3) For each interval of **Q** we apply the bisection procedure and again we form the set **Q**, until the length of intervals of **Q** becomes equal to $\mathbf{2^m}$

4) For each interval, that defines increasing sorted segment, we apply the procedure "binary search" with time complexity $m = Log_2 2^m$ to find maximum element, not surpassing **c** in a sorted array.

4.1 Analysis of A1

The efficiency of this algorithm depends considerably on **m** - „sorting period", and on the value of **c** too. The greater **m** the less is the number of monotonic segments. If we denote with **z** the number of intervals containing **c**, the efficiency of the algorithm is **O(z.m)**.
Surely, the power of **z** can be reached in the worst case $\mathbf{2^{n-m}}$. But in these cases we can restrict ourselves to considering only and only the allowed number of segments. At last, the solution for each segment is allowable and can be considered as a solution, near to the optimal.

5 Algorithm A2

Prior to explaining this algorithm, the classes K_l *and subclass* K_l^p are considered. Let find the powers of these sets. These are the codes of K_l^p:

0….0 1 0 0… 0 0 1
0….0 1 0 0… 0 1 0
……..
0….0 1 1 0… 0 0 0
0….0 1 1 0… 0 1 0
……
0….0 1 1 1… 0 0 0
……
0….0 1 1 1… 1 1 1
l+1 l l-1.....1

It is not difficult to compute that $|K_l^p| = \frac{l(l-1)}{2}, l = 2,..,n-1, |K_n^p| = \frac{n(n+1)}{2}, a\ |K_l| = 2^{n-l-1}.\frac{l(l-1)}{2},\ l = 2,...,n-1, |K_n| = \frac{n(n+1)}{2}$

For the classes considered we will prove the following theorem:

Theorem 2. *If* $x, y \in K_l^p$ *and* $x < y$ *then* $s(x) \le s(y)$

Proof: Let $x < y$. This means that for j<l

$$pos(l,x) = pos(l,y) = 1, pos(l-1,x) = pos(l-1,y) = 1, ..., pos(l-j,x) = pos(l-j,y) = 1,$$
$$and\ \ pos(l-j-1,x) = pos(l-j-1,y) = 0$$

Therefore, if the position of next remaining single **1** for **x** is **p**, and for **y** is **q**, then it is evident that pos[q,y]>pos[p,x] and $w_p \le w_q$, from which we can conclude that

$$s(x) = \sum_{i=l-j}^{l} w_i + w_p,\ s(y) = \sum_{i=l-j}^{l} w_i + w_q \ \text{ or } \ s(x) \le s(y)$$

The theorem just proved, it follows that the set $\{s(x) \mid x \in K_l^p\}$ is sorted for each l=2,3,….,n.

The solution of (1) we search sequentially in „pure classes" $K_2^p, K_3^p, ..., K_n^p$, that are sorted. And that makes the algorithm heuristic, i.e. the solution is not always optimal but is very close to optimal. Computer-based experiments confirmed the latter conclusion. On finding the best „pure" solution, the latter can be improved if necessary. That is the short description of A2:

1) $x_l = \max\{s(x), x \in K_l^p\}, l = 2,3,...,n$

2) $x_l = \max\{x_2, x_3, ..., x_n\};\ s_{\max} = s(x_l)$

3) if $s_{\max} = c \rightarrow Stop$,else 4)

4) for $x < x_l, x \in K_l^p$ we search $\max\{w_{l+2}, w_{l+3}, ..., w_n\}$, so that

$$w_i \le c - s_{\max},\ w_i \in \{w_{l+2}, w_{l+3}, ..., w_n\}$$

if we find solution in the i-bit of $x, i \in \{l+2, l+3, ..., n\}$ we insert 1, hence $x \in K_l \setminus K_l^p$. For the new **x** we apply 4). i.e. we search $\max\{w_{i-1},\ w_{i-2}, ..., w_{l+2}\}$ so that

$$w_j \le c - s_{\max} - w_i,\ w_j \in \{w_{i-1},\ w_{i-2}, ..., w_{l+2}\} \text{ etc.}$$

Step 4) is performed using "binary search", applying an idea from [2].

5.1 Analysis of A2

Considering that $|K_l^p| = \frac{l(l-1)}{2}, l = 2,3,....., n-1, |K_n^p| = \frac{n(n+1)}{2}$ finding out $s_{\max}$ has time complexity $O(\sum_{k=2}^{n-1} \log_2 \frac{k(k-1)}{2})$, which is in the order of magnitude of $\ln(G[n]G[n-1])$, where G is the well known special Euler function. Even if we take for granted that each class K_l^p is with length **n**, processing **n** sorted classes will take time much less than $2n\log_2 n$ or

$$\ln(G[n]G[n-1]) < 2n\log_2 n$$

In Figure 2 are shown graphics of the two functions for n≤5000.

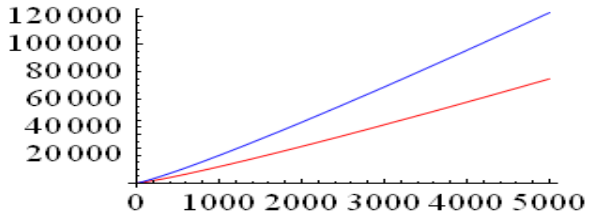


Figure 2

6 Algorithm A3

This algorithm is a result of the application of the so called „greedy algorithm" that is implemented by repeatedly applying the "binary search" and backtracking. Step-by-step description of the algorithm follows:

1) $w_{i_1} = \max\{w_n, w_{n-1}, ..., w_1\} \le c$

$w_{i_2} = \max\{w_{i_1-1}, w_{i_1-2}, ..., w_1\} \le c - w_{i_1}$

……

$w_{i_k} = \max\{w_{i_{k-1}-1}, w_{i_{k-1}-2}, ..., w_1\} \le c - w_{i_1} - w_{i2} - ... - w_{i_{k-1}}$

2) code $\mathbf{p_1}$ is generated with $pos(i_1, p_1) = 1; pos(i_2, p_1) = 1; .....; pos(i_k, p_1) = 1$

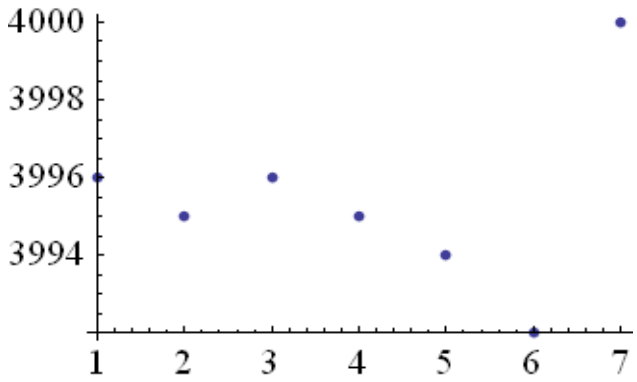
3) if $w_{i_1} + w_{i_2} + ... + w_{i_k} = c \rightarrow Stop$

4) $pos(i_k, p_1) = 0$

5) steps 1), 2), 3) are repeated for the set $\{w_{i_k-1}, w_{i_k-2}....., w_1\}$ and $c = c - w_{i_k}$

At last, a sequence of codes $\mathbf{p_1, p_2, \ldots}$ is obtained that reaches to the solution.

In the graphics below is shown how it can be reached to the solution of (1) passing through 7 iterations for an instance of **n=50** , **c=4000** and $\mathbf{w_i}$ generated in random.



7 Conclusion

Each of the algorithms presented has its advantages and disadvantages.
In algorithm A1 the original idea is the bisection of set S to sorted segments and search for the solution in a part of these segments. This idea is subject to future development.
Algorithm A2 has the most effective time complexity, when searching for a solution in the so called „pure" sorted classes $K_2^p, K_3^p, ..., K_n^p$, but it not always yields optimal result, no matter how close to it. This conclusion is confirmed by the numerical experiments made.
Algorithm A3 is a combination of „greedy algorithm", "binary search" and „backtracking". It always yields the optimal solution, but it is very difficult to evaluate its efficiency. The numerical experiments demonstrated polynomial complexity when $\max \sum_{i=1}^{n} w_i x_i = c$.

Reference:

[1] M. R. Garey, D. S. Johnson, *Computers and intractability. A Guide to the Theory of NP- Completeness*, San Francisco 1979, Moscow 1982

[2] Andrea Bianchini, *A Polynomial-time Exact Algorithm for the Subset Sum Problem*, Engineer, www.bianchiniandrea.it

Petko Lalov
University of Mining and Geology "St. Ivan Rilski"
Head of chair of Informatics
Students' Town, Sofia
BULGARIA
E-mail: petko@mgu.bg

Stefan Dimitrov
University of Sofia "St. Kliment Ohridski"
University Computer Centre
5, J. Bourchier, Str., Sofiq
BULGARIA
E-mail: stefan@ucc.uni-sofia.bg

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Examine the patterns that emerge when considering the taxonomy and identify models of recommender applications in E-commerce

Ionela Maniu, George Maniu

Abstract

Recommender systems are a technology that can help businesses, such systems allow search engines and advertising companies to suggest advertisements or offers to display based on consumer behavior. Product-associated recommendations allow businesses to respond to each customer's current interests and allow the natural associations among different products to guide customers to the right purchase. In this paper, we examine and synthesize data underlying patterns of recommender applications.

1 Introduction

Technology has dramatically reduced the barriers of publishing and distributing information. Now is time of technologies that can help us shift through all the available information to find that which is most valuable to us. In everyday life, people rely on recommendations from other people by spoken words, reference letters, news reports from news media, travel guides, etc. Recommender systems assist and augment this natural social process to help people sift through available books, articles, webpages, movies, music, restaurants, jokes, grocery products, etc. to find the most interesting and valuable information for them [10]. For e-commerce, conventionally, a recommendation system is considered as a success if clients purchase the recommended products. Personalized service requires firms to understand customers and offer goods or services that meet their needs**. Successful firms are those that provide the right products to the right customers at the right time and for the right price.** Recommendation systems apply data mining techniques to determine the similarity among thousands or even millions of data. One such techniques is collaborative filtering(CF). There are three major processes in the recommendation systems: object data collections and representations, similarity decisions, and recommendation computations. Collaborative filtering aims at finding the relationships among the new individual and the existing data in order to further determine the similarity and provide recommendations [13].

2 Examples of recommendation systems

Collaborative filtering has been very successful in both information filtering application and E-commerce applications.

Commercial sites that implement collaborative filtering systems include:

- *Amazon.com* is a commercial book shop/recommendation service.

- *eBay* is a commercial system which allows buyers and sellers to contribute to profiles of other customers with whom they have done business.
- *Rind* is a recommender system to help with buying a PC.
- *Ski-europe.com* recommends ski holidays.

Non-commercial sites that implement collaborative filtering systems include:

- *SurfLen* monitors user browsing and recommends web pages.
- *Tapestry* is an email recommendation system.
- *GroupLens* collaboratively recommends Usenet newsgroup articles.

3 Examine data patterns

3.1 Data Representation

User representation:

- by user attributes : demographic data such as gender, birth date, salary
- by associated items: the products the user has expressed interest in, has given ratings to or actually purchased
- by transactions: attributes extracted from the user's transaction history such as time, frequency, and amount, can partially represent a user's behavior pattern
- by items or item attributes associated with the user reflected in the feedback data- e.g., a user may be characterized as liking romantic stories and favoring low prices based on the attributes of the books she has purchased or simply as a set of books she has purchased

Items representation:

- by item attributes - such as price, content, brand, etc.
- by associated users – the customers who have purchased this item before

There are recommendation algorithms that directly operate on the *user-item interaction matrix* and do not explicitly derive any intermediate user or item representations

Transactions representation :

- by the transaction attributes: such as time, amount, etc.
- by items in the transactions.

Researchers have also included some transaction attributes such as time and place as additional dimensions and support a different type of recommendation that may be based on different combinations of dimensions, such as the recommendation of web content to a particular customer on weekends, or the recommendation of the best time to promote certain products to a particular customer. However, most existing recommender systems focus on the analysis of the two dimensions of users and items. More information about data representation can be found at the adress [14].

Other approach for data representation consist of exploiting product classification taxonomies. Semantic product classification corpora for diverse fields are becoming increasingly popular, facilitating smooth interaction across company boundaries.

Several online retailers employed *product taxonomy* (PT) to give a clear view of their product lines in tree structure. PT is practically represented as a tree and categorizes a set of products at a low level into a more general product at a higher level. The leaves of the tree denote the product instances, and non-leaf nodes denote product classes obtained by combining several nodes at a lower level into one parent node . The root node labeled by 'All' denotes the most general product category

For a given category, attributes are then determined in 3 steps. First, create basic list of attributes, this list is modified through asking users to express their opinions about importance of items. Finally, it's generate final list according to preliminary list and users opinions. There are also determine levels and possible values for final attributes.

Once attributes of categories are determined, we should place this attributes in product taxonomy. There by child nodes will inherit parent node attributes while it has own attributes. Examples and other information can be found in [5] and [16].

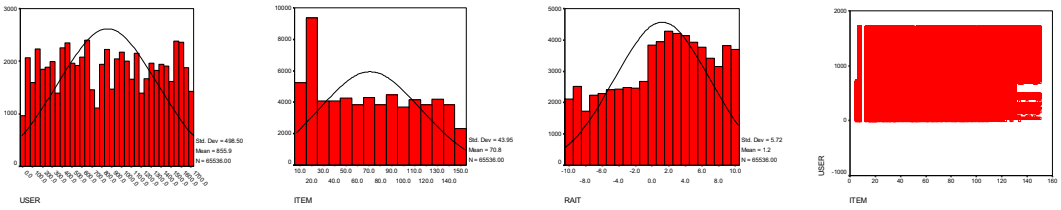
3.2 Data distribution

- **many items.** If there are only a few items to choose from, the user can learn about them all without need for computer support.
- **many ratings per item.** If there are only a few ratings per item, there may not be enough information to provide useful predictions or recommendations.
- **more users rating than items to be recommended.** If there are few ratings per user, you'll need many users. Lots of systems are like this. The ratings distribution is almost always very skewed: a few items get most of the ratings, a long tail of items that get few ratings. Items in this long tail will not be confidently predictable.
- **users rate multiple items.** If a user rates only a single item, this provides some information for summary statistics, but no information for relating the items to each other.
- **homogenous items.** Music albums are like this. Most are similarly priced, similar to buy, of a similar length. Books or research papers are also like this. Items sold at a department store are not like this: some are cheap, some very expensive. For example, if you buy a hammer, perhaps you should not be recommended a refrigerator.

3.3 Experimental study

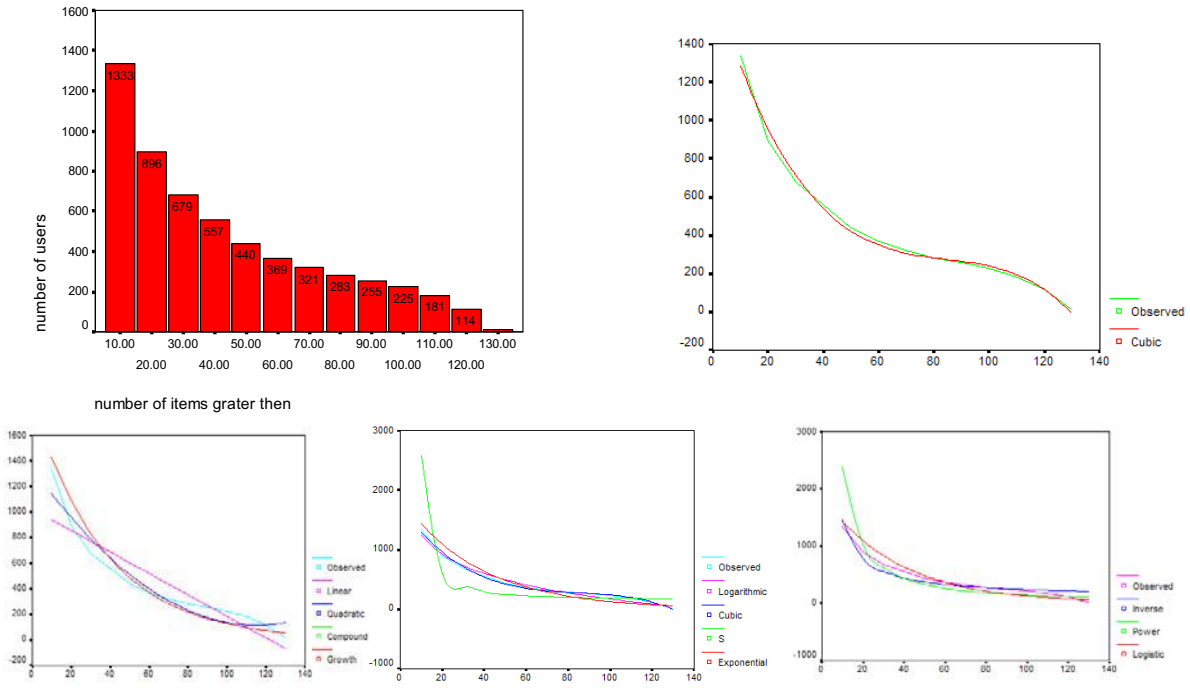
For examine this data pattern (data distribution) we used Jester data set example**.** The dataset comes from Ken Goldberg's joke recommendation website, Jester [11]. In Jester, users rate a core set of jokes, and then receive recommendations about others that they should like. The database we used has 150 jokes, and records of 1708 users. Some users end up reading and rating all the jokes. Jester has a rating scale from -10 to 10, the matrix user-item has 65536 raitings. Ratings are implemented with a slider, so Jester's scale is continuous.

In order to establish data distribution (user, item, raitings) we achieved the histogram chart below and display normal curve and scatter chart :

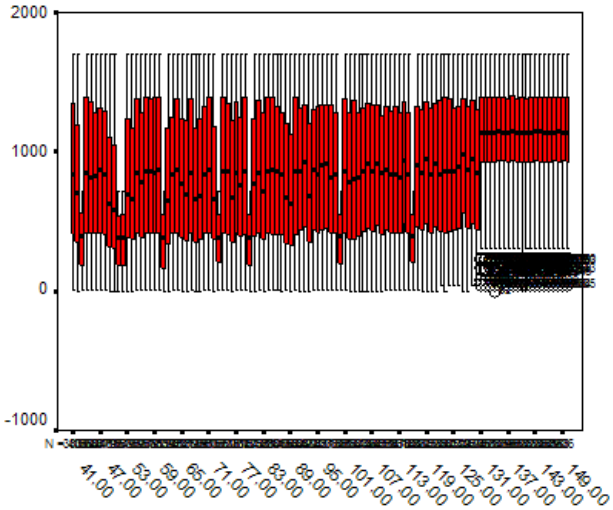


As we can see Jester is a dense dataset (more dense than other datasets) which is why we should try to normalize the data (for example raiting matrix) and after that clustering the data (using clustering algoritms).

Chart below highlights the fact that the number of users decreases if the number of raiting items increase(with a 10 step), this phenomenon following an cubic curve, result that was obtained after comparison with several types of curves.

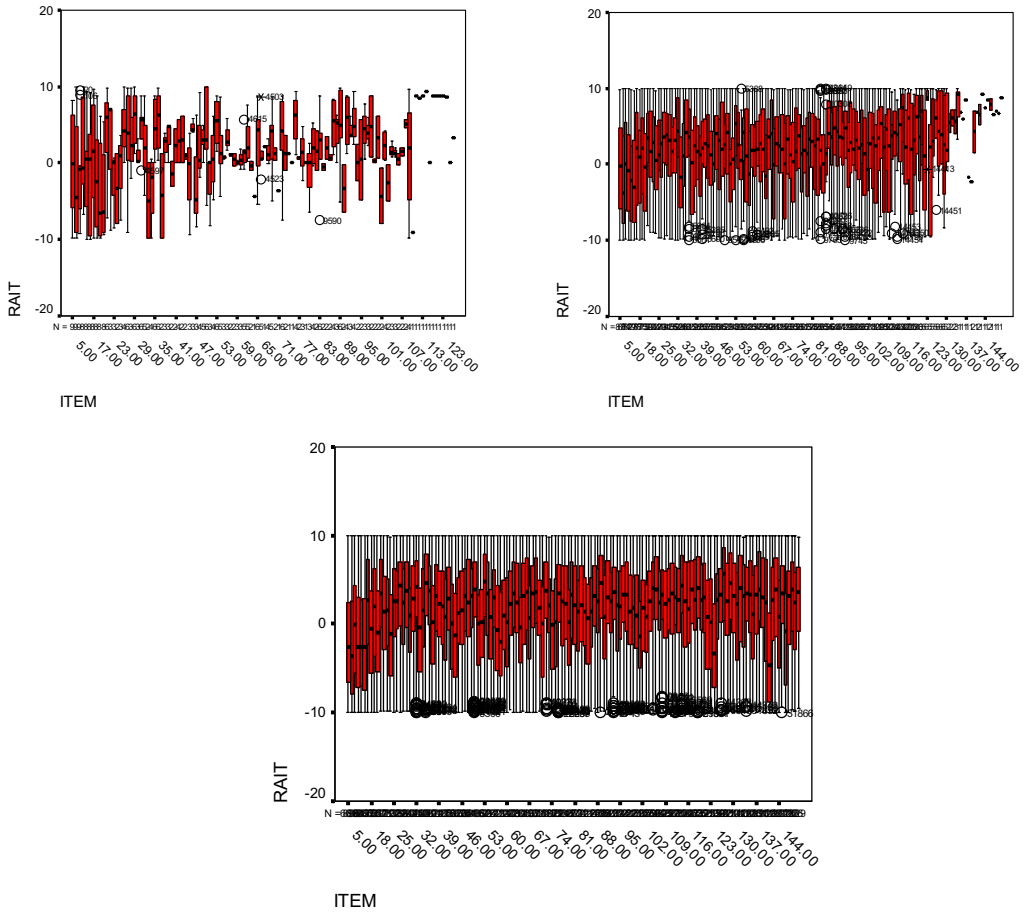


We also can see that those items that are on the last position in the items list are those with the fewest raitings (jokes which the user has not rated from the known ratings of jokes from all users). Consequently this items that have been rated by only few people would be recomanded very rarely even if those few users gave high ratings to them. Also for the user whose tastes are unusual compared to the rest of the population, there will not be any other users who are particularly similar, leading to poor recommendations.



Organizing data into clusters can be a solution in this situation too. In [12], the author proposed a Bregman co-clustering algorithm whose main objective is to find a partition of m rows and n columns of a data matrix into k row clusters and l column clusters such that the distance between the original matrix and the reconstructed matrix is minimized. Another solution is to use user/item profile information when calculating user/item similarity.

Regarding the fact that it is important to be **more users rating than items to be recommended,** it can be seen in the graphs that follow, that increasing the number of users has as consequence the average raitings homogenization and thus increase the accuracy of the recommendation. There were taken into account 3 situations: less than 10 users, less than 100 users, less than 1,000 users.



4 Conclusion

Recommender systems' performance can be easily affected when there are no sufficient item preferences data provided by previous users. This paper suggests, after examine a part of the patterns of data, another information source, item taxonomies, in addition to item preferences for assisting recommendation making. Item taxonomic information has been popularly applied in diverse ecommerce domains for product or content classification, and therefore can be easily obtained and adapted by recommender systems.

References

[1] Basu, C., Hirsh H., and Cohen, W. , *Recommendation as classification: Using social and content-based information in recommendation*, Workshop on Recommender Systems. AAAI Press, Reston, 11-15, 1998

[2] TSO, H. L. K. and Schmidt-Thieme, *Attribute-Aware Collaborative Filtering*, Annual Conference of the German Classification Society, Magdeburg, Germany, 2005

[3] Ziegler, C., Schmidt-Thieme, L. and Lausen, G., *Exploiting Semantic Product Descriptions for Recommender Systems*, 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop , UK, 2004

[4] http://en.wikipedia.org/wiki/Collaborative_filtering

[5] http://www.amazon.com

[6] L.P. Hung, *A personalized recommendation system based on product taxonomy for one-to-one marketing online*, Expert Systems with Applications 29, pp. 383–392, 2005

[7] S.H. Choi, S. Kang and J.Y. Jeon, *Personalized recommendation system based on product specification values*, Expert Systems with Applications 31 , pp. 607–616, 2006

[8] www.movielens.org

[9] www.grouplens.org
[10] http://www.hindawi.com/journals/aai/2009/421425.html
[11]K. Goldberg, D. Gupta, M. Digiovanni, and H. Narita. Jester 2.0 : *Evaluation of a new linear time collaborative filtering algorithm*. In ACM SIGIR, August 1999. Poster Session and Demonstration.
[12]Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. *A generalized maximum entropy approach to bregman co-clustering and matrix approximation*. Journal of Machine Learning Research, 8:1919–1986, August 2007.
[13] http://sir-lab.usc.edu/publications/121new.pdf
[14] http://www.personal.psu.edu/~huz2/Zan/docs/zanhuang.dissertation.pdf
[15] http://www.cs.uni.edu/~schafer/publications/CF_AdaptiveWeb_2006.pdf
[16] http://deposit.ddb.de/cgibin/dokserv?idn=975319213&dok_var=d1&dok_ext=pdf&filename=975319213.pdf
[17] http://www.cs.uni.edu/~schafer/publications/CF_AdaptiveWeb_2006.pdf

Maniu Ionela
Univ. Lucian Blaga Sibiu
Informatics Departament
Dr. Ratiu No. 5-7, Sibiu
Romania
E-mail: mocanionela@yahoo.com

Maniu George
Univ. Spiru Haret Brasov
Management Faculty
Turnului No.7
Romania
E-mail: costelmaniu@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Adaptive Test Generation for E-Learning Systems

Ana-Maria Mirea[1], Constantin Teodorescu-Mihai, Doina Preda, Mircea Preda

Abstract

Computer based testing can provided knowledge evaluation that is tailored to the specific of each student with costs and quality performances that exceeds the human based testing. Usually, testing performed by the humans is a tradeoff between the involved costs and accuracy. Usually, an accurate test will take longer and if it is performed by a human examiner it will cost more. Also, longer tests can made the human examiners more prone to be affected by other factors like fatigue that can have an adverse impact on quality. Computer based testing can permanently adapt to the specific of the students, it is not affected by human specific factors and has low costs on long run.

1 Introduction

Testing is a major component of the learning process, it allows to evaluate the progress of learning. Classical testing involves human examiners and it is costly, subjective, error prone and not ready available in every situation. E-testing systems are permanently available for the students, are not biased by human factors like fatigue, provides fast and accurate results, the feedback for the students is comprehensive and a system can cover a large amount of topics being able to replace several human examiners. Also, e-testing systems provides auto evaluation possibility to the students.

Several major attempts to build Computer Adaptive Testing (CAT) systems can be mentioned. Their mathematical background is provided by the Item Response Theory (IRT) [1] that is concerned with the application of mathematical models to data from questionnaires and tests as a basis for measuring abilities, attitudes, or other variables. It is used for statistical analysis and development of assessments, often for high importance tests. The main assumption behind IRT is that the probability to obtain a correct answer to an exercise is a mathematical function of the characteristics of the person who takes the exercise and the features of the exercise itself.

In [4] it is described a web-based tool to assist teachers and instructors in the assessment process. The tests are generated according to teachers specifications and are adaptive, that is, the questions are selected intelligently to fit the students level of knowledge. In this way, are obtained more accurate estimations of students knowledge with significantly shorter tests. The knowledge level of the student is represented and measured by a single variable θ. Using as input data a set of responses of the students to a set of questions, the level of knowledge of the student is estimated using statistical methods. Then, the estimation $\hat{\theta}$ is used to determine the most informative item to ask next. These steps are repeated until some stopping criterion is met. Different statistical methods to estimate θ and to select the next best question to ask give different IRT models that are used in the article. By using applets to present exercises, rich interactions between students and testing system are possible. Some adaptive test generation methods involve costly time and human resources for preparation and have complex mathematical formulas. These methods are available only in large educational institutes and in professional testing centers. In [6], it

[1] All authors contributed equally to the paper.

is proposed a CAT method tailored for classrooms and for small business daily routine. The evaluation problem is considered as a sequential statistical hypothesis test where the hypothesis is *the topis is mastered*. As a solution is employed the Sequential Probability Ratio Test (SPRT), originally used in quality control but later reformulated for use in computerized testing of human examinees. The authors obtain a system that is easy to implement and does not require a large number of training sessions for parameter estimation. [7] provides an excellent coverage of computerized adaptive testing (CAT). It presents all necessary statistic and psihometric concepts and provides also practical foundations. [3] and [2] discusses the general problematic of adaptive educational systems. [5] presents the main learning styles that can be encountered at the students and the implications of these styles in building adaptive learning systems.

This paper formalizes an e-testing system with advanced capabilities that include adaptivity to each particular student and a database of exercises that is permanently evolving. The classic structure of the CAT methods is followed: an iterative algorithm where at each round, new exercises are presented to the examinee. If he performs well, more difficult questions will be presented, otherwise, if the performances are poor, simpler questions will be used. The proposed formalism is centered around the notion of difficulty degree and difficulty degree updating. Both an individual exercises and tests have attached difficulty degrees. The system maintains for each user the most appropriate difficulty degree. Initially, the difficulty degree for a student can be obtained by data mining methods starting from the difficulty degrees of similar users on a predefined initial value can be used. After each interaction between the testing system and user, the difficulty degree is updated. For a specified difficulty degree, the system will generate every time a different test. Other advanced features of the system include a permanently evolving exercises database, possibility to include short resumes and full lessons on covered topics and rich possibilities of interaction with the students. Exercises database evolves by generating new exercises using predefined patterns and by incorporating exercises proposed by students.

The main differences over the already developed CAT systems are:

- Most CAT methods estimate the knowledge level of the students but not the features of the exercises. The newly introduced method estimates the characteristics (difficulty) of the exercises and these estimations are permanently updated. The degree of mastery attained by a student on a topic is given by the difficulty of the exercises that were successfully resolved.

- Usually, the CAT systems are not able to provide detailed assessments on subtopics, they provide only a single estimation of the knowledge level. In the presented settings, the topic of the test can be divided in several subtopics and the degree of mastery on each subtopic is estimated.

- Measures of similarity are used to identify the exercises and students that are closely related. In this way, the system can transfer the learned knowledge between similar users and similar exercises and can substitute an exercise with another one.

The next section of the paper presents the theoretical model of the proposed adaptive testing system. After that, there presented implementation considerations, algorithms and further extensions of the system capabilities.

2 Specifying adaptive tests

In the following paragraphs, S will represents the set of the students that use the e-learning system and $s \in S$ is a particular student.

Let us consider Ex the set of exercises that are available for the testing and evaluation system. A test is a subset of Ex, $T \subseteq Ex$ that satisfies a specified set of restrictions. For example, the number of exercises from the test can be limited by a superior threshold, $Card(T) \leq M$, where $Card(.)$ is the cardinal function that provides the number of elements from a set and $M \in \mathbb{N}$.

A function $Diff : Ex \rightarrow \mathbb{R}$ will be used to evaluate the difficulty degree of the exercises from tests, where for an exercise $e \in Ex$, $Diff(e)$ represents the difficulty of the exercise e. Function $Diff$ can be

extended to sets of exercises:

$$Diff : 2^{Ex} \rightarrow \mathbb{R}$$
$$Diff(T) = \sum_{e \in T} Diff(e)$$

In accordance with the above definition, the difficulty of a test is the sum of the difficulties of the exercises that appear in the test. Alternately, the difficulty of a test can be the average difficulty of the exercises that form the test. The definition is as follows:

$$Diff : 2^{Ex} \rightarrow \mathbb{R}$$
$$Diff(T) = \frac{\sum_{e \in T} Diff(e)}{Card(T)}$$

The difficulty of a test will be limited by a constant $d_{\max}$, $\forall T, Diff(T) \leq d_{\max}$. $d_{\max}$ represents the maximum difficulty for a test.

2.1 Estimating difficulty of the exercises

A first estimation for the difficulty of the exercises will be provided by the human operator,

$$e \in Ex \underset{human}{\longrightarrow} Diff(e)$$

To simplify the task of the human operator, we can suppose that the first estimation of the difficulty degrees will use only a discrete set of qualitative values like *easy, average, high* and these values will be after that converted to numbers.

The first estimation will be subsequently refined using the performances of the students during tests. Let us consider $e \in Ex$ an exercise and $Result(e)$ a measure of the performance of a student for e where

$$Result(e) = \begin{cases} -1 & \text{if student failed to resolve the exercise} \\ 1 & \text{if student succeded to resolve the exercise} \end{cases}$$

Then, a new estimate for the difficulty of e can be obtained as follows:

$$Diff(e)^{new} \longleftarrow Diff(e)^{old} + \alpha(e) \cdot Result(e) \cdot Diff(e)^{old} \quad (1)$$

where $\alpha(e) \in [0, 1]$ is a parameter named *learning rate* for the exercise e. Usually, $\alpha(e) \rightarrow 0$ when $t(e) \rightarrow \infty$ where $t(e)$ represents the number of appearances in tests for the exercise e (over time the updates of the difficulty degree for the exercise e should be smaller because the already existent estimation will be accurate). A common formula for the learning rate is

$$\alpha(e) = \frac{\alpha_0(e)}{1 + \frac{t(e)}{TE}}$$

where $\alpha_0(e)$ and TE are constants.

2.2 Similarity measure between exercises

For adaptive tests generation it is important to establish how similar are two exercises. For example, if a student fails to resolve an exercise, similar exercises can be proposed to him in the subsequent tests to accurately measure the level of knowledge of the student in the area. Moreover, with a similarity measure between exercises, different tests can be generated for different students, tests that cover same topics and have similar levels of difficulty. In this way, the evaluation process will be consistent and in same time the students will have no possibility to know what exercises will appear in a test.

A measure of the distance between two exercises is introduced as:

$$d : Ex \times Ex \rightarrow \mathbb{R}_+$$

with the following properties:

$$d(e_1, e_2) \geq 0$$
$$d(e, e) = 0$$
$$d(e_1, e_2) \leq d(e_1, e_3) + d(e_3, e_2)$$

2.3 Classes of exercises

A category or a class of exercises is a subset of the full set of exercises $C \subseteq Ex$ that includes exercises with similar properties. Let us denote by $\mathcal{C} = C_1, ..., C_k$ the set of the all categories that are available for the system. It is possible that same exercise e to apart to two or more categories. This means that two categories can have a non empty intersection $C_i \cap C_j \neq \emptyset$. We will suppose that an exercise $e \in Ex$ will apart to at least one category, and, consequently,

$$\bigcup_{i=1}^{k} C_i = Ex.$$

Let us suppose that a distance measure can be devised between categories:

$$dc : \mathcal{C} \times \mathcal{C} \rightarrow \mathbf{IR}_+,$$

where $dc(C_i, C_j)$ represents how different are the categories C_i and C_j. In these conditions, a distance between exercises can be devised based on the distance between categories:

$$d(e_1, e_2) = \max_{\substack{c_1, c_2 \\ e_1 \in C_1 \\ e_2 \in C_2}} dc(C_1, C_2) \tag{2}$$

2.4 Generating adaptive tests

Let us consider $T \subseteq Ex$ a test that was performed by a student s and $Success(T, s) \subseteq T$ and $Failure(T, s) \subseteq T$ represent the exercises that were successfully resolved and, respectively, failed. The level of knowledge of the student s assessed by the test will be defined as:

$$TrainingLevel(T, s) = \frac{Diff(Success(T, s))}{Diff(T)} \in [0, 1]. \tag{3}$$

Otherwise stated, the training level of a student s regarding to a test T is the ration between the complexity of the exercises that were correctly resolved and the total complexity of the test.

The difficulty degree of the next test can be established in accordance with the algorithm 1.

Algorithm 1: The algorithm used to adjust the difficulty degree of the tests in accordance with the student's performances.

SELECTING THE DIFFICULTY DEGREE OF THE NEXT TEST()
(1) **if** $TrainingLevel(T, s) > \beta_1$
(2) increase the difficulty level of the next test
(3) **else if** $TrainingLevel(T, s) < \beta_2$
(4) decrease the difficulty level of the next test
(5) **else**
(6) maintain the current difficulty level

Here, β_1 and β_2 are two constant thresholds that are used to trigger the update of the test's difficulty level for a student.

Difficulty level adjustment can be also done on each category. Let us define

$$TrainingLevel(T, s, C) = \frac{Diff(Success(T, s) \cap C)}{Diff(T \cap C)} \in [0, 1]. \tag{4}$$

to be the knowledge level of the student $s \in S$ for the category $C \in \mathcal{C}$ assest by the test $T \subseteq Ex$. The difficulty of the exercises selected from the category C can be updated using the mentioned algorithm.

2.5 Updating the difficulty degree

The algorithm from previous section involves update operations on the difficulty degree of the entire test or of a specified category. These update operations will be performed as is prescribed by the following guidelines. Let us consider $d_{\min}$ the minimum degree of difficulty and $d_{\max}$ the maximum degree of difficulty, and, d_{crt} the current one, $d_{\min} \leq d_{crt} \leq d_{\max}$. Three situations are possible:

- Difficulty degree should be increased. Then, the next difficulty will be selected using a Gaussian distribution with mean $(1-\gamma)d_{crt} + \gamma \cdot d_{\max}$ and variance 1. Probability density of this function is represented in the figure 1.

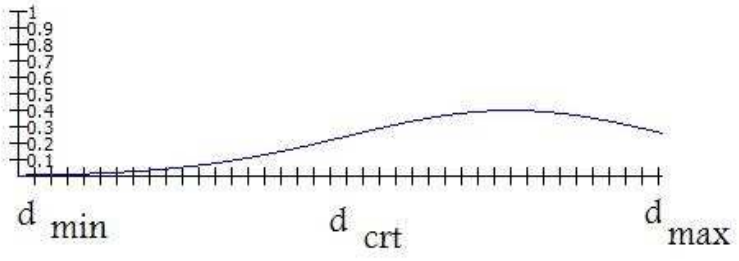


Figure 1: Density function for the Gaussian probability distribution used to select the newly increased difficulty degree with $\gamma = 0.5$.

- Difficulty degree should be decreased. Then, the next difficulty will be selected using a Gaussian distribution with mean $(1-\gamma)d_{crt} + \gamma \cdot d_{\min}$ and variance 1. The process is presented in the figure 2

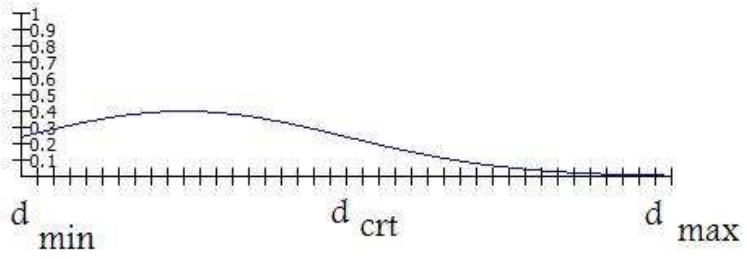


Figure 2: Density function for the Gaussian probability distribution used to select the newly decreased difficulty degree with $\gamma = 0.5$.

- Difficulty degree should be maintained. Then, the next difficulty will be selected using a Gaussian distribution with mean d_{crt} and variance 1.

Here, $\gamma \in [0, 1]$ is a parameter named *update rate*. γ can be a constant, for example, $\gamma = 0.5$, or, γ can have decreasing values, $\gamma \to 0$ when the number of difficulty updates goes to ∞.

2.6 Generating a test with a specified difficulty

Let us consider $\mathcal{C}(T) \subseteq \mathcal{C}$ the categories of exercises that will be considered for the test. If $Card(\mathcal{C}(T)) = 1$ then the test will involve only one topic (category). If $\mathcal{C}(T) = \mathcal{C}$ then the test will involve all topics that

are available for the system. Let us denote by

$$P(C) = \frac{Card(C)}{\sum\limits_{C_i \in \mathcal{C}(T)} Card(C_i)}, C \in \mathcal{C}(T). \tag{5}$$

$P(C) \in [0,1], \forall C \in \mathcal{C}(T)$ represents a discrete probability distribution over the set $\mathcal{C}(T)$ of the categories (topics) involved in the test.

If the d_{crt} is the difficulty degree of the test that should be constructed, then the algorithm 2 is used to achieve the desired result.

Algorithm 2: The algorithm used to create a test with a specific difficulty degree. The iterative procedure selects at each step a category and, then, an exercise from that category.

CONSTRUCTING A TEST WITH A SPECIFIED DIFFICULTY DEGREE(d_{crt})

(1) $T \leftarrow \emptyset$ (the initial test is empty)
(2) **while** true
(3) Select the category $C \in \mathcal{C}(T)$ of the exercise that will be added to the test in accordance with the probability distribution $P(C)$ (the categories with a larger number of exercises will have a greater probability to be selected).
(4) Select $e \in C$ an exercise in accordance with the target difficulty d_{crt} and with the current difficulty of the test $Diff(T)$.
(5) $T \leftarrow T \cup \{e\}$ (the exercise is added to the test).
(6) **if** $StopConditions(T, d_{crt})$ are satisfied
(7) **return** T

When an exercise is to be selected from a category in order to be added at a test, there are two situations:

- The difficulty of the test is considered the sum of the difficulties of the exercises from test $Diff(T) = \sum\limits_{e \in T} Diff(e)$. A current difficulty level $d_{crt}(C)$ will be maintained for each category $C \in \mathcal{C}(T)$. The exercise will be selected using a Gaussian distribution with the mean $d_{crt}(C)$ and variance 1. In this way, all exercises from the category have a chance to be selected and the exercises with the difficulty around $d_{crt}(C)$ are preferred. Initially, the intra category difficulty $d_{crt}(C)$ should be low or medium low, because, pedagogy indicates that is recommendable to start with relatively low complexity exercises. $d_{crt}(C)$ will be updated after each test performed by the student in accordance with his results.
- The difficulty of the test is the average of the difficulties of the exercises from test $Diff(T) = \frac{\sum\limits_{e \in T} Diff(e)}{Card(T)}$. In this case, the exercise will be selected using a Gaussian distribution with the mean d_{crt} and the variance 1.

To define $StopConditions(T, d_{crt})$ same situations as above should be considered:

- The difficulty of the test is the sum of difficulties of the exercises that form the test. Then

$$StopConditions(T, d_{crt}) = \begin{cases} true & \text{if } Diff(T) \geq d_{crt} \\ false & \text{if } Diff(T) < d_{crt} \end{cases}$$

  Otherwise stated, test building process stops when the difficulty of the current test exceeds the target difficulty d_{crt}. The employed algorithm is similar with the algorithm for the knapsack problem, exercises are added until the target difficulty, which is the capacity of the knapsack, is exceeded.
- Test difficulty is the average of the exercises difficulties. In these conditions, test building process stops when some test admissibility conditions are satisfied. Examples of such admissibility conditions are:

– Test should have at least Min exercises, $Card(T) \geq Min$.

– A test is admissible if it includes an exercise for each category, $\forall C \in \mathcal{C}(T)$, $\exists e \in T$ such that $e \in C$.

2.7 Adaptive test generation

Adaptive test procedure for a student is described in the algorithm 3 and it synthesizes the theoretical developments from the previous sections.

Algorithm 3: The adaptive testing procedure. At each step a test is generated in accordance with the recommended difficulty level for the student. The test is applied and after that the difficulty level is updated in accordance with the test results.

ADAPTIVE TEST PROCEDURE FOR A STUDENT($s \in S$)
(1) $d_{crt} \leftarrow d_{init}(s)$ the initial difficulty level for the student s
(2) **while** stop conditions not satisfied for s
(3) generate a test with the difficulty d_{crt}
(4) apply the test to student and record the results
(5) update the difficulty level d_{crt} in accordance with the results.

Regarding the stop conditions for a student s several scenarios can be considered:

- The student s uses the system on regular basis to improve or maintain his knowledge level. In this case, the testing loop can continue forever. The student should have the possibility to reset the adaptive system in order to start the training again from the initial conditions.
- The testing system is used by an institution to evaluate the knowledge level of the student s. In this case, the system should stop when the variance on the last K tests is under a threshold $\delta(K)$. A low variance means that the difficulty level, which is optimal for the student s, was found and the system is stabilizing around this level. In this case, the average difficulty level on the last K tests can be used as a measure of the knowledge for s. A limit of the number of tests that can be applied to s, *MaxTestsNum* can be specified in this situation to cope with the cases when the difficulty level has strong oscilations for s.

3 Implementing testing systems with advanced capabilities

In this section, it will be discussed implementation details for the previously presented theoretical model and other features that can improve the e-learning system.

3.1 Personalizing the system for each student

Our system should consider each student as an individual entity with specific requirements. In order to do that, the system should store data to be able to differentiate between students. Maintained data will include:

- Personal information about student. Ideally, this information should allow to identify the similar students.
- Information about the performed tests. Minimally, this information will consist from the success percentages obtained at the tests. If the storage possibilities allow, the information can also include even the exercises that composed the tests, at the results for each exercise.
- Information on the evolution of the difficulty degree d_{crt} of the administrated tests. The intra category difficulty degree evolution $d_{crt}(C)$ can be also included for each category $C \in \mathcal{C}$.

3.2 Providing extensive feedback to the students

Students should receive complete statistics and graphical representations on their tests' performances. The evolution of the tests' difficultly degree should be also included. For each failure, the student will receive full explications on the topic.

3.3 Passing from a testing system to a learning system

The system can be extended with short summaries and even with full lessons on the covered topics. Different techniques can be used to ensure that the students really read and understand the presented information. A such technique can be for example to impose a minimum time of presentation of the information on screen. Also, each test can be prefaced by the summaries of the covered topics. Also, access to some tests can be allowed only to the users that completed a specified set of lessons and/or summaries.

3.4 A permanently evolving system driven by students

For some types of systems and students, it can be considered the possibility that the exercise database to be extended with the exercises proposed by examinees. The similarity measure between exercises can be used to automatically classify into categories the newly added exercises. Of course, the quality of these exercises is a concern and should be carefully assessed. Several methods to ensure the quality can be considered:

- All new exercises will be moderated by human administrators.
- Only the students with high performances can propose new exercises.
- Feedback from other users will be employed to measure the quality of the exercises.
- Social trust measuring methods can be employed, and only the trusted users can extend the exercise database.

3.5 Measuring the quality of the proposed tests

The system can request to users to provide feedback information about the quality of the exercises from tests and about the overall quality of the tests. The requested feedback should be carefully minimized in order to not place an unpleasant burden on the students. The exercises that in average have on long run low quality marks will be removed from the system.

3.6 Adapting to the learning styles of the students

The proposed model does not impose any restriction on the modality of presentation for the exercises. However, it can be supposed that the presentation with several alternate answers will predominate. Our system can accommodate variants of presentations that are predominant visual, auditive, practical or combinations between them. These styles can be easily applied to lessons and resumes but also can appear in the testing procedure. The system should establish the main learning style of a student using an initial set of questions or based on the student performances on different types of presentations. Ideally, the system should use for each student a variant of presentation that is suited to the student learning style.

4 Conclusion

The paper proposes a testing system that adapts the difficulty of the tests to the training level of the students. The system is highly extensive, it can adapt several learning styles, can be enriched with short summaries and full lessons and has an exercise database that is permanently evolving. The system

always will present different tests to students even the difficulty level has not changed. The system is configurable, both administrators and students can decide how testing is performed and in particular when the testing procedure is finished. Future improvements will include:

- Automatic generation of exercises on specified topics.
- Automatic generation of summaries from full lessons.
- Rich interaction possibilities for the students (hyper-media).
- Improved data mining procedures to identify commonalities between students and exercises.
- Day time usage statistics to identify patterns in the results obtained by students.

References

[1] Frank B. Baker, *The Basics of Item Response Theory*, ERIC Clearinghouse on Assessment and Evaluation, 2001.

[2] Peter Brusilovsky. A Distributed Architecture for Adaptive and Intelligent Learning Managament Systems. *In Proceedings of the AIED 2003 Workshop Towards Intelligent Learning Management Systems*, Sydney, 5–13, 2003.

[3] Peter Brusilovsky. Adaptive navigation support: from adaptive hypermedia to the adaptive Web and beyond. *PsychNology Journal*, Vol. 2, No. 1, 7-23, 2004.

[4] Conejo, R., Guzman, E., Millan, E., Trella, M., Perez-De-La-Cruz, J. L., and Rios, A. SIETTE: A Web-Based Tool for Adaptive Testing. *Int. J. Artif. Intell.*, Ed. 14, 1, 29–61, 2004.

[5] Elvira Popescu, Philippe Trigano, Costin Badica. Towards a Unified Learning Style Model in Adaptive Educational Systems. *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, 804–808, 2007.

[6] Tao, Y.-H., Wu, Y.-L., Chang, H.-Y. A Practical Computer Adaptive Testing Model for Small-Scale Scenarios. *Educational Technology and Society*, 11(3), 259-274, 2008.

[7] Howard Wainer, Neil J. Dorans, Ronald Flaugher, Bert F. Green, Robert J. Mislevy, Lynne Steinberg, David Thissen *Computerized Adaptive Testing: A Primer*, Lawrence Erlbaum, 2000.

Ana-Maria Mirea
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, Craiova, 200585
Romania
E-mail: *ammirea@acm.org*

Constantin Teodorescu-Mihai
"Petrache Poenaru" High School
Department of Computer Science
Bălceşti, Vâlcea, 245400
Romania
E-mail: *constantintm@ymail.com*

Doina Preda
"Matei Basarab" High School
Department of Computer Science
Vasile Alecsandri street, 113, Craiova, 200463
Romania
E-mail: *dpreda@acm.org*

Mircea Preda
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, Craiova, 200585
Romania
E-mail: *mpreda@acm.org*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

On the approximate solution of a functional-integral equation

Viorica Mureşan

Abstract

In this paper we consider the following functional-integral equation with linear modifications of the arguments:

$$u(x,y)=\int_0^x\int_0^y K(s,t,u(s,t),u(\lambda s,t),u(s,\mu t))dsdt,(x,y)\in[0,a]\times[0,b],$$

where $0<\lambda<1, 0<\mu<1, \quad K\in C([0,a]\times[0,b]\times\mathbb{R}^3)$.

Using the Picard operators' technique we obtain existence and uniqueness results for the solution of this equation.

By applying the successive approximations method and by using a cubature formula we give an algorithm for the approximate solution.

2000 AMS Subject Classification: 47H10, 34K05, 34K15
Key words: functional-integral equations, Picard operators, approximate solution

1 Introduction

Many problems from astronomy, chemistry, biology, economics, engineering lead to mathematical models described by functional-integral equations.

The theory of functional-integral equations has been developed in the last fifty years. Many monographs appeared: Bellman and Cooke [3] (1963), Halanay [14] (1965), Elsgoltz and Norkin [9] (1971), Bernfeld and Lakshmikantham [4] (1974), Hale [12] (1977), Lakshmikantham [19] (1984), Azbelev, Maksimov and Rahmatulina [1] (1991), Corduneanu [7] (1991), Gopalsamy [10] (1992), Hale and Verduyn Lunel [13] (1993), Guo, Lakshmikantham and Liu [11] (1996) such as a large number of papers. We quote here [2], [5], [15], [25], [31], [33].

A special class is represented by the integral equations with affine modifications of the arguments, which can be with delay or with linear modifications of the arguments. The latter equations have been developed in connection with the pantograph equation $x'(t)=ax(\lambda t)$ and in connection with problems for the mentioned equation and for some of its generalizations (see [17], [18], [24], [32]).

Various Darboux-Ionescu problems for some equations with deviating arguments were presented by I.A. Rus in [27]. These problems are equivalent to some functional -integral equations.

In this paper we consider the following functional-integral equation with linear modifications of the arguments:

$$u(x,y)=\int_0^x\int_0^y K(s,t,u(s,t),u(\lambda s,t),u(s,\mu t))dsdt,(x,y)\in[0,a]\times[0,b],$$

where $0 < \lambda < 1, 0 < \mu < 1, \quad K \in C([0,a] \times [0,b] \times \mathbb{R}^3)$.

Using the Picard operators' technique (see I.A.Rus [29]), we obtain existence and uniqueness results for the solution of this equation.

By applying the successive approximations method and by using a cubature formula (see D.V. Ionescu [16]) we give an algorithm for the approximate solution.

2 Existence and uniqueness of the solution

Let (X,d) be a metric space and $A : X \to X$ an operator. We denote by
$F_A := \{x \in X | A(x) = x\}$ - the fixed point set of A;
$A^0 := 1_X, A^1 := A, A^{n+1} := A \circ A^n, \ n \in \mathbb{N}$.

Definition 1 *(Rus [29]) The operator A is a **Picard operator** if there exists $x^* \in X$ such that:*

(i) $F_A = \{x^*\}$;

(ii) the sequence $(A^n(x_0))_{n\in\mathbb{N}}$ converges to x^*, for all $x_0 \in X$.

Definition 2 *(Rus [29]) The operator A is a **weakly Picard operator** if the sequence $(A^n(x_0))_{n\in\mathbb{N}}$ converges for all $x_0 \in X$ and its limit (which may depend on x_0) is a fixed point of A.*

Remark 1 *If the operator A is a weakly Picard operator and $F_A = \{x^*\}$, then A is a Picard operator.*

Now, we consider the functional-integral equation:

$$u(x,y) = \int_0^x \int_0^y K(s,t,u(s,t),u(\lambda s,t),u(s,\mu t))dsdt, (x,y) \in [0,a] \times [0,b], \quad (2.1)$$

where $0 < \lambda < 1, 0 < \mu < 1, \quad K \in C([0,a] \times [0,b] \times \mathbb{R}^3)$.
We consider the Banach space $(C[0,a] \times [0,b], ||.||_B)$, where

$$||u||_B = \max_{(x,y)\in[0,a]\times[0,b]} |u(x,y)|e^{-\tau(x+y)}, \tau \in \mathbb{R}_+,$$

and the operator $A : (C[0,a] \times [0,b], ||.||_B) \to (C[0,a] \times [0,b], ||.||_B)$, defined by

$$A(u)(x,y) := \int_0^x \int_0^y K(s,t,u(s,t),u(\lambda s,t),u(s,\mu t))dsdt.$$

We can write the equation (2.1) as a fixed point problem of the form: $u = A(u)$.
We have

Theorem 1 *We suppose that:*
(i) $K \in C([0,a] \times [0,b] \times \mathbb{R}^3)$;
(ii) there exists $L > 0$ such that

$$|K(s,t,u_1,v_1,w_1) - K(s,t,u_2,v_2,w_2)| \leq L(|u_1 - u_2| + |v_1 - v_2| + |w_1 - w_2|),$$

for all $(s,t) \in [0,a] \times [0,b]$ and all $u_i, v_i, w_i \in \mathbb{R}, i = 1,2$.

Then the equation (2.1) has a unique solution in $C([0,a] \times [0,b])$ and this solution can be obtained by the succesive approximation method, starting from any $u_0 \in C([0,a] \times [0,b])$.

Proof. We have

$$|A(u)(x,y)-A(v)(x,y)| \leq$$

$$\leq L(\int_0^x\int_0^y(|u(s,t)-v(s,t)|e^{-\tau(s+t)}e^{\tau(s+t)}dsdt+$$

$$+\int_0^x\int_0^y(|u(\lambda s,t)-v(\lambda s,t)|e^{-\tau(\lambda s+t)}e^{\tau(\lambda s+t)}dsdt+$$

$$+\int_0^x\int_0^y(|u(s,\mu t)-v(s,\mu t)|e^{-\tau(s+\mu t)}e^{\tau(s+\mu t)}dsdt) \leq$$

$$\leq \frac{L}{\tau^2}(1+\frac{1}{\lambda^2}+\frac{1}{\mu^2})e^{\tau(x+y)}||u-v||_B.$$

Therefore,

$$|A(u)(x,y)-A(v)(x,y)|e^{-\tau(x+y)} \leq \frac{L}{\tau^2}(1+\frac{1}{\lambda^2}+\frac{1}{\mu^2})||u-v||_B$$

for all $(x,y)\in[0,a]\times[0,b]$.

It follows that

$$||A(u)-A(v)||_B \leq \frac{L}{\tau^2}(1+\frac{1}{\lambda^2}+\frac{1}{\mu^2})||u-v||_B,$$

for all $u,v\in C([0,a]\times[0,b])$.

By choosing $\tau\in\mathbb{R}_+$ large enough, we have that A is a contraction. So A is a Picard operator. □

Now, we are looking for the solution of (2.1) in the following set

$$\begin{aligned} Y &= \{u\in C^2([0,a]\times[0,b],J)|||u||_C\leq R_1,||\frac{\partial u}{\partial x}||_C\leq R_2,||\frac{\partial u}{\partial y}||_C\leq R_3,\\ ||\frac{\partial^2 u}{\partial x^2}||_C &\leq R_4,||\frac{\partial^2 u}{\partial x\partial y}||_C\leq R_5,||\frac{\partial^2 u}{\partial y^2}||_C\leq R_6,\\ R_i &> 0, i=\overline{1,6}, J=[-r,r], r>0\}. \end{aligned}$$

Here $||.||_C$ is the Tchebyschev norm.

Consider the Banach space $(Y,||.||_B)$ and the operator $C:(Y,||.||_B)\to(C^2([0,a]\times[0,b],J),||.||_B)$ defined by

$$C(u)(x,y):=\int_0^x\int_0^y K(s,t,u(s,t),u(\lambda s,t),u(s,\mu t))dsdt,$$

where $K\in C^2([0,a]\times[0,b]\times J^3)$.

We denote

$$M=\max_{[0,a]\times[0,b]\times J^3,|\beta|\leq 2}|\frac{\partial^{|\beta|}K}{\partial s^{\beta_1}\partial t^{\beta_2}\partial u^{\beta_3}\partial v^{\beta_4}\partial w^{\beta_5}}|.$$

Therefore

$$||C(u)(x,y)||_C\leq Mab,||\frac{\partial}{\partial x}C(u)(x,y)||_C\leq Mb,||\frac{\partial}{\partial y}C(u)(x,y)||_C\leq Ma,$$

$$||\frac{\partial^2}{\partial x^2}C(u)(x,y)||_C\leq Mb[1+R_2(\lambda+2)],||\frac{\partial^2}{\partial y^2}C(u)(x,y)||_C\leq Ma[1+R_3(\lambda+2)],$$

$$||\frac{\partial^2}{\partial x\partial y}C(u)(x,y)||_C\leq M.$$

We have

Theorem 2 *We suppose that*
(i) $K \in C^2([0,a]\times[0,b]\times J^3)$;
(ii) there exists $L>0$ *such that*

$$|K(s,t,u_1,v_1,w_1)-K(s,t,u_2,v_2,w_2)| \leq L(|u_1-u_2|+|v_1-v_2|+|w_1-w_2|),$$

for all $(s,t)\in[0,a]\times[0,b]$ *and all* $u_i,v_i,w_i\in J, i=1,2$.
(iii) $Mab\leq R_1,\ Mb\leq R_2,\ Ma\leq R_3,\ Mb[1+R_2(\lambda+2)]\leq R_4,\ M\leq R_5,\ Ma[1+R_3(\lambda+2)]\leq R_6$.
Then the functional-integral equation (2.1) *has a unique solution in* Y.

Proof. The equation (2.1) can be written as a fixed point problem $u=C(u)$. The condition *(iii)* insures us that Y is an invariant subset for the operator C. Similarly as above, by using *(ii)* we obtain that C is a Picard operator. □

3 The approximate solution

We give an algorithm for the approximate solution of the equation (2.1).

We suppose that the conditions in Theorem 5 are satisfied.

Let $u^*\in Y$ be the unique solution of this equation. This solution can be obtained by the succesive approximations method starting from any $u_0\in Y$. Consider $u_0(x,y)=u_0$, where $u_0\in\mathbb{R}$. Then

$$u_1(x,y):=\int_0^x\int_0^y K(s,t,u_0,u_0,u_0)dsdt, (x,y)\in[0,a]\times[0,b];$$

$$u_2(x,y):=\int_0^x\int_0^y K(s,t,u_1(s,t),u_1(\lambda s,t),u_1(s,\mu t))dsdt, (x,y)\in[0,a]\times[0,b];$$

.....

$$\begin{aligned} u_n(x,y) &: = \int_0^x\int_0^y K(s,t,u_{n-1}(s,t),u_{n-1}(\lambda s,t),u_{n-1}(s,\mu t))dsdt,\\ (x,y) &\in [0,a]\times[0,b]; \end{aligned}$$

.....

We use the cubature formula (see D.V. Ionescu [16])

$$\begin{aligned} \int_0^a\int_0^b f(x,y)dxdy &= \frac{ba}{2pq}[\sum_{j=0}^{p-1} f(x_j,0)+\sum_{j=1}^{p} f(x_j,b)+\sum_{l=1}^{q-1} f(0,y_l)+\\ &+\sum_{l=1}^{q-1} f(a,y_l)+2\sum_{j=1}^{p-1}\sum_{l=1}^{q-1} f(x_j,y_l)]+R_f. \qquad (3.1) \end{aligned}$$

An upper bound for the remainder R_f is given by

$$|R_f|\leq\frac{ab}{12}(\frac{a^2}{p^2}+3\frac{ab}{pq}+\frac{b^2}{q^2})M_2,$$

where

$$M_2=\max_{[0,a]\times[0,b]}\left\{|\frac{\partial^2 f}{\partial x^2}|,|\frac{\partial^2 f}{\partial x\partial y}|,|\frac{\partial^2 f}{\partial y^2}|\right\}.$$

Here

$$\begin{aligned} 0 &< x_1 < ... < x_{j-1} < x_j < ... < x_{p-1} < a, \\ 0 &< y_1 < ... < y_{l-1} < y_l < ... < y_{q-1} < b, \end{aligned}$$

and $x_j = \frac{a}{p}j, j = \overline{1,p}, y_l = \frac{b}{q}l, l = \overline{1,q}$.
We have

Theorem 3 *We suppose that all the conditions in Theorem 5 are satisfied. The values of the succesive approximations sequence on the knots* $(x_i, y_k) \in [0,a] \times [0,b]$, $i = \overline{0,p}, k = \overline{0,q}$ *are*

$$\begin{aligned} u_n(x_i,y_k) &= \frac{ba}{2pq}[\sum_{j=0}^{i-1} K(x_j,0,u_{n-1}(x_j,0),u_{n-1}(\lambda x_j,0),u_{n-1}(x_j,0)) + \\ &+\sum_{j=1}^{i} K(x_j,y_k,u_{n-1}(x_j,y_k),u_{n-1}(\lambda x_j,y_k),u_{n-1}(x_j,\mu y_k)) + \\ &+\sum_{l=1}^{k-1} K(0,y_l,u_{n-1}(0,y_l),u_{n-1}(0,y_l),u_{n-1}(0,\mu y_l)) + \\ &+\sum_{l=1}^{k-1} K(a,y_l,u_{n-1}(a,y_l),u_{n-1}(\lambda a,y_l),u_{n-1}(a,\mu y_l)) + \\ &+2\sum_{j=1}^{i-1}\sum_{l=1}^{k-1} K(x_j,y_l,u_{n-1}(x_j,y_l),u_{n-1}(\lambda x_j,y_l),u_{n-1}(x_j,\mu y_l))] + \\ &+R_{n,i,k}, \end{aligned} \tag{3.2}$$

where $i = \overline{1,p}$, $k = \overline{1,q}$, $n \in \mathbb{N}^*$, *and*

$$|R_{n,i,k}| \leq \frac{ab}{12}(\frac{a^2}{i^2} + 3\frac{ab}{ik} + \frac{b^2}{k^2})M_0,$$

where M_0 *is a constant not depending on* n.

Proof. We have

$$\begin{aligned} u_n(x_i,y_k) &= \int_0^{x_i}\int_0^{y_k} K(s,t,u_{n-1}(s,t),u_{n-1}(\lambda s,t),u_{n-1}(s,\mu t))dsdt, \\ i &= \overline{0,p}, k = \overline{0,q}, n \in \mathbb{N}^*. \end{aligned}$$

By using the cubature formula (3.1), we obtain (3.2).
For $x_m \leq \lambda x_j < x_{m+1}$, we consider
$u_{n-1}(\lambda x_j, y_k) := u_{n-1}(x_m, y_k), m = \overline{0,i-1}$, and for $y_r \leq \mu y_l < y_{r+1}$, we consider
$u_{n-1}(x_j, \mu y_l) := u_{n-1}(x_j, y_r), r = \overline{0,k-1}$.
Here

$$|R_{n,i,k}| \leq \frac{x_i y_k}{12}(\frac{x_i^2}{i^2} + 3\frac{x_i y_k}{ik} + \frac{y_k^2}{k^2})M_{2,n,i,k}$$

and

$$M_{2,n,i,k} = \max_{[0,x_i]\times[0,y_k]} \left\{ |\frac{\partial^2 K_n}{\partial s^2}|, |\frac{\partial^2 K_n}{\partial s \partial t}|, |\frac{\partial^2 K_n}{\partial t^2}| \right\},$$

where,

$$K_n(s,t) := K(s,t,u_{n-1}(s,t),u_{n-1}(\lambda s,t),u_{n-1}(s,\mu t)).$$

We denote by $u_{n-1}(s,t)=\alpha, u_{n-1}(\lambda s,t)=\beta,$ and $u_{n-1}(s,\mu t)=\gamma.$ So we obtain

$$\frac{\partial K_n}{\partial s}(s,t,\alpha,\beta,\gamma)=\frac{\partial K}{\partial s}+\frac{\partial K}{\partial \alpha}\frac{\partial u_{n-1}}{\partial s}(s,t)+\lambda\frac{\partial K}{\partial \beta}\frac{\partial u_{n-1}}{\partial s}(\lambda s,t)+\frac{\partial K}{\partial \gamma}\frac{\partial u_{n-1}}{\partial s}(s,\mu t);$$

$$\frac{\partial^2 K_n}{\partial s^2}(s,t,\alpha,\beta,\gamma)=\frac{\partial^2 K}{\partial s^2}+\frac{\partial^2 K}{\partial s\partial \alpha}\frac{\partial u_{n-1}}{\partial s}(s,t)+$$
$$+\lambda\frac{\partial^2 K}{\partial s\partial \beta}\frac{\partial u_{n-1}}{\partial s}(\lambda s,t)+\frac{\partial^2 K}{\partial s\partial \gamma}\frac{\partial u_{n-1}}{\partial s}(s,\mu t)+$$
$$+(\frac{\partial^2 K}{\partial s\partial \alpha}+\frac{\partial^2 K}{\partial \alpha^2}\frac{\partial u_{n-1}}{\partial s}(s,t)+\lambda\frac{\partial^2 K}{\partial \alpha\partial \beta}\frac{\partial u_{n-1}}{\partial s}(\lambda s,t)+$$
$$+\frac{\partial^2 K}{\partial \alpha\partial \gamma}\frac{\partial u_{n-1}}{\partial s}(s,\mu t))\frac{\partial u_{n-1}}{\partial s}(s,t)+\frac{\partial K}{\partial \alpha}\frac{\partial^2 u_{n-1}}{\partial s^2}(s,t)+$$
$$+\lambda[\frac{\partial^2 K}{\partial s\partial \beta}+\frac{\partial^2 K}{\partial \alpha\partial \beta}\frac{\partial u_{n-1}}{\partial s}(s,t)+\lambda\frac{\partial^2 K}{\partial \beta^2}\frac{\partial u_{n-1}}{\partial s}(\lambda s,t)+$$

$$+\frac{\partial^2 K}{\partial \gamma\partial \beta}\frac{\partial u_{n-1}}{\partial s}(s,\mu t)]\frac{\partial u_{n-1}}{\partial s}(\lambda s,t)+\lambda\frac{\partial K}{\partial \beta}\frac{\partial^2 u_{n-1}}{\partial s^2}(\lambda s,t)+$$

$$+[\frac{\partial^2 K}{\partial s\partial \gamma}+\frac{\partial^2 K}{\partial \alpha\partial \gamma}\frac{\partial u_{n-1}}{\partial s}(s,t)+\lambda\frac{\partial^2 K}{\partial \beta\partial \gamma}\frac{\partial u_{n-1}}{\partial s}(\lambda s,t)+$$
$$+\frac{\partial^2 K}{\partial \gamma^2}\frac{\partial u_{n-1}}{\partial s}(s,\mu t)]\frac{\partial u_{n-1}}{\partial s}(s,\mu t)+\frac{\partial K}{\partial \gamma}\frac{\partial^2 u_{n-1}}{\partial s^2}(s,\mu t).$$

Because

$$\alpha=u_{n-1}(s,t)=\int_0^s\int_0^t K(s,t,u_{n-2}(s,t),u_{n-2}(\lambda s,t),u_{n-2}(s,\mu t))dsdt,$$

we have

$$\frac{\partial \alpha}{\partial s}=\int_0^t K(s,t,u_{n-2}(s,t),u_{n-2}(\lambda s,t),u_{n-2}(s,\mu t))dt,$$

$$\frac{\partial^2 \alpha}{\partial s^2}=\int_0^t(\frac{\partial K}{\partial s}+\frac{\partial K}{\partial \alpha}\frac{\partial u_{n-2}}{\partial s}(s,t)+\lambda\frac{\partial K}{\partial \beta}\frac{\partial u_{n-2}}{\partial s}(\lambda s,t)+\frac{\partial K}{\partial \gamma}\frac{\partial u_{n-2}}{\partial s}(s,\mu t))dt,$$

and

$$|\frac{\partial \alpha}{\partial s}|\le Mb, |\frac{\partial^2 \alpha}{\partial s^2}|\le Mb[1+Mb(\lambda+2)].$$

It follows that

$$|\frac{\partial^2 K_n}{\partial s^2}|\le M+M^2b+\lambda M^2b+M^2b+(M+M^2b+\lambda M^2b+M^2b)Mb+$$
$$+M^2b(1+\lambda Mb+2Mb)+\lambda(M+M^2b+\lambda M^2b+M^2b)Mb+$$
$$+\lambda M(Mb+\lambda M^2b^2+2M^2b^2)+(M+M^2b+\lambda M^2b+M^2b)Mb+$$
$$+M^2b(1+\lambda Mb+2Mb)=M_1.$$

Similarly as above we have

$$|\frac{\partial^2 K_n}{\partial t^2}|\le M_2 \quad and \quad |\frac{\partial^2 K_n}{\partial s\partial t}|\le M_3.$$

We choose

$$M_0=\max\{M_1,M_2,M_3\}.$$

□

References

[1] N.V. Azbelev, V.P. Maksimov, L.F. Rahmatulina, *Introduction to Functional-differential Equations Theory*, MIR, Moscow, 1991 (in Russian)

[2] J. Banas, I.J. Cabrera, *On existence and asymptotic behaviour of solutions of a functional integral equation,* Nonlinear Anal. 66(2007), no. 10, 2246-2254

[3] R.E. Bellman, K.L. Cooke, *Differential Difference Equations*, Acad.Press, New York, 1963

[4] S.R. Bernfeld, V. Lakshmikantham, *An Introduction to Nonlinear Boundary Value Problems*, Acad.Press, New York, 1974

[5] H. Brunner, *The discretization of Volterra functional integral equations with proportional delays,* Difference and differential equations, 3-27, Fields Inst. Commun, 42, Amer. Soc., Providence, RI, 2004

[6] Gh. Coman, G. Pavel, I. Rus, I.A. Rus, , *Introduction to the Theory of Operatorial Equations*, Ed. Dacia, Cluj-Napoca, 1976 (in Romanian)

[7] C. Corduneanu, *Integral Equations and Applications,* Cambridge University Press, 1991

[8] G. Dezso, *The Darboux-Ionescu problem for a third order hyperbolic equation with modified argument,* ACAM, 11(2002), no. 2, 69-72

[9] L.F. Elsgoltz, S.B. Norkin, *Introduction to the Theory of Differential Equations with Deviating Arguments,* MIR, Moscow, 1971 (in Russian)

[10] K. Gopalsamy, *Stability and Oscillations in Delay Differential Equations of Population Dynamics,* Kluwer, Academic Publisher, 1992

[11] D. Guo, V. Lakshmikantham, X. Liu, *Nonlinear Integral Equations in Abstract Spaces,* Kluwer, Dordrecht, 1996

[12] J.K. Hale, *Theory of Functional-differential Equations,* Springer Verlag, 1977

[13] J.K. Hale, Sjoerd M. Verduyn Lunel *Introduction to Functional-differential Equations,* Springer Verlag,New York, 1993

[14] A. Halanay, *Differential Equations: Stability, Oscillations, Time Lags,* Academic Press, New York, 1965

[15] A. Horvat-Marc, *Positive solutions of nonlinear functional integral equations, Carpathian J. Math., 19, (2003), no. 1, 67-72*

[16] D.V. Ionescu, *Cubature formula. Application to the numerical integration of the second order partial differential equations of hyperbolic type*, Studii şi Cercetări de matematică, 1960, 35-78

[17] A. Iserles, On the generalized pantograph functional-differential equation, *European J.Appl.Math.* (1992), 1-38

[18] T. Kato, J.B. McLeod, The functional-differential equation $y'(x) = ay(\lambda x) + by(x)$, *Bull.Amer.Math.Soc.*, 77, 6(1971), 891-937

[19] V. Lakshmikantham (ed), *Trends in the Theory and Practice of Nonlinear Differential Equations*, Marcel Dekker, New York, 1984

[20] V. Mureşan, *Integral equations of Volterra type with deviating argument,* Babeş-Bolyai University, Faculty of Mathematics, Preprint 3, 1984, 42-50

[21] V. Mureşan, *Succesive approximation method for a Volterra integral equation with deviating argument*, Bul. St. IPC-N, 27, 1984, 9-16 (In Romanian)

[22] V. Mureşan, *Differential Equations with Affine Modification of the Argument*, Transilvania Press, Cluj-Napoca, 1997 (in Romanian)

[23] V. Mureşan, *Functional-integral Equations*, Ed. Mediamira, Cluj-Napoca, 2003

[24] J.R. Ockendon, *Differential equations and industry*, The Math. Scientist. 5(1980) no.1, 1-12

[25] I.K. Purnaras, *A note on the existence of solutions to some nonlinear functional integral equations,* Electron. J. Qual. Theory Differ. Equ. 2006, no. 17, 24 pp. (electronic)

[26] I.A. Rus, *Metrical Fixed Point Theorems*, Univ. of Cluj-Napoca, 1979

[27] I.A. Rus, *On the problem Darboux-Ionescu,* Babeş-Bolyai University, Faculty of Mathematics, Preprint 1, 1981

[28] I. A. Rus, *Weakly Picard mappings,* Comment. Math. Univ. Caroline, 34, 4(1993), 769-773

[29] I. A. Rus, *Picard operators and applications*, Scientiae Math. Japonicae, 58(2003), Nr.1, 191-219

[30] I. A. Rus, *Weakly Picard operators and applications,* Seminar on Fixed Point Theory, 2(2001), 41-58

[31] I. A. Rus, *A class of nonlinear functional-integral equations, via weakly Picard operators*, Anal.Univ.Craiova, ser.Mat-Inf., 28(2001), 10-15

[32] J. Terjeki, *Representation of the solutions to linear pantograph equations,* Acta Sci. Math. Szeged, 60(1995), 705-713

[33] T.K. Yuldashev, J.A. Artykova, *Volterra functional integral equation of the first kind with nonlinear right-hand side and variable limits of integration,* Ukrain Math. J., 58 (2006), no.9, 1458-1461

Viorica Mureşan
Technical University of Cluj-Napoca
Faculty of Automation and Computer Science
Department of Mathematics
15 C. Daicoviciu Street
ROMANIA
E-mail: *vmuresan@math.utcluj.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

The computer's vulnerabilities

Mircea Iosif Neamţu

Abstract

The vulnerabilities are „exchange tricks" used by hackers, which allow them to enhance access possibilities by exploiting a logic error in one of the computer's code. A vulnerability can be caused by a project-error, by an error occurring after using some COM type objects, by an limit-checking-error that can occur while interpreting HTML tags to which a large number of scrip-functions has been associated to (thousands) etc. We can say that the different types of vulnerabilities and the ways in which they are exploited are random and chaotic while every way has, apparently, an unexpected outcome. My intention, through this article, is to draw attention upon the fact that vulnerabilies are an imminent threat and at the same time, to convince that this threat can be removed.

1. Weaknesses in computers

By analysing vulnerabilities, we observe that these are weaknesses within the security-plan, an error that can be exploited to overcome security.

Security is the basic structure that prevents unauthorized access to the system. When a vulnerability is exploited, the person using this vulnerability can gain some extra-influences within the system and thus, compromising its integrity.

In conclusion, there are four types of basic vulnerabilities: logic errors (e.g. an attack on a protocol), weaknesses (e.g. a program that captures packages from the web), social engineering (e.g. false phone calls) and surveillance-politics – while being related with two factors:

- the specific target of the vulnerabilities in terms of the computer or the person,
- how fast a vulnerability can be achieved.

Therefore the current features of these attributes are independent and some problems can be identified the same as the four types, mentioned above.

Most software-types inherit the vulnerabilities of the operating system. There are vulnerabilities of the operating system that represent the most direct attack-methods, showing almost instant reaction-types with unpredictable results.

A vulnerability-example for an operating-system is the ***chroot*** vulnerability. Before reporting and solving this vulnerability, it could be found within Unix systems. This vulnerability could have been used for changing access-privileges, by creating a password-file and offering a neutral password for the root account.

Applications can be represented starting from a game all the way to a web-server. The programs can also be written both by programmers with exceptional abilities and by amateurs.

Let's take into consideration an error within the Microsoft internet browser Internet Explorer.

Both the application-errors, as well as the errors from within the operating system can be found on the highest executing speed. There are errors that don't appear until a certain condition is fulfilled that the attacker requires specifying the computer that runs the application-error.
As an example we use the game-application LARN (in a package of more versions of the operating system BSD), accidentally programmed with a vulnerability that allows the administrator access to host.

Description of the game's LARN vulnerability; BSD 4.4. For e.g.: if a person scores 263 points, the game forces the system to send an e-mail to the user. The e-mail-sending process causes an IFS vulnerability that can be used to exploit the root-access.
This attack-example doesn't happen instantaneously, although it was given as an example to show that situations that are not automatically provoked can also occur.

Another feature that generates discussions, similar with vulnerabilities, is the notion of weakness of computers that often creates confusions.
Vulnerability always has a remedy, while a weakness might never have a remedy. Regarding the computer weaknesses, they say that: „a chain is only as strong as its weakest link".
Critical security-elements targeted by weaknesses are:

- security by obscurity
- encrypting
- password-security
- secured hash-functions
- old software
- old hardware
- people

Each of those are elements will break down in time, although they can be upgraded to correct that weakness.

2. The WI-FI - WEP, WPA, WPA2 vulnerabilities

WI-FI (Wireless Fidelity) is part of the top-technologies from the gamma transmissions without a cable. The support for the Wi-Fi technology is already inserted within a multitude of equipments: laptops, PDAs, mobile phones, etc., but unfortunately, the security-aspect has passed on unobserved. Let's now closely analyze the level of safety for the encrypting methods used within modern implementations of the Wi-Fi technology.

Even if the security methods are activated on the Wi-Fi equipments, in most cases, the used WEP protocol is a low level encrypting-protocol. An analysis of the WEP protocol demonstrates the vulnerability of this system by using a WEP-key. We observe the need for introducing a new security-architecture in the form of the 802.11i standard for solving some of the vulnerabilities of the Wi-Fi technology.

2.1.1. WEP (Wired Equivalent Privacy)

The WEP encrypting-protocol wasn't created by experts from the security field or from the encrypting area, fact that has thus generated the vulnerability within the RC4 algorithm problem. This problem was first described by David Wagner, Scott **F**luher, Itsik **M**antin and Adi **S**hamir who presented the two vulnerabilities of the RC4 algorithm. The two attacking methods rely on the fact that for some key-values it's possible that some bits from the actual bit depend only on a couple of bits from the encrypting-key (normally, each key has a 50% chance to be different from the latter). WEP has been the implicit encrypting protocol first introduces in the first standard IEEE 802.11 in 1999. At the base of the WEP protocol lies the RC4 encrypting algorithm, this includes a secret key on 40 or on 104 bits combined with an initializing vector (IV) on 24 bits for encrypting the text message (M) and the control bits (checksum, ICV–Integrity Check Value).

The encrypted message (C) was determined by using the following formula:

$$C = [M \,||\, ICV(M)] + [RC4(K \,||\, IV)] \quad (1)$$

Where: „||" is the concatenating operator and „+" is the XOR operator.

The initializing vector (IV) is the WEP security-key and is used for maintaining a medium security level and to minimize detection.

The IVs must be enhanced within every package so that all following packages are encrypting with different encrypting-keys. Unfortunately for the WEP security, the IVs are transmitted only in a simple text within the 802.11 standard and don't require incrementing the IVs allowing the implementers of the particular wireless terminals to chose the needed security measure (access points and WLAN board).

After presenting this vulnerability, the fact that the WEP protocol offers an acceptable level of security only for its use within applications of lesser importance, was realized. Afterwards, this level of security was crushed once attacks appeared that allow arbitrary packages to be decrypted without knowing the key (KoreK and Arbaugh) by using the injection-method. It was proven that cracking tools such as Aircrack and Weplab using this method succeed in recuperating the key encrypted by 128 bits in less than 10 minutes. As a recommendation, the WEP protocol shouldn't be used, not even with the rotation of the encrypting key.

2.1.2. Treating vulnerabilities within the WiFi 802.11i technology

January 2001 the „i" group within the IEEE was created to enhance security of the authentication and encryption of data within the 802.11 standard. April 2003, the WiFi alliance published a recommendation regarding wireless security within companies. Concern was shown regarding the consumers' availability to change their existing equipments. June 2004 the last update of the IEEE 802.11i standard was adopted, being given the commercial name WPA2 from the WiFi alliance. This standard introduced some fundamental changes amongst which the most important being the separation of the user's authentication from the integrity of the message. This creates a robust security-architecture and scalable applicable to any kind of network. By using the new standard, the new architecture for wireless networks called Robust Security Network (RSN) uses another key distribution and a new integrity mechanism. Because the RSN architecture is more complex, it offers safety-solutions and scalability for wireless communication. Generally, a RSN only accepts RSN-capable equipments. The new standard defines also a network security architecture to which both RSN and WEP systems can participate, thus allowing users to change their equipment in time. The authentication procedures or the association procedures that are used between stations for the communication method in 4ways (4-way handshake) are called RSNA (Robust Security Network Association).

In order to achieve a secured connection, four well defined fazes must be run through.

1. Convention upon the security policy.
2. Authentication 802.1x.
3. Grading keys and distribution.
4. RSNA, confidentiality and integrity of data.

2.1.3. Weaknesses of the WPA/WPA2 protocols

Since publishing the WPA/WPA2 protocols, a small number of minor security deficiencies were found, none being though too dangerous while respecting the conditions for a simple security recommendation.

Regarding vulnerabilities, the fiercest is the attack against the PSK key for WPA/WPA2 protocols. The PSK offers an alternative to the 802.1 X PMK generations by using an authentication server. For using a known algorithm such as PSK a sequence of 256 bits is needed or an access password containing 8 to 63 characters for generating the line.

The algorithm PSK = PMK = PBKDF2 (password, SSID, length SSID, 4096, 256), where: PBKDF2 is the used method, 4096 is the number of hashes, and 256 is the exit length. Worth mentioning would be

that the second message within the information exchange in all four ways can be subjected both to dictionary attacks, as well as to the bruteforce –type ones. This is why a utilitarian was invented to exploit this error, its source-code being used and improved by Christophe Devine in Aircrack. The Aircrack utilitarian can be used for PSK type attacks (dictionary and bruteforce) on wireless networks with WPA security protocols.

Because of the protocol's component (4096 hashes for each password try) a bruteforce attack is very slow (with a processor of the latest generation with a single nucleus only about 100 passwords can be loaded per second).

The PMK can't be pre-calculated because the password is encrypted based on the ESSID. To protect ourselves from this vulnerability we must have a password with a minimum of 20 characters which isn't a word from a dictionary.

The attack consists in capturing the exchange of information in the 4 ways (4-way Handshake) by passively monitoring a wireless network or by using an authentication attack to speed up this process. Actually, the first two messages are required to start guessing the PSK values.

The second important weakness of the WPA protocol is the DoS possibility during the exchange of information in the 4 ways. John Mitchell noticed that the first message within the 4 ways isn't authenticated; each client must save his message until receiving a signed message leaving the client vulnerable until exhausting the memory. By listening to the first message sent by the access point, the attacker can perform a DoS to the client with more simultaneous sessions.

The last known weakness is the possibility of a theoretic attack against the temporary hashes of the WPA key protocol. This attack implies a low level complexity in some circumstances and also knowing more RC4 keys.

WPA/WPA2 also are subject to vulnerabilities that affect other mechanism of the 802.11i standard, such as: e.g. the listening-attack of 802.1X messages (EAPoL Logoff, EAPoL Start, EAP Failure etc.), first described by William Arbaugh and Arunesh Mishra; this being possible in case of a lacking authentication.

Not last, it's important to know that the usage of the WPA/WPA2 protocol doesn't provide a current protection against the attacks that stand at the base of these technologies, such as blocking radio frequencies.

2.1.4. Implementing WPA/WPA2 in SO

Within the operating system Windows XP, the support for the WPA2 protocol isn't built in. An update of the operating system Windows XP SP2 (KB893357) was performed on the 29th of April 2005 containing the support for the WPA2 protocol with an improvement of detecting wireless networks.

Other Microsoft operating systems must use an external applicant (commercial or Windows open source). The Linux and BSD platforms, *wpa_supplicant* were already prepared for WPA2 as soon as the 802.11 standard was published.

The external applicant only bears a large number of EAP methods and key management-methods for WPA, WPA2, WEP. More networks can be declared with different types of encryptions, key management and EAP methods. The default location of the wpa_supplicant configuration file is /etc/wpa_supplicant.conf and the file must only be accessed by the root.

On the Macintosh platform the WPA2 protocol already had a support at release of the 4.2 update of the Apple AirPort software.

2.1.5. Firewall protection

A frequent encountered error is considered to be the *firewall* application which can detect and block a message mail based attack.

The *firewall* applications control the network connectivity and normally don't verify the traffic taken place on the standard e-mail port (25). The network administrator can add rules that allow specifying the types of traffic that have the permission of passing the firewall. For example, the traffic

on the 25th port is seeing as an e-mail, which might allow an attacker the use of the 25th port for launching an attack.

2.2. Conclusion

It's obvious that encrypting with the WEP protocol doesn't offer a sufficient wireless security and can only be used with high level encrypting solutions (e.g.: VPN).

The WPA protocol is a secure solution for equipments that still don't have the support for the WPA2 protocol, protocol that will become a standard wireless security. Let's not forget the fact that the wireless equipment must be placed in a filtered area always prepared for a cable connection for critical networks because the blockage of radio frequencies and low level attacks can still be devastating for a wireless network with the encrypting protocol WPA2.

3. Case-study: Treating method of vulnerabilities infecting memory sticks

Because the use of memory sticks has increased, the malwares have started to take advantage of this situation to spread rapidly. Because a single memory stick can be used on many computers, this has become a problem for users. The following presented method can be helpful in diminishing the risk of infecting a memory stick used daily on more than one computer.

The malwares use two techniques for spreading. The first technique is characterized by the fact that it infects the executable files from the stick and by executing those files they migrate to the computers.

The most common and dangerous method is spreading through the ***„autorun.inf"*** file, file which the operating Windows system automatically executes after inserting the memory stick into the computer. Many malwares use this spreading technique (e.g. the *Conficker* worm).

3.1. Solution

One solution would be to deactivate the *autorun* mechanism from within the Windows SO, this requires although technical IT knowledge.

Because most users don't have this certain knowledge, preventing the execution of the *autorun.inf* file must be directly solved on the memory stick.

Before applying this method we can also say that there also exists a possibility of acquiring a type of memory sticks that contain a switch for „read only" or „write-read" modes. These sticks thus can be blocked so that transfer of viruses can't be made from the computer to the stick or the other way around. If the file system of the stick is a FAT32 (< 8GB+) type, than a small trick can be applied presented next. This trick needs a hexa editor and some basic knowledge regarding the director table of the FAT32 filing system.

Step 1. A blank file is created on the stick, called „autorun.inf",

Step 2. The stick is being opened with the hexa editing program.

Observation: It's not important whether the physical disc or the local partition is being opened but we must assure that the stick was opened with the required read/write permissions and that no other program is accessing it in that moment.

As an example, I used an HxD, a portable hexa editor. If we don't have a new stick and our stick already contains information, it's recommended that this information is copied on another support and the stick to be formatted. After creating the empty file called „autorun.inf" and opening the stick with HxD, the hexa editor, we than start a search for „AUTORUN", as a Unicode string text. If the stick was empty, then it must be found right at the beginning of the disc.

The interest area is the following:

41 55 54 4F 52 55 4E 20 49 4E 46 20
A U T O R U N I N F

The first 8 bytes stand for the name of the file (followed by a space), followed by 3 bytes (the file extension), and followed by 1 byte containing the attributes of the file. This last byte is the relevant one. The current value of this byte is (0x20), meaning that the file has only the archive byte set.
Our objective is to change this byte into 0x40 (setting the bit of the equipment).
The area then changes into:

41 55 54 4F 52 55 4E 20 49 4E 46 40
A U T O R U N I N F @

Once this change has been saved on the disc, ignoring any warning that might corrupt the disc, we first remove it from the computer and then we reattach it to the computer.
Now, the file *autorun.inf* is visible but can't be deleted, edited, overwritten or have its attributes changed.

When this stick is connected to an infected computer that tries creating a *autorun.inf* file on the stick, it will inform that the „file can't be created" which means that it can't be infected through this method (*autorun.inf)* and can't spread an infection to other computers.

4. Conclusion

This method is addressed to all users of memory-sticks. If this technique will start being used by more and more users, it will surely assure a simple but efficient method of protection against the distribution of the M-stick viruses.

References

1. Protected Access and 802.11i (John Edney, William A. Arbaugh) – Addison Wesley – ISBN: 0-321-13620-9,
2. The Insecurity of 802.11 (Borisov, Goldberg, Wagner),
3. Criptografia şi Securitatea Reţelelor de Calculator (Victor-Valeriu Patriciu), Editura Tehnică
4. Protecţia şi Securitatea Informaţiilor (Dumitru Oprea), Editura Polirom
5. Insider Attack and Cyber Security Beyond the Hacker (Salvatore J. Stolfo, Steven M. Bellovin), Springer
6. Securitatea Sistemelor Linux (Dragos Acostachioaie), Editura Polirom
7. Assessing Network Security (Kevin Lam, David LeBlanc, Ben Smith), Microsoft Press
8. Hack Proofing Your Network (Internet Tradecraft), Syngress Publishing
9. The Hacker Crackdown. Law and Disorder on the Electronic Frontier (Sterling B.)
10. Building Internet Firewalls (Chapman B.), O'Reilly and Associates
11. Practical UNIX and Internet Security (Garfinkel S., Spafford G.), O'Reilly and Association
12. Hackers Beware (Cole E.), New Riders Publishing
13. Designing Secure Web-Based Applications for Windows 2000 (Howard), Microsoft Press

Weblinks

1. http://standards.ieee.org/getieee802/download/802.11i-2004.pdf – IEEE 802.11i standard,
2. http://www.awprofessional.com/title/0321136209 – Real 802.11 Security Wi-Fi
   http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm – An inductive chosen plaintext attack against WEP/WEP2 (Arbaugh),
3. http://www.cr0.net:8040/code/network/aircrack/ – Aircrack (Devine)

Mircea Iosif Neamţu
"Lucian Blaga" University
Computer Science Department
Str. Dr. Ioan Raţiu nr.5-7 ROMÂNIA
e-mail: neamtu12@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

The art of teaching and learning online

Cristina Pop, Vasile Cornita

Abstract

Good coaching or moderation of online courses by teachers, trainers and experts is often a critical success factor of e-Learning, especially when working with groups, who are new to computer technologies and online learning. E-Learning combines education with the medium internet used to impart knowledge. Principal components of online learning programs are contents in the form of texts, pictures, animations, audio and video sequences. The access to the course contents can take place without any time or regional limit and the learner can study with the help of the computer at his work place or even at home. As the contents are dynamic they can be combined optionally and can get updated easily, too. Via communication services like e-mail, chat or forums the learner can communicate with the teacher and also with other learners. Computer, information, and communication technologies certainly play a major role and are a basic requirement for e-Learning.

1 Introduction

This paper examines how online learning is changing the traditional forms of teaching used in education. It is argued that, if managed correctly, online learning methods can add value to traditional face-to-face methods and provide opportunities for reducing some of the weeknesses of traditional teaching methods. Online learning takes on its forms, structures and discourse, in the same way as traditional classroom-based and face-to-face learning. What is fundamentally different is the nature of the medium and its added dimensions in time and place. The change of medium offers new opportunities to move some of the contemporary and the traditional approaches towards learning in more interesting and efficient ways.

2 General aspects of e-Education, e-Teacher, and e-Learning

E-Education involves e-teaching and e-learning along with the various administrative and strategic measures needed to support teaching and learning in an Internet environment. It will incorporate a local, regional, national and international view of education. E-teachers are the new generation of

teachers who will work in an Internet environment in both regular and virtual classroom situations. They will build new concepts of working in time and space. E-teachers collaborate, build and discover new learning communities and explore resources as they interact with information, materials and ideas with their students and colleagues. E-learning is learning which takes place as a result of experiences and interaction in an Internet environment. It is not restricted to a regular school day and can take place in a variety of locations including home, school and community locations e.g. libraries, cafes etc. An effective e-learning strategy must be more than the technology itself and the content it carries. E-learning is flexible learning using Internet resources, tools and applications, and focusing on interaction among teachers, learners, and the online environment and on collaborative learning.

3 Advantages of e-Learning

The target group for the e-Learning is represented by scholarships and students. They need to accumulate a large variety of knowledge, to learn how to think, to connect all school value with the life reality, to transform the theory into practice and so on. For this reasons, the e-content prepared to be delivered by IT&C infrastructure, designed for self learning or assisted by the teacher, has to be in full accordance with the didactical and pedagogical principles.

- Students participate from a variety of locations and may "attend" multiple learning institutions and/or their local school.
- Students may determine the times when they access e-learning opportunities.
- Students can choose to work individually or collaboratively with people who may or may not be in their regular class.
- Classes may be synchronous or asynchronous.
- Students may take classes from more than one school.
- Students may set their own objectives and explore their own learning needs and agendas.
- Students can follow a non-linear path at a pace that meets their individual needs at that time, i.e. just-in-time learning. The teacher is facilitating the activity
- E-teachers can work in more than one school.
- Students can proceed at their own pace
- Students can replay audio lectures or video clips
- Slower students do not slow down their classmates

4 Comparative analysis of the most popular elearning systems

This paragraph presents some advantages and disadvantages of the most popular elearning systems. Geoplan-Geospace is a 2D and 3D interactive geometry software to create and manipulate interactive figures for eLearning. It features euclidean geometry objects (lines, circles), numerical functions, numerical sequences, geometrical transformations. Geoplan-Geospace is a mathematical software which is dedicated to math-teaching/learning from primary school level up to University. Its new multilingual interface is particularly adapted to international exchanges and to English-speaking schools. Geoplan-Geospace is the result of over ten years of a work led by a team of math-teachers and didacticians, and has become a standard in French educational system. Geoplan-Geospace offers an opportunity to produce and manipulate numerical and geometrical objects in plane as well as in space. Geoplan-Geospace is a software for mathematical constructions that allows dynamic and interactive representations. Another eLearning system for mathematics is WMI (WebMathematics Interactive). WMI is an open web-based eLearning system for grammar school and/or university students, including basic mathematical functions (equation solving, function plotting) and thematic modules. It is like a computer algebra system, but with an easy accessibility on the web and support for educational purposes.[7]

Interact is a platform for the delivery and support of online learning. It differs from many other elearning platforms in that its aim is to concentrate on the social/interactive aspects of teaching and learning. It is an open source online community environment designed with the intention of making it easy for people to interact online, based around constructivist views of learning. The key advantage of Interact is the ability to structure a learning space the way you want it, rather than being stuck with a predefined separation of content and interaction. The user starts off with a blank slate and can build their space up anyway they see fit using a selection of 'Components'. There is no preset structure that forces the user into any particular pedagogy. Interact is a flexible, open source, online learning and community management environment. It can be used as a full intranet/portal as well as an online community environment. Features include a user-centred interface that works on most browsers, and powerful, feature-rich, interactive content and multimedia components for creating new and dynamic online experiences. In essence this means that people learn best as they interact and engage with others. With Interact, use of discussion forums, blogs, and sharing areas mean that communication occurs frequently between members. Interact is committed to providing tools that encourage and facilitate online collaboration and interaction.

V-MAP is a visual mapping tool for generating ePortfolios for Personal Development Plans, implemented on the Java platform. It is a project funded by JISC as part of their eLearning tools initiative. The objectives of the project are to:

- Enable the learner to interact with a pre-defined institutional template for preparing e Portfolios in a format that is most accessible to the learner;
- Enable the learner to plan, construct and update an e Portfolio through the use of a visual mapping interface, on their own personal desktop;
- Enable the learner to share, publish and disseminate an e Portfolio through the use of a visual mapping interface, to their institutional system.

Dokeos is an elearning and course management web application (LMS), translated in 34 languages, already helping thousands of organisations worldwide to manage learning and collaboration activities. It has many tools, is light and flexible, and free software. Advantages of Dokeos system: [7]

- SCORM is the norm for e-learning content import/export. Most professional courses today are delivered as SCORM packages.
- Convert a Powerpoint or Openoffice slideshow into an online SCORM course, add audio on the slides and tests between the slides.
- Build web pages with diagrams, animations, video, audio etc. in an online editor relying on templates and media galleries.
- Multiple choice, fill-in-the-gaps, open question, hotspot, choose the type of question in accordance to the competence to assess.
- Assemble multimedia content, tests and interaction activities into a SCORM learning path and get accurate reporting on completion, time and score step by step.
- Import all types of documents into the course and organize the info using folders and sub-folders.
- Register users individually or by groups in the system and in the courses through the Dokeos administration panel.
- Manage groups of users and associate them to courses in time-based sessions under the supervision of a coach.
- Teach or meet remotely live using audio and video duplex + a whiteboard and a shared presentation.

5 Proposed architecture of an advances e-learning framework

The general architecture of the proposed e-learning framework uses a typical 3 layered software architecture. The main responsibilities of server include learning modules storage on a pertaining web server and delivery of e-learning data content on client software application request. The remote client application sends requests to Server and snapshots with student actions on predefined time intervals set by the trainer. For a particular client request, the server decides on and takes the appropriate action.

This can be represented graphically as below:

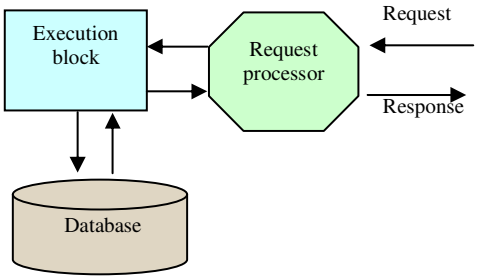


Fig.1. e-Learning server general architecture

The main functions of each block are explained below:

- Request processor takes as input a request and interprets it. After all request characteristics have been determined, it is up to the execution block to act appropriately.
- Execution block communicates with both the request processor and the e-learning Database content in order to execute the client request. It must be mentioned that this layer executes when the request characteristics have been determined by the request processor layer and only then.

Both, requests and responses can take message or file transfer forms, depending on the specific context.

A simplified graphical representation of the Client application architecture is depicted below:

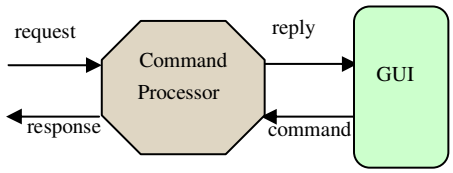


Fig.2. Client software application general architecture

As it can be easily seen, all that happens is driven by user commands, making use of a well-designed graphical user interface. The command processor is a logical entity that transforms client requests into an appropriate format for the server to understand and process, including here http request to the web server contained by the e-Learning server. All these requests are transported between client and the server encapsulated in a general request type, this ensuring both, flexibility and extensibility for the request transport level. The communication mechanism between client and server uses both messages and files. The data interchange operation is realized via file transfer making use of standard XML language. Also, when a client request a connection to the e-Learning server a user and password has to be provided in order to fulfil the connection. For each user predefined settings can be set by the corresponding teacher such as lessons to attend, obtained marks, to ensure a controlled learning environment, but leaving in the same time the learner the possibility to advance in his own way.

The novelty of this proposed architecture is that web content is delivered via an graphical user interface that is capable both , of visualization of e-learning data from server and of server monitoring of user actions like intervals between mouse clicks on different e-learning content, navigation tree and verification questions via which the e-learning server is capable of deciding which level of e-learning content to deliver on subsequent lessons as well as giving the monitoring teacher the possibility to visualize the attention given to a specific lesson by a group of student.

For implementing the web Server part of the proposed e-Learning server and connecting Client software applications general programming books [8] [9], C++ programming books [10 [11], and socket programming books [12] [13] have been used.

6 Conclusions

Online learning is the solution for people who want to gain education but don't have time to go to school. The infusion of the ICT in the educational systems, at different levels and different forms, as a possible solution for identified problems, became familiar. But use of ICT in the educational landscape assembled a lot of other questions marks concerning the educational value of this use in terms of learning quality. Instruction appears to be a complex matter and simple “look and feel” in front of computer does not signify learning and/or teaching. The literature focused on ICT use in education report its benefits in enhancing aspects of the teaching/learning processes. But those benefits are not only the result of putting a student in front of a computer. The computer must provide a learning environment, a space where learning process is enhanced in some aspects of teaching process are also included. The aim is to provide an overview concerning the development of the learning environments provided true computers, emphasizing the design phase that has the most important role in the prediction of the learning quality.

References

[1] Maggi Savin-Baden, *Problem-based learning online,* Open University Press 2006, pp. 121-132.
[2] Shirliey Bach, Phillip Hayness, Jennifer Lewis Smith, *Online learning and teaching in higher education,* Open University Press 2007, pp. 180-191.
[3] Sorin-Aurel Moraru, *Preparatory Module Course,* Lux Libris 2004, pp. 112-117.
[4] Doina Fotache, *Groupware. Metode, tehnici si tehnologii pentru grupuri de lucru,* Polirom 2002, pp. 78-82.
[5] Mihaela Brut, *Instrumente pentru e-learning. Ghidul informatic al profesorului modern,* Polirom 2006, pp. 110-115.
[6] Implementation of *Education and Training 2010* work programme, European Commision Directorate General for Education and Culture.
[7] http://sourceforge.net
[8] Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest (2001). Introduction to algorithms, 2nd edition, The MIT Press.
[9] Anthony Jones, Jim Ohlund (2002). Network Programming for Microsoft Windows, Microsoft Press.
[10] Shaharuddin Salleh, Albert Y. Zomaya, Sakhinah A. Bakar (2007). Computing for Numerical Methods Using Visual C++, 1st edition.
[11] Bjarne Stroustrup (2000). The C++ Programming Language: Special Edition, 3rd Edition.
[12] Dave Roberts. Developing for the Internet with WinSock, Bk&CD-Rom edition, Coriolis Group Books.
[13] Jeffrey Richter, Christophe Nasarre (2007). Windows via C/C++ (Pro - Developer), 1st edition.

Cristina Pop
Academy of Economic Studies of Bucharest
Faculty of Cybernetics, Statistics and Informatics
Calea Dorobantilor 15-17, Sector 1
ROMANIA
E-mail: cristinel19@yahoo.com

Vasile Cornita
University Politehnica of Bucharest
Faculty of Electronics, Telecommunications and Information Technology
B-dul Iuliu Maniu 1-3, Sector 6
ROMANIA
E-mail: cornita_vasile@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

A comparison between two collocations methods for linear polylocal problems - a Computer Algebra based approach

Daniel N.Pop , Radu T. Trîmbiţaş

Abstract

Consider the problem:

$$\begin{aligned} -y''(t) + q(t)y(t) &= r(t), \qquad t \in [a,b] \\ y(c) &= \alpha \\ y(d) &= \beta, \qquad c, d \in (a,b). \end{aligned}$$

The aim of this paper is to present two approximate solutions of this problem based on B-splines and first kind Chebyshev polynomials, respectively. The first solution uses a mesh based on Legendre points, while the second uses a Chebyshev-Lobatto mesh. Using computer algebra techniques and a Maple implementation, we obtain analytical expression of the approximations and give examples. Chebyshev method has a smaller error, but for large number of mesh points the B-spline method is faster and requires less memory.

1 Introduction

Consider the problem:

$$-y''(t) + q(t)y(t) = r(t), \qquad t \in [a,b] \tag{1}$$
$$y(d) = \alpha \tag{2}$$
$$y(e) = \beta, \qquad d, e \in (a,b), d < e. \tag{3}$$

where $q, r \in C[a,b]$, $\alpha, \beta \in \mathbb{R}$. *This is not a two-point boundary value problem*, since $d, e \in (a,b)$.

If the solution of the two-point boundary value problem

$$\begin{aligned} -y''(t) + q(t)y(t) &= r(t), \qquad t \in [d,e] \\ y(d) &= \alpha \\ y(e) &= \beta, \end{aligned} \tag{4}$$

exists and it is unique, then the requirement $y \in C^2[a,b]$ assures the existence and the uniqueness of (1)+(2)+(3).

We have two initial value problems on $[a,d]$ and $[e,b]$, respectively, and the existence and the uniqueness for (4) assure existence and uniqueness of these problems. It is possible to solve this problem by dividing it into the three above-mentioned problems and to solve each of these problem separately, but we are interested to a unitary approach that solve it as a whole.

In 1966, two researchers from Tiberiu Popoviciu Institute of Romanian Academy, Cluj-Napoca, *Dumitru Ripianu* and *Oleg Arama* published a paper on a polylocal problem, see [9].

2 Principles of the method

The implementation is inspired from [4, 5].

2.1 B-spline method

Our first method is based on collocation with nonuniform cubic B-splines [2, 10]. For properties of B-spline and basic algorithms see [5].

Consider the mesh (see [1])

$$\Delta : a = x_0 < x_1 < \cdots < x_m < x_{m+1} = b, \tag{5}$$

and the step sizes

$$h_i := x_{i+1} - x_i, \qquad i = 0, \ldots, m.$$

Within each subinterval we insert k points

$$0 \leq \rho_1 < \rho_2 < \cdots < \rho_k \leq 1,$$

which are the roots of the kth Legendre's orthogonal polynomial on $[0, 1]$ [6, 8].

Finally, the mesh has the form

$$\xi_{i,j} := x_i + h_i \rho_j, \qquad j = 1, \ldots, k, \qquad i = 0, \ldots, m.$$

The number of mesh points is now $N = (m + 1)k$.

We shall choose the basis such that the following conditions hold:

- the solution verifies the differential equation (1) at $\xi_{i,j}$;
- the solution verifies the conditions (2), (3).

We need a basis having $N + 2$ cubic B-spline functions.

One renumbers the points such that the first point is x_0 and the last is x_{n+1}.

In order to impose the fulfillment of (1) at a and b we complete the mesh with points $x_{-k}, x_{-k+1}, \ldots, x_{-1}$ and $x_{n+2}, x_{n+3}, \ldots, x_{n+k+1}$.

The form of solution is

$$y(t) = \sum_{i=-1}^{n+2} b_i B_i(t), \tag{6}$$

where $B_i(t)$ is the B-spline with knots $x_{k-2}, x_{k-1}, x_k, x_{k+1}, x_{k+2}$.

The conditions on solution yield a linear system with $n + 4$ equations and $n + 4$ unknowns (the coefficients b_i, $i = -1, \ldots, n + 2$).

The system matrix is banded with at most 4 nonzero elements on each line (3 nonzero at each mesh point and four at d and e).

2.2 Chebyshev method

Our second method is based on first kind Chebyshev polynomials [6, 8]. We consider the mesh

$$\frac{b-a}{2} \cos \frac{k\pi}{n} + \frac{a+b}{2}, \qquad k = 0, \ldots, n \tag{7}$$

(the extremes of Chebyshev #1 polynomials, or equivalently the roots of Chebyshev #2 polynomials) completed with inner points c and d. The form of the solution is

$$y(t) = \sum_{i=0}^{n+1} c_i T_i(t); \tag{8}$$

where $T_i(t)$ is the k-th degree first kind Chebyshev polynomial on interval $[a, b]$. As in the previous section, the fulfillment of (1), (2) and (3) leads us to a system of $n + 2$ equations and $n + 2$ unknowns (the coefficients c_i, $i = 0, \ldots, n + 1$). This time the matrix is dense.

3 Maple implementation

We implement our ideas in Maple 10. For necessary details on Maple see [7].
Both methods return the approximation in analytic form.

3.1 B-spline method

The basic functions are computed using the function `BSpline` of the package `CurveFitting`. The B-spline basis is obtained through Maple sequence

```
> S:=(x,u,k)->eval(BSpline(4,t,
> knots=[seq(u[i],i=k-2..k+2)]),
> t=x):
```

`S(x,u,k)` computes the cubic B-spline in variable `x`,
with knots `u[k-2],...,u[k+2]`.

The procedure `genspline` computes the B-spline solution. It accepts the mesh `x`, the number of points `n`, the functions `q` and `r`, the points `d`, `e` and the values at `d` and `e`, `alpha` and `beta`, respectively. It returns the solution `y`, given by (6). The matrix of the system and the right-hand side vector are constructed element by element and the solution is computed using the function `LinearSolve` from `LinearAlgebra` package. This is a fast and flexible solution, and allows the selection of the solution method and gaining additional information, like condition number. Here is the Maple code.

```
> genspline:=proc(x,n,q,r,d,e,
> alpha,beta)
> local k, i, A, y, poze, pozd, ii,p,xe,xd, Y;
> global S, b;
> A:=Matrix(n+4,n+4); y:=Vector(n+4):
> b:=Vector(n+4):
> ii:=1;
> for i from 0 to n+1 do
> for k from max(i-1,-1) to i+1 do
> A[ii,k+2]:=(-eval(diff(S(t,x,k),
> t$2), t=x[i])+q(x[i])*
> eval(S(t,x,k),t=x[i]));
> end do:
> y[ii]:=r(x[i]);
> if (x[i]<d and x[i+1]>d) then
> ii:=ii+1; pozd:=ii; xd:=i;
> end if:
> if (x[i]<e and x[i+1]>e) then
> ii:=ii+1; poze:=ii;  xe:=i;
> end if:
> ii:=ii+1;
> end do:
> p:=xd;
> for k from p-1 to p+2 do
> A[pozd,k+2]:=eval(S(t,x,k),t=d);
> end do;
> y[pozd]:=alpha;
> p:=xe;
> for k from p-1 to p+2 do
> A[poze,k+2]:=eval(S(t,x,k),t=e);
> end do;
> y[poze]:=beta;
> b:=LinearSolve(A,y);
> Y:=0:
> for k from -1 to n+2 do
> Y:=Y+b[k+2]*S(t,x,k):
> end do:
> return Y:
> end proc:
```

The procedure `genspline` accepts the mesh given in array form. The procedure `gendivLeg` generates the mesh as shown in Section 2.1. It calls the procedure `genpoints`. It computes the Legendre polynomial, solve it, and generates mesh points using an affine transform. The Legendre polynomials are generated using the `orthopoly` package, and their roots are obtained via `solve` function. Here is the code for `genpoints`:

```
> genpoints:=proc(a,b,N,k)
> local L,i,j,xu,xc,pol,pol2,sol,
> h,nL,x;
> L:=[a];
> h:=(b-a)/(N+1);
> xc:=a-h; pol:=P(k,t);
> pol2:=expand(subs(t=2*x-1,pol));
> sol:=fsolve(pol2);
> for i from 0 to N+2 do
> xu:=xc+h;
> for j from 1 to k do
> L:=[op(L),xc+(xu-xc)*sol[j]];
> end do;
> xc:=xu;
> end do;
> L:=[op(L),b];
> L:=sort(L);
> return L;
> end proc:
```

The code for `gendivLeg` closes the section.

```
> gendivLeg:=proc(a,b,n,k)
> local h,x,Y,L,nn,j:
> L:=genpoints(a,b,n,k);
> L:=convert(L,rational,exact);
> nn:=nops(L)- 2*k;
> x:=Array(-k..nn+k-1,L):
> return x;
> end proc:
```

3.2 Chebyshev method

The Chebyshev polynomials are generated via the `orthopoly` package. The Maple sequence

```
> S:=(x,k,a,b)->T(k,
> ((b-a)*x+a+b)/2):
```

computes the k-th degree Chebyshev polynomial on interval $[a, b]$. The following Maple procedure `genceb` is the analogous of `genspline`. It uses `solve` to compute the Chebyshev coefficients.

```
> genceb:=proc(x,n,q,r,c0,d0,
> alpha,beta)
> local k, ecY, ecd, C, h, Y, c, a, b;
> global S;
> a:=x[0]; b:=x[n-1];
> Y:=0;
> for k from 0 to n+1 do
> Y:=Y+c[k]*S(t,k,a,b);
> end do;
> Y:=simplify(Y);
> ecY:=-diff(Y,t$2)+q(t)*Y=r(t):
> ecd:=Array(0..n+1);
> for k from 0 to n-1 do
> ecd[k]:=eval(ecY,t=x[k]):
> end do;
> ecd[n]:=eval(Y,t=c0)=alpha:
> ecd[n+1]:=eval(Y,t=d0)=beta:
> C:=solve({seq(ecd[k],k=0..n+1)},
> [seq(c[k],k=0..n+1)]);
> assign(C):
> return Y:
> end proc:
```

The mesh points are the roots of the n-th degree second kind Chebyshev polynomials (formula (7)) and the points c and d.

4 Numerical examples

We present two examples: one with a nonoscillating solution and the other with oscillating solution. A problem with a nonoscillating solution is simple and does not require a large computational effort. A problem with an oscillating solution is harder, and requires a mesh with a large number of points. The methods do not depend on conditions on $q(x)$. We solved our examples using both methods. For each example and method we plot the exact and the approximate solution and generate the execution profile (with the pair `profile` - `showprofile`). The first example is from [3, page 560]

$$\begin{aligned} -y'' - y &= x, \qquad x \in [0,1] \\ y\left(\frac{1}{6}\right) &= -\frac{1}{6}\frac{-6\sin\frac{1}{6} + \sin 1}{\sin 1}, \\ y\left(\frac{3}{4}\right) &= -\frac{1}{4}\frac{-4\sin\frac{3}{4} + 3\sin 1}{\sin 1}. \end{aligned}$$

The exact solution is $Z(t) = -\frac{-\sin(t)+t\sin 1}{\sin 1}$, and we computed it using `dsolve`. We chose $n = 10$ for both methods and $k = 3$ for the first method. Figure 1 shows the exact and the approximate solution computed using the first method. The error plot in a semilogarithmic scale is given in Figure 2.

The corresponding graphs for Chebyshev methods are illustrated in Figures 3 and 4.

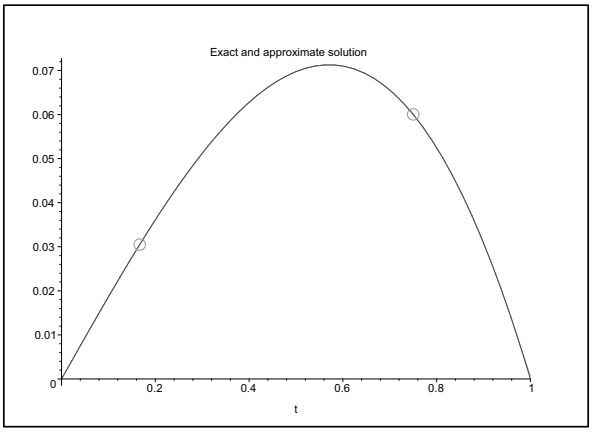


Figure 1. The graph of exact and approximate solution, nonoscillating problem, B-spline method, $n = 10$**,** $k = 3$

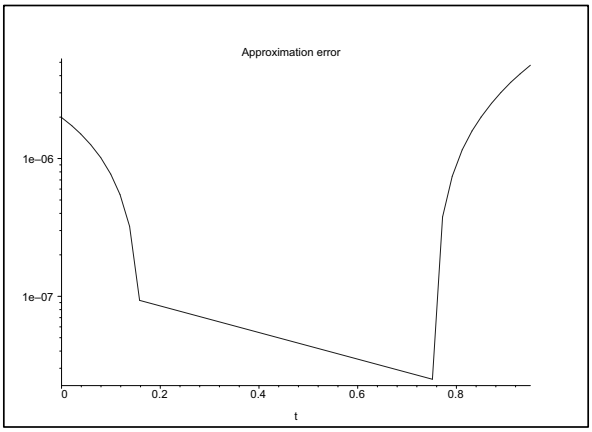


Figure 2. Error plot, nonoscillating problem, B-spline method, $n = 10$**,** $k = 3$

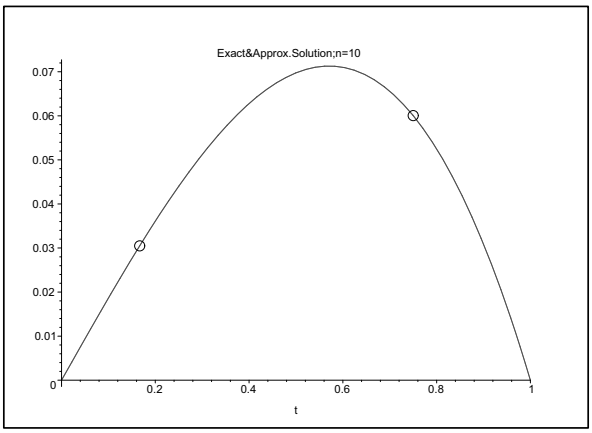


Figure 3. Exact and approximate solution, nonoscillating problem,Chebyshev method,$n = 10$

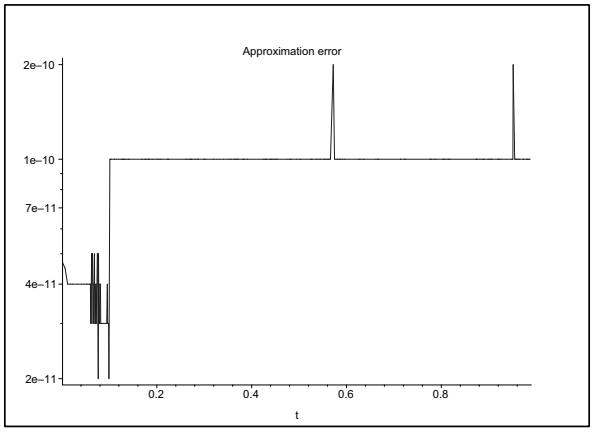


Figure 4. Error plot, nonoscillating problem, Chebyshev method, $n = 10$

Here are the profiles for the procedures in the case of nonoscillating problem. The function `showprofile` for the B-spline method gives the following results:

```
function            depth    calls     time    time%         bytes   bytes%
-------------------------------------------------------------------------
genspline               1        1    2.496   100.00      92768440   100.00
-------------------------------------------------------------------------
total:                  1        1    2.496   100.00      92768440   100.00
```

The profile for Chebyshev method is:

```
function            depth    calls     time    time%         bytes   bytes%
-------------------------------------------------------------------------
genceb                  1        1    0.249   100.00       9291004   100.00
-------------------------------------------------------------------------
total:                  1        1    0.249   100.00       9291004   100.00
```

The second example has an oscillating solution:

$$\begin{aligned} -y'' - 243y &= x, \qquad x \in [0,1] \\ y\left(\frac{1}{6}\right) &= -\frac{1}{1458}\frac{-6\sin\frac{3}{2}\sqrt{3} + \sin 9\sqrt{3}}{\sin 9\sqrt{3}} \\ y\left(\frac{3}{4}\right) &= -\frac{1}{972}\frac{-4\sin\frac{27}{4}\sqrt{3} + 3\sin 9\sqrt{3}}{\sin 9\sqrt{3}}. \end{aligned}$$

The exact solution, provided by `dsolve` is $Z(t) = -\frac{1}{243}\frac{-\sin 9\sqrt{3}t + t\sin 9\sqrt{3}}{\sin 9\sqrt{3}}$. We chose $n = 100$ for both methods and $k = 3$ for the first method. Figure 5 gives the graph of exact and approximate solution for the oscillating problem. The error plot appear in Figure 6. The corresponding graphs for Chebyshev methods are given in Figures 7 and 8, respectively.

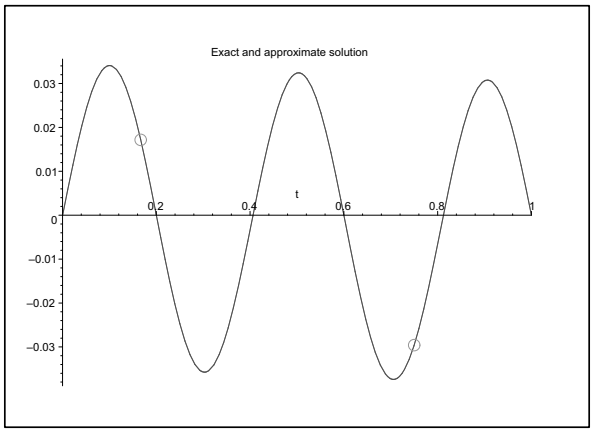


Figure 5. Exact and approximate solution, oscillating problem, B-spline method, $n = 100$**,** $k = 3$

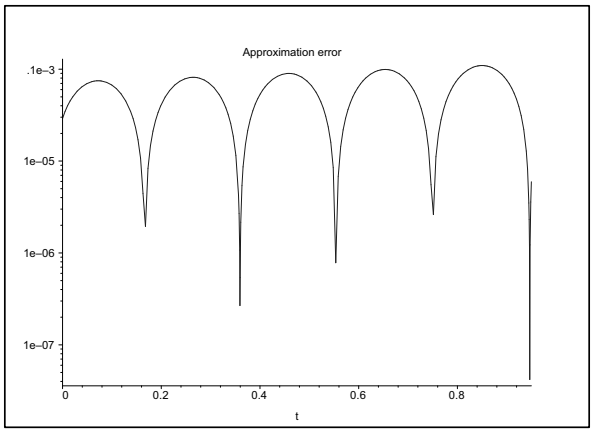


Figure 6. Error plot, oscillating problem, B-spline method, $n = 100$**,** $k = 3$

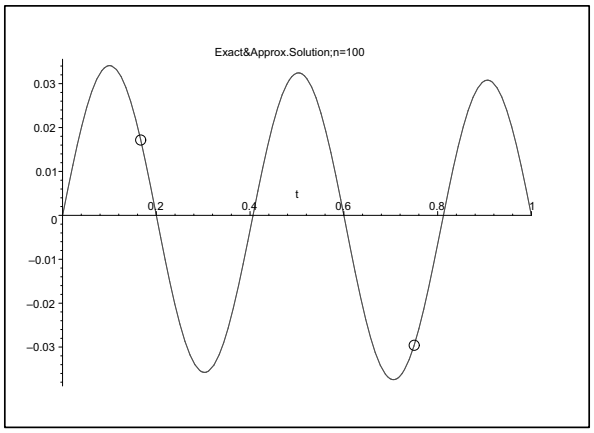


Figure 7. Exact and approximate solution, oscillating problem, Chebyshev method, $n = 100$

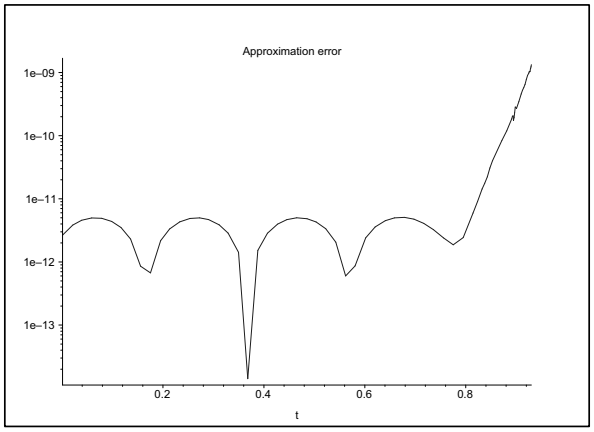


Figure 8. Error plot, oscillating problem, Chebyshev method, $n = 100$

Here are the profiles for the procedures in the case of oscillating problem.

```
function            depth    calls     time    time%         bytes   bytes%
-------------------------------------------------------------------------
genspline               1        1   36.691   100.00     990061444   100.00
-------------------------------------------------------------------------
total:                  1        1   36.691   100.00     990061444   100.00

function            depth    calls     time    time%         bytes   bytes%
-------------------------------------------------------------------------
genceb                  1        1  174.814   100.00    4365866236   100.00
-------------------------------------------------------------------------
total:                  1        1  174.814   100.00    4365866236   100.00
```

5 Conclusions

The Chebyshev method has a smaller error (see error plots, Figures 2, 4, 6, 8). For the nonoscillating solution and a mesh with a small number of subintervals Chebyshev method is faster and requires less memory. If the number of points increases the B-spline method is faster and requires less memory. The reason is that for the B-spline method the matrix of the system that provides the coefficients is a band matrix with at most 4 nonzero elements per line, while for Chebyshev method the matrix is dense. The example with oscillating solution supports this conclusion.

Our approach based on computer algebra has the following advantages:

- The choice of mesh points is arbitrary.
- The degree of Legendre polynomial can be changed.
- We need not bother with differentiation, equation building, ordering and so on.
- The analytic form of the solution allow to compute the approximation at any point, to plot it and to use it further as input for other problems.

References

[1] U.Ascher,J.Christiansen and R. Russel , A collocation Solver for Mixed Order Systems Of Boundary Value Problems ,*Mathematics of computations.*33 nr:146,pp 659-679,1979

[2] U.Ascher,S.Pruess and R.D. Russel , On spline basis selection for solving differential equations, *SIAM,J.Numer.Anal.*20,no.1,pp 121-142, 1983

[3] L.R. Burden,J.D. Faires , *Numerical analyses*, PWS-Kent Publishing Company,New York,1985.
[4] C.deBoor, Package for calculating with B-splines, *SIAM J.Numer.Anal.* 14(1977)no.3,pp.441-472,1977.
[5] C.deBoor, *A practical guide to splines,* Springer-Verlag,Berlin,Heidelberg,NewYork,1978.
[6] W.Gautschi, *Numerical Analysis,An Introduction*,Birkhauser,Basel,1997.
[7] A.Heck, *Introduction to Maple*,Springer-Verlag ,Berlin,Heidelberg,1997.
[8] Al. Lupas, *Numerical Methods*, Constant,Sibiu,2000,(in Romanian).
[9] D. Ripeanu,O.Arama, Qulque recherche actuelles consernant l'equtions de Ch.de la Valee Poussin relative au problem polylocal dans la theorie des equations differentials, *Studia Mathematica* 8(31), pp 19-28,1966.
[10] M.H Schultz , *Spline Analysis,* Prentice Hall ,Inc.Englewood Clifs,New Jersey,1972

Daniel N.Pop
Romanian-German University Sibiu
Faculty of Economics and Computers
Calea Dumbravii nr: 28-32
Romania
E-mail: *danielnicolaepop@yahoo.com*

Radu T. Trîmbiţaş
Babe ş-Bolyai University,Cluj-Napoca
Department of Applied Mathematics
Mihail Kogalniceanu street nr:1
Romania
E-mail: *tradu@math.ubbcluj.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Invisible Web Search: Case Study Deep Web Search Tools

IOAN POP

Abstract

Some users have the sensation that they can found anything on the World Wide Web by using common search engines. This approach is not correct, as powerful as these search engines are, they do not index everything on the World Wide Web. This is the massive content that is publicly available, but hidden from regular search engines. The spiders or crawlers do not have the capability to index a big part of the web documents. In order to find the information from the web document repository it is necessary the special techniques and resources. In this paper we propose some approaches to mining the invisible web through: some strategies to access and mine the invisible web; many web search tools; techniques and resources to search the deep web; webometrics for the deep web.

1 Introduction

The term "Invisible Web" (or "Deep Web" or "Cloaked Web") is recently accepted by the people that research the web. Invisible web mainly addresses to the growth repository of the documents that you can't access with search engines and directories. To be specific: the Invisible Web is comprised of hundreds billion web pages that are not stored as static web pages. Instead, the Invisible Web is made of on-demand database content pages which exist only as reports of changing data. Today, robot crawlers are not advanced enough to read these private databases. Only a human reader can see these "invisible pages" by directly visiting these sites and making direct database requests [7].

To dig deeper into the Web, a new breed of search engine has cropped up that takes a different approach to Web page retrieval. Instead of broadly scanning the Web by indexing pages from any links they can find, these search engines are devoted to drilling further into specialty areas i.e.: medical sites, legal documents, even Web pages dedicated to jokes and parody. A few search engines have tried to take that step, with mixed results. But again, many Web users do not know that the narrow searching tools exist. So reference librarians and library Web sites are now directing their patrons to those areas on the Web.

BrightPlanet is the leader in harvesting high quality content from inaccessible Deep Web and Surface Web sources. With over 10 years of Deep Web extraction expertise, the company has developed a heuristic, rule-based expert system for communicating with Deep Web sources that does not require one-off scripts to be built by hand, which are often prone to failure. The fully automatic configuration system configures about 80% of known Deep Web sources without any user intervention. Another 15% of Deep Web sources can be configured with only minor user intervention, requiring only about 5 seconds per source. All remaining Deep Web sources can be configured using an extensive scripting language that also supports password protected and JavaScript sites [7].

A recent study, BrightPlanet's white paper indicates that the invisible web is 400-550 times larger than the traditional (surface or open) web. They estimate that there are more than 7,500 terabytes of information contained in the invisible web, compared to the 19 terabytes of the surface web. Most of these sites are accessible to the public, free of charge [2].
Major components of the invisible web include: non-HTML files (i.e.: PDF files, Flash files, etc.), sites requiring registration of login, archives (magazines, newspapers, etc), interactive tools (calculators, translators, etc.).

2 Strategies to access and mine the invisible web

The researchers often need more than Google and Wikipedia to get the job done. To find what you're looking for, it may be necessary to tap into the invisible web, the sites that don't get indexed by broad search engines. The following resources were designed to help you do just that, offering specialized search engines, directories, and more places to find the complex and obscure.

2.1 A stratified view of the web

An illustration of the Web "*Content Layers*" is presented in the figure 1.

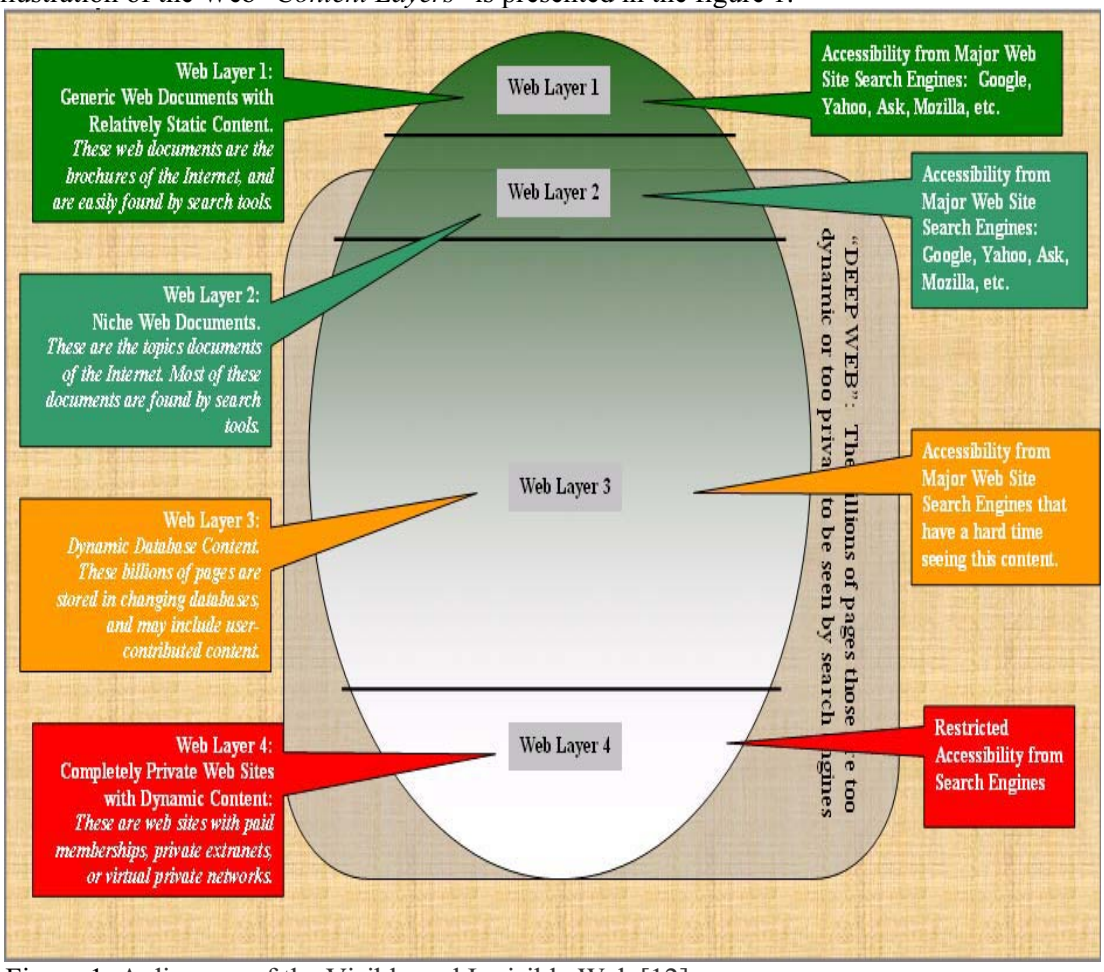


Figure 1. A diagram of the Visible and Invisible Web [12].

There are two pole tips of the Web: **a**. the surface web (layer 1 and layer 2) that is visible by the commonly search engines; **b**. the invisible web (layer 3 and layer 4) that is not visible by the commonly search techniques. The content of the deep web can only be found by sending a direct query to a database. Traditional search engines are not able to detect the contents of these databases as their crawlers are collecting information for their catalogs. However, more and more traditional search engines are adding deep web searching capabilities to their sites.

2.2 Deep Web Search Strategies

Currently there are two ways the deep web can be searched: on the one hand we want instant access to information throught a specialized deep web search tools, and on the other hand throught a specialty search engine passed on to you.
First, let's fin some deep web search engines. Some are easy to find in the surface web by going to a search engine and searching "Deep Web search engines". From time to time you will pick up a good deep web site by "word of month", at a conference or in a deep web class such as this one. These sites may or may not show up in a normal deep web search because of their specialization or newness, or they may be password protected.
For example, the steeps that can be followed in the try to mine the deep web are the following: **a**. being aware that the deep web exists; **b**. using a general search engine for broad topic searching; **c**. using a searchable database for focused searches; **d**. register on special sites and use their archives; **e**. call the reference desk at a local college if you need a proprietary web site; **f**. many college libraries subscribe to these services and provide free on-site searching (and a friendly trained librarian to help you); **g**. check the web site of your local public library; **h**. many libraries offer free remote online access to commercial and research databases for anyone with a library card.

3 Techniques for Search the Deep Web

The web sites that usual search engines can't find is the following types of: sites with dynamic scripting, private web pages, sites that require a registration, temporary web pages, blocked sites by local webmasters, sites blocked by search engine policy, sites with special formats, searchable databases [18].

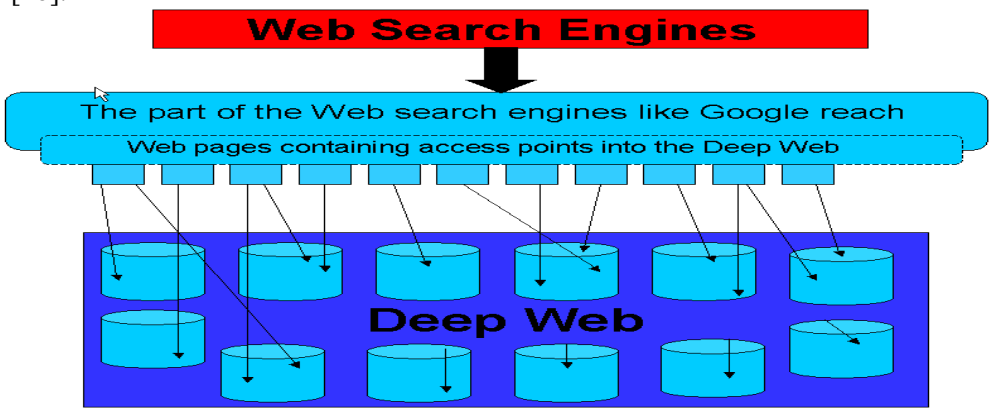


Figure 2. A diagram of the Deep Web Search Engines[14].

3.1 Searching the Invisible Web

In order to search the invisible web the search engines vendor now make tools available. Rather than retrieving web pages or documents, invisible web search engines direct the user to an appropriate searchable database. Some even generates a search form for your convenience. Searching the invisible web is processes in two-steps. According [19] these two steps are:
First, locate the appropriate database to search. Browsing is the easiest method. Invisible web search engines are accompanied by a categorized list or subject directory. Browse the appropriate category to make your selection. Or, you can use keywords, including natural language, to search for a database using the search engine. Searches should be kept simple.
Second, search the database you just located. Review the HELP screens for tips on improving the effectiveness of your search query.
According the same reference, [19], a selective list of Invisible Web Search Tools is:
Complete Planet (http://www.completeplanet.com/index.asp)
Direct Search (http://www.freepint.com/gary/direct.htm)
ProFusion (http://www.profusion.com)

FirstGov (http://www.firstgov.gov) -- access to federal government databases
Digital Librarian - http://www.digital-librarian.com
Infomine - http://infomine.ucr.edu/ - a virtual library of Internet resources relevant to faculty, students, and research staff at the university level.
MagPortal - http://magportal.com - Find individual articles from many freely accessible magazines.

3.2 Multiple Resources to Mine the Deep Web

The various deep web search resources can be classified such as: search engines, databases, catalogs, directories, social media and more, and guides.

Search engines
Whether you're looking for specific science research or business data, these search engines will point you in the right direction.

Databases
Tap into these databases to access government information, business data, demographics, and beyond.

Catalogs
If you're looking for something specific, but just don't know where to find it, these catalogs will offer some assistance.

Directories
Get hand-picked links to high quality research sources with these directories.

Social Media and More
Social media sites are a great way to find content that's obscure or hasn't quite made it to the search engines yet. Use these tools and more to round out your arsenal.

Guides
Use these guides to learn how to fine-tune your search on the invisible web. For example: “The Deep Web” **a** guide quickly discusses the deep web and offers a few tips for finding deep web information.

4 Case Study: Deep Web Search Tools

4.1 Webometrics for the Deep Web

There are many metrics for asses deep web searching. The main metrics proposed are from the following area [4]:

- PageRank
- Graph theory
- Network mapping
- Search engine optimization
- Impact factor

For example, one relatively straightforward measure is the "Web Impact Factor" (WIF) introduced by Ingwersen [2]. The WIF measure may be defined as the number of web pages in a web site receiving links from other web sites, divided by the number of web pages published in the site that are accessible to the crawler. However the use of WIF has been disregarded due to the mathematical artifacts derived from power law distributions of these variables.

4.2 Deep Web Search Tools

In order to be succesfull while searching the deep Web we provide you, learn how to use the three websites described below:
CompletePlanet™ uses a query based engine to index 70,000+ deep Web databases and surface Web sites. Appendix A lists 60 of the largest deep Web databases which contain 10% of the information in the deep Web, or 40 times the content of the entire surface Web. These 60 databases are included in CompletePlanet’s indexes. The interface is intuitive and easy to use. You can do a keyword search on all 70,000+ databases to find which databases to use for your search. You can also browse by category, and then search databases of interest.

ProFusion is a combination of query based engine and a deep Web directory portal. The directory structure is accessed by clicking on Specialized Searches. With an account, you can setup custom "My Search Groups" to search customized lists of websites and/or databases of your choice. For example, you could create a group called Technology and add all the databases and websites of interest to you. This group is saved to your profile. You could then, at any future time, search this group on a research topic with keywords. This is a great time saver. Their query based engine is called SmartDiscovery®.
SurfWax also uses a site's existing search capability as part of the meta-search process to tap the deep Web. They use proprietary algorithms to interpret the site's search criteria (Boolean, etc). With an account, you can also setup custom SearchSets to search customized lists of websites and/or databases of your choice. Surfwax also has a news accumulator feature with over 50,000 news topics in 84 categories. This news accumulator feature is a godsend providing high quality results. These are some useful news accumulator categories: all topics, networking, technology, telecommunication, and web services. In addition this site has WikiWax which takes the online encyclopedia Wikipedia to the next level. WikiWax does advanced look-aheads on Wikipedia searches to speed your keyword choices.

Finding Deep Web Resources

In addition to other methods discussed in this presentation, Schlein [10] shares several techniques to help the researcher find deep Web resources.

Pre-emptive search to find deep Web databases, use a search engine or search a site containing both surface and deep Web content. For example, to find a database containing information on viruses use this search term (exact syntax may vary among search engines):

- On Google or InfoMine search for virus *(database OR repository OR archive)* has this additional method specific for the Teoma search engine
- On Teoma search for: virus *(resources OR meta site OR portal OR pathfinder).*

Reverse-Link Searching: Find out which pages link to a database you already find useful and see if those sites have further recommendations. To do this, use the "link" operator in the search engine. For example, Google uses "link: yourURL." If you want to find out what sites link to NTIS, type this in the Google search bar: **link:http://www.ntis.gov**

Find Experts: When you do a search with Teoma, experts and enthusiasts for your keywords are listed to the right of the results column. Go to these sites and see what resources are recommended to help you "mine" for deep Web resources.

Search by document type: Search engines are now indexing heretofore "deep" files, like PDF files. In Google, by preceding your search terms with "filetype:ext" (where "ext" is the 3 character file extension), only those files will appear in the results. More **about Google:** When you do a search, the results are not only in the window you are viewing, but also simultaneously in the associated windows under the topics listed at the top of the search page, namely, *Web*, *Images*, *Groups*, *News*, *Froogle*, *Local*, etc. For example, if you search for the word "virus," under *Web* are the websites found for virus, under *Images* are the graphics found for virus, under *Groups* are the discussion groups on virus, etc. - all of this is available without you doing anything extra on your part other than click each topic link in succession.

The following tables illustrate some (not all) information situations where search engines are usually appropriate [17]:

Search For...	Choose
General topic	Google, Yahoo, MSN Search
Limited image search	Google, Yahoo, Picsearch, Ithaki, Ditto
Results clustered by subject for refining	Vivisimo, Clusty, Kartoo
Background information on an unfamiliar topic	Google, Yahoo, MSN Search
Find facts, calculate problems, convert formulas	Google, MSN Search
Multiple News Sources (current)	Google, Yahoo, World News Network
Country Information (Statistics, government,	CIA World Factbook, InfoNation

population, comparisons etc)	
Driving Directions, Maps	Google Maps, Mapquest, Mapmachine
Search by domain (.org, .com, .gov, .mil)	Google, Yahoo, MSN Search
Search for pages in specific languages	Google, Yahoo, MSN Search

Table 1. When to Use a usually Search Engine [17].

According to [17], searching into the invisible web is easier to make in the following conditions:

- Familiar with your topic
- Need a specific answer
- Need credible and verifiable information
- Need information contained in databases
- Information dynamically generated or changes frequently
- Topic is narrow
- Real-time information is needed (stock quotes, flight times, and other examples)
- Familiarity with specific research tools

Searching For...	**Choose Invisible Resources**
Is my flight on time?	Flight Arrivals, Yahoo Travel: Check Flight Status
What is the path of a hurricane?	National Hurricane Center
What concerts will be playing in Bucharest next month?	Pollstar, Ticketmaster
How do I find recent magazine articles for free on the web?	FindArticles, MagPortal
I would like to download sheet music of Bach's works.	Choral Public Domain Library

Table 2. When to Use the Invisible Web [17].

As an example of applying some of the principles in this presentation, let's do a search on "web mining" using a surface search engine and a Deep Web Database.

First let's do a surface search on Google. The result is hundreds million hits. I don't really have time to look at hundreds million hits; even a million might take a while. Realistically, I'll look at the first hundred or so and perhaps adjust my keywords then search again. The results show too many vendor sites and only a few dozen sites that might have good information. These results were above average for ten minutes of work. However, I will need to evaluate these few dozen sites and that could take a few hours – maybe I will find something useful among these, maybe not.

Now let's try a deep Web site like Educause, again using the same keywords, "web mining." There are round about 3,000 hits. I look at the first hundred or so and none of these are vendors. There are however many PDF files that look like they contain useful information.

Comparing the quality of the results between the two methods, for this search, the deep Web results have more substance and credibility. Of course this will not always be the case. The surface and deep Web each have their advantages and disadvantages depending on the search topic. You need both aspects of the Web plus a phone to call people (not all information is on the Web). In this example, within three minutes, the deep Web search revealed a goldmine of high-quality information very relevant to the search topic.

4.3 Search Engine Inconveniances

People's access to content on the Web has been greatly enhanced by search engines also, these search engines make a research work succesful and they are an important component of the research procedure. In table 3 are presented a list of the limits for work with search engines [19].

Inconveniences	**Details**
Search Engine Display Limit	The results displayed by the search engines from a specific site may have a limit. For instance, only one or

	two results, may be displayed in the search results for each site Google indexes. But, unless you click on more results other relevant pages from that site will not be displayed even though they might be indexed.
Quality	It is not considered when sites whose content might not be accurately reflected by their name will appear.
Quantity	SE returns tens of thousands of results with no way of separating the quality sites from the useless ones.
Search Engine Preferences	Depending on user preferences, a search engine might be able to find available documents but the user preferences are screening it out.
Surface Indexing	A search engine might index the home page and maybe a page or two after that from a large site. It will not return all the content from a quality web site with many levels of information.
Business & Popularity	Most search engines rank pages according to popularity. Google, in particular, returns results based on the most popular and best-known sites for your topic (how many pages linking to a site are a big part of Google's results). Business organizations pay big bucks to have their sites appear near the top of a search. Many search engines target consumers looking to buy products. Information seekers have become secondary targets for some search engine companies.

Table 3. Search engine inconveniences.

Conclusion

The search tools for exploring the Invisible Web need the new valuable resources because the usual search engines can't be available to enlighten searchers. With enhanced resources for the hidden contents of the deep web documents the search process is more efficient and productive.
We can get benefits beyond just academic, by knowing the Invisible Web. Through the Invisible Web we can gain information concerning medicine, law, and other areas. Also, people can become informed consumers. One of the benefits of mining to use the invisible web resources in conjunction with search engines is how to be able to discern viable information from information targeting consumers.
In our case study we emphases multiple resources; which is, for now, an accepted solution for exploring the Invisible Web. Our contributions from this case study and the trials we made by using the "Federated Search" techniques in searching the Invisible Web further on, encourage us to improve new tools for mining the Deep Web without using up too many resources.

References

[1] S. Brin, L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, http://infolab.stanford.edu/~backrub/google.html
[2] P. Ingwersen, *The calculation of web impact factors*, Journal of Documentation, Vol. 54, No. 2, March 1998.
[3] E. T. Peterson, *Website Measurement Hacks,* Published by O'Reilly & Associates, 2005.
[4] I. Pop, *Web Metrics using Web Quality Model in Social Media*, New Trends in Approximation, Optimization and Classification, Proceedings of International Workshop, Sibiu, 2008, ISBN: 978-973-739-678-5 pp. 58-67, 2008.

[5] V. Shkapenyuk, T. Suel, *Design and Implementation of a High-Performance Distributed Web Crawler*, Technical Report TR-CIS-2001-03, CIS Department Polytechnic University Brooklyn, NY 11201, 2001.
[6] J. Sterne, *Web Metrics: Proven Methods for Measuring Website Success*, Published by John Wiley & Sons. Inc, 2002.
[7] P. Tin, T. T. Zin, H. Hama, *A Novel Interdisciplinary Approach to Deep Web Search Engine*, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.11, November 2008
[8] W. Warnick, *Searching the Deep Web*, D-Lib Magazine January 2001, Vol. 7 N# 1, 2001.
[9] http://www.philb.com/webse.htm
[10] http://techdeepweb.com/4.html
[11] http://www.webometrics.info/methodology.html
[12] http://www.uaf.edu/Library/instruction/handouts/Invisible_Web.html
[12] http://netforbeginners.about.com/library/diagrams/n4layers.htm
[13] http://en.wikipedia.org/wiki/Deep_Web
[14] http://www.searchengineshowdown.com/
[15] http://searchenginewatch.com/.
[16] http://www.websearchguide.ca/research/
[17] http://eastlrc.valencia.cc.fl.us/invisibleweb/comparing_search_engines_to_the.htm/
[18] http://www.webdesignerforum.co.uk/index.php?autocom=blog&blogid=85&showentry=153/
[19] http://www.valenciacc.edu/library/east/invisible_reasons.cfm

IOAN POP
"Lucian Blaga" University - Sibiu
Informatics Department
Dr. Ioan Rațiu street, nr. 5-7
ROMANIA
E-mail: me.ioanpop@gmail.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

On Particular Class of Location-Transition Petri Nets: State Machine

Marin Popa, Mariana Popa, Mihaita Dragan

Abstract

This article discusses important properties of a particular class of Petri nets, class state machine. This class allows finit automate and sequential processes modeling. A deficiency of this class is not allows synchronization of independent processes. Obtain here some theoretical results in a bimarked state machine and shows that any program can be associated in a Petri net's of this class. Some properties of such programs for their accuracy can be studied using results of the class state machine.

Keywords: state machine monomarked, state machine bimarked, viability, surety, S-conflict matrix, S- confluence matrix, T-conflict matrix, T-confluence matrix.

This particular class of Petri Net's the automats modeling and finite sequential processes. Operation of a state machine simulating a multitude of independent processes which unfortunately can not be synchronized in this particular class.

1. PRELIMINARY NOTIONS

In this section we present some concepts and theoretical results of the theory of Petri networks wich are required to demonstrate results that determine the class type state machine of Petri networks.

DEFINITION 1.1 [3] Be $\Sigma = (S, T, pre, post)$ a Petri net and $\overline{\Pr e}$, $\overline{Post}$, two finite matrix size respectively constructed as follows:

$$\overline{\Pr e}(s,t) = \begin{cases} 1, \Pr e(s,t) \neq 0 \\ 0, \Pr e(s,t) = 0 \end{cases}, \quad \overline{Post}(s,t) = \begin{cases} 1, Post(s,t) \neq 0 \\ 0, Post(s,t) = 0 \end{cases}$$

a) We call the matrix T - symmetric conflict matrix $CT = \overline{\Pr e}^t \bullet \overline{\Pr e}$, of size $|T| \times |T|$.

b) We call S-matrix symmetric conflict matrix $CS = \overline{Post}^t \bullet \overline{Post}$ $CS = \overline{Post}^t \cdot \overline{Post}$, of size $|S| \times |S|$.

c) We call T-matrix confluence symmetric matrix $TC = \overline{Post} \bullet \overline{Post}^t$, of size $|T| \times |T|$.

d) We call S-matrix confluence of symmetric matrix $SC = \overline{\Pr e} \bullet \overline{\Pr e}^t$, of size $|S| \times |S|$.

e) We call the matrix T - precedence symmetric matrix $TP = \overline{Post} \bullet \overline{\Pr e}$, of size $|T| \times |T|$.

f) We call the matrix S - precedence symetric matrix $TP = \overline{\Pr e} \bullet \overline{Post}$, of size $|S| \times |S|$.

PROPOSITION 1.2 [3]

Be a Σ finite Petri net, CT, CS, TC, SC, TP, SP conflict matrix, precedence matrix, respectively confluence matrix defined above and $s,\bar{s} \in S$, $t,\bar{t} \in T$. Then :

a) $CT(t,\bar{t}) = \left|{}^{\bullet}t \cap {}^{\bullet}\bar{t}\right|$, and $CT(t,t) = \left|{}^{\bullet}t\right|$

b) $CS(s,\bar{s}) = \left|{}^{\bullet}s \cap {}^{\bullet}\bar{s}\right|$, and $CS(t,s) = \left|{}^{\bullet}s\right|$

c) $TC(t,\bar{t}) = \left|t^{\bullet} \cap \bar{t}^{\bullet}\right|$, and $TC(t,t) = \left|t^{\bullet}\right|$

d) $SC(s,\bar{s}) = \left|s^{\bullet} \cap \bar{s}^{\bullet}\right|$, and $SC(s,s) = \left|s^{\bullet}\right|$

e), $TP(t,\bar{t}) = \left|{}^{\bullet}t \cap {}^{\bullet}\bar{t}\right|$, and $TP(t,t)$ = number of loops formed by t and a location to some.

f), $SP(s,\bar{s}) = \left|{}^{\bullet}s \cap {}^{\bullet}\bar{s}\right|$, and $SP(s,s)$ = number of loops formed by s and a transition to some.

DEFINITION 1.3 [5]

Be Σ_M = (S, T, Pre, Post) a marked Petri net, and $A(\Sigma_M, \mu_0)$ multitude of accessible marks of the network. It's called the graph of marks the digraph accessible labeled: $G_A(\Sigma) = (A(\Sigma_M, \mu_0), T, \Gamma)$ where for label $\forall \mu \in A(\Sigma_M, \mu_0)$, $\Gamma_\mu = \{ \mu' \in A(\Sigma_M, \mu_0) \mid \exists t \in T$ so that $\mu[t > \mu' \}$, and T is the multitude of labels for the digraph arcs.

The arch μ [> μ 't is labeled with the $t \Leftrightarrow \mu[t > \mu'$. It notes that $G_A(\Sigma)$ can be finite or infinite as so as $A(\Sigma_M, \mu_0)$ is finite or not. Because from μ_0 we can reach to all the markings of the $A(\Sigma_M, \mu_0) \Rightarrow G_A(\Sigma)$ is a related digraph.

DEFINITION 1.4 [5]

Let be $\sum_M=(\sum, \mu_0)$ a Petri net marked and $t \in T$ a transition of the network.

a) We say that transition *t* it is cvasiviable $\Leftrightarrow \exists \mu \in A(\sum_M)$ so that $\mu[t>$.

b) Petri net $\sum_M$ it is cvasiviable $\Leftrightarrow \forall\, t \in T$, transition *t* it is cvasiviable.

c) We say that transition *t* it is viable $\Leftrightarrow \forall \mu \in A(\sum_M)$, transition *t* it is cvasiviable $(\sum, \mu)$

d) Petri net $\sum_M$ it is viable $\Leftrightarrow \forall\, t \in T$, transition *t* it is viable.

e) Petri net it is viable $\Leftrightarrow \exists \mu \in \mathbf{N}^{|S|}$ so that the network $(\sum, \mu)$ to be viable.

Cvasiable of a transition that gives the opportunity to occur at least once and so that the operation for which is represented in the system modeled is not important in functioning of the system. The viability of a transition expresses that in any moment in the evolution of the transition can occur to a specific mark which is a characteristic of systems with continuous operation and for which a unavailability of a operation corresponds to a feather (error) of the system.

DEFINITION 1.5 [5]

Let be $\sum_M = (S,T,Pre,Post,\mu_0)$ a Petri net marked and $s \in S$ a given location.

a) We say that *s* it is k-bordered ($k \in \mathbf{N}^*$) $\Leftrightarrow \forall \mu \in A(\sum_M)$ we have $\mu(s) <= k$.

For k = 1 we say that *s* it is binary.

We say that *s* is bordered $\Leftrightarrow \exists\, k \in \mathbf{N}^*$ that *s* it is *k*-bordered.

b) $\sum_M$ is bordered $\Leftrightarrow \forall s \in S, \exists\, k \in \mathbf{N}^*$ that *s* it is *k*-bordered.

c) A Petri net $\sum$ is bordered $\Leftrightarrow \forall \mu \in N^{|S|}$, the marked network $\sum_M = (\sum,\mu)$ it is bordered.

d) We say that $\sum$ it is sure if throughout its evolution any location of the network it is binary.

PROPOSITION 1.6 [5]

For any $\sigma \in T^*$ $\exists\, \mu \in N^{|S|}$ marking a network so that $\mu[\sigma\rangle$.

PROPOSITION 1.7 [5]

Let be $\sum_M = (\sum, \mu_0)$ a Petri net marked and $t \in T$ a transition of his life. The transiton *t* is viable $\Leftrightarrow \forall \mu \in A(\sum_M) \exists \mu' \in A(\sum, \mu)$ and $\exists\, \sigma \in T^*$ repetitive sequence containing *t* and such that $\mu'[\sigma>$.

2. THEORETICAL CONSIDERATIONS

DEFINITION 2.1 [1] Let $\Sigma = (S, T, Pre, Post)$ a PT - Petri net {0,1} - value.

a) a) We say that Σ is the state machine $\Leftrightarrow \forall t \in T$ we have $\left|{}^\bullet t\right| = \left|t^\bullet\right| = 1$ (equivalent to $\forall t \in T$, $\sum_{s \in S} Pre(s,t) = 1$ and $\sum_{s \in S} Post(s,t) = 1$).

b) We say that the machine state Σ is monomarked $\Leftrightarrow$ for $\forall \mu_0$ initial marking $\exists! s \in S$ so that $\mu_0(s) = 1$.

If an initial marking μ_0 is allowed a single mark on a particular network location, then the input transitions that have concession at this location μ_0 and may produce one of them. Thus, the mark will move from location to location designating the current state of the machine. If permitted in μ_0 several grades, then by producing the transition grid, these brands moving independently simulating a variety of independent processes on the same program. State machines but do not allow the synchronization of these processes.

OBSERVATION 2.2.

Any scheme logic associated with a program is transformed into a PT-network (0,1)-valuated. Indeed, let P a program (a set of instructions that can be executed in a certain order) and let be SP = (B, Q) a logical schem associated with a logical layout program, when B it is a multitude of blocks for scheme logical , and Q it is a multitude of its arcs.

We associate to location P a PT-net $\sum$ P = (S, T, Pre, Post) as follows:

Each location corresponds to an arch from Q or a lot of arcs with same node terminal , each transition corresponds to a block b from B if it is not test block (block predicative) or transitions k from T correspond to a test block b from B, where k it is aritate for b- and Pre, Post:$S \times T \to \mathbf{N}$ are given by:

$\Pr e(s,t) = \begin{cases} 1, \exists t' \in T, s = (t', t) \\ 0, otherwise \end{cases}$ (t it is final node of s) $Post(s,t) = \begin{cases} 1, \\ 0, otherwise \end{cases}$ t it is intial node of arc s

Obtain such a PT net in wich transitions correspond to the scheme blocks , and the locations correspond arcs of a logical scheme. If $\sum_P$ is a state machine monomarked, then by producing the transitions, mark is moving from location to location and indicating a instruction wich to be executed next .

PROPOSITION 2.3 [1]

Let be $\sum$ a state machine monomarked and hard-related, μ_0 an initial marking, the $G_{\sum}=(S \cup T, \Gamma)$ associated the digraph and his $\sum$ and GA $G_A(\sum)=(A(\sum, \mu), T, \overline{\Gamma})$ graph marking accessible $\sum$'s definition given in 2.2. Then $G_A(\sum)$ and T-labeled graph :

G=(S, T, Γ^2) are isomorphic.

PROOF

Because $\sum$ it is monomark by producing the transitions, the unique mark of the network transitions moving from location to location. How $\left|t^{\bullet}\right|=1$ for $\forall t \in T$ result that $\forall \mu \in A(\sum, \mu_0),(\sum, \mu)$ it is monomark and for $\left|t^{\bullet}\right|=1$, $\forall t \in T$ and $\sum$ it is hard-related result that $\forall t \in T$ can producing at a specific marker (to which the only marked location, is it his entry) and that the number of different $\sum$ marking is equal to the number of locations in S. Thus $\left|A(\sum, \mu_0)\right|=|S|$ and so there is a correlation between S and $A(\sum, \mu_0)$ which associating biunivoc a marker μ_s of a location s and is it given by $\mu_s(s')=\begin{cases}1, s'=s \\ 0, s' \neq s\end{cases}$.

Since $\overline{\Gamma} \mu=\{\overline{\mu} \in A(\sum, \mu_0) \mid \exists t \in T, \mu[t>\overline{\mu}\}$ we have that $\forall \mu, \overline{\mu} \in A(\sum, \mu_0)$ we have $\overline{\mu} \in \overline{\Gamma} \mu \Leftrightarrow \exists s, \bar{s} \in S, \exists t \in T$ that $\mu=\mu_s$ $\overline{\mu}=\mu_{\bar{s}}$ şi $\mu_s[t>\overline{\mu}_{\bar{s}} \Leftrightarrow \exists s, \bar{s}$ şi $\exists t \in T$ that $\Gamma(s)=t$ and $\Gamma t=\bar{s} \Leftrightarrow \exists s, \bar{s} \in S$ şi $\exists t \in T$ that $\bar{s}=\Gamma t=\Gamma(\Gamma s)=\Gamma^2 s \Leftrightarrow \exists s, \bar{s} \in S$ that $\bar{s} \in \Gamma^2 s$. It follows that $\forall(\mu, \overline{\mu})$ arch in $G_A(\sum)$, $\exists(s, \bar{s})$ arch in G corresponding from bijective above him $(\mu, \overline{\mu})$.

Conversely, it is obvious because $(s, \bar{s})$ the arch in G, take the $(\mu_s, \mu_{\bar{s}})$ arch in $G_A(\sum)$.

COROLLARY 2.4

Let be $\sum$ a state machine monomarked with n = | S | location and marking μ some of it. Then the number of the accessible markings is finished, and in addition $\left|A(\sum, \mu_0)\right| \leq n$.

PROOF.

Obviously by moving a marking unique from location to location we obtained new markings of the network, in which one location is marked. So each marker has a single element 1 and 0 the rest, and this element can occupy at most n-positions, can be obtained at most n-different markings. The maximum number is reached when the state machine it is hard-connex, because of the isomorphism established in the previous theorem.

PROPOSITION 2.5.

Let be $\sum$ a machine state monomarked and hard-connex. Then $\sum$ is it a PT-net viable and sure.

PROOF.

Let be $\forall \mu_0$ a initial marker of a $\sum$. We will show that $(\sum, \mu_0)$ it is viable and sure. Since $\sum$ it is hard-connex $\Rightarrow$ the graph G=(S, T, Γ^2) it is hard-connex and according to previous 2.5, the graph of the marks accessible $G_A(\sum)$ it is hard-connex, which means $\forall \mu, \overline{\mu} \in A(\sum, \mu_0), \exists \sigma \in T^*$ that $\mu[\sigma > \overline{\mu}$.

It follows that after a certain sequence of procedures is it possible re-obtaining any marking $\mu \in A(\sum, \mu_0)$. How in a state machine $\forall t \in T$ is permitted at a marker μ (ie one that assigns 1 only its entry) and how μ to obtain the sequence σ what containing t, so a repetitive sequence, resulting in 1.7 sentence that it is viable. $\sum$ therefore follows that is it viable.

Fact that $\sum$ it is sure resulting from definition of a $\sum$ namely from $\sum_{s \in S} Post(s,t) = 1$ we obtain that $\exists! s \in S$ so that Post(s, t)=1. In other words, by producing t marking unique of the network is transformed from entry of *t* of exit its *s* ie and $\forall \mu \in A(\sum, \mu_0)$ şi $\forall s \in S$ we have $\mu(s) \leq 1$.

THE EXAMPLE 2.6.

$\sum$ Petri net of figure 1 is a state machine with initial marking $\mu_0 = (100)^t$.

It is noted that: $\mu_0[a > \mu_1 = (010)^t$
$\mu_1[b > \mu_2 = (001)^t$
$\mu_2[c > \mu_1, \mu_2[d > \mu_0$

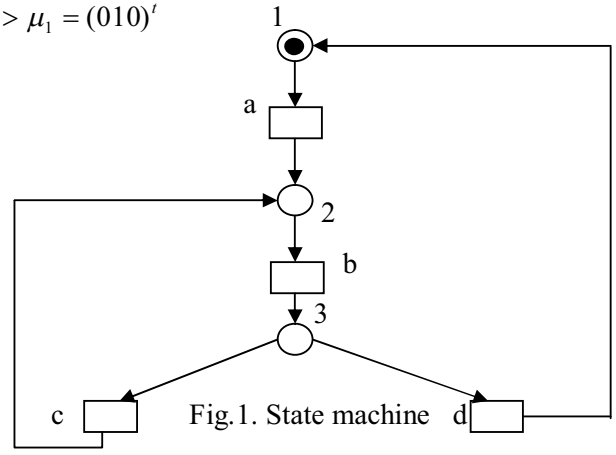
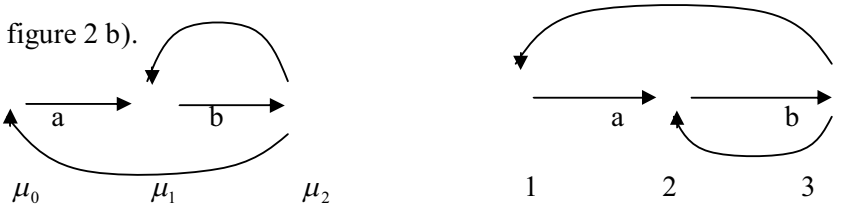


Fig.1. State machine

So : $A(\Sigma, \mu_0) = (\mu_0, \mu_1, \mu_2)$

The graph's accessible markings it is in the figure 2 a) and graph G=(A, T, Γ^2) it is in figure 2 b).



a) the graph's accessible markings b) graph G=(A, T, Γ^2)

Fig. 2. The isomorphic graphs

Obviously the two graphs are isomorphic and $\sum$ is viable and sure.

Follows a characterization theorem of the machine state by a matrix T-conflict and T-confluence defined in 1.1.

THEOREM 2.7 Let be $\sum$= (S, T, Pre, Post) a PT-Petri network (0, 1)-value and CS, SC, CT, TC matricx S-conflict, S-confuence, T-conflict, T-confluence defined in 1.1. Then:

a) $\sum$ is it a machine state $\Leftrightarrow$ CT(t, t)=1,TC(t, t)=1 for $\forall t \in T$ for (have all elements of the diagonally 1)

b) If $\sum$ it is the state machine then CS and SC are matrix diagonal.

c) If CS and SC are diagonal matrix then $\sum$ it is state machine $\exists t \in T$ that $t^\bullet = \Phi$ or ${}^\bullet t = \Phi$.

PROOF.

a) Using proposition 1.2 a) and c) we have CT(t, t)=$\left|{}^\bullet t\right|$ and TC(t, t)=$\left|t^\bullet\right|$ for $\forall t \in T$. From definition we have $\sum$ state machine $\Leftrightarrow \left|{}^\bullet t\right|=\left|t^\bullet\right|=1 \Leftrightarrow$ TC(t, t)=CT(t, t)=1 for $\forall t \in T$.

b) If $\sum$ it is a state machine $\Rightarrow \forall t \in T$ has one entrance and one exit and so t can not be many entries in two different locations would be any different because $t \in T$ that will be $\left|t^\bullet\right|$=2>1. It follows therefore that $\forall s, \bar{s} \in S, s \neq \bar{s}$ we have ${}^\bullet s \cap {}^\bullet \bar{s} = \Phi$ which means that CS(s, $\bar{s}$)=$|{}^\bullet s \cap {}^\bullet \bar{s}|$=0.

Analog $\forall t \in T$ output can not be common to two different locations that otherwise would have

that $\left|t^\bullet\right|$=2>1 and hence $\forall s, \bar{s} \in S, s \neq \bar{s}$ we have $s^\bullet \cap \bar{s}^\bullet = \Phi$. Thus we deduce from 1.2. item d) SC(s, $\bar{s}$)=$\left|s^\bullet \cap \bar{s}^\bullet\right| = 0$, $\forall s, \bar{s} \in S, s \neq \bar{s}$.

c) If CS and SC are diagonal we have $\Rightarrow \forall s, \bar{s} \in S, s \neq \bar{s}$ avem CS(s, $\bar{s}$)=SC(s, $\bar{s}$) = 0 and hence the 1.2. gain ${}^\bullet s \cap {}^\bullet \bar{s} = \Phi$ and ${}^\bullet s \cap \bar{s}^\bullet = \Phi$. It follows that $\forall t \in T$, ${}^\bullet t$ şi $t^\bullet$, and can contain no more than one location, ie $\left|{}^\bullet t\right| \leq 1$ and $\left|t^\bullet\right| \leq 1$. This means that time $\sum$ it is a state machine or that $\exists t \in T$ that t has no input and / or any output.

COROLLARY 2.8.

Let be $\sum$ a PT- net {0, 1}-value and hard-connex. Then $\sum$ it is the state machine $\Leftrightarrow$ SC and CS are diagonals matrix.

PROOF.

Because $\sum$ it is hard-connex, resutls that $\forall t \in T$ we have $t^{\bullet} \neq \Phi$ and ${}^{\bullet}t \neq \Phi$ and then point c) of the theorem says that SC and CS are diagonals $\Rightarrow \sum$ it is machine state. The reciprocal is even point b) of the theorem

Follows an interesting result by fact that it specifies an upper edge for the number of the accessible markings in a machine state bimarked.

DEFINITION 2.9

Let be $\sum$= (S, T, Pre, Post) a state machine. We say that $\sum$ it is bimarked $\Leftrightarrow \forall \mu_0$ initial marking of the network, μ_0 puts on network locations only two marks .

PROPOSITION 2.10

Let be $\sum$= (S, T, Pre, Post) a bimarked state machine, with n locations and μ_0 is it marking initial.

Then the lot of markings accessible is finished and in addition $\left|A(\sum, \mu_0)\right| \leq \frac{n(n+1)}{2}$.

PROOF.

The two markings which they puts on the network locations could both be attributed to a location one or one to two different locations. In the first case, the transitions from output of that location have concession at μ_0 and producing one of them, one of a marking is moved to a new location. In this situation by producing one or another of the transitions located at the exit of the two locations, are obtained new markings that does not change the number of locations marked and this because $\forall t \in T$ we have $\left|{}^{\bullet}t\right| = 1$ which means that the number of locations is not diminishing and we have $\left|t^{\bullet}\right| = 1$ which says that the number of locations marked increase not. It follows therefore that any marking μ of the network will mark the only two locations and that μ may not have than the elements 0, 1 and 2. Since the number of possible combinations of these values on the n-locations of the network, so their sum be 2 is finite, results that the number of markings accessible is finite .

Obviously the maximum value is obtained when the machine state is hard-connex because otherwise if there is at least a location with no one exit when a marking arrive into it will she not be able to move never from s, and so certain marking can not be obtained (it is as

if a marking can be in a fixed location and the other would cover a part or all of the other locations).

In this case will be n-markings that assigns each locations with two markings, and the others leave them unmarked .

Assume that the locations are: S={s_1, s_2,……, s_n}.

If a mark is puts in s_1 then other mark can occupy any of the other n-1 free locations , and so, in this case we obtain n-1 distinct markings by the firsts. Similarly, if a mark is placed in the s_2 then the other can handle any of the other n-2 locations s_3,....s_n free (do not use and not to s_1 with a marker already obtained in the previous step).

By recursive procedure is apparent that if one marking the deals locationsn s_{n-1} then the other can only deal on sn obtain the marking $(00\ldots011)^t$. So the total number of markings obtained in this case is :

$$n+(n-1)+\ldots..+1=\frac{n(n+1)}{2}.$$

PROPOSITION 2.11

Let be $\sum$ a bimarked state machine and hard-connex. Then $\sum$ it is a PT-net viable and 2-bordered.

PROOF.

We will show that for any marking initial μ_0 of the network, $(\sum, \mu_0)$ is viable and 2-bordered. Since $\sum$ it is hard-connex, results and graph of the marking accessible $G_A(\sum)$ is hard-connex, reduce to the absurdity that if there is a marking from which no longer leaves any arc to another marking would mean that at least one of two locations indicated that the marking has no output arc to a transition of any network, meaning that $\exists s \in S$ that $\forall t \in T$, Pre(s, t)=0. But this means that s and t can not be linked by road and therefore would not be $\sum$ loud related. $G_A(\sum)$ is hard-connex that $\Rightarrow \forall \mu, \overline{\mu} \in A(\sum, \mu_0)$, $\exists \sigma \in T^*$ that $\mu[\sigma > \overline{\mu}$. Because in a state machine $\forall t \in T$ is allowed to μ a bookmarking and how μ to re-obtain after a repetitive sequence σ containing t resulting from 2.1. that it is viable and how it is certain, $(\sum, \mu_0)$ it is viable.

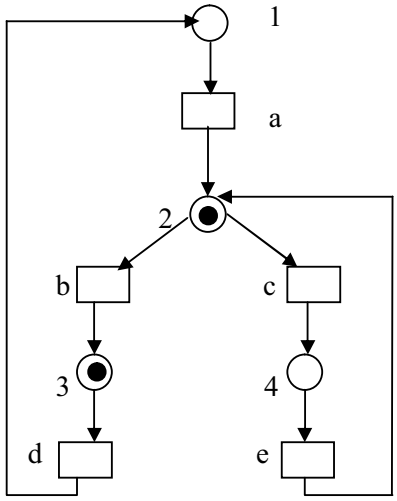
That $\sum$ that is 2-bordered resulted from the proof previous, where was shown earlier that $\forall \mu \in A(\sum, \mu_0)$ şi $\forall s \in S$ we have $\mu(s) \leq 2$.

THE EXAMPLE 2.12

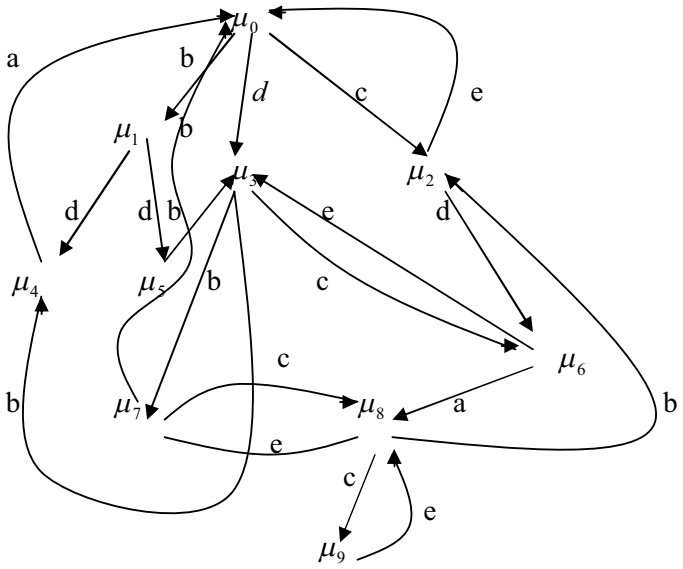
Let the hard-connex state machine from figure 3.6.a) in wich $\mu_0 = (0110)^t$.

It is noted that, $A(\sum, \mu_0) = \{\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7, \mu_8, \mu_9\}$, where $\mu_1(0020)^t, \mu_2 = (0011)^t, \mu_3 = (1100)^t$, $\mu_4 = (1010)^t$, $\mu_5 = (2000)^t$, $\mu_6 = (1001)^t$, $\mu_7 = (0200)^t, \mu_8 = (0101)^t$, and $\mu_9 = (0002)^t$. Graph marking accessible $\sum$'s is given in figure 3.6.b). Since $\sum$ it is hard-connex observed that $G_A(\sum)$ is hard-connex.

Applying the proposition 2.10 we get for n = 4.10 marked $\sum$'s accessible, which is easily seen on the figure. Machine state is viable and 2-bordered as seen on the figure.



a) Bimarked state machine



b) Graph marking accessible

Fig. 3. State machine

In what follows we prove that any program P can be associated state machine. For this we use the results of structured programming [] including the structure theorem of Bohn and Jacopini.

DEFINITION 2.13 [4]

We call the basic structures of structured programming structures: SEQVENCE noted $\pi(a,b)$ (sequential structure), WHILE-DO noted $\Omega(\alpha,a)$ (repetitive structure subject before) and IF-THEN-ELSE noted $\Delta(\alpha;a,b)$ (alternative structure) where a and b are functional blocks (operations on certain basic variables), and α the block is predicative which selects the next operation to be run.

We will show in figure 4 the basic structures of the above.

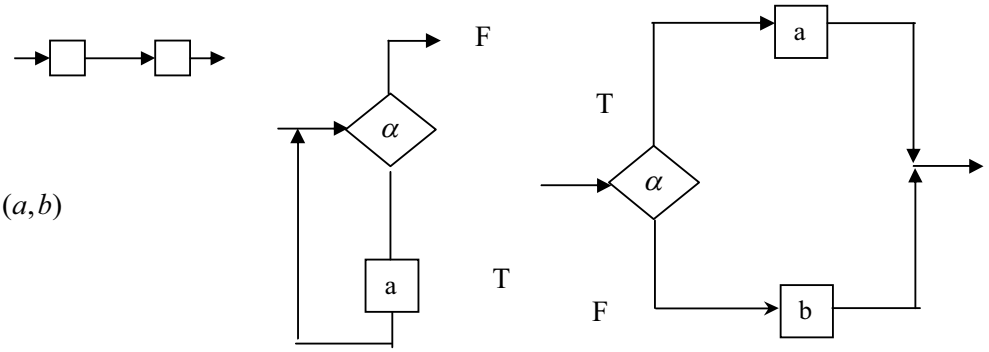


Fig 4 The basic structures of structured programming

DEFINITION 2.14. [2]

Let SP = $(F \cup P, Q)$ schedule an associated program logic P where F is many blocks fonction of the scheme, many blocks predicative P, $F \cap P = \Phi$ and Q many arcs. We introduce three functional blocks F, T, K$\notin F$ and a block predicative $\omega \notin P$ in the following manner. Block F transforms any object x in the pair (0, x), T transforms any object x in the pair (1, x) where 0 and 1 values are indeed true and false respectively associated K block any pair (v, x) with $v \in \{0,1\}$ the of the second component to x. So F (x) = (0, x), T (x) = (1, x),

K(v, x) = x.

Predicative block ω is defined by $\omega(v,x)=1 \Leftrightarrow v=1$ (ω checked $\Leftrightarrow$ first component of the pair (v, x) is 1).

THEOREM 2.15. [2]

Any logical schema SP = $(F \cup P, Q)$ related to a program P can be transformed into a logical schema structured SP '=$(F' \cup P', Q')$ using basic structures defined in 2.13. and where $F' = F \cup \{F, T, K\}, P' = P \cup \{\omega\}, Q = Q' = Q \cup A$, A is a lot of arcs required legării blocks F, T, K, ω with blocks of SP.

PROOF. We found in [2].

THEOREM 2.16.

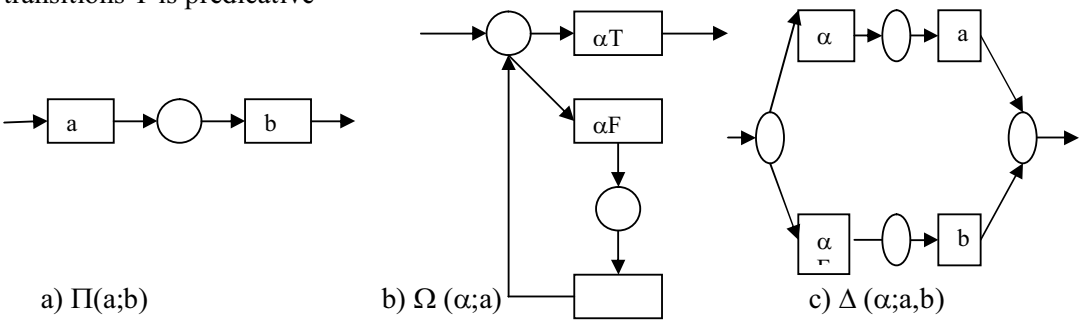
Let P be a program and SP = (B, Q) a logical scheme associated with P..

Whether $\sum$ P = (S, T, Pre, Post) PT-Petri net associated with SP included in note 2.2.

Then $\sum$ P is a state machine.

PROOF. Using theorem 2.15. will be sufficient to prove that the basic structures π, Ω, Δ are associated Petri subrețele are sections of a state machine. Using the construction in remark 2.2. basic structures are associated with sections of Petri networks fig 5 [a), for π, b) for Ω, c) for Δ].

It is noted that a block with two predicative values of truth and is associated if the two transitions T is predicative



a) Π(a;b) b) Ω (α;a) c) Δ (α;a,b)

Fig. 5. Sections Petri networks associated with the basic structures of structured programming true and αF if false. Also note that all three sections of Petri networks are state machines because each transition has one input and one output.

OBSERVATION 2.16.

It is not necessary for structured logic diagram SP' by theorem 2.15. to use all auxiliary blocks placed in the definition 2.14. In fact if the initial schedule does not contain cycles (return to blocks already completed the transition from START to STOP) will not be any need for auxiliary block structure but will be used procedure codes halving [39] which consists of repeating certain blocks on every branch of $\Delta(\alpha;a,b)$ when this is necessary for the structuring and Π and Δ. If the schema contains original ciclări who can not speak and only Π and Δ can use technique of Boolean variable described in [123] leading to the use of auxiliary blocks T, F, K and for structuring ω.

THE EXAMPLE 2.17**.**

Let P be the following program:

Let be determine max (x, y, z) for x, y, z read from the keyboard, and print the result.

A logical schema SP associated with P is given in Fig. 6

Since isn't need for recovery SP it is structured only by Π and Δ.

The network Petri, associated with SP, as we can see in construction from remark 2.2 is given in Fig. 7 and notes on drawing that present a state machine.

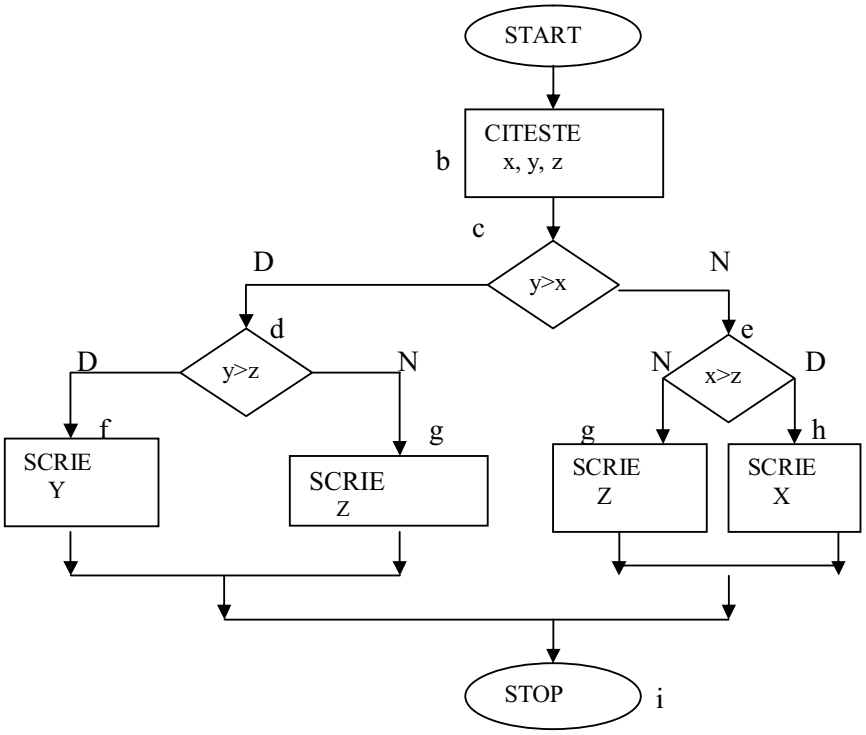


Fig. 6. Logical structural scheme

It is noted that, for the structuring is necessary of g block dedublarea . in the state machine associated with this block is associated not two transition, only one transition.

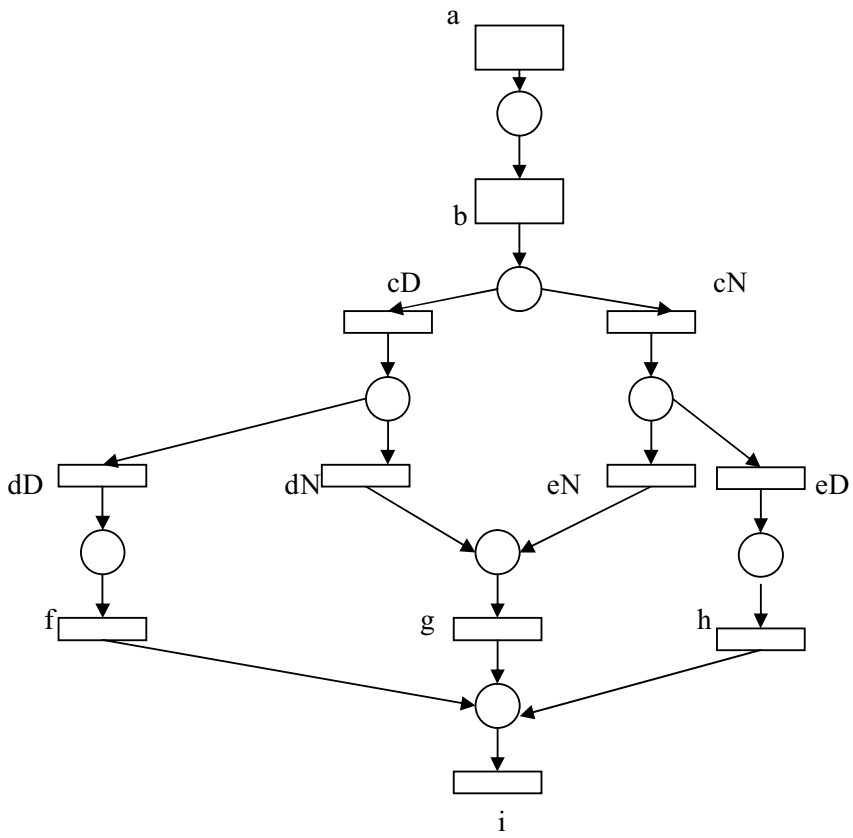


Fig.7. State machine associated to a program

CONCLUSIONS

This article highlights the importance of Petri networks in modeling discrete systems. Even with this class of the Petri net called the state machine class, which is one of the simplest classes, it is noted of this article, that can be obtained results interesting in complex research fields such as automata theory, sequential processes and accuracy study programs.

REFERENCES

[1]. E. Best, A. Merceron-Brecht - *"Some properties of nonsequential processes",* IFS Report 82. 07 (august 1982), Gesellschaft fur mathematik und Datenverarbeitung MBH, Bonn.

[2]. M. Popa I. Vaduva - *"Culegere de probleme de sisteme informatice",* Tipografia Universitatii Bucuresti, 1988.

[3]. J. L. Johnson, T. Murata - *"Structure matrix for Petri nets and their applications*", Journal of Franklin Institute, Vol 319 Nr. 3 march, 1985.

[4]. L. Dumitrascu , A. Ioachim - *"Tehnici de construire a programelor cu structuri alternative*", Editura Academiei, 1981..

[5]. G. Brams - *"Reseaux de Petri: Theory et Practique*", Mason, Paris, 1983

THE AUTHORS

MARIN POPA

UNIVERSITATEA BUCURESTI
DEP. TEHNOLOGII
FAC. : TEHN. INFORMATIEI
BLVD. M.KOGALNICEANU
NR. 36-46, SECTOR 5
COD POSTAL 70709
BUCURESTI, ROMANIA
marpopa2002@yahoo.com

MARIANA POPA

UNIVERSITATEA BUCURESTI
DEP. TEHNOLOGII
FAC. : TEHN. INFORMATIEI
BLVD. M. KOGALNICEANU
NR. 36-46, SECTOR 5
COD POSTAL 70709
BUCURESTI, ROMANIA
marpopa2002@yahoo.com

MIHAITA DRAGAN

UNIVERSITATEA BUCURESTI
DEP. TEHNOLOGII
FAC. : TEHN. INFORMATIEI
BLVD. M. KOGALNICEANU
NR. 36-46, SECTOR 5
COD POSTAL 70709
BUCURESTI, ROMANIA
dragan_mihaita@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Formal and Informal Methods: Different Approaches, Similar Conclusions

Cristina-Elena Popenţa

Abstract

This paper presents a synthesis from a series of articles in which formal and informal methods are used in different case studies from different domains. Formal methods seem to be difficult, expensive, and not widely useful. [2]. A method is formal if it involves mathematical specification, given by a formal language, that ensures precise definition, specification, implementation and correctness. Formal methods are more often used internally within the analysis and for communicating the specification. Informal methods are more often used to communicate with clients and for easier understanding of the formal specifications. The term of informal methods in education is often used as informal learning, and it is serves as a supplement to classical formal methods of teaching.

1 Introduction

From the papers taken into consideration, one presents case studies that indicate that analysts believe that users or clients cannot understand the more formal models such as Object Modelling Technique (OMT) or Unified Modelling Language (UML) diagrams, and that variations on use cases, ad hoc diagrams or rich pictures are more appropriate informal models with which to communicate the specification to users. A research approach is presented in the paper, as well as case studies and findings and interpretation of them, along with a discussion about all the research.

Another study focuses on formal methods, presenting most well-known seven myths about these formal methods. Each of them is debated, with pro and against arguments. As a result, the formal methods can be better understood at large. They are powerful tools, effective and useful for a large variety of applications.

From educational approach, formal and informal learning are considered to be complementary contexts. After common evaluation and research questions, the presented study reaches to conclusion that informal science learning might be well integrated in formal science learning.

Although at first side the it has nothing in common with the other articles, another presented study on a data set about an individual or a group uses two modes of data combination for a predictive or diagnostic purpose. The clinical method relies on human judgment that is based on informal contemplation and, sometimes, discussion with others. The mechanical method involves a formal, algorithmic, objective procedure to reach the decision. Empirical comparisons of the

accuracy of the two methods show that the mechanical method is almost invariably equal to or superior to the clinical method.

All these articles will be summarised below, and in the end we should notify if they lead to similar concussions. Each part of the article was entitled with the title of the article that is support for summarization.

2 "Formal and Informal Methods in Object-Oriented Requirements Engineering"

The summary below synthesised the most important ideas from the article that presents a case study of formal and informal methods in object-oriented requirements engineering.

Various definitions and approaches to the requirements engineering processes are suggested in literature, and in the article. For the beginning, define software engineering as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software.

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirements analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Systematic requirements analysis is also known as requirements engineering. It is sometimes referred to as requirements gathering, requirements capture, or requirements specification.

Requirement engineering is a sub-discipline of systems engineering and software engineering that is concerned with determining the goals, functions, and constraints of hardware and software systems. In some life cycle models, the requirement engineering process begins with a feasibility study activity, which leads to a feasibility report. If the feasibility study suggests that the product should be developed, then requirement analysis can begin. If requirement analysis precedes feasibility studies, which may foster outside the box thinking, then feasibility should be determined before requirements are finalized.

Object-oriented methods for information system development lead to a need for the development of object-oriented approaches to requirements engineering.

In an object oriented modelling processes several models are usually produced, categorised as either static models or dynamic models.

Static models describe objects, their characteristics and the relationships between them, e.g class and object diagrams, component notation and templates, object models, class cards, hierarchies and collaborations, object/class models, object and layer models.

Dynamic models define states of objects, state transition, message passing and event handling, e.g. state transition and event diagrams, state diagrams, object charts, interaction diagrams, object communication models.

The research approach used was multiple sequential-case studies. It involved taped semi-structured interviews with individual practicing professional requirements engineers.

The case studies were opportunistically selected, participants were recruited through industry. Some participants provided contact for subsequent participants. All the participants were currently working in the field of object-oriented requirements specification.

Although the core of the data was gathered from taped and transcribed in depth interviews, several other data sources were used: phone, email, questions in need of clarification which emerge from the transcript process, comments from participants.

One should distinguish between formal models and informal models: formal models are considered to be those models that require training in order to be understood or explained: models that contain specific, graphical notations, such as OMT models, UML models, interaction models or state models; informal models are considered to be models that can be understood and explained without specific training: most common are natural language models including text descriptions, use case scripts, ad hoc diagrams and interactive demonstration models as often produced for prototypes.

The findings from the case study indicated that analysts believe that users or clients find formal models much too complex, both conceptually and technically, to understand and that the use of informal models such as rich pictures, diagrams and use cases, particularly use case scripts which are closer to natural language models, are perceived to be better models for communicating and validating specifications with clients.

Requirements engineering process involves two groups: the users/clients and the professional consultants. The specification needs to be validated as correct from both points of view – the formal or consultant's point of view and the client's informal point of view. For this agreement to take place there needs to be two levels of modelling: informal modelling for communicating the specification to the user for information and validation; and formal modelling for the analyst team to pass on to the design and implementation team.

The implications for practice in these findings lie in the recognition of the social aspects of the requirements specification process. The errors which arise due to inconsistencies, omissions and ambiguities in functional specifications often result in the costly maintenance or failure of software systems. If, as the findings of the research project suggest, the models used for validation of the specification with the clients are different to the models used in design and implementation, then this may indicate that inconsistencies, omissions and ambiguities might arise. The extension of use case models and concrete scenarios may assist in addressing these issues.

In conclusion, understanding the way models and methods are used in practice should lead to

- Improving existing modelling techniques and tools
- Developing new modelling techniques and tools appropriate for a creative and social process
- Developing new modelling techniques and tools appropriate for new environments such as electronic commerce

3. "Seven Myths of Formal Methods"

For most people from user/clients category, formal methods are unfamiliar and difficult to understand. Practical use of formal methods in a software-engineering company lead to conclusion that the myths about formal methods are not true. Seven myths about the use of formal methods were considered to be:

1. Formal methods can guarantee software is perfect.
2. They work by proofing programs are correct.
3. Only high-critical systems benefit from their use
4. They involve complex mathematics
5. They increase the cost of development
6. They are incomprehensible to clients
7. Nobody uses them for real projects

Each of these myths is largely discussed (either approved or partially dissaproved) related to a CASE project that lasted 90 weeks and involved the effort of 450 people's effort, in which formal specifications were applied.

Instead of perpetuating the seven myths, the author proposes seven facts to replace them, accordingly.

1. formal methods are very helpful at finding errors early, and can easily eliminate certain classes of errors.
2. they work largely by making you think very hard about the system you propose to built.
3. they are useful for almost any application
4. they are based on mathematical specifications, which are much easier to understand than programs.
5. they can decrease the cost of development
6. they can help clients understand what they are buying
7. they are being used successfully on practical projects in industry.

4. "Bridging the gap between formal and informal Science learning"

Informal learning main characteristics were summarised and compared to formal learning characteristics as follows:

Formal learning	Informal learning
Compulsory	Voluntary
Structured	Unstructured
Sequenced	Unsequenced
Assessed	Nonassesed
Evaluated	Unevaluated
Close-ended	Open-ended
Teacher-lead	Learner -lead
Teacher-centered	Learner -centered
Classroom –context	Out-of-school context
Curriculum based	Non-curriculum based
Fewer unintended outcomes	Many unintended outcomes
Empirical mesured outcomes	Less directly measurable outcomes
Solitary work	Social intercourse
Teacher directed	Non directed or learner directed

From the summary of the article, we mention some ideas below:

The article begins with a discussion of the importance of motivation and varying institutional techniques in school learning.

Evidence are presented that informal science experience can be effectively used to advance science learning. An important distinction between learning context and learning methods is emphasized. Although connected in the past (e.g. compulsory school with formal learning methods and free choice context with informal learning methods), learning context and learning methods link is artifficial, because a person-s knowledge cannot be limited to what is learned in schools. Instead, learning context and learning methods should be mixted to provide a good learning experience. The integration of informal learning experience within the formal school curriculum could make an important contribution in dealing with the issue of this mixing.

5. "Comparative Efficiency of Informal (Subjective, Impressionistic) and Formal (Mechanical, Algorithmic) Prediction Procedures: The Clinical - Statistical Controversy"

The main idea of the article is summarized in its abstract. As further details about the study in the article are not needed, we should focus on the main idea of the article, which best fits the interest of our article.

Given a data set about an individual or a group (e.g., interviewer ratings, life history or demographic facts, test results, self-descriptions), there are two modes of data combination for a predictive or diagnostic purpose. The clinical method relies on human judgment that is based on informal contemplation and, sometimes, discussion with others (e.g., case conferences). The mechanical method involves a formal, algorithmic, objective procedure (e.g., equation) to reach the decision. Empirical comparisons of the accuracy of the two methods (136 studies over a wide range of predictants) show that the mechanical method is almost invariably equal to or superior to the clinical method: Common antiactuarial arguments are rebutted, possible causes of widespread resistance to the comparative research are offered, and policy implications of the statistical method's superiority are discussed.

4. Conclusions… to conclusions

At first sight all articles mentioned and quoted in this paper seem not to have anything or very few in common. The first common thing that is easily noticed is the presence of words "formal" and "informal" in every one of these articles. The study upon articles led to formality can be enlarged, and can better help to the conclusions of this article.

Anyway, from the diversity of the chosen articles, we can express some of the conclusions as follows:

Formality was always associated with difficult accessibility, difficult understanding, and it is reserved to specialists that can deal with it. Informality can be better understood by majority of common people, but is has its lacks. Although they can be regarded as totally opposite terms, a link between them can be seen in each of the articles. The "inaccessibility" of formal methods can be resolved by finding a way to make them easier to be understood by common users. The informal methods are useful as they offer the first information due to common sense observation. The development of formal models starts with gathering information, or, at least, they are necessary to verify the developed algorithms.

Formalism appeared from the necessity of precision, general application in every context, general support for any user who request it. It's main characteristic can be considered the possibility of adapting it, starting from its generality.

Applications of formal methods in economy, medicine, law, education, etc. are easy to find and become more and more accessible to a larger category of people.

One thing that should be mention, last but not least, is a definition of formalism accessible to all web users. According to wikipedia, formalism is a theory that holds that statements of mathematics and logic can be thought of as statements about the consequences of certain string manipulation rules. Formalism is associated with rigorous method. In common use, a formalism means the out-turn of the effort towards formalisation of a given limited area. In other words, matters can be formally discussed once captured in a formal system, or commonly enough within

something formalisable with claims to be one. Complete formalisation is in the domain of computer science… and the applicability of computer science has no longer any limit in any domain.

References

[1] Filipe, J., Sharp, B. and Miranda, P., *Enterprise Information Systems III – Chapter"Formal and Informal Methods in Object-Oriented Requirements Engineering"*, Kluwer Academic Publisher, Netherlands, 2002.
[2] A Hall , *Seven myths of formal methods*, IEEE software, 1990
[3] A Hofstein, S Rosenfeld, *Bridging the gap between formal and informal science learning*, Studies in Science Education, 28 (1996) 87-112.
[4] William M. Grove and Paul E. Meehl, *Comparative Efficiency Of Informal (Subjective, Impressionistic) And Formal (Mechanical, Algorithmic) Prediction Procedures: The Clinical-Statistical Controversy*, Psychology, Public Policy, and Law, Vol. 2, No. 2,293-323, 1996
[5] www.wikipedia.org

POPENȚA Cristina-Elena
University "Lucian Blaga" Sibiu
Faculty of Science
Department of Informatics
Str. Dr. Ion Ratiu, no. 5-7
ROMANIA
E-mail: cristina.popenta@yahoo.com
E-mail: cristina.popenta@gmail.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Parallel-Distributed Programming Tool for the Linda Programming Model

Ernest Scheiber

Abstract

This note presents an implementation of the Piranha model to program parallel - distributed application and a development tool. We use the TSpaces framework from I.B.M. as a middleware for data exchanges. Our implementation does not introduce a new API but requires certain program structure.

1 Introduction

Even if there is a lack of agreements on what exactly grid computing or cloud computing are, the parallel - distributed programming API's or tools are one of the main ingredient to build such computing environment or application.

Linda is not a programming model but a set of operations that can be added to any language. Linda gives a way to communicate and synchronize different processes. We known the following Linda type development frameworks in Java: *TSpaces* from I.B.M., [6] and *JavaSpaces* from SunMicrosystems.

The *Piranha* model is a parallel - distributed computing model based on Linda, [2]. The paradigm behind Piranha is that of dispatcher-worker. The dispatcher coordinates the activities while the workers solve the received tasks. Between them, the communications are programmed using the Linda operations.

Other known parallel - distributed computing models are MPI, [5, 1], map-reduce.

This note presents an implementation of the Piranha model to program parallel - distributed application and a development tool. We use the TSpaces framework, but instead other messaging service may be used. The actual messaging services (*Sun MessageQueue, apache-activemq* or *qpid*) own the required functionality of a Linda framework. The reason of our development activities consists in the lack of free correspondent software. A preliminary version was reported in [3].

An application to be used with the tool is constrained to a specific structure; it is composed from a dispatcher thread and worker threads. Between the dispatcher thread and the worker threads there are asynchronous message changes. These messages are kept by a TSpaces server until they are consumed. The tool allows us to state the network of workstations involved in the computation, to deploy and to launch an application.

2 The Structure of an Application

We suppose that several computers in a network will perform the required computation. The Java application contains:

- A *Console* class - The Console launches to run the *Dispatcher.*

- On any workstation there is a *Service* class, it instantiates and starts the *Worker*s.
- The application associated TSpace server, it keeps the messages until they are consumed.

The structure of an application is outlined in Fig. 1.

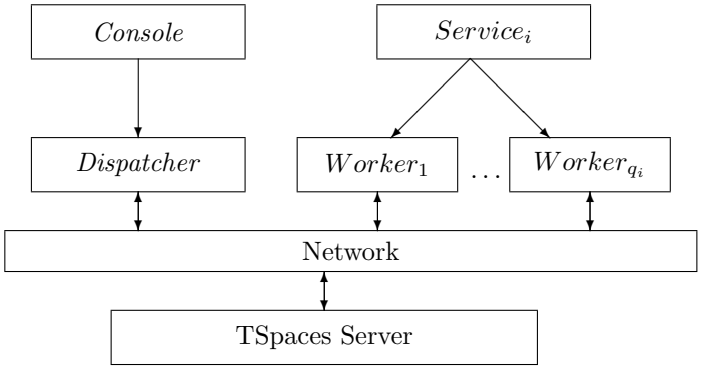


Figure 1: The structure of an application.

The *Dispatcher* and the *Worker* classes are threads. The constructors of the dispatcher and of the worker classes are responsible to establish the connection with the TSpaces server. The `run` methods of these Java threads contain the specific activities to solve the given problem. On a workstation it is possible to start several worker threads. The i index in Fig. 1 denotes a workstation on which q_i worker threads are run. The *Console* and *Service* classes are independent from an application.

Between the dispatcher thread and the worker threads there are asynchronous message exchanges. These messages are kept by a TSpaces server. The data to be exchanged between the dispatcher and the workers are wrapped into objects. To distinguish between different kinds of data, a tag field may be introduced. A message consumer (dispatcher or worker) waits until all the required messages are available. In this way the synchronization problems are solved as well as the coordination of the activities.

3 The anatomy of our implementation

We shall analyze the codes of the above mentioned classes for a very simple application: the dispatcher sends a greeting message to each worker. After receiving the message, a worker responds indicating the name of its computer.

Let us suppose that we use n workstations and that on the $i-$th station there will be launched q_i workers ($\sum_{i=1}^{n} q_i = q$).

Some environment data are fixed in two property files:

propConsole		propService	
Name		Name	
AppName		AppName	
Host		Host	
Port		Port	
TasksNumber	q	TasksNumber	q
		ComputerTaskNumber	q_i
		ComputerFirstTaskIndex	$\sum_{j=1}^{i-1} q_j,\ q_0 = 0$

where *AppName, Host, Port* identifies the name of the application TupleSpace and respectively the host of the *TSpaces* server with the corresponding port number.

As we have mention, the dispatcher is launched by the *Console* class, while the workers are launched by the *Service* class. These two classes are independent of an application.

The *Console* class

```
package ******;
import java.util.Properties;
import java.io.*;

public class Console{
  Properties p=null;

  public Console(String path2Prop) {
    try{
                                        // Loads the properties
      FileInputStream fis=new
        FileInputStream(path2Prop+"propConsole");
      p=new Properties();
      p.load(fis);
    }
    catch(Exception e){
      System.out.println(e.getMessage());
      System.exit(1);
    }
  }

  public static void main(String[] args){
    String fs=System.getProperties().
      getProperty("file.separator");
    String path2Prop;
                    // An argument may contain the path to the property file
    if(args.length>0)
      path2Prop=args[0]+fs;
    else
      path2Prop="";
    Console obj=new Console(path2Prop);
                                       // Set the properties
    String appName=obj.p.getProperty("AppName");
    String host=obj.p.getProperty("Host");
    String sPort=obj.p.getProperty("Port");
    String sTasks=obj.p.getProperty("TasksNumber");









    int port=Integer.parseInt(sPort);
    int tasks=Integer.parseInt(sTasks);


// The dispatcher is launched into execution

    Dispatcher dispacher=new Dispatcher(appName,
      host,port,tasks);
    dispacher.start();




  }
}
```

The *Service* class

```
package ******;
import java.util.Properties;
import java.io.*;

public class Service{
  Properties p=null;

  public Service(String path2Prop)
    try{
                                        // Loads the properties
      FileInputStream fis=new
         FileInputStream(path2Prop+"propConsole");
      p=new Properties();
      p.load(fis);
    }
    catch(Exception e){
      System.out.println(e.getMessage());
      System.exit(1);
    }
  }

  public static void main(String[] args){
    String fs=System.getProperties().
      getProperty("file.separator");
    String path2Prop;
                    // An argument may contain the path to the property file
    if(args.length>0)
      path2Prop=args[0]+fs;
    else
      path2Prop="";
    Service obj=new Service(path2Prop)
                                       // Set the properties
    String appName=obj.p.getProperty("AppName");
    String host=obj.p.getProperty("Host");
    String sPort=obj.p.getProperty("Port");
    String sTasks=obj.p.getProperty("TasksNumber");
    String sIndex=obj.p.getProperty(
      "ComputerFirstTaskIndex");
    String sComputerTasks=obj.p.getProperty(
      "ComputerTaskNumber");
    int port=Integer.parseInt(sPort);
    int tasks=Integer.parseInt(sTasks);
    int index=Integer.parseInt(sIndex);
    int computerTasks=Integer.parseInt(sComputerTasks);
// The required number of workers are
// launched into execution
    Worker[] worker=new Worker[computerTasks];
    for(int i=0;i<computerTasks;i++){
      worker[i]=new Worker(appName,host,
      port,tasks,index+i);
      worker[i].start();
    }
  }
}
```

The wrapper class that contain the data to be exchanged between the dispatcher and workers may be

```
1  package tshello;
2  import java.io.Serializable;

4  public class DataWrapper implements Serializable{
5    // tag=0 message sent by the dispatcher
6    // tag=1 message sent by a worker
7    public int tag;
8    public String mesaj;

10   public DataWrapper(String mesaj, int tag) {
11     this.mesaj=mesaj;
12     this.tag=tag;
13   }
14 }
```

The connection to the TupleSpace server is made in the constructors of the *Worker* and *Dispatcher* classes.

The code of the *Worker* class is

```
package tshello;
import com.ibm.tspaces.*;
import java.net.*;
import java.io.*;

public class Worker extends Thread{
  private int id;
        private TupleSpace ts=null;
        private String tsName;
  private int tasks;
  private PrintStream f;

  public Worker(String tsName,String host,int port,int tasks,int id){
    this.tsName=tsName;
    this.id=id;
    this.tasks=tasks;
    ts=startTupleSpace(tsName,host,port);
    try{
      f=new PrintStream("worker"+id+".txt");
    }
    catch(Exception e){
      System.out.println("File error :"+e.getMessage());
    }
  }

  public void run(){
    Tuple tuple=null;
    DataWrapper d=null;
    String source=tsName+id;
    String dest=tsName+" "+id;
    try{
      Tuple template=new Tuple(source,new FieldPS(DataWrapper.class));
      tuple=(Tuple)ts.waitToTake(template);
      d=(DataWrapper)tuple.getField(1).getValue();
      if(d.tag==0){
        String msg="Mesage received by "+id+" from the dispatcher:\n"+d.mesaj;
        System.out.println(msg);
        f.println(msg);
        InetAddress addr=InetAddress.getLocalHost();
        String s=addr.getHostName();
        String mesOut="Hello from "+id+" at "+s;
        d=new DataWrapper(mesOut,1);
        FieldPS ps=new FieldPS(d);
        ts.write(dest,ps);
        f.close();
      }
    }
    catch(Exception e){
      System.out.println("Exception-Worker : "+e.getMessage());
    }
  }

  private TupleSpace startTupleSpace(String tsName, String host,int port){
    TupleSpace ts=null;
    try{
      Tuple active=TupleSpace.status(host,port);
      if((active==null)||(active.getField(0).getValue().equals("NotRunninng"))){
        System.out.println("TupleSpace Server is not available");
        System.exit(1);
      }
      ts=new TupleSpace(tsName,host,port);
    }
    catch(TupleSpaceException e){
      System.out.println("TupleSpaceException "+e.getMessage());
                        System.exit(1);
    }
                return ts;
  }
}
```

Finally, in the *Dispatcher* class, there are send messages to the workers (the *scatter* method) and are waiting to receive the corresponding responses (the *gather* method):

```
1  package tshello;
2  import com.ibm.tspaces.*;
3  import java.io.*;

5  public class Dispatcher extends Thread{
6    private int tasks;
7    private TupleSpace ts=null;
8    private String tsName;
9    private PrintStream f;

11   public Dispatcher(String tsName,String host,int port,int tasks) {
12     this.tasks=tasks;
13     this.tsName=tsName;
14     ts=startTupleSpace(tsName,host,port);
15     try{
16       f=new PrintStream("dispatcher.txt");
17     }
18     catch(Exception e){
19       System.out.println("File error :"+e.getMessage());
20     }
21   }

23   public void run(){
24     try{
25       scatter();
26       gather();
27       ts.cleanup();
28     }
29     catch(Exception e){
30       System.out.println(e.getMessage());
31     }
32   }

34   private void scatter(){
35     String mesOut="Hello from the dispatcher !";
36     DataWrapper d=new DataWrapper(mesOut,0);
37     try{
38       FieldPS ps=new FieldPS(d);
39       Tuple multi=new Tuple();
40       for(int i=0;i<tasks;i++){
41         String dest=tsName+i;
42         Tuple nextTuple=new Tuple(dest,ps);
43         multi.add(new Field(nextTuple));
44       }
45       TupleID[] ids=ts.multiWrite(multi);
46     }
47     catch(TupleSpaceException e){
48       System.out.println("TupleSpaceException-scatter0 "+e.getMessage());
49     }
50     System.out.println("Scatter OK");
51   }

53   private void gather(){
54     Tuple tuple,template;
55     DataWrapper d=null;
56     try{
57       for(int i=0;i<tasks;i++){
58         String source=tsName+" "+i;
59         template=new Tuple(source,new FieldPS(DataWrapper.class));
60         tuple=(Tuple)ts.waitToTake(template);
61         d=(DataWrapper)tuple.getField(1).getValue();
62         if(d.tag==1){
63           System.out.println(d.mesaj);
64           f.println(d.mesaj);
65         }
66       }
67       f.close();
68     }
69     catch(Exception e){
70       System.out.println("Exception-gather "+e.getMessage());
71     }
72   }

74   private TupleSpace startTupleSpace(String tsName, String host,int port){. . .}
75 }
```

4 An example

The sixteen grid problem (www.ams.org/ams/16-grid.html). Each of the numbers 1,2,...,16 is used exactly once in the empty cells to form arithmetic expressions connected by symbols for the four basic operations. Each row (column) is an arithmetic expression, read and performed left to right (top to bottom), disregarding the usual order of operations, to yield the result at the right (bottom).

	-		×		×		=-60
×	■	+	■	×	■	÷	
	+		+		-		=29
+	■	+	■	-	■	×	
	-		-		-		=-14
+	■	-	■	+	■	-	
	×		-		-		=32
=28		=1		=57		=27	

A solution, presented in [4] is based on a parallelization scheme of the backtracking algorithm. Based on the domain decomposition method, there is a more efficient way to solve this problem. In the set on n order permutations, $\mathcal{P}_n$, considering the lexicographic order the first permutation is $(1, 2, \ldots, n)$ and the last is $(n, n-1, \ldots, 1)$. Relative to this order, it may compute the m-th permutation, $1 \leq m \leq n!$.

The first $q_1 = (n-1)!$ elements of $\mathcal{P}_n$ have 1 on the first position, the next q_1 elements have 2 on the first position and so on. In the group having i on the first position, there are $n-1$ subgroups with $q_2 = (n-2)!$ elements with the second element equals respectively with $1, 2, \ldots, i-1, i+1, \ldots, n$. If the first $s-1$ elements are fixed, $i_1, \ldots, i_{s-1}$ then there exits $n-s+1$ groups with $q_s = (n-s)!$ elements, having on the i-th position an element of $\{1, \ldots, n\} \setminus \{i_1, \ldots, i_{s-1}\}$.

To find the s-th element of the m-th permutation, we compute the group number t_s having the i_s element as constant, $t_s = \lceil \frac{m}{q_s} \rceil$, and finally i_s will be the l_s-th element of the sequence $1, \ldots, n$, after we have deleted $i_1, \ldots, i_{s-1}$, where $l_s \equiv t_s (\text{mod } n-s+1),\ l_s \in \{1, 2, \ldots, n-s+1\}$.

If there are p workers, then the set of all $n!$ permutations are equally divided into p intervals. Each worker generates sequentially the permutations of its attached interval and verifies the restrictions of the problem.

The unique solution of the sixteen grid problem is

1	-	2	×	4	×	15	=-60
×	■	+	■	×	■	÷	
12	+	8	+	14	-	5	=29
+	■	+	■	-	■	×	
13	-	7	-	9	-	11	=-14
+	■	-	■	+	■	-	
3	×	16	-	10	-	6	=32
= 28		= 1		= 57		= 27	

5 The Development Tool

To run the application a lot of activities are to be performed: to deploy the application, to launch the Service program on each workstation and finally to launch the Console on the host station.

To assist these activities, a development tool was created. The *Service* program becomes a servlet and we use the *apache-tomcat* as a servlet container Web server. The *Console* program which starts the dispatcher makes the requests to the *Service* servlets, too.

To the network of workstations it is associated a second TSpaces server to keep the names of the involved computers - denoted as the tuple space of the network.

The tool contains two parts:

- The *Advertiser* serves to state the network of workstations. Using the *Advertiser* a computer is linked to the network of workstations to perform a parallel-distributed computing.

The graphical interface of the *Advertiser* allows to

- Declare the computer as a member of the network of workstations - i.e. a tuple with the name of the computer is written into the tuple space of the network;
- Remove the computer from the network - i.e. remove the above define tuple from the tuple space of the network;
- Show the list of the computers in the network of workstations.

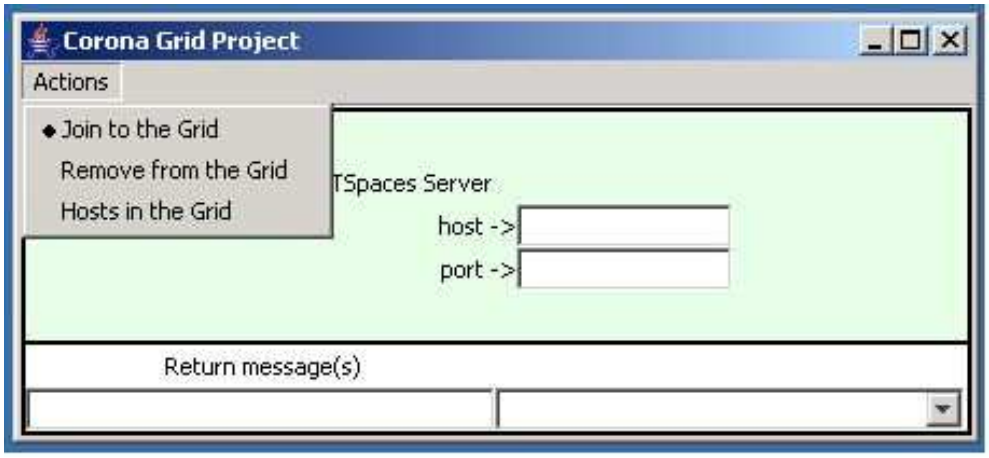


Figure 2: The window of the *Advertiser* tool.

- The *Developer* allows to
  - Generate the folders of the application;
  - Compile and archive the service part of the application;
  - Deploy the service part to the workstations of the network. The deployment is done with the *apache-tomcat-deployer;*
  - Compile the console part of the application.
  - Launch the console to run. In the *Console* class the request are programmed using the *apache commons-httpclient* software;
  - Undeploy the service part of the application.

  These targets are executed through *apache-ant.*

  The *Developer* is installed only on the host workstation.

  A number of parameters are required: The name of the application, The host and the port of the application tuple space, The number of the worker threads, The path to the *apache-ant* and The username and password for the `manager` application of *apache-tomcat.*

  The list of the computers name in the network is required, too. This list may be generated with the *Advertiser,* contained in the *Developer,* too.

The scenario to run a parallel-distributed application using the framework involves:

1. To set up the network of workstations:
   (a) On launches the tuple space associated to the network;

(b) On each workstation starts the *apache-tomcat* Web server and, with the Advertiser, declares the availability of that computer to join to the network of workstations.

2. Using the Developer, on the host workstation, we can install, deploy, start the tuple space associated to the application and execute the parallel-distributed computation launching the Console.

The current version of the framework may be downloaded from the author's Web page `http://cs.unitbv.ro/site/pagpers/scheiber.` The archive contains several other examples: a numerical integration, quicksort, the queens' problem, the graphic representation of the Mandelbrot set, the dinning philosophers' problem.

Conclusions. An implementation of the Piranha model for a parallel-distributed application is developed. The implementation does not introduce a new API but requires certain program structure. In addition we have created a tool to assist the development of an application that can be ported to any Java enabling platform. As a drawback of our approach is that, if a workstation dies then a running application never finishes. We intend to work on this problem. The security constraints are that of the *apache-tomcat* Web server.

References

[1] R. BISSELING, *Parallel Scientific Computation. A structured approach using BSP and MPI,* Oxford Univ. Press, 2004.

[2] G. A. PAPADOPOULOS, F. ARBAB, Coordination models and languages. CWI Report, SEN-R9834, 1998.

[3] SCHEIBER E., 2007, A TSpaces Based Framework for Parallel - Distributed Applications. *Knowledge Engineering Principles and Techniques* 1 (2007), Cluj University Press, 341-345.

[4] SCHEIBER E., 2009, A Parallelization scheme of Some Algorithms. Knowledge Engineering Principles and Techniques 2 (2009), *Studia Universitas Babeş-Bolyai, Informatica,* 244-248.

[5] M. SNIR, D. OTTO, S. HUSS-LEDERMAN, D. WALKER, J. DONGARRA, *MPI: The Complete Reference.* MIT Press, Cambridge, MA, 1996.

[6] * * * , TSpaces-User's Guide & Programmer's Guide, Distributed with the software, Version 2.1.2, 2000.

Ernest Scheiber
Transilvania University of Braşov
Faculty of Mathematics and Computer Science
str. Iuliu Maniu 50
ROMANIA
E-mail: *scheiber@unitbv.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Modelling and optimization of static and dynamic power systems

Stanislaw Sieniutycz

Abstract

This research treats power optimization for energy converters, such like thermal, solar and chemical engines. Thermodynamic analyses lead to converter's efficiency and limiting power. Steady and dynamic systems are investigated. Static optimization of steady systems applies the differential calculus or Lagrange multipliers, dynamic optimization of unsteady systems uses variational calculus and dynamic programming. The primary result of the first is the limiting value of power, whereas that of the second is a total generalized work potential. The generalizing quantity depends on thermal coordinates and a dissipation index, h, i.e. the Hamiltonian of the problem of minimum entropy production. It implies stronger bounds on work delivered or supplied than the classical work of thermodynamics.

1 Introduction

In this research we treat power limits in static and dynamical energy systems driven by fluids that are generally restricted in their amount or magnitude of flow, i.e. are certain resources. A power limit is an upper (lower) bound on power produced (consumed) in the system. A resource is a valuable substance or energy used in a process; its value can be quantified by specifying its exergy, a maximum work obtained when the resource relaxes to the equilibrium. Reversible relaxation of the resource is associated with the classical exergy. When dissipative phenomena prevail generalized exergies arise which quantify deviations of the system's efficiency from the Carnot efficiency. An exergy is obtained as the principal component of solution to the variational problem of extremum work under suitable boundary conditions. Other components of the solution are the optimal trajectory and optimal control. In thermal systems without chemical changes the trajectory is characterized by the temperature of the resource fluid, $T(t)$, whereas the control, is an innovative quantity, called Carnot temperature $T'(t)$. This control was defined in our previous work [1,2] to get rid less suitable while more popular controls such as heat flux or temperatures of circulating fluids (Fig.1) which are constrained by balances of mass and entropy. In opposition to these conventional controls Carnot temperature is a free control. For the engine in Fig. 1 Carnot temperature is

$$T' \equiv T_2 T_{1'} (T_{2'})^{-1} .$$

In chemical case also Carnot chemical potential(s) μ'(t) enter

$$\frac{\mu'}{T'} = \frac{\mu_2}{T_2} + \frac{T_{2'}}{T_2}\left(\frac{\mu_{1'}}{T_{1'}} - \frac{\mu_{2'}}{T_{2'}}\right) .$$

Whenever $T'(t)$ and $\mu'(t)$ differ from $T(t)$ and $\mu(t)$ then the power-producing resource relaxes with a finite rate, and with an efficiency vector different from the perfect efficiency. Only when $T' = T$ and $\mu' \doteq \mu$ the efficiency is pefect, but this corresponds with an infinitely slow relaxation rate of the resource to the thermodynamic equilibrium. Carnot variables T' and μ' are two free, independent control variables applied in power maximization of steady and dynamical generators.

The structure of this paper is as follows. Section 2 discusses various aspects of optimization with resources. Properties of steady systems are outlined in Sec. 3, whereas those of dynamical ones - in Sec. 4. Section 5 develops analyses of power yield with resource downgrading (in the first reservoir) and outlines origin of "work potentials" for finite rates. Sections 6-8 discuss various Hamilton-Jacobi-Bellman equations (HJB equations) for optimal work functions, as solutions of power yield problems. Extensions for fuel cells as electrochemical flow systems are outlined in Sec. 9.

The size limitation of the paper does not allow for inclusion of all derivations to make the paper self-contained, thus the reader may need to turn to some previous works, [1] - [5] of which ref. [3] discusses convergence of numerical algorithms solving HJB equations and role of Lagrange multipliers.

2 Thermodynamic Aspects of Finite Resources

Limited amount or flow of a resource working in an engine causes a decrease of the resource potential in time (chronological or spatial). This is why studies of the resource downgrading apply the dynamical optimization methods. From the optimization viewpoint, dynamical process is every one with sequence of states, developing either in the chronological time or in (spatial) holdup time. The first group refers to unsteady processes in non-stationary systems, the second group may involve steady state systems.

In a process of energy production two resting reservoirs do interact through an energy generator (engine). In this process power flow is steady only when two reservoirs are infinite. When one, say, upper, reservoir is finite, its potential must decrease in time, a result from the energy balance. Any finite reservoir is thus a resource reservoir. It is the resource property that leads to the dynamical behavior of the fluid and its relaxation to the equilibrium with an infinite lower reservoir (usually the environment).

Alternatively, fluid at a steady flow can replace resting upper reservoir. The resource downgrading is then a steady-state process in which the resource fluid flows through a pipeline or stages of a cascade and the fluid's state changes along a steady trajectory. As in the previous case the trajectory is a curve describing the fluid's relaxation towards the equilibrium between the fluid and the lower reservoir (the environment). This is sometimes called "active relaxation" as it is associated with the simultaneous work production. It should be contrasted with "dissipative relaxation", a well-known, natural process between a body or a fluid and the environment without any power production. Relaxation (either active or dissipative) leads to a decrease of the resource potential (i.e. temperature) in time. An inverse of the relaxation process is the one in which a body or a fluid abandons the equilibrium. This cannot be spontaneous; rather the inverse process needs a supply of external power. This process refers to thermal upgrading of the resource, which can be accomplished with a heat pump.

3 Power and Power Limits in Steady Systems

The great deal of research on power limits published to date deals with stationary systems, in which case both reservoirs are infinite. To this case refer steady-state analyses of the Chambadal-Novikov-Curzon-Ahlborn engine (CNCA engine [6]), in which energy exchange is described by Newtonian law of cooling, or the Stefan-Boltzmann engine, a system with the radiation fluids and the energy exchange governed by the Stefan-Boltzmann law [7].

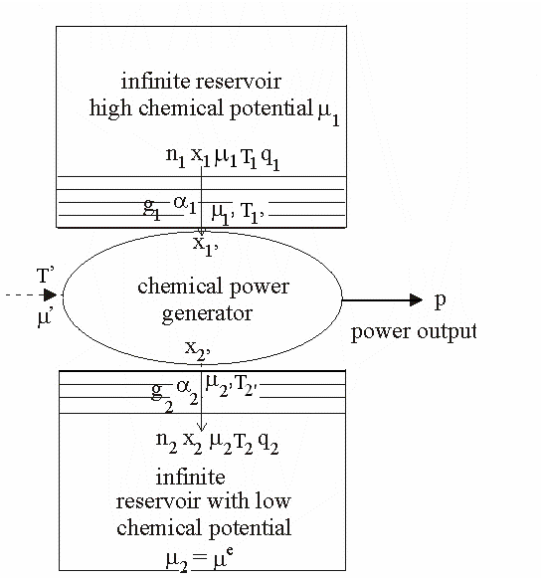


Fig. 1. A scheme of an engine controlled by a suitable choice of Carnot variables $T'$and μ'.

Due to their stationarity (caused by the infiniteness of both reservoirs), controls maximizing power are lumped to a fixed point in the state space. In fact, for the CNCA engine, the maximum power point may be related to the optimum value of a free (unconstrained) control variable which can be efficiency η or Carnot temperature T'. In terms of the reservoirs temperatures T_1 and T_2 and the internal irreversibility factor Φ one finds $T'_{opt} = (T_1 \Phi T_2)^{1/2}$ [4]. For the Stefan-Boltzmann engine exact expression for the optimal point cannot be determined analytically, yet, this temperature can be found graphically from the chart $P=f(T')$. Moreover, the method of Lagrange multipliers can successfully be applied [8]. As their elimination from a set of resulting equations is quite easy, the problem is broken down to the numerical solving of a nonlinear equation for the optimal control T'. Finally, the so-called pseudo-Newtonian model [4, 5], which uses state or temperature dependent heat exchange coefficient, $\alpha(T^3)$, omits, to a considerable extent, analytical difficulties associated with the Stefan-Boltzmann equation. Applying this model in the so-called symmetric case, where both reservoirs are filled up with radiation, one shows that the optimal (power maximizing) Carnot temperature of the steady radiation engine is that for the CNCA engine, i.e. [4]. This equation is, in fact, a good approximation under the assumption of transfer coefficients dependent solely on bulk temperatures of reservoirs.

4 Power and Its Limits in Dynamical Systems

Dynamical energy yield requires the knowledge of an extremal curve rather than an extremum point, i.e. is associated with application of variational metods in place of static optimization methods. For example, the use of the pseudo-Newtonian model to quantify the dynamical energy yield from radiation, gives rise to an extremal curve describing the radiation relaxation to the equilibrium. This curve is non-expotential, the consequence of the nonlinear properties of the relaxation dynamics. Non-expotential are also other curves describing the radiation relaxation, e.g. those following from exact models using the Stefan-Boltzmann equation (symmetric and hybrid, [4,5]).

Analytical difficulties associated with dynamical optimization of nonlinear systems are severe; this is why diverse models of power yield and diverse numerical approaches are applied. Optimal (e.g. power-maximizing) relaxation curve $T(t)$ is associated with the optimal control curve $T'(t)$; they both are components of the dynamic optimization solution to a continuous problem. In the corresponding discrete problem, formulated for numerical purposes, one searches for optimal temperature sequences $\{T^n\}$ and

$\{T'^{n}\}$. Various discrete optimization methods involve: direct search, dynamic programming, discrete maximum principle, and combinations of these methods.

Minimum power supplied to the system is described by function sequences $R^n(T^n, t^n)$, whereas maximum power produced – by functions $V^n(T^n, t^n)$. Importantly, energy limits of dynamical processes are connected with the exergy functions, the classical exergy and its rate-dependent extensions. To obtain classical exergy from power functions it suffices to assume that the thermal efficiency of the system is identical with the Carnot efficiency. Yet, non-Carnot efficiencies lead to generalized exergies. The latter depend not only on the classical thermodynamic variables but also on their rates. These generalized exergies refer to state changes in a finite time, and can be contrasted with the classical exergies that refer to reversible quasistatic processes evolving infinitely slowly. The benefit obtained from generalized exergies is that they define stronger energy limits than those predicted by classical exergies. In this case Fig. 1 above is suitably applied to calculation of dynamical systems. Introduced is a cascade of engines, where each stage is controlled by suitable choice of Carnot variables T' and μ'. The system stage comprises: a resource at flow, engines and the environment. At each stage power is generated and total flux of resulting power is maximized. In the chemical case the control is the propelling mass flux of the fuel flowing to the power generator.

5 Finite-Rate Exergies as Work Potentials

Two different works, the first associated with the resource downgrading during its relaxation to the equilibrium and the second – with the reverse process of resource upgrading, are essential. During the approach to the equilibrium engine mode takes place in which work is released, during the departure-heat-pump mode occurs in which work is supplied. Work W delivered in the engine mode is positive by assumption ("engine convention"). Sequence of irreversible engines such as the one in Fig.1 (CNCA or Stefan-Boltzmann engines) serves to determine a rate-dependent exergy extending the classical exergy for irreversible, finite rate processes. Before maximization of a work integral, process efficiency η has to be expressed as a function of state T and a control, i.e. energy flux q or rate $\mathrm{d}T/\mathrm{d}\tau$, to assure the functional property (path dependence) of the work integral. The integration must be preceded by maximization of power or work at flow (the ratio of power and flux of driving substance) w to assure an optimal path. The optimal work is sought in the form of a potential which depends on the end states and duration. For appropriate boundary conditions, the principal function of extremum work coincides with the notion of an exergy, the function that characterizes quality of resources.

The idea of an infinite number of infinitesimal CNCA steps, necessary for exergy calculations, can be developed. Each step is a work-producing (consuming) stage with the energy exchange between two fluids and the thermal machine through finite "conductances". For the radiation engine it follows from the Stefan-Boltzmann law that the effective transfer coefficient α_1 of the radiation fluid is necessarily temperature dependent, $\alpha_1 = \propto T_1^3$. The second or low-T fluid represents the usual environment, as defined in the exergy theory. This fluid possesses its own boundary layer as a dissipative component, and the corresponding exchange coefficient is α_2. In the physical space, the flow direction of the resource fluid is along the horizontal coordinate l. The optimizer's task is to find an optimal temperature of the resource fluid along the path that extremizes the work consumed or delivered.

Total power obtained from an infinite number of small engines is determined as the Lagrange functional

$$\dot{W}[\mathbf{T}^i, \mathbf{T}^f] = \int_{t^i}^{t^f} f_0(T, T')dt = -\int_{t^i}^{t^f} \dot{G}c(T)\eta(T, T')\dot{T}dt \tag{1}$$

where f_0 is power generation intensity, $\dot{G}$ - resource flux, $c(T)$-specific heat, $\eta(T, T')$ -efficiency in terms of state T and control T, further $\mathbf{T}$ − enlarged state vector comprising state and time, t −time variable (residence time or holdup time) for the resource contacting with heat transfer surface. Sometimes one uses a non-dimensional time τ, identical with the so-called number of the heat transfer units. Note that,

for constant mass flow of a resource, one can extremize power per unit mass flux, i.e. the quantity of work dimension called "work at flow". In this case Eq. (1) describes a problem of extremum work. When the resource flux is constant a work functional describing the thermal exergy flux per unit flux of resource can be obtained from Eq. (1)

$$w_{\max\limits_{dT/dt}} = -\int_{T^i=T}^{T^f=T^e} c(T)\left(1-\frac{T^e}{T'(T,dt/dT)}\right)dT\ . \tag{2}$$

Note that the independent variable in this equation is T, i.e. it is different than that in Eq. (1). The function f_0 in Eq. (1) contains thermal efficiency function, η, described by a practical counterpart of the Carnot formula. When $T > T^e$, efficiency η decreases in the engine mode above η_C and increases in the heat-pump mode below η_C. At the limit of vanishing rates, $dT/dt = 0$ and $T' \to T$. Then work of each mode simplifies to the common integral of the classical exergy. For the classical thermal exergy

$$w_{\max\limits_{dT/dt\to 0}} = -\int_{T^i=T}^{T^f=T^e} c(T)\left(1-\frac{T^e}{T}\right)dT = h-h^e-T^e(s-s^e)\,. \tag{3}$$

Nonlinearities can have both thermodynamic and kinetic origins; the former refer, for example, to state dependent heat capacity, $c(T)$, the latter to nonlinear energy exchange. Problems with linear kinetics (Newtonian heat transfer) are an important subclass. In problems with linear kinetics, fluid's specific work at flow, w, is described by an equation

$$w[\mathbf{T}^i,\mathbf{T}^f] = \dot{W}/\dot{G} = -\int_{T^i}^{T^f} c(T)\left(1-\frac{T^e}{T}\right)dT - T^e\int_{t^i}^{t^f} c(T)\frac{(T'-T)^2}{T'T}\,d\tau \tag{4}$$

where

$$\tau \equiv \frac{x}{H_{TU}} = \frac{\alpha' a_v F}{\dot{G}c}\,x = \frac{\alpha' a_v F v}{\dot{G}c}\,t = \frac{t}{\chi} \tag{5}$$

is non-dimensional time of the process. Equation (5) assumes that a resource fluid flows with velocity v through cross-section F and contacts with the heat transfer exchange surface per unit volume a_v [1]. Quantity τ is identical with the so-called number of the heat transfer units.

Solutions to work extremum problems can be obtained by:
a) variational methods, i.e. via Euler-Lagrange equation

$$\frac{\partial L}{\partial T} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{T}}\right) = 0\ . \tag{6}$$

In the example considered above, i.e. for a linear thermal system

$$T\frac{d^2T}{d\tau^2} - \left(\frac{dT}{d\tau}\right)^2 = 0 \tag{7}$$

which corresponds with the optimal trajectory

$$T(\tau,\tau^f,T^i,T^f) = T^i(T^f/T^i)^{\tau/\tau^f}\ . \tag{8}$$

($\tau^i = 0$ is assumed in Eq. (8).) However, the solution of the Euler-Lagrange equation does not provide any information about the optimal function V. Hence another method must be used as described below.
b) dynamic programming via Hamilton-Jacobi-Bellman equation (HJB equation, [9]) for the 'principal function' (V or R), also called extremum work function. This is described below.

6 HJB Equations for Selected Power Systems

For the linear kinetics considered

$$\frac{\partial V}{\partial \tau} - \max_{T'}\left\{(-\frac{\partial V}{\partial T} - c(1-\frac{T^e}{T'}))(T'-T)\right\} = 0\,. \tag{9}$$

The extremal work function V is a function of the final state and total duration. After evaluation of optimal control and its substitution to Eq. (9) one obtains a nonlinear equation

$$\frac{\partial V}{\partial \tau} - c\left\{\sqrt{T^e} - \sqrt{T(1+c^{-1}\partial V/\partial T)}\right\}^2 = 0 . \tag{10}$$

which is the Hamilton-Jacobi equation of the problem. Its solution can be found by the integration of work intensity along an optimal path, between limits T^i and T^f. A reversible, path independent part of V is the classical exergy $A(T, T^e, 0)$. Details of models of multistage power production in sequences of engines are known from previous papers [1]-[5].

We shall display further some Hamilton-Jacobi-Bellman equations for power systems described by nonlinear kinetics. A suitable example is a radiation engine whose power integral is approximated by a pseudo-Newtonian model of radiative energy exchange associated with optimal function

$$V(T^i,t^i,T^f,t^f) \equiv \max_{T'(t)}\left(-\int_{t^i}^{t^f} \dot{G}_m c_m (1-\Phi'\frac{T^e}{T'}) \upsilon\,(T',T)dt \right) \tag{11}$$

where $\upsilon = \alpha(T^3)(T'-T)$. An alternative form uses Carnot temperature T' explicit in υ [5]. Optimal power (11) becomes

$$\dot{W} = -\int_{T}^{T_0} \dot{G}_m \left(c_{hm}(T) - c_{vm}(T)\frac{T^e}{T} \right) \upsilon dt - \int_{T}^{T_0} T^e \dot{G}_m \left(c_{vm}(T)\left(\frac{\chi\upsilon^2}{T(T+\chi\upsilon)} + (1-\Phi)\frac{\upsilon}{T+\chi\upsilon}\right)\right) dt . \tag{12}$$

This process is described by a pseudolinear kinetics $dT/dt = f(T, T')$ consistent with $\upsilon = \alpha(T^3)(T'-T)$ and a general form of HJB equation for work function V in the form

$$. \quad -\frac{\partial V}{\partial t} + \max_{T'(t)}\left(f_0(T,T') - \frac{\partial V}{\partial T} f(T,T') \right) = 0 \tag{13}$$

where f_0 is defined as the integrand of Eq. (11) or (12).

A more exact model or radiation conversion relaxes the assumption of the pseudo-Newtonian transfer and applies the Stefan-Boltzmann law. For a *symmetric* model of radiation conversion (both reservoirs composed of radiation)

$$\dot{W} = \int_{t^i}^{t^f} \dot{G}_c(T)\left(1-\frac{\Phi T^e}{T'}\right)\beta\frac{T^a - T'^a}{(\Phi'(T'/T^e)^{a-1}+1)T^{a-1}}dt . \tag{14}$$

The coefficient is $\beta = \sigma a_v c_h^{-1}(p_m^0)^{-1}$ is related to molar constant of photons density p_m^0 and Stefan-Boltzmann constant σ. In the physical space, power exponent a=4 for radiation and a=1 for a linear resource. With a state equation

$$\frac{dT}{dt} = -\beta\frac{T^a - T'^a}{(\Phi'(T'/T_2)^{a-1}+1)T^{a-1}} \tag{15}$$

[5] applied in general Eq. (13) we obtain a *HJB* equation

$$-\frac{\partial V}{\partial t} + \max_{T'(t)}\left\{\left(\dot{G}_c(T)(1-\Phi\frac{T^e}{T'}) + \frac{\partial V}{\partial T}\right)\beta\frac{T^a - T'^a}{(\Phi'(T'/T_2)^{a-1}+1)T^{a-1}}\right\} = 0 . \tag{16}$$

Dynamics (15) is the characteristic equation for Eq. (16).

For *a hybrid model* of radiation conversion (upper reservoir composed of the radiation and lower reservoir of a Newtonian fluid, [5]) the power is (17)

$$\dot{W} = -\int_{\tau'}^{t^f} G_c(T)\left(1 - \frac{\Phi T^e}{(T^a + \beta^{-1}T^{a-1}u)^{1/a} + \Phi\beta^{-1}T^{a-1}ug_1/g_2}\right)u dt \tag{17}$$

and the corresponding Hamilton-Jacobi-Bellman equation is

$$-\frac{\partial V}{\partial t^f} + \max_{T'(t)}\left\{-\left(\dot{G}_c(T)(1 - \frac{\Phi T^e}{(T^a + \beta^{-1}T^{a-1}u)^{1/a} + \Phi\beta^{-1}T^{a-1}ug_1/g_2}) + \frac{\partial V}{\partial T^f}\right)u\right\} = 0\,. \tag{18}$$

7 Some Analytical Solutions of HJB Equations

In all HJB equations extremized expressions are some Hamiltonians, H. The maximization of H leads to two equations. The first expresses optimal control T' in terms of T and $z = -\partial V/\partial T$. For the linear kinetics of Eq. (9) we obtain

$$\frac{\partial V}{\partial T} - \frac{\partial f_0(T,T')}{\partial T'} = \frac{\partial V}{\partial T} + c(1 - \frac{T^e T}{T'^2}) = 0 \tag{19}$$

whereas the second is the original equation (9) without maximizing operation

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial T}(T' - T) + c(1 - \frac{T_2}{T'})(T' - T) = 0\,. \tag{20}$$

To obtain optimal control function $T'(z, T)$ one should solve the second equality in equation (19) in terms of T', The result is Carnot control T' in terms of T and $z = -\partial V/\partial T$,

$$T' = \left(\frac{T^e T}{1 + c^{-1}\partial V/\partial T}\right)^{1/2}\,. \tag{21}$$

This is next substituted into (20); the result is the nonlinear Hamilton-Jacobi equation

$$-\frac{\partial V}{\partial \tau} + cT\left(\sqrt{1 + c^{-1}\partial V/\partial T} - \sqrt{T^e/T}\right)^2 = 0 \tag{22}$$

which contains the energylike (extremum) Hamiltonian

$$H(T, \frac{\partial V}{\partial T}) = cT\left(\sqrt{1 + c^{-1}\partial V/\partial T} - \sqrt{T^e/T}\right)^2\,. \tag{23}$$

Expressing extremum Hamiltonian (23) in terms of state variable T and Carnot control T' yields an energylike function satisfying the following relations

$$E(T,u) = f_0 - u\frac{\partial f_0}{\partial u} = cT^e\frac{(T' - T)^2}{T'^2} \tag{24}$$

E is the Legendre transform of the work lagrangian $l_0 = -f_0$ with respect to the rate $u = \mathrm{d}T/\mathrm{d}\tau$.

Assuming a numerical value of the Hamiltonian, say h, one can exploit the constancy of H to eliminate $\partial V/\partial T$. Next combining equation $H=h$ with optimal control (21), or with an equivalent result for energy flow control $u=T'-T$

$$u = \left(\frac{T^e T}{1 + c^{-1}\partial V/\partial T}\right)^{1/2} - T\,. \tag{25}$$

yields optimal rate $u = \dot{T}$ in terms of temperature T and the Hamiltonian constant h. An optimal trajectory which applies to systems with internal dissipation (factor Φ) and applies to the pseudo-Newtonian model of radiation has the form

$$\dot{T} = \left(\pm \sqrt{\frac{h_\sigma}{\Phi c_v(T)}} \left(1 - \pm \sqrt{\frac{h_\sigma}{\Phi c_v(T)}} \right)^{-1} \right) T \equiv \xi(h_\sigma, \Phi, T) T \quad , \tag{26}$$

where ξdefined in the above equation, is an intensity index and $h_\sigma = h/T$. This holds for the temperature dependent heat capacity $c_v(T)=4a_0T^3$. Positive ξ refer to heating of the resource fluid in the heat-pump mode, and the negative - to cooling of this fluid in the engine mode. Thus, the optimal power is associated with the temperature relaxation

$$\dot{T} = \xi(h_\sigma, T, \Phi) T \tag{27}$$

Equations (26 and (27) describe the optimal trajectory in terms of variables T and h. The related optimal (Carnot) control is

$$T' = \left(1 + \xi(h_\sigma, \Phi, T)\right) T \tag{28}$$

Thus, in comparison with the linear systems, the pseudo-Newtonian relaxation is not exponential.

9 Some Data for Electrochemical Systems

In chemical engines mass transports participate in transformation of chemical affinities into mechanical power [10-13]. Yet, in chemical and electrochemical engines generalized reservoirs are present, capable of providing both heat and substance. Fuel cells are electrochemical engines propelled by chemical reactions. Units producing power are engines, whereas those which consume power are electrolyzers. Their main advantage in comparison to heat engines is that their efficiency is not a major function of device size. Figure 2 illustrates a solid oxide fuel cell (SOFC) which works in the power yield mode.

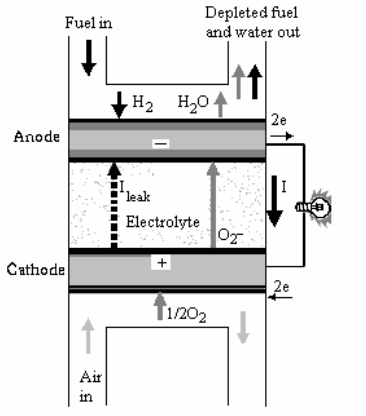


Fig. 2. Principle of a solid oxide fuel cell

The basic structure of fuel cells includes electrolyte layer in contact with a porous anode and cathode on either side. Gaseous fuels are fed continuously to the anode (negative electrode) compartment and an oxidant (i.e., oxygen from air) is fed to the cathode (positive electrode) compartment. Electrochemical reactions at the electrodes produce an electric current. The effect is the oxidation of fuel, e.g. hydrogen, and reduction of oxidant, e.g. oxygen. This makes fuel cells similar to an engine in Fig. 1.

Voltage lowering in fuel cells below the idle run value is a suitable measure of their imperfection, Fig.3. With the concept of effective resistances operating voltage of a fuel cell can be represented as the departure from the idle run voltage E_0 [14]

$$V = E_0 - V_{\text{int}} = E - V_{\text{act}} - V_{\text{conc}} - V_{\text{ohm}} \tag{29}$$

The losses, called polarization, include three main sources: activation polarization (V_{act}), ohmic polarization (V_{ohm}), and concentration polarization (V_{conc}). Large number of approaches for calculating polarization losses has been reviewed [15]. Activation and concentration polarization occurs at both electrodes locations, while the resistive polarization represents ohmic losses. As the losses increase with current, the initially increasing power finally begins to decrease, so that power maxima emerge (Fig. 3).

The voltage equation used in [14] for the purpose of the power calculation is
:

$$V = E_0(T, p_{H_2}) - iA_R(p_{H_2})\exp\left(\frac{\Delta E}{RT}\right) + B\ln\left(1 - \frac{i}{i_L(T, p_{H_2})}\right) \quad , \tag{30}$$

where a limiting current is introduced defined by an equation

$$i_L = C_1 T^{-1} \exp(\frac{-E_a}{RT}) p_{H_2} \tag{31}$$

in which C_1 is a experimentally determined parameter. Power density is simply the product of voltage V and current density i. In an ideal situation (no losses) the cell voltage is defined by the Nernst equation. Yet, while the first term of Eq. (30) defines the voltage without load, it nonetheless takes into account losses of the idle run, which are the effect of flaws in electrode constructions and other imperfections which cause that the open circuit voltage will in reality be lower than the theoretical value. Activation polarization V_{act} is neglected in this model. The losses include ohmic and concentration polarization. The second term of Eq. (30) quantifies ohmic losses associated with electric resistance of electrodes and flow resistance of ions through the electrolyte. The third term refers to mass transport losses. Quantity i_L is the current arising when the fuel is consumed in the reaction with the maximum possible feed rate.

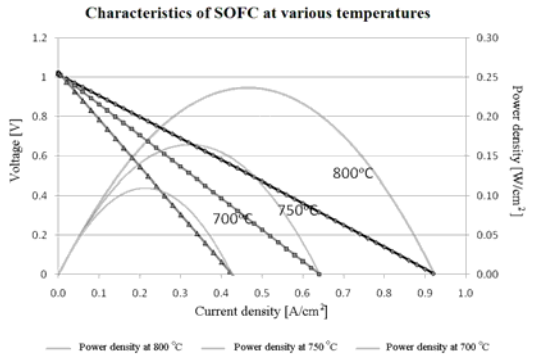


Fig.3. Voltage-current density and power-current density characteristics of the SOFC for various temperatures. Continuous lines represent the Aspen Plus™ calculations testing the model consistency with the experiments. These lines were obtained in Wierzbicki's MsD thesis supervised by S. Sieniutycz and J. Jewulski [14]. Points refer to experiments of Wierzbicki and Jewulski in Warsaw Institute of Energetics ([14] and ref [18] therein).

In the literature there are many experimental and theoretical examples showing power maxima in fuel cells and proving the suitability of the theory to chemical and electrochemical systems. For example, data obtained in L. Chen's research group [15] are consistent with those of Wierzbicki [14].

10 Concluding Remarks

This research provides data for power production bounds (limits) which are enhanced in comparison with those predicted by the classical thermodynamics. When infinite reservoirs assure constancy of chemical potentials, problems of extremum power are static optimization problems. For finite reservoirs, however, amount and chemical potential of an active reactant decrease in time, and considered problems are those of dynamic optimization and variational calculus. In fact, thermostatic bounds are often too far

from reality to be really useful. Generalized bounds, obtained here by solving HJB equations, are stronger than those of thermostatics. As opposed to classical thermodynamics, they depend not only on state changes but also on irreversibilities, ratios of stream flows, stream directions, and mechanism of all transfers. The methodology familiar for thermal machines can be applied to electrochemical engines. Extensions are available for multicomponent, multireaction chemical systems [11]).

A real work supply can only be larger than the finite-rate bound obtained by the optimization. Similarly, the real work delivered from a nonequilibrium work-producing system (with the same boundary states and duration but with a suboptimal control) can only be lower than the corresponding finite-rate bound. Indeed, the two bounds, for a process and its inverse, which coincide in thermostatics, diverge in thermodynamics. With thermokinetic models, we can confront and surmount the limitations of applying classical thermodynamic bounds to real processes.

Acknowledgement: This paper is funded by the Polish Ministry of Science grant NN208 019434: Thermodynamics and Optimization of Chemical and Electrochemical Energy Generators with Applications to Fuel Cells. Results in Fig. 3 were found with M. Wierzbicki during the present author's supervising of his MsD thesis. Thesis co-supervising by dr J. Jewulski of Warsaw Institute of Energetics is also appreciated. Finally, the author thanks dr P.Kuran for his kind help in the manuscript preparation.

References

[1] S. Sieniutycz, Carnot controls to unify traditional and work-assisted operations with heat & mass transfer, *International J. of Applied Thermodynamics*, 6, 2003, 59-67.
[2] S. Sieniutycz, A synthesis of thermodynamic models unifying traditional and work-driven operations with heat and mass exchange, *Open Sys. & Information Dynamics*, 10, 2003, 31-49..
[3] S. Sieniutycz, Dynamic programming and Lagrange multipliers for active relaxation of fluids in non-equilibrium systems, *Applied Mathematical Modeling*, 33, 2009, 1457-1478.
[4] S. Sieniutycz and P. Kuran, Nonlinear models for mechanical energy production in imperfect generators driven by thermal or solar energy, *Intern. J. Heat Mass Transfer*, 48, 2005, 719-730.
[5] S. Sieniutycz and P. Kuran, Modeling thermal behavior and work flux in finite-rate systems with radiation, *Intern. J. Heat and Mass Transfer* 49, 2006, 3264-3283.
[6] F.L. Curzon and B. Ahlborn, Efficiency of Carnot engine at maximum power output, *American J. Phys*. 43, 1975, 22-24.
[7] A. de Vos, *Endoreversible Thermodynamics of Solear Energy Conversion*, Oxford University Press, 1994, p.30.
[8] P. Kuran, *Nonlinear Models of Production of Mechanical Energy in Non-Ideal Generators Driven by Thermal or Solar Energy*, PhD thesis, Warsaw University of Technology, 2006.
[9] R. E. Bellman, *Adaptive Control Processes: a Guided Tour*, Princeton, University Press, 1961, pp.1-35.
[10] S. Sieniutycz, An analysis of power and entropy generation in a chemical engine, *Intern. J. of Heat and Mass Transfer* 51, 2008, 5859–5871.
[11] S. Sieniutycz, Complex chemical systems with power production driven by mass transfer, *Intern. J. of Heat and Mass Transfer*, 52, 2009, 2453–2465.
[12] G. Lin, J. Chen and E. Bruck, Irreversible chemical-engines and their optimal performance analysis, *Applied Energy* 78, 2004, 123–136.
[13] S. Sieniutycz, Thermodynamics of simultaneous drying and power yield, *Drying Technology*, 27, 2009, 322 – 335.
[14] Wierzbicki, M., 2009. *Optimization of SOFC based energy system using Aspen Plus^{TM}*, Thesis (MsD) supervised by S. Sieniutycz (Faculty of Chemical and Process Engineering, Warsaw University of Technology) and J. Jewulski (Laboratory of Fuel Cells, Institute of Energetics, Warsaw).
[15] Zhao, Y., Ou, C., Chen, J., 2008. A new analytical approach to model and evaluate the performance of a class of irreversible fuel cells. *International Journal of Hydrogen Energy*, 33, 2008, 4161- 4170.

STANISLAW SIENIUTYCZ
Warsaw University of Technology
Faculty of Chemical and Process Engineering
1 Waryńskiego Street, 00-645 Warszawa, POLAND
E-mail: sieniutycz@ichip.pw.edu.pl

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

E-learning solutions in curve and surface design

Corina Simian

Abstract

The aim of this paper is to present a tool which can be integrated in an e-learning system, allowing students to learn advanced topics in curve and surface design. The tool is designed using MATLAB platform. The design of this tool implies advanced knowledge related to CAGD. We used Bezier curves and surfaces and introduced a new interpolation algorithm for obtaining Bezier spline curves of C^1 class.

1 Introduction

The design of curves and surfaces is a topic of great interest in CAGD. A large number of courses from bachelor and master level are dedicated to this topic. The wide area of applications made this topic more attractive. The theory of Bezier curves and surfaces and Coons patches, together with B-spline methods were the reference points in CAGD. Many interactive applications for construction of Bezier curves and surfaces, most of them implemented in Java, can be found on Internet. The aim of this article is to present a tool which can be used for accomplish many e-learning tasks in the field of Bezier curves and surfaces. For implementation we used MATLAB platform.

The paper is organized as follows: in chapter 2 we briefly present the problem of approximation of curves and surfaces using spline curves and surfaces in Bezier form and few algorithmic aspects. Section 3 presents the advantages of MATLAB for implementation. The main result, MathTool, is presented in section 4. Section 5 contains conclusions and further developments.

2. Interpolation using Bezier curves and surfaces

A Bezier curve of degree n is defined using a set of control points, b_i, i=0,...,n. The parametric form of a Bezier curve is given using the Berstein basis.

$$f(t) = (x(t), y(t), z(t)) = \sum_{i=0}^{n} b_i B_i^n(t), t \in [0,1], b_i = (x_i, y_i, z_i) \quad (1)$$

The Bernstein polynomials are defined as:

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i; t \in [0,1]; i = 0,1,...,n \quad (2)$$

The most important case, from a practical point of view is the case of cubic Bezier curves, obtained for n=3.
More details regarding the Bezier curves can be found in [3].
The problem of interpolation using Bezier curves is the following one: given *n+1* points, P_i, *i=0,...,n*, find a curve *C*, in Bezier form, such that $P_i \in C, \forall i = 0,...,n$, satisfying given conditions of continuity and derivability, .

A kind of "pseudo-interpolation" consists in finding the control points of the curve *C* (see [6], [7]). Let consider the cubic case. We have known 4 points from a curve and find the 4 control points of a cubic Bezier curve which contains these 4 points. We suppose that the given points correspond to equidistant values of parameter *t*. The interpolation conditions lead to:

$$c(0) = b_0 = P_0; \qquad c(\frac{1}{3}) = \frac{8}{27}b_0 + \frac{12}{27}b_1 + \frac{6}{27}b_2 + \frac{1}{27}b_3 = P_1; \quad (3)$$

$$c(\frac{2}{3}) = \frac{1}{27}b_0 + \frac{6}{27}b_1 + \frac{12}{27}b_2 + \frac{8}{27}b_3 = P_2; \qquad c(1) = b_3 = P_3$$

The equalities (3) can be written in the form (4):

$$b = A^{-1} * p, \qquad (4)$$

$$A = \frac{1}{27}\begin{pmatrix} 27 & 0 & 0 & 0 \\ 8 & 12 & 6 & 1 \\ 1 & 6 & 12 & 8 \\ 0 & 0 & 0 & 27 \end{pmatrix}; P = \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = \begin{pmatrix} Px_0 & Py_0 & Pz_0 \\ Px_1 & Py_1 & Pz_1 \\ Px_2 & Py_2 & Pz_3 \\ Px_4 & Py_4 & Pz_4 \end{pmatrix}; b = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix}$$

If the number of interpolation points is greater, we use a spline Bezier curve. The j-th component of a spline curve of degree *n*, composed by *m* pieces, has the equation:

$$c_j(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \sum_{i=0}^{n} b_i^j B_i(t - j + 1), t \in [j-1, j], j = 1,...,m \qquad (5)$$

A spline Bezier curve is of class G^0 (see [3]), iff $b_n^j = b_0^{j+1}, \forall j = 1, \ldots, m-1$.

A spline Bezier curve is of class G^1, if the control points $b_{n-1}^j, b_n^j = b_0^{j+1}, b_1^{j+1}$ are collinear. There are different approaches for reaching G^1 class of a spline Bezier curve (see [3]).

The tensor product Bezier surface patch is controlled using a mesh of control points, b_{ij} and is defined as:

$$s(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \sum_{i=0}^{n}\sum_{j=0}^{m} b_{ij} B_j^m(u) B_i^n(v); (u,v) \in [0,1] \qquad (6)$$

The bi cubic tensor product Bezier surface is given by the relations:

$$s(u,v) = Bv' \cdot b \cdot Bu \qquad (7)$$

with *Bv* the column matrix of Bernstein polynomials in variable *v*, *Bv'* the transpose matrix of *Bv*, *Bu* the column matrix of Bernstein polynomials in variable *u* and $b=[b_{ij}]$, the 4x4 matrix, with the coordinates of the 16points of the control mesh of the Bezier surface.

3. MATLAB in curve and surface design

MATLAB offers many advantages in implementation of the algorithms from CAGD and in particular for algorithms which handle Bezier curves and surfaces:

a) Easy matrix formulation of problems in MATLAB. Best performance of the programs which use specific operations with matrix in MATLAB
b) Graphical capabilities. MATLAB provides high-level functions for displaying graphics.
c) Specific toolboxes can be developed from users.
d) MATLAB provides powerful tools for Graphical User Interfaces design.

The most important advantage is the possibility of matrix formulation of problems in MATLAB. All the relations presented in section 2, can be expressed in a matrix form, which enable to write performing programs and easy to understand. The representation of a Bezier curve using (1), is illustrated forward

```
function CurbaBezierf(b)
%computation
t=0:0.01:1;%parameter
B0=(1-t).^3
B1=3*(1-t).^2.*t
B2=3*(1-t).*(t.^2)
B3=t.^3
%Bernstein polynomial matrix
B=[B0;B1;B2;B3]
%Bezier curve
f=b*B;
%graphical representation
hold on
title(Bezier curve');
plot(b(1,:),b(2,:),'r-')
plot(f(1,:),f(2,:),'b')
hold off
```

The representation of a bi cubic tensor product Bezier surface patch is made very easy using the matrix form given in (7). The function *bez3surf*, uses the coordinates of the control points of the surface and return the coordinates of a mesh of points from the Bezier surface. Using this mesh, the surface can be represented using the MATLAB graphical function surf, or mesh.

```
function [x,y,z] = bez3surf(bx,by,bz)
u=0:0.01:1; %parameter
v=0:0.01:1;
B0u=(1-u).^3;

B1u=3.*(1-u).^2.*u;
B2u=3.*(1-u).*u.^2;
B3u=u.^3;
%matrix of Bernstein polynomial in variable u
Bu=[B0u;B1u;B2u;B3u];

B0v=(1-v).^3;
B1v=3.*(1-v).^2.*v;
B2v=3.*(1-v).*v.^2;
B3v=v.^3;
%matrix of Bernstein polynomial in variable v
Bv=[B0v;B1v;B2v;B3v];
```

```
%coordinates of points from the Bezier surface
x=Bv'*bx*Bu;
y=Bv'*by*Bu;
z=Bv'*bz*Bu;
```

In the case of interpolation using a bi cubic tensor product Bezier surface patch, we use 16 interpolation points and find the 16 points from the control mesh of the Bezier surface. We repeat first on lines and then on columns the procedure for interpolation of a curve using 4 interpolation points, given in relation (4). The function *bsplines* computes the coordinates of the control points of the Bezier interpolation curve. The function *bezinterpsurf* applies the tensor product method for obtaioning the coordinates of the control points of interpolation Bezier surface.

```
function[bx]= bsplines(px)
%px is 1*4 matrix with a coordinate (x,y or z) of 4 points from the surface
%bx is a matrix with the same coordinate of the control points of Bezier
%curve
A=(1/27)*[27. 0 0 0; 8. 12. 6. 1.; 1. 6. 12. 8.; 0 0 0 27.];
bx = inv(A)*px';
bx=bx';
```

```
function [bx] = bezinterpsurf(px)
%px is a 16x16 matrix, containing one of the coordinates (x,y or z) of the
%interpolation points.
%function bsplines is applied on lines
b1=ones(4,4);
b11 = bsplines(px(1,:));
b12 = bsplines(px(2,:));
b13 = bsplines(px(3,:));
b14 = bsplines(px(4,:));
b1=[b11; b12; b13; b14]
%function bsplines is applied on columns
b=ones(4,4);
bc1=(b1(:,1))';
bc2=(b1(:,2))';
bc3=(b1(:,3))';
bc4=(b1(:,4))';
b21 =bsplines(bc1);
b22 =bsplines(bc2);
b23 =bsplines(bc3 );
b24 =bsplines(bc4);
bx=[b21; b22; b23; b24];
bx=bx';
```

The functions presented before are called from the function *beziertensor*, which realizes the graphical representation of the interpolation surface.

```
function beziertensor(px,py,pz)
bx= bezinterpsurf(px);
by= bezinterpsurf(py);
bz=bezinterpsurf(pz);
[sx,sy,sz] = bez3surf(bx,by,bz);
%bz=bz'
surf(sx,sy,sz);hold on
shading interp;
```

```
colormap cool;
alpha 0.5;
plot3(px,py,pz,'ro');
hold on
```

The examples presented in this section proved the advantages of MATLAB in curve and surface design using Bezier curves and surfaces.

4 Main results - MathTool

4.1 General presentation

Our aim was to allow a collaborative learning in the field of curve and surface design (using Bezier curves and surfaces) and to develop algorithmic and programming skills. Our tool, named MathTool, is a web based tool, which includes many sections. Theoretical elements are grouped in 4 sections, containing many items and providing a guide in curve and surface design. The sections are:
1) Theoretical fundamentals: contains basic mathematical elements regarding multivariate polynomial interpolation, spline interpolation, Bernstein polynomials.
2) Curves and surfaces in CAGD: presents a short history of CAGD, parametric representation of curves and surfaces and different types of curves and surfaces used in CAGD.
3) Bezier curves: includes theoretical, algorithmic and computational aspects related to Bezier curves and interpolation using Bezier curves.
4) Bezier surfaces: contains theoretical, algorithmic and computational aspects related to Bezier surfaces and interpolation using spline surfaces in Bezier form. It is presented the tensor product method. The ruled surfaces, cylindrical surfaces, sweep surfaces are taken into account.

Every section has a corresponding MATLAB GUI, to illustrate the presented notions.
The user has access to the source code, after registration in the MathTool interface. The call-back functions from the GUI are easy to change and to adapt such that new applications can be obtained. We included a guide which contains a detailed description of the algorithms used, implementation details and instructions to use the MATLAB GUIs.
References for a deeper research in the domain of curve and surface design are given in a special section.
For exchanging ideas or posing questions, a discussions forum is included and can be used.

The MathTool Interface is presented in fig. 1. In fig. 2 is presented a list of the sources which can be accessed by a user

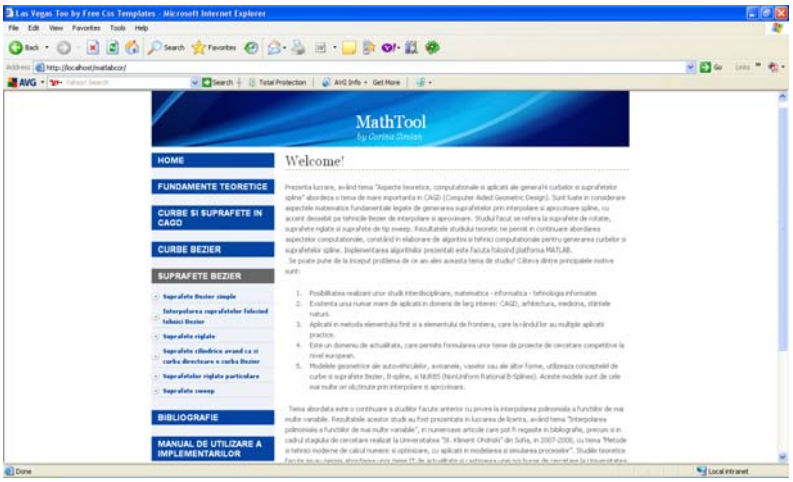


Fig. 1 – MathTool Interface

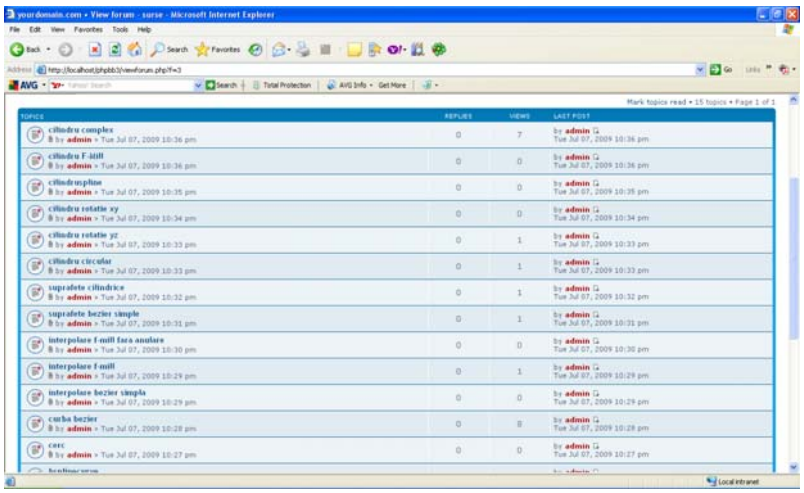


Fig. 2 –List of the MATLAB files

4.2 MATLAB GUIs

The aim of this subsection is to present the most important implementations included in MathTool. The implementations are made in MATLAB and offer a friendly Graphic User Interface (GUI).

4.2.1 Interactive Bezier Curve

A special pop-up menu created in the MATLAB figure windows allows different representation of a cubic Bezier curve.

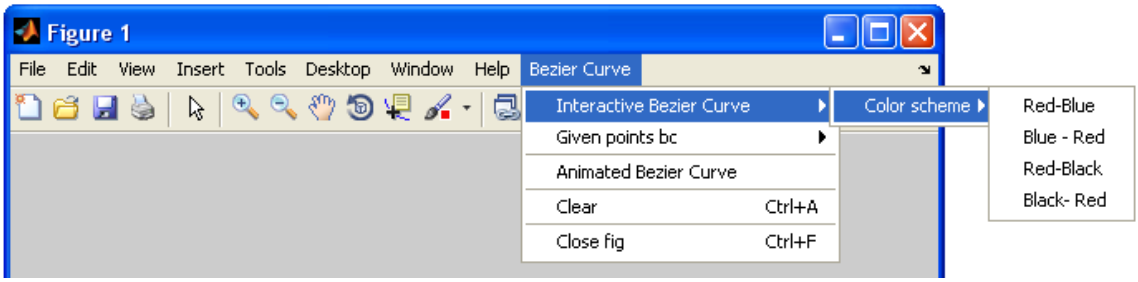


Fig. 3 Pop–up Bezier curve menu

Interactive representation of a cubic Bezier curve is presented in fig. 4.

4.2.2 Interpolation using Bezier curves

Many GUI were designed for interpolation using Bezier curves: F-Mill interpolation, interpolation using spline curves in Bezier form of class G^0 and G^1(see [1]-[3], [6] for more details of these methods).

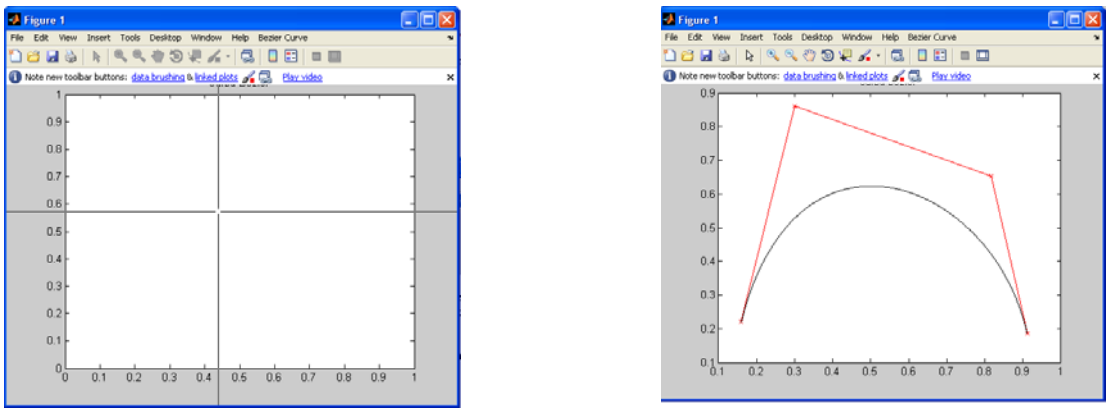


Fig. 4 Interactive representation of a cubic Bezier curve

All GUIs allow the input from the graphical interface of the interpolation points coordinates and provide the spline interpolation curve, the control polygon and the interpolation points. As an application, a GUI for approximation of one or two circular arcs of given sweep is realized.

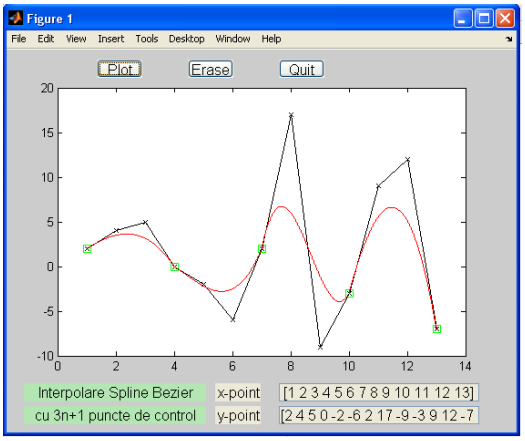


Fig. 5 Bezier spline curve with 3n+1 control points

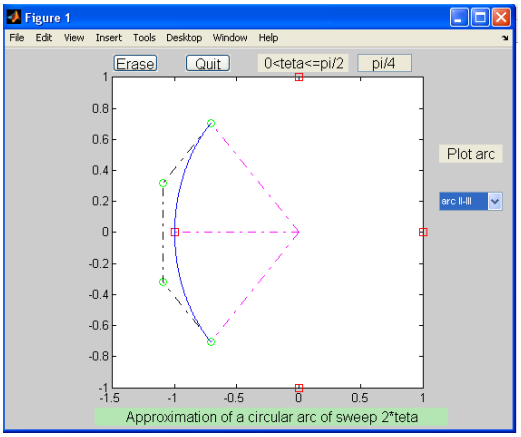


Fig. 6 Approximation of circular arc

4.2.3 Interpolation surfaces in Bezier form

Many GUIs for obtaining interpolation surfaces in Bezier form were designed: interpolation bicubic Bezier surfaces, cylindrical surfaces, rotation surfaces.

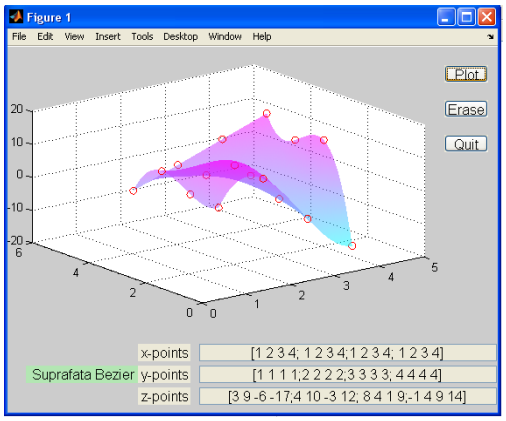


Fig. 7 Bicubic interpolation Bezier surface

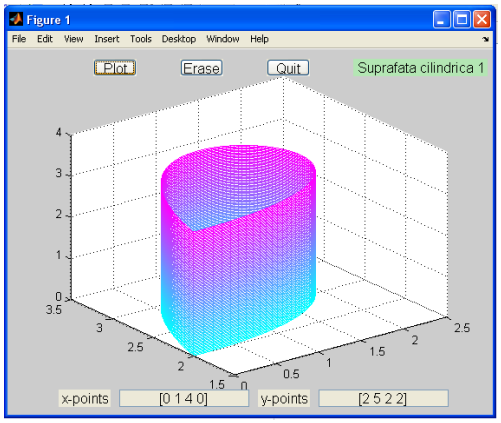


Fig 8- Cylindrical surface based on Bezier curve

4.3 E-learning goals of MathTool

MathTool was created in order to accomplish many e-learning tasks:
1. Computer-based learning of topics from Computational Geometry and especially from CAGD. Both the theoretical elements and the graphical interfaces from MathTool are useful in distance learning, or in blended learning (see [10]).
2. Computer-based learning of MATLAB programming and MATLAB GUI design. All implementations are relevant for MATLAB programming. The source can be visualized and all details about the algorithms and implementations are given in the User Implementation Guide.
3. Computer – based training in Bezier curves and surfaces and their applications in CAGD.
4. Computer-supported collaborative learning. To encourage students to work together and to cooperate, a forum of discussions was included in the MathTool.

5. Conclusions

In this article we present a web based tool which can be used in computer based-learning and training in topics from CAGD and MATLAB programming. It offers also a collaborative learning support. A great advantage is the modularization of the applications and the possibility of easy changing and adapting the modules for other goals.
A further development of our tool will offer the possibility to extended the set of programs and knowledge from the users and to introduce a section of open problems.

References

[1] W. Boehm, G. Farin and J. Kahmann, *A survey of curve and surface methods in CAGD Computer Aided Geometric Design*, 1–60, 1984.
[2] C. de Boor, K. Hollig and M. Sabin, High accuracy geometric Hermite interpolation. *Computer Aided Geometric Design*, 4:4,269–278, 1987.
[3] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design. A practical guide*, Academic Press, Boston, 1996.
[4] G. Farin, Class A Bezier Curves,*Computer Aided Geometric Design*, 23, Elsevier Press, pp.573-581, 2006
[5] M. A. Khan, Approximation of Circle by Cubic Bezier Curves, http://www.mathworks.com/matlabcentral/fileexchange/6844
[6] D. E Ulmet., Swept surfaces in Computer Aided Geometric Design, *Proceedings of the International Workshop New approaches, Algorithms and Advanced Computational Techniques in Approximation Theory and its Applications*, Ed. Univ. Lucian Blaga, Sibiu, pp. 19-31, 2007.
[7] D. E. Ulmet, Customized Reflection Lines for Surface Interrogation in Car Body Design, *Proceedings of the SYNASC,* Timisoara, 2007
[8] eLearn Magazine, ACM, http://www.elearnmag.org/subpage.cfm?section=archives
[9] Journal of e-Learning, http://www.ejel.org/index.htm
[10] http://en.wikipedia.org/wiki/E-learning#Computer-based_learning

CORINA SIMIAN
Zurich University
Institute for Mathematics
Winterthurerstrasse 190, CH-8057 Zurich
SWITZERLAND
E-mail: corina.simian@math.uzh.ch,
corinafirst@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Wasp based algorithms and applications

Dana Simian

Abstract

The aim of this paper is to present the wasp based computational model and many applications of wasp based algorithms. A general frame for designing a wasp based algorithm, starting from the classical problem of task allocation in a factory, is realized. The most important characteristics of the wasp computational model are underlined and the way of particularization of these characteristics for each problem is presented. Original applications of wasp based algorithms in modeling multi agent systems, in solving optimization problems and in building a reinforcement scheme for a stochastic learning system are presented.

1 Introduction

In the last ten years methods and models inspired from the behavior of social insects like ants and wasps have gained increasing attention. Computational analogies to adaptive natural multi-agent systems have served as inspiration for multi-agent optimization and control algorithms in a variety of domains and contexts. Self-organization, direct and indirect interactions between individuals are important characteristics of these natural multi-agent systems. Metaheuristics inspired from nature represent an important approach to solve NP-difficult problems. It is important to identify when a problem can be solved using these kind of methods. It is the goal of this article to identify some type of problems which can be solved using wasp computational based algorithms and to give a general frame for design these algorithms. The remainder of this paper is organized as follows: in section 2 we present the wasp computational model and the classical problem of task allocation in a factory. Starting from this problem we realize a frame for design models based on wasp behavior and present many models for multi agents systems, from different fields. In section 3 we present a reinforcement scheme for stochastic learning automata, based on wasp behavior. In section 4 we present a wasp based algorithm for improving the performances of a co-mutation operator. The co-mutation operator is used in a hybrid approach for building multiple SVM kernels. Conclusions are presented in section 5.

2 Wasp behavior based algorithms in modeling multiagent systems

The self organization model that takes place within a colony of wasps was used for solving large complex problems, most of them with a dynamic character. In [18], Theraulaz et al. present

a model for the self-organization within a colony of wasps. The main characteristics of wasp behavior is the stimulus-response mechanism which governs the interaction between individuals from the nest and the environment. Every wasp has a response threshold for each zone of the nest. The brood located in one zone broadcasts a stimulus. Based on a wasps threshold and the amount of stimulus, a wasp may be or not engaged in the task of foraging for this zone.

An algorithm based on wasp behavior is essentially a system based on agents that simulates the natural behavior of insects. Wasp behavior algorithms are used especially in building multiagent systems for solving task allocation problems, distributed coordination of resources, dynamic scheduling and control tasks. These kind of algorithms are used for solving optimization problems related to distributed factory coordination ([5]-[8]), job shop dynamic scheduling ([4]), self organization of robot groups ([19]), distribution of tasks in a multiagent e-learning system ([10],[9]), distribution of patients in a health sanatorium system ([14]), dynamically allocations of tasks and resources in monitoring process within a site Natura 2000 ([15]), etc. Each of these problems has analogies with the classical problem of tasks allocation in a distributed manufacturing system with specialized machines. We shortly characterize this problem.

In this classical problem, any machine have associated an artificial wasp. An artificial wasp will have a response threshold for every possible task(new command) and any task will have associate a stimulus. Let denote the set of response thresholds for the wasp i with

$$W_i = \{w_{i,j},\ j = 1, \ldots, n\}, \tag{1}$$

where $w_{i,j}$ is the response threshold of wasp i for the task j. The threshold value $w_{i,j}$ may vary in an interval $[w_{min}, w_{max}]$. A task j which has not been assigned yet to a machine broadcasts in the system a stimulus S_j, which is proportional to the length of time the task has been waiting for assignment to a machine.

An artificial wasp, will probabilistically decide if it bids or nor for a task. The probability is dependent of the level of the threshold and of the stimulus. The general formula for the probability is

$$P(i,j) = \frac{S_j^\gamma}{S_j^\gamma + w_{i,j}^\gamma} \tag{2}$$

The exponent γ is a system parameter. If, in (2), $\gamma \geq 1$, than as lower the response thresholds is, as bigger the probability of binding a task is, but a wasp can bid for a task if a hight enough stimulus is emitted. In [18] is used such a rule for task allocation with $\gamma = 2$.

One of the elements which particularizes an wasp based algorithm is the way in which the response thresholds are updated. First, in [3] these thresholds remain fixed over time. Later, in [18] is considered that a threshold for a given task decreases during time periods when that task is performed and increases otherwise. In [5] Cicirello and Smith, consider three ways in which the response thresholds are updated. The first two ways encourage a wasp to take a task for which the associated machine is specialized. The goal of the third one is to encourage a wasp associated with an idle machine to take whatever jobs rather than remaining idle. The rules for thresholds updating are given in (3)-(6):
If the machine i is specialized in realizing a task of the same type as the task j then

$$w_{i,j} = w_{i,j} - \delta_1 \tag{3}$$

If the machine i is specialized in realizing a task of other type, then

$$w_{i,j} = w_{i,j} + \delta_2 \tag{4}$$

If the machine i is currently idle then

$$w_{i,j} = w_{i,j} - \delta_3^{\tau}, \tag{5}$$

where τ is the length of time the machine has been idle and is an exponent.
The δ_1, δ_2 and δ_3 are positive system constants. The response thresholds are reinforced as to encourage the artificial wasp to bid on task for which the associated machine is specialized and to avoid idle machines.

More, additional conditions can be added in order to restrict the access of a wasp to a specific task.

In the case of distribution of tasks in a multiagent e-learning system, each student will have associated a wasp agent. The tasks in e-learning systems can be projects, grants, group activities, individual activities, etc. In [10], rules like (3)-(6) are given, but a specific condition restricts a wasp to bid for a task if no sufficient specialization of the student for this task exists.

In [14], two models for patients distribution in a health network sanatoria are given. In the first one, patients are associated with the machines from the classical model of task distribution in a factory. There is a artificial wasp for each patient, which bids for a place in a sanatoria taking into account a complex set of criteria, including the type and gravity of the disease of a patient and the specific of the sanatorium. There is only one rule for updating the response threshold. If the sanatorium is contraindicated for a type of disease that the patient have than the response threshold is increased and the probability decreases. The response thresholds are reinforced to encourage the routing wasp to bid on a place in a sanatorium which maximize the effects of the treatment for all types of the diseases that a patient have. As greater the number of the types of diseases of a given patient can be treated in a sanatorium as lower the response threshold of the patient wasp for this sanatorium is. A restrictive condition is imposed to not allow a wasp to bid for a sanatorium contraindicated for a disease that patient have.

The second model consider sanatoria associated with the machines in a factory, that is, every sanatorium will have associated an artificial wasp biding for patients. This model has only a theoretical importance, be cause it does not match the most important requirements of the patients distribution: the efficiency of the treatment.

In [15] the response threshold is update using an adaptive procedure. The rule for updating the response threshold contains 3 parameters greater or equal to 0. The values of parameters are modified taking into account the local policy and the previous results.

Another important characteristics of wasp based algorithm is the way in which the conflicts are solve. When the probability to bid for a task, of two ore more wasps is the same, a conflict occurs. Specific mechanism for solving the conflicts must exist for the efficiency of the algorithm. In nature, wasps within the colony self-organize themselves into a dominance hierarchy. When two individuals of the colony encounter each other, the wasp with the higher social rank will have a higher probability of dominating in the interaction. A probability function will solve the domination conflicts in a wasp based algorithm. In [5] is introduced a model for defining the social rank of a wasp using a force function F_i. The probability $P_c(i,j)$ in a domination contest between wasp i and j, depends on the forces F_i and F_j.

$$P_c(i,j) = P(Wasp\ i\ win\ |F_i, F_j) = \frac{F_j^{\alpha}}{F_i^{\alpha} + F_j^{\alpha}} \tag{6}$$

The name "force" for the function F_i is improper, be cause as lower F_i is as bigger the probability that the wasp i gains is. That is the social rank of the wasp i is inverse proportional with

the force F_i. The parameter α is a system parameter, usually chosen equal 2. The definition of F_i is specific for each problem.

In [10] the force is defined taking into account the specialization of a student and the complexity of activities in which the student is involved, such that, the routing wasps associated with students of equivalent specializations and equivalent complexity of the activity in their queue, have equal probabilities of getting the activity.

In [14] the force depends on two system parameters which define the policy of the patients distribution in the sanatoria system. Different values for these parameters will make a hierarchy of the model design criteria: gravity of diseases, results already obtained by the patient in the sanatorium.

Fitness functions can also be used for solving domination contests, even that they are not specific for wasp models.

3 Automatic control based on wasp behavior model

In [16] we describe a stochastic learning system for intelligent vehicle control. The system is based on two automata. The system can learn the best possible action based on the data received from on-board sensors, or from roadside-to-vehicle communications. The action probabilities of the learning automata are functions of the status of the physical environment. We supposed that an intelligent vehicle is capable to do the following actions: LEFT (shift to left lane), RIGHT (shift to right lane), LINE_OK (stay in current lane), ACC (accelerate), DEC (decelerate) and SPEED_OK (keep current speed). The first three actions represent the lateral actions and the other three represent the longitudinal actions. We assume that there are four sensors modules for evaluating the information received from the on-board sensors or from the highway infrastructure: the headway module, two side modules (left and right module) and a speed module. The sensors modules send a response to the longitudinal and lateral automaton. The sensors modules are decision blocks that calculate the response (reward/penalty), based on the last chosen action of automaton. A reward is obtained when the action is correct and a penalty in other case. By example, a penalty response is received from the left sensor module when the action is LEFT and there is a vehicle in the left or the vehicle is already traveling on the leftmost lane. The actions received from the two automata are handled by the regulation layer in a distinct manner, using for each of them a regulation buffer. A rewarded action will be introduced in the regulation buffer of the corresponding automaton, else in buffer will be introduced a certain value which denotes a penalized action by the physical environment. The regulation layer carries out an action only if it is recommended l times consecutively by the automaton, where l is the length of the regulation buffer.

We proposed a reinforcement scheme for stochastic learning automata, based on the computational model of wasp behavior. The longitudinal and lateral automaton has an associated wasp. Each wasp has a response threshold for each possible action. Let denote by $\theta_{i,j}$ the response threshold of automaton i for the action $A_{i,j}$, $i \in \{1,2\}$, $j \in \{1,\ldots,r\}$, r r denotes the number of actions of the automaton. The threshold values $\theta_{i,j}$ may vary between θ_{min} and θ_{max}. An action $A_{i,j}$ broadcasts to the automaton a stimulus $S_{i,j}$ which is equal to the number of occurrences of action in the regulation buffer of the automaton i.

The probability that the automaton i picks the action $A_{i,j}$ is

$$P(i,j) = \frac{S_{i,j}^2}{S_{i,j}^2 + \theta_{i,j}^2} \tag{7}$$

The rules for updating the threshold values are the following:
- if automaton i will execute the action $A_{i,j}$ then $\theta_{i,j} = \theta_{i,j} - \delta_1$;
- for each action $A_{i,k}$, with $k \neq i$ the threshold $\theta_{i,k}$ is updated according to: $\theta_{i,k} = \theta_{i,k} + \delta_2$.

After updating the action probability vectors in both learning automata, using the new reinforcement scheme presented before, the outputs from stochastic automata are transmitted to the regulation layer. After an action is executed, the action probability vector is initialized to $\frac{1}{r}$, where r is the number of actions. When an action is executed, regulation buffer is initialized also.

We made an implementation of a simulator for the Intelligent Vehicle Control System (see [16]). The implementation was realized in Java and based on JADE platform. Used within this simulator of an Intelligent Vehicle Control System, the reinforcement scheme, based on a computational model of wasp behavior, has proved its efficiency.

4 Other applications for wasp behavior algorithms

Wasp behavior can be used in optimization algorithms, not only in a system based on agents. A probabilistic choice in an optimization problem can be made using a stimulus-response mechanism. We will present next an example of a multiple kernel optimization using wasp based algorithm. We start from the problem of binary classification using SVM. If the data set is separable we obtain an optimal separating hyperplane with a maximal margin (see [20]). In the case of no separable data the kernel method is used for projected the data in a space with higher dimension in which they are separable by a hyperplane. Kernel functions can be interpreted as representing the inner product of data objects mapped into a nonlinear feature space. It is sufficient to calculate the inner product in the feature space without knowing explicit the mapping function. Simple kernel functions usually used are: polynomial, RBF, sigmoidal, defined in the following equalities:

$$K_{pol}^{d,r}(x_1, x_2) = (x_1 \cdot x_2 + r)^d, \ r, d \in Z_+ \tag{8}$$

$$K_{RBF}^{\gamma}(x_1, x_2) = \exp\left(\frac{-1}{2\gamma^2}|x_1 - x_2|^2\right) \tag{9}$$

$$K_{sig}^{\gamma}(x_1, x_2) = tanh(\gamma \cdot x_1 \cdot x_2 + 1) \tag{10}$$

The real problems require more complex kernels. The problem of obtaining an appropriate complex SVM kernel for given type of data is an important and current problem. We studied in [12], [13], many models for building optimal multiple SVM kernels, using a hybrid method, based on a genetic algorithm, working on two levels. In the first level, the genetic algorithm builds the multiple kernel, choosing the types and the parameters of the simple kernels. Every chromosome code a multiple kernel. In the second level, the quality of chromosomes is evaluated using a SVM algorithm. The fitness function for the genetic algorithm is represented by the classification accuracy rate of the SVM algorithm on a validation set of data. The validation set of data is part of the training subset of data for the SVM algorithm. The multiple kernels are built using a set of operations from $\{+, *, exp\}$ which preserve the kernel properties. Many details about the method can be found in [11]-[13].

The most complex model we built and analyze in [12], [13] contains at most 4 simple kernels of types described in (8)-(10). The multiple kernel can be graphic illustrated using a tree which terminal nodes contain a single kernel and the other nodes contains the operations. If a node

contains the operation *exp* only one of its descendants is considered (the "left" kernel). Each chromosome codes the expression of a multiple kernel. The tree representation of multiple kernel $(K_1 op_2 K_2) op_1 (K_3 op_3 K_4)$ is given in figure 1.

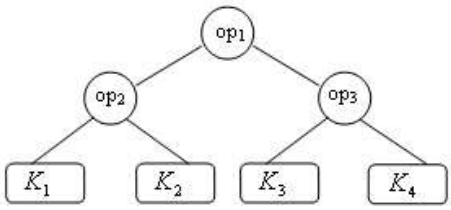


Fig.1 Representation of a multiple kernel

We use a linear structure of the chromosome which codes the multiple kernel described above. The linear structure contains 78 genes. The first 6 genes code the operations type, each operation op_i is represented using 2 genes. A kernel is defined by its type, t_i stored using 2 genes and the parameters values. If the kernel is of polynomial type, for the degree d_j are allocated 4 genes and the parameter r_i is represented using 12 genes. If the associated kernel is not polynomial, 16 genes are used to represent a real value of parameter γ_i. Therefore for each kernel are allocated 18 genes. The linear structure of the chromosome is depicted in figure 2.

Fig.2 Linear chromosome's structure

The operations of the genetic algorithm are the following ones:
Initialization: We initialize randomly population of chromosomes, P(t), with P elements.
Evaluation: Chromosomes are evaluated using the classification accuracy of the multiple kernel coded in them, on a validation set of data.
Co-Mutations: We select randomly one element among the best T% from P(t). We mutate it using the co-mutation operator $LR - M_{ijn}$, defined in [17].
In the linear structure of the chromosome, the set of operations is coded on only 6 chromosome, while the multiple kernels parameters are coded using 16 genes. Therefore, in the co-mutation process, the probability of changing the multiple kernels parameters is bigger the the probability of changing the operations from the multiple kernel. In order to allow a often faster changing of the operations in the chromosome structure we optimize the $LR - M_{ijn}$ operator using a scheme based on a computational model of wasp behavior.

Each chromosome C, can be associated with a wasp, which will bid for a unique task: changing the set of operations coded in chromosome structure within the co-mutation operation of the genetic algorithm. Each "wasp - chromosome" has a response threshold θ_C. The threshold values θ_C may vary in the range $[\theta_{min}, \theta_{max}]$. The set of operations coded within chromosome broadcasts a stimulus S_C, equal to the difference between maximum classification accuracy (100) and the current classification accuracy obtained using the multiple kernel coded in the chromosome:

$$S_C = 100 - CA_C \tag{11}$$

The modified $LR-M_{ijn}$ operator will perform a mutation that will change the operations coded within chromosome with probability:

$$P_C = \frac{S_C^2}{S_C^2 + \theta_C^2} \tag{12}$$

The update of the chromosome threshold is made in the evaluation part of the genetic algorithm, using the following rule:
-if the classification accuracy of the new chromosome C is lower than in the previous step, then

$$\theta_C = \theta_C - \delta,\ \delta > 0, \tag{13}$$

-if the classification accuracy of the new chromosome C is greater than in the previous step, then

$$\theta_C = \theta_C + \delta,\ \delta > 0, \tag{14}$$

In this way is increased the probability of changing the set of operations for the chromosomes with lower classification accuracy.

Using the "leukemia" data set we obtained the following values for classification accuracy: 67.65 for the the standard libsvm package, 91.18 for multiple kernels obtained using genetic approach, based on co-mutation operator $LR - M_{ijn}$, 94.12 for multiple kernels obtained using genetic approach, based on modified co-mutation operator $LR - M_{ijn}$ using the wasp computational model. The dimension of population was 35 and the number of generations was 30.

5 Conclusions

In this article we made a general presentation of the main characteristics of a wasp behavior based algorithm and illustrated how, starting from classical wasp computational model, we can solve many different problems. The applications we realized prove the efficiency of wasp computational model, not only in modeling systems based on agents but also in solving NP difficult optimization problems. The idea of solving all practical problems starting from the classical problem of tasks allocation in a factory and establishing analogies, allows quick adaptation of the computational wasp model for solving problems from many different fields.

References

[1] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm intelligence: From natural to artificial systems.*Santa Fe Institute Studies of Complexity, Oxford University Press, 1999.

[2] E. Bonabeau, M. Dorigo, G. Theraulaz, Inspiration for optimization from social insect behaviour, *Nature* , vol. 406, 2000, pp. 39–42.

[3] E. Bonabeau, G. Theraulaz, J.I. Demeubourg, Fixed response thresholds and the regulation of division of labor in insect societies, *Bull. Math. Biol.*, vol. 60, 1998, pp. 753–807.

[4] Y. Cao, Y. Yang, H. Wang, Integrated Routing Wasp Algorithm and Scheduling Wasp Algorithm for Job Shop Dynamic Scheduling, *Electronic Commerce and Security 2008 International Symposium* on 2008, pp. 674–678.

[5] V. A. Cicirelo, S.F. Smith, Wasp-like Agents for Distributed Factory coordination, *Autonomous Agents and Multi-Agent Systems*, Vol. 8, No. 3,2004, pp. 237–267.

[6] V. A. Cicirelo, S.F. Smith, Distributed Coordination of Resources via Wasp-like Agents, *First NASA GSFC/JPL Workshop on Radical Agent Concepts* (WRAC), January, 2002.

[7] V. A. Cicirello, S. F. Smith, Wasp nests for self-configurable factories, *Agents 2001, Proceedings of the Fifth International Conference on Autonomous Agents*, ACM Press, May-June 2001.

[8] V. A. Cicirelo, S.F. Smith, Insect Societies and Manufacturing, *The IJCAI-01 Workshop on Artificial Intelligence and Manufacturing: New AI Paradigms for Manufacturing*, August, 2001.

[9] I. Moisil, I. Pah, D. Simian, Advanced Modelling of Tutor Inteligent Systems for Distance Learning Applications, *International Journal of Computers, Communications & Control*, A Quarterly Journal, Year: 2008, Volume: III, Agora University Editing House

[10] D. Simian, C. Simian, I. Moisil, I. Pah, Computer mediated communication and collaboration in a virtual learning environment based on a multi-agent system with wasp-like behavior, *Lecture Notes in Computer Science*, Springer Berlin Heildelberg, LSCC 2007, 2007, pp. 606–614.

[11] D. Simian, F. Stoica, Optimization of Complex SVM Kernels Using a Hybrid Algorithm Based on Wasp Behaviour, *Lecture Notes in Computer Science*, LNCS 5910, I. Lirkov, S. Margenov, and J. Wasniewski (Eds.): LSSC 2009, Springer-Verlag Berlin Heidelberg, 2010, pp. 361368.

[12] D. Simian, F. Stoica, An evolutionary method for constructing complex SVM kernels, *Recent Advances in Mathematics and Computers in Biology and Chemistry, Proceedings of the 10th International Conference on Mathematics and Computers in Biology and Chemistry, MCBC'09, Prague, Chech Republic, 2009*, WSEAS Press 2009, pp. 172-178.

[13] D. Simian, F. Stoica, Evaluation of a hybrid method for constructing multiple SVM kernels, *Recent Advances in Computers, Proceedings of the 13th WSEAS International Conference on Computers, Recent Advances in Computer Engineering Series*, WSEAS Press 2009, pp. 619–623.

[14] D. Simian, F. Stoica, C. Simian, Models For A Multi-Agent System Based On Wasp Like Behaviour For Distributed Patients Repartition, *Advanced Topics On Evolutionary Computing - proceeding of the 9-th Conference on Evolutionary Computing EC'08*, WSEAS Press 2008, pp. 82–87.

[15] D. Simian, F. Stoica, A. Curtean-Bănăduc, Multi-Agent System models for monitoring optimization within a site Natura 2000, *Mathematics and Computers in Biology and Chemistry, Proceedings of 9th WSEAS Int. Conf. on Mathematics and Computers in Biology and Chemistry MCBC'08, Bucharest, Romania 2008*, WSEAS Press, 2008, pp. 212–217.

[16] F. Stoica, D. Simian, Automatic control based on Wasp Behavioral Model and Stochastic Learning Automata, *Proceedings of 10th WSEAS int. Conf. On Mathematical Methods, Computational Techniques and Intelligent Systems, Corfu, 2008*, WSEAS Press, 2008, pp. 289–295.

[17] F. Stoica, D. Simian, C. Simian, A new co-mutation genetic operator, *Advanced topics on evolutionary computing, Proceeding of the 9-th Conference on Evolutionay Computing, Sofia, 2008*, WSEAS Press, 2008, pp. 76-82.

[18] G. Theraulaz , E. Bonabeau, J. I. Demeubourg, Response threshold reinforcement and divizion of labour in insects societies, *Proc. R Spob London B.*,vol. 265, no. 1393,1998, pp. 327–335.

[19] G. Theraulaz , E. Bonabeau, J. Gervet , J. I. Demeubourg , Task differention in policies wasp-colonies.A model for self-organizing groups of robots, *From animals to Animats: Proceedings of the First International Conference on Simulation of Adaptiv behaviour*, 1991, pp. 346–355.

[20] Vapnik V., *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

Dana Simian
Lucian Blaga Unuversity
Faculty of Sciences
5-7, I. Ratiu str., Sibiu
ROMANIA
E-mail: *dana.simian@ulbsibiu.ro, d_simian@yahoo.com*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

A multilingual conversational agent with speech recognition and text-to-speech capabilities

Mihai Stancu

Abstract

Conversational agents are computer programs that utilize natural language technologies to engage users in human-like, text-based, information-seeking and task-oriented "dialogs". They can support a broad range of applications in business enterprises, education, government, healthcare, and entertainment. In this paper we propose and implement a conversational agent based on AIML, with speech recognition and text-to-speech capabilities and a virtual avatar. We describe the challenges we had to face in order to improve the artificial intelligence, and the quality of the speech synthesis for the Romanian language.

1 Introduction

The most natural form of communication is face-to-face communication. This is why using virtual avatars and natural speech is very useful for human-computer interaction [1]. Many artificial conversational entities or conversational agents, speech recognition and text-to-speech engines, as well as virtual 2d or 3d characters have been created for this purpose [1],[2],[3],[4]. Today there are numerous applications of conversational agents in the following areas [3],[5]:

- businesses:
  - customer service: responding to customers' general questions about products and services
  - help desk: responding to internal employee questions
  - website navigation: guiding customers to relevant portions of complex websites.
  - guided selling: providing answers and guidance in the sales process
  - technical support: responding to technical problems
- telecommunications: eliminating the need for human-assisted support
- education: creating virtual tutors and trainers for interactive learning
- healthcare: creating information experts and virtual communication partners for therapy
- entertainment: creating life-like videogame characters and virtual actors

The main goal of this paper is to propose and implement a multilingual conversational agent with speech recognition and text-to-speech capabilities. Though there are many similar applications, the majority of which are clones after the original, what this paper brings to this field is a new way of utilizing existing technologies to achieve a better human-computer interaction.

The paper has the following contents: section 2 presents a short classification of the different conversational agents up to date, section 3 describes shortly the systems used in order to reach our goal, in section 4 we have the design and implementation of the conversational agent, and finally, in the last section we give our results and a plan for future developments.

2 The main types of conversational agents

Conversational agents can be classified by the following categories [3],[4]:

1. The technology used for the conversational engine (AIML, bayessian networks, etc.)
2. The degree of interactivity (text-based interface, SR and TTS capabilities, avatars, etc.)
3. The goal for which it was created (the applications mentioned in the first section)

If the first agents only allowed a text-based console for the conversation to take place, once the technology developed, new ways of interaction based on speech services and 2D and 3D graphics have evolved, reaching closer and closer to true human-like appearances.

More advanced means of comparing embodied conversational agents consist in evaluating the users' reactions and emotional responses while interacting with the agent [4]. In other words, it is a way of measuring the realism of the conversational agent, similar to the famous Turing test [6].

In 1990 Hugh Loebner agreed with The Cambridge Center for Behavioral Studies to underwrite a contest designed to implement the Turing Test. Dr. Loebner pledged a Grand Prize of $100,000 and a Gold Medal for the first computer whose responses were indistinguishable from a human's. Each year, a prize of $2000 and a bronze medal is awarded to the most human-like computer. The winner of the annual contest is the best entry relative to other entries that year. [7]

3 Technologies used for developing the conversational agent

We have used three technologies that have proved their value through the extremely large number of applications that utilize them today. We will now shortly present these technologies and in the following section we will show exactly how we combined them and came up with new techniques in order to achieve our goal.

Additionally we have used software applications such as the trial versions of Adobe Photoshop and Adobe Flash Professional in order to create the graphics files, the gestures and lip-sync animations of our avatar.

3.1 AIML

AIML (Artificial Intelligence Markup Language) is an XML compatible, easy to learn language, that facilitates the rapid creation of a knowledge base for a conversational agent. [8] Developed by Richard Wallace and a worldwide free software community between the years of 1995 and 2002, it formed the basis for A.L.I.C.E ("Artificial Linguistic Internet Computer Entity"), which won the annual Loebner Prize Contest for Most Human Computer three times, and was also the Chatterbox Challenge Champion in 2004 [9].

Because the A.L.I.C.E. AIML set was released under the GNU GPL, and because most AIML interpreters are offered under a free or open source license, many "Alicebot clones" have been created based upon the original implementation of the program and its AIML knowledge base, and this fact has eased very much the development of our own application [10]. Free AIML sets in several languages have been developed and made available by the user community. It was these sets that made possible for our application to support multiple languages.

The most important AIML tags are:

<aiml> which marks the beginning and the end of an AIML document;

<category> which marks a unit of the knowledge base;

<template> which marks the response to a question.

There are over 20 other tags which can be used, that allow the formation of so-called personalized predicates with the help of wildcards. Also, it is possible for a category to be called by another, and the language allows for the directing of a conversation to a specific topic.

Other advantages consist in using recurrence with the help of the <srai> operator. Without going into too much detail, we list the other possibilities the language has to offer: symbolic reduction, divide et impera, synonym substitution, keyword detection and implementation of conditionals.

3.2 MBROLA

MBROLA (Multi Band Re-synthesis Overlap-Add) is an algorithm for speech synthesis, and software which is distributed at no financial cost but in binary form only.

Initiated by the TCTS Lab of the Faculté Polytechnique de Mons in 1996, MBROLA is now a worldwide collaborative project. The MBROLA project web page [11] provides resources for a large number of spoken languages and voices.

The MBROLA software is not a complete text-to-speech system for all those languages. The text must first be transformed into phoneme and prosodic information in MBROLA's format. The input consists of a list of phonemes, together with the duration of the phonemes and a piecewise linear description of pitch, and produces speech samples on 16 bits, at the sampling frequency of the diphone database used. Separate software to do this is available for some but not all of MBROLA's languages This is the reason we implemented our own TTS system that supports multiple languages and developed for the first time a pipeline for processing raw text for Romanian speech synthesis, which will be discussed in the following section of our paper.

Although diphone-based, the quality of MBROLA's synthesis is considered to be higher than that of most diphone synthesizers. This is due in part to the fact that it is based on a preprocessing of diphones (imposing constant pitch and harmonic phases), which enhances their concatenation while only slightly degrading their segmental quality [12].

MBROLA is a time-domain algorithm, as PSOLA (Pitch Synchronous Overlap-Add) created by Paul Taylor [13], which implies very low computational load at synthesis time. Unlike PSOLA, however, MBROLA does not require a preliminary marking of pitch periods. This feature has made it possible to develop the MBROLA project around the MBROLA algorithm, through which many speech research labs, companies, or individuals around the world have provided diphone databases for 34 languages, a number which is by far a world record for speech synthesis.

3.3 MS Agent

Microsoft Agent is a set of software services that enable developers to incorporate interactive animated characters into their applications. [14] These characters can speak, via a text-to-speech engine or recorded audio, and accept spoken voice commands through speech recognition.

Microsoft Agent was first introduced through Microsoft Bob in 1995, which used an early version of Agent technology internally referred to as "Microsoft Actor". Microsoft Agent became popular as the initial version of the Office Assistant in Office 97, sometimes dubbed "Clippit" or "Clippy". [15] The first version of Microsoft Agent was released on MSDN in 1998 and the MS Agent version 2.0 core components were available for download on Microsoft’s website in 2003.

Microsoft Agent includes an ActiveX control that makes its services accessible to any programming language that supports this type of control. This means that interaction with the characters can be programmed even in HTML.

The speech engine used by MSAgent is driven by the Microsoft Speech API (SAPI), version 4 and above. Microsoft SAPI provides a control panel for easily installing and switching between various available text-to-speech and speech recognition engines, as well as voice training and scoring systems to improve quality and accuracy. Because the MBROLA voices are not compatible with SAPI, one of our future goals is to create a SAPI compliant TTS engine for all 34 languages available on MBROLA’s website.

Microsoft Agent characters are stored in files with the .ACS extension, and can be stored in a number of compressed .ACF files for better Internet distribution. Microsoft provides four agent characters for free, which can be downloaded from the Microsoft Agent website. These are called Peedy, Merlin, Genie, and Robby. New Agent characters can also be created using Microsoft's development tools, including the Agent Character Editor which we used for creating the character for our application.

4 Designing and implementing the conversational agent

In this section we present the main algorithms we used and the difficulties we had to surpass in order to make our program as modular and as flexible as possible.

The system is implemented in C# and works as follows: the user inputs text from the keyboard which is then compared to one of the predefined commands. Then the text is sent to the AI module. This module uses the AIML language to store a database of knowledge, in order to generate a response. The response is then sent to the TTS module which will perform the speech synthesis. Once the generated audio file is ready to be played, a corresponding animation sequence for the virtual avatar is rendered. The process repeats itself until termination of the program. These steps are illustrated in figure 1.

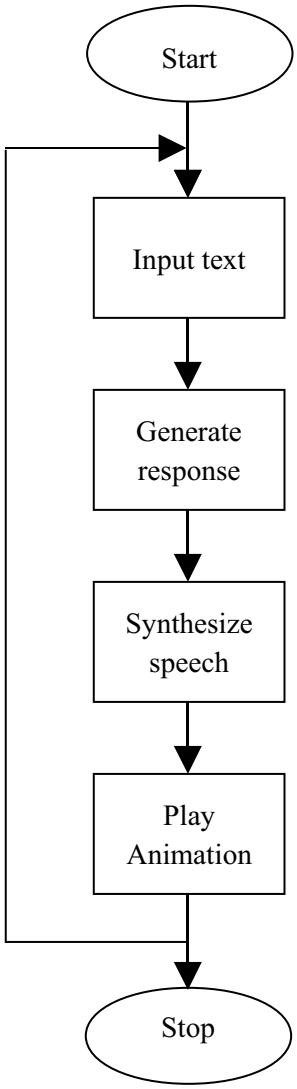


Figure 1. Functioning principle of the conversational agent

Because there is no freely available SAPI compliant Romanian TTS engine, the most difficult part was to create such an engine. In the following lines we describe how our TTS system works. Inspired by [16],[17],[18], the engine contains a natural language processing mechanism. Text is transmitted to a pre-processor, then the program generates phoneme and prosodic information and finally, the MBROLA system is called with these parameters and generates the speech audio files. These steps are illustrated in figure 2.

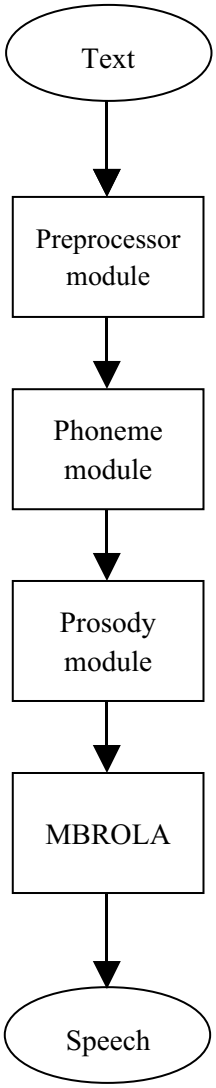


Figure 2. Functioning principle of the Romanian TTS engine

The architecture of this TTS engine allows multiple language support. Because the application we are building is a work in progress, there is no compatibility with the Speech Application Interface from Microsoft yet. In order to take advantage of SAPI, we can use other Microsoft voices and languages for speech synthesis until the final release of our conversational agent. In the next lines we give further details about every module of our TTS engine, along with a few examples written for the Romanian language, to better describe what is happening at each step.

4.1 Preprocessor module

This module uses an abbreviations dictionary and also a symbols dictionary in order to substitute all abbreviations, mathematical symbols, punctuation marks, etc. with the corresponding words or phrases in the current language. This module also replaces all Arabic and Roman numerals, all the times and dates with the corresponding text strings and changes all uppercase letters to lowercase.

As an example, the statement: "În data de 27.08.2003 locuiam pe str. Pieții, bl. 18, sc. C, et. IV, or. Victoria, jud. Brașov, România" will become "în data de douăzeci și șapte august două mii trei locuiam pe strada pieții bloc optisprezece scara ce etajul patru orașul Victoria județul brașov românia".

4.2 Phoneme module

The phoneme module is responsible for coding the preprocessed text in a sequence of phonemes. The phoneme symbols are the same SAMPA (Speech Assessment Methods Phonetic Alphabet) symbols used for the diphone database of a certain language. In order to do this, we use a dictionary of phonetic rules and a dictionary of phonetic exceptions. For the Romanian language these dictionaries have quite a small file size, unlike for English or other languages, because the Romanian language is mostly a phonetic language, i.e. words are spelled as they are written. This is also the reason why Romanian speech synthesis has a better quality than for other languages.

4.3 Prosody module

This represents the final step of our natural language processing pipeline. The prosody module gives the correct intonation for every word of the phonemized text output from the previous module. In the case of the Romanian language, we can apply a few simple rules in order to give the voice more naturalness. First every word is separated into syllables and the last but one is accented, i.e. the pitch for the sounds corresponding to that syllable is increased. Of course, not all the words in the Romanian language are accented this way, and thus a dictionary of accenting exceptions helps to accent all the words in the correct manner. This module also determines the speed at which the text is "read". By modifying the duration of the sounds corresponding to every phoneme, we can adjust the rate of speech, and also increase or shorten the pause between words. Another aspect that is currently in development is creating emotional states for the voices, again by changing the pitch values of the generated sounds, according to specific, predetermined patterns.

In figure 3 we show, for clarification, the contents of the file bonjour.pho supplied as an example for the French FR1 diphone database.

```
_ 51 25 114
b 62
o~ 127 48 170
Z 110 53 116
u 211
R 150 50 91
_ 91
```

Figure 3. Contents of bonjour.pho file

This shows the format of the input data required by MBROLA. Each line contains a phoneme name, a duration (in ms), and a series (possibly none) of pitch pattern points composed of two integer numbers each : the position of the pitch pattern point within the phoneme (in % of its total duration), and the pitch value (in Hz) at this position.

Hence, the first line of bonjour.pho : _ 51 25 114 tells the synthesizer to produce a silence of 51 ms, and to put a pitch pattern point of 114 Hz at 25% of 51 ms.

5 Conclusions and future developments

In this paper we have presented the design and implementation of a multilingual conversational agent with speech recognition and text-to-speech capabilities. Numerous efforts have been made in order to increase the interactivity and the quality of the speech synthesis for the Romanian language. Nonetheless there are many things that can be improved, that are planned for future versions of our application. Among these we mention the following:

- extending the supported languages and the knowledge base;
- incorporating a logic module, an arithmetic module, and a learning module;
- creating an agent editor and new virtual characters

References

[1] A. Paiva (ed.). *Affective Interactions: Towards a New Generation of Computer Interfaces*. Springer, Berlin, 2000.
[2] T. Rist, R. Aylett, D. Ballin, J. Rickel (eds.). *Intelligent Virtual Agents*. Springer, Berlin, 2003.
[3] H. Predinger, M. Ishizuka (eds.). *Life-Like Characters: Tools, Affective Functions, and Applications*. Springer, Berlin, 2004.
[4] J. Cassel, J. Sullivan, S. Prevost, E. Churchill. Embodied Conversational Agents. MIT Press, Cambridge, Massachussetts, 2000.
[5] J. Lester. *PracticalHandbook of Internet Computing*. CRC Press, Boca Raton, 2009.
[6] http://en.wikipedia.org/wiki/Turing_test
[7] http://www.loebner.net/Prizef/loebner-prize.html
[8] http://www.alicebot.org/aiml.html
[9] http://en.wikipedia.org/wiki/AIML
[10] http://aimlbot.sourceforge.net
[11] http://tcts.fpms.ac.be/synthesis/mbrola.html
[12] http://en.wikipedia.org/wiki/MBROLA
[13] P. Taylor. *Text-to-Speech*. Cambridge University Press, Cambridge, 2009.
[14] http://www.microsoft.com/products/msagent/main.aspx
[15] http://en.wikipedia.org/wiki/Microsoft_Agent
[16] D. Burileanu, C. Dan, M. Sima, Mihai, *A Parser-based Text Preprocessor for Romanian Language TTS Synthesis,* Eurospeech'99, Budapest, 1999
[17] O. Buza, G. Toderean, *A Romanian Syllable-Based Text-to-Speech System,* 6th WSEAS International Conference, Corfu Island, 2007
[18] A. Ferencz, T. Ratiu, M. Ferencz, T.C. Kovacs, I. Nagy, D. Zaiu, *Romvox – Experiments Regarding Unrestricted Text-to-Speech Synthesis for the Romanian Language,* 6th EWONLG, Duisburg, 1998

Stancu Mihai
Universitatea Lucian Blaga din Sibiu
Facultatea de Stiinte
Bd-ul. Victoriei, Nr.10, Sibiu
Romania
stancu_meehigh@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

A comparative analysis on the potential of SVM and k-means in solving classification tasks

Luminiţa State, Iuliana Paraschiv-Munteanu

Abstract

The method based on support vectors aims to increase the efficiency in approximating multidimensional functions. The basic idea in a SVM approach is twofold. On one hand it aims to determine a classifier that minimizes the empirical risk, that is to encode the learning sequence as good as possible with respect to a certain architecture, and the other hand to improve the generalization capacity by minimizing the generalization error. In case of non-linear separable data the SVM is combined with kernel based technique which transforms the data in a linear separable data by mapping the initial data on to higher dimensional space of features. This mapping is performed in terms of special tailored kernels that allow to keep the computations at a reasonable complexity level.

The aim of the research reported in the paper is to obtain an alternative approach in using SVM in case of non-linearly separable data based on using the k-means algorithm instead of the standard kernel based approach. The potential of the proposed approach is pointed out on experimental basis in the final section of the paper. The tests were performed on data generated from multi-dimensional normal repartitions yielding to linearly separable and non-linearly separable samples respectively. The results encourage the research toward integrating the k-means technique in a SVM-based learning scheme.

1 Introduction

The Support Vector Machine (SVM) is a relatively new concept in machine learning and it was introduced by Vapnik ([10], [11]). In designing a classifier, two main problems have to be solved, on one hand the option concerning a suitable structure and on the other hand the selection of an algorithm for parameter estimation.

The algorithm for parameter estimation performs the optimization of a convenable selected cost function with respect to the empirical risk which is directly related to the representativeness of the available learning sequence. The choice of the structure is made such that to maximize the generalization capacity, that is to assure good performance in classifying new data coming from the same classes. In solving these problems one has to establish a balance between the accuracy in encoding the learning sequence and the generalization capacities because usually the over-fitting prevents the minimization of the empirical risk.

2 Supervised learning using SVM

Let us assume that the data is represented by examples coming from two categories or classes such that the true provenance class for each example is known. We refer such a collection of individuals as being a

supervised learning sequence, and it is represented as

$$\mathcal{S}=\left\{ (x_i,y_i) \mid x_i=(x_{i1},\ldots,x_{id})^T \in \mathbf{R}^d,\ y_i \in \{-1,1\},\ i=\overline{1,N} \right\}. \tag{1}$$

The values 1, -1 are taken as labels corresponding to the classes. We say the data is linearly separable if there exists a linear discriminant function $g:\mathbf{R}^d \longrightarrow \mathbf{R}$,

$$\forall\, x\,, \quad g(x)=b+w_1x_1+\ldots+w_dx_d\,, \tag{2}$$

where $x=(x_1,\ldots,x_d)\in\mathbf{R}^d$, such that for any $(x_i,y_i)\in\mathcal{S}$, $y_i g(x_i)>0$.

Denoting by $w=(w_1,\ldots,w_d)^T$ the vector whose entries are the coefficients of g, we say that $\mathcal{S}$ is separated without errors by the hyperplane

$$H_{w,b}\ :\quad w^Tx+b=0\,. \tag{3}$$

Obviously all examples coming from the class of label 1 belong to the positive semi-space, and all examples coming from the class of label -1 belong to the negative semi-space defined by $H_{w,b}$. For this reason, $H_{w,b}$ is called a solution of the separating problem.

In a SVM-based approach, the search for a solution $H_{w,b}$ is usually formulated as a constraint optimization problem on the objective function $\Phi(w)=\dfrac{1}{2}\|w\|^2$,

$$\begin{cases} \min \Phi(w) \\ y_i\left(w^Tx_i+b\right)\geq 1\,, \quad i=\overline{1,N}\,. \end{cases} \tag{4}$$

If w^* is a solution of (4), then H_{w^*,b^*} is called an optimal separating hyperplane, where the computation of w^* and b^* is carried out using the following algorithm

Algorithm $SVM1$ ([9])

`Input:` $\mathcal{S}=\left\{ (x_i,y_i) \mid x_i\in\mathbf{R}^d,\ y_i\in\{-1,1\},\ i=\overline{1,N}\right\}$

Step 1. `Compute the matrix` $D=(d_{ik})$ `of entries,` $d_{ik}=y_iy_k\,(x_i)^T\,x_k$, $i,k=\overline{1,N}$ `;`

Step 2. `Solve the constrained optimization problem`

$$\begin{cases} \alpha^*=\texttt{arg}\left(\max\limits_{\alpha\in\mathbf{R}^N}\left(\alpha^T\mathbf{1}-\dfrac{1}{2}\,\alpha^TD\alpha\right)\right), \\ \alpha_i\geq 0\,, \quad \forall\, 1\leq i\leq N\,, \\ \sum\limits_{i=1}^{N}\alpha_iy_i=0\,, \end{cases} \tag{5}$$

`If` $\alpha_i^*>0$ `then` x_i `is called the support vector.`

Step 3. `Select two support vectors` x_r, x_s `such that` $\alpha_r^*>0$, $\alpha_s^*>0$, $y_r=-1$, $y_s=1$.

Step 4. `Compute the parameters` w^*,b^* `of the optimal separating hyperplane, and the width of the separating area` $\rho(w^*,b^*)$,

$$\begin{cases} w^*=\sum\limits_{i=1}^{N}\alpha_i^*y_ix_i\,, \\ b^*=-\dfrac{1}{2}\,(w^*)^T\,(x_r+x_s)\,, \\ \rho(w^*,b^*)=\dfrac{2}{\|w^*\|} \end{cases} \tag{6}$$

`Output:` $w^*,b^*,\rho(w^*,b^*)$.

A linear separable sample is represented in figure 1a. The straight lines d_1, d_2, d_3 and d_4 are solutions for the separating problem of $\mathcal{S}$, d_4 corresponds to the optimal separating hyperplane. The examples placed at the minimum distance to the optimum separating hyperplane are the support vectors.

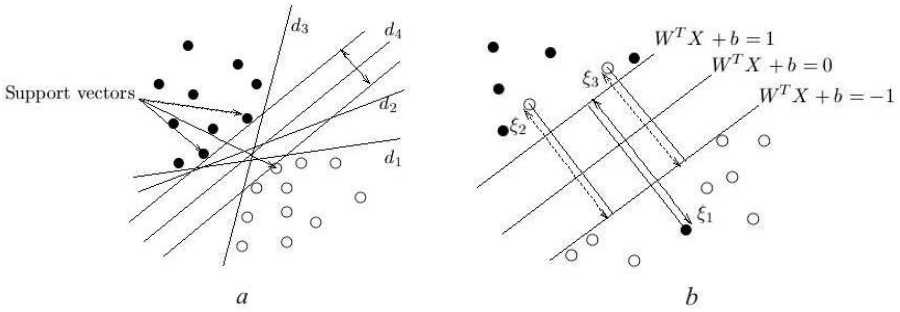


Figure 1: *a) Optimal separating hyperplane; b) Classification errors.*

In case of non-linearly separable samples the idea is to determine a separating hyperplane such that the number of misclassified examples is minimized.

The problem of designing the optimal hyperplane in case of non-linearly separable samples has been approached several ways. The approach introduced by Cortes and Vapnik ([3]) uses the error function

$$\Phi_\sigma(\xi) = \sum_{i=1}^{N} \xi_i^\sigma \,, \tag{7}$$

where the slack variables $\xi_i \,, 1 \leq i \leq N$, are taken as indicators for the classification errors (see figure 1*b*), and σ is a positive real number.

The optimality is expressed in terms of the objective function $\Phi : \mathbf{R}^d \times \mathbf{R}^N \longrightarrow [0, +\infty)$

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + c\, F\left(\sum_{i=1}^{N} \xi_i^\sigma\right) = \frac{1}{2} \sum_{j=1}^{n} w_j^2 + c\, F\left(\sum_{i=1}^{N} \xi_i^\sigma\right) \,, \tag{8}$$

where c is a given positive constant, $\xi = (\xi_1, \ldots, \xi_N)$, and F is a monotone convex function, $F(0) = 0$.

The idea is to compute a subset of $\mathcal{S}$, say $\{(x_{i_1}, y_{i_1}), \ldots, (x_{i_k}, y_{i_k})\}$, by minimizing $\Phi_\sigma(\xi)$, such that there exists an optimal hyperplane for $\mathcal{S} \setminus \{(x_{i_1}, y_{i_1}), \ldots, (y_{i_k}, y_{i_k})\}$. This optimal hyperplane is referred as the soft margin hyperplane ([3]).

The soft margin hyperplane is a solution of the constrained optimization problem

$$\begin{cases} \arg\left(\min\limits_{w \in \mathbf{R}^d} (\Phi(w, \xi))\right) \\ \\ y_i \left(w^T x_i + b\right) \geq 1 - \xi_i \,, \; \forall\, 1 \leq i \leq N \,, \\ \xi_i \geq 0 \,, \; \forall\, 1 \leq i \leq N \,, \end{cases} \tag{9}$$

The samples represented in figure 1*b*, correspond to the non-linearly separable case. The soft margin hyperplane, the separating area, and slack variables are indicated in figure 1*b*.

The computation of soft margin hyperplane is carried out by the following algorithm.

Algorithm $SVM2$ ([9])

`Input:` $\mathcal{S} = \left\{ (x_i, y_i) \mid x_i \in \mathbf{R}^n \,,\; y_i \in \{-1, 1\} \,,\; i = \overline{1, N} \right\}, \quad c \in (0, \infty)$;

Step 1. `Compute the matrix` $D = (d_{ik})$ `of entries,` $d_{ik} = y_i y_k \left(x_i\right)^T x_k \,,\; i, k = \overline{1, N}$;

Step 2. `Solve the constrained optimization problem`

$$\begin{cases} \alpha^* = \mathtt{arg}\left(\max\limits_{\alpha \in \mathbf{R}^N} \left(\alpha^T \mathbf{1} - \frac{1}{2}\, \alpha^T D \alpha - \frac{\left(\alpha_{max}\right)^2}{4\, c}\right)\right) \,, \\ \alpha_i \geq 0 \,, \quad \forall\, 1 \leq i \leq N \,, \\ \sum\limits_{i=1}^{N} \alpha_i Y_i = 0 \,, \end{cases} \tag{10}$$

where $\alpha_{max} = \max\{\alpha_1, \ldots, \alpha_N\}$

Step 3. Select two support vectors x_r, x_s such that $\alpha_r^* > 0$, $\alpha_s^* > 0$, $y_r = -1$, $y_s = 1$.

Step 4. Compute the parameters w^*, b^* of the soft margin hyperplane, and the width of the separating area $\rho(w^*, b^*)$, according to (6).

Output: $w^*, b^*, \rho(w^*, b^*)$.

3 Unsupervised learning (clustering) using the k-means method

Center-based clustering algorithms are very efficient for clustering large databases and high-dimensional databases. They have own objective functions which define how good a clustering solution is, the goal being to minimize the objective function. Clusters found by center-based algorithms have convex shapes and each cluster is represented by a center. The k-means algorithm introduced by MacQueen ([8]) was designed to cluster numerical data, each cluster having a center called the *mean*.

Let $\mathcal{D} = \{x_1, \ldots, x_N\} \subset \mathbf{R}^d$ be the data set, k a given positive integer, and $\mathcal{C}_1, \ldots, \mathcal{C}_k$ pairwise disjoint clusters of $\mathcal{D}$, that is, $\bigcup_{i=1}^{k} \mathcal{C}_i = \mathcal{D}$, $\mathcal{C}_i \cap \mathcal{C}_j$, $\forall\, i \neq j$. If we denote by $\mu(\mathcal{C}_i)$ the center of $\mathcal{C}_i$ then *the inertia momentum (error)* is expressed by

$$\varepsilon = \sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} d^2(x, \mu(\mathcal{C}_i)) \,, \tag{11}$$

where d is a convenable distance function on $\mathbf{R}^d$. In the following we take d as being the Euclidean distance on $\mathbf{R}^d$, $d(x, y) = \|x - y\|$.

The k-means methods proceeds, for a given initial k clusters, by allocating the remaining data to the nearest clusters and then repeatedly changing the membership of the clusters according to the error function until the error function does not change significantly or the membership of the clusters no longer changes.

The k-means algorithm can be treated as an optimization problem where the goal is to minimize a given objective function under certain constrains.

We denote by $\mathcal{C}$ the set of all subsets of $\mathbf{R}^d$ of cardinal k; any particular $Q = \{q_1, \ldots, q_k\} \in \mathcal{C}$ is called a set of possible centers.

A system of k pairwise disjoint clusters of $\mathcal{D}$ can be obviously represented in terms a matrix $W = (w_{il}) \in \mathcal{M}_{N \times k}(\mathbf{R})$ such that

$$\begin{aligned} &(i) \quad w_{il} \in \{0, 1\}\,, \quad i = \overline{1, N}\,,\ l = \overline{1, k} \\ &(ii) \quad \sum_{l=1}^{k} w_{il} = 1\,, \quad i = \overline{1, N}\,. \end{aligned} \tag{12}$$

The k-means algorithm can be formulated as the constrained optimization problem:

$$\begin{cases} \min\limits_{W \in \mathcal{M}_{N \times k}(\mathbf{R}),\ Q \in \mathcal{C}} P(W, Q) \\ w_{il} \in \{0, 1\}\,, \quad i = \overline{1, N}\,,\ l = \overline{1, k}\,, \\ \sum\limits_{l=1}^{k} w_{il} = 1\,, \quad i = \overline{1, N}\,, \end{cases} \tag{13}$$

where the objective function is defined as

$$P(W, Q) = \sum_{i=1}^{N} \sum_{l=1}^{k} w_{il} \|x_i - q_l\|^2 \,. \tag{14}$$

The problem (13) can be solved by decomposing it into two simpler problems P_1 and P_2, and then iteratively solving them, where

P_1. Fix $Q = \widehat{Q} \in \mathcal{C}$ and solve the reduced constrained optimization problem for $P\left(W, \widehat{Q}\right)$.

P_2. Fix $W = \widehat{W} \in \mathcal{M}_{N\times k}(\mathbf{R})$ and solve the reduced unconstrained optimization problem for $P\left(\widehat{W}, Q\right)$.

The solutions of these problems can be derived by straightforward computations, and they are given by the following theorems:

Theorem 1 *For any fixed $\widehat{Q} = \{\widehat{q}_1, \ldots, \widehat{q}_k\}$ a set of centers, the function $P\left(W, \widehat{Q}\right)$ is minimized if and only if W satisfies the conditions*

$$\begin{aligned} &w_{il} = 0 \Longleftrightarrow \|x_i - \widehat{q}_l\| > \min_{1\le t\le k} \|x_i - \widehat{q}_t\| \ , \\ &w_{il} = 1 \Longrightarrow \|x_i - \widehat{q}_l\| = \min_{1\le t\le k} \|x_i - \widehat{q}_t\| \ , \\ &\sum_{j=1}^{k} w_{ij} = 1 \, , \ \textit{for any} \ \ i = \overline{1,N} \, , \, l = \overline{1,k}. \end{aligned}$$

Note that in general, for any given $\widehat{Q}$ there are more solutions of $W^{(0)}$ type because any particular data x_i can be at minimum distance to more than one center of $\widehat{Q}$.

Theorem 2 *For any fixed $\widehat{W}$ satisfying the constrains of (13), the function $P\left(\widehat{W}, Q\right)$ is minimized if and only if*

$$q_l = \frac{\sum_{i=1}^{N} \widehat{w}_{il} x_i}{\sum_{i=1}^{N} \widehat{w}_{il}} \, , \quad l = \overline{1,k}.$$

The k-means algorithm viewed as an optimization process for solving (13) is as follows

The algorithm k-MOP

```
Input:  D - the data set,
        k - the pre-specified number of clusters,
        d - the data dimensionality,
        T - threshold on the maximum number of iterations.
Initializations:  Q^(0), t ⟵ 0
Solve P(W, Q^(0)) and get W^(0)
sw ⟵ false
repeat
        Ŵ ⟵ W^(t)
        solve P(Ŵ, Q) and get Q^(t+1)
        if P(Ŵ, Q^(t)) = P(Ŵ, Q^(t+1)) then
              sw ⟵ true
              output (Ŵ, Q^(t+1))
        else
              Q̂ ⟵ Q^(t+1)
              solve P(W^(t), Q̂) and get W^(t+1)
              if P(W^(t), Q̂) = P(W^(t+1), Q̂) then
                    sw ⟵ true
                    output (W^(t+1), Q̂, )
              endif
```

endif
$t \longleftarrow t+1$
until sw or $t > T$.

Note that the computational complexity of the algorithm k-**MOP** is $\mathcal{O}(Nkd)$ per iteration. The sequence of values $P\left(W^{(t)}, Q^{(t)}\right)$ where $W^{(t)}$, $Q^{(t)}$ are computed by k-**MOP** is strictly decreasing, therefore the algorithm converges to a local minimum of the objective function.

4 The combined separating technique based on SVM and the k-means algorithm

At first sight, it seems unreasonable to compare a supervised technique to an unsupervised one, mainly because they refer to totally different situations. On one hand the supervised techniques are applied in case the data set consists of correctly labeled objects, and on the other hand the unsupervised methods deal with unlabeled objects. However our point is to combine SVM and k-means algorithm, in order to obtain a new design of a linear classifier.

The aim of the experimental analysis is to evaluate the performance of the linear classifier resulted from the combination of the supervised SVM method and the 2-means algorithm.

Our method can be applied to whatever data, either linear separable or non-linear separable. Obviously in case of non-linear separable data the classification can not be performed without errors and in this case the number of misclassified examples is the most reasonable criterion for performance evaluation. Of a particular importance is the case of linear separable data, in this case the performance being evaluated in terms of both, misclassified examples and the generalization capacity expressed in terms of the width of separating area. In real live situations, usually is very difficult or even impossible to established whether the data represents a linear/non-linear separable set. In using the $SVM1$ approach we can identify which case the given data set belongs to. For linear separable data, $SVM1$ computes a separation hyperplane optimal from the point of view of the generalization capacity. In case of a non-linear separable data $SVM2$ computes a linear classifier that minimizes the number of misclassified examples. A series of developments are based on non-linear transforms represented be kernel functions whose range are high dimensional spaces. The increase of dimensionality and the convenable choice of the kernel aim to transform a non-linear separable problem into a linear separable one. The computation complexity corresponding to kernel-based approaches is significantly large therefore in case the performance of the algorithm $SVM1$ proves reasonable good it could be taken as an alternative approach of a kernel-based SVM. We perform a comparative analysis on data consisting of examples generated from two dimensional Gaussian distributions.

In case of a non-linear separable data set, using the k-means algorithm, we get a system of pairwise disjoint clusters together with the set of their centers representing a local minimum point of the criterion (13), the clusters being linear separable when $k = 2$. Consequently, the $SVM1$ algorithm computes a linear classifier that separates without errors the resulted clusters.

Our procedure is described as follows

Input: $\mathcal{S} = \left\{ (x_i, y_i) \mid x_i \in \mathbf{R}^n,\ y_i \in \{-1, 1\},\ i = \overline{1, N} \right\}$

Step 1. Compute the matrix $D = (d_{ik})$ of entries, $d_{ik} = y_i y_k \left(x_i\right)^T x_k,\ i, k = \overline{1, N}$;
$sh \longleftarrow true$

Step 2. If the constrained optimization problem (5) does not have solution then
$sh \longleftarrow false$
input $c \in (0, \infty)$, for hyperplane soft margin
Solve the constrained optimization problem (10)
endif

Step 3. Select x_r, x_s such that $\alpha_r^* > 0$, $\alpha_s^* > 0$, $y_r = -1$, $y_s = 1$;
Compute the parameters w^*, b^* of the separating hyperplane,

and the width of the separating area, $\rho(w^*, b^*)$ according to (6);
Compute the width of the separating area, $\rho(w^*, b^*) = \frac{2}{\|w^*\|}$;

Step 4. if *not sh* then
compute nr_err1 - the numbers of examples incorrect classified
compute $err1$ - error classification
endif

Step 5. The set $\mathcal{D} = \{x_i \mid x_i \in \mathbf{R}^d,\ i = \overline{1,N}\}$ is divided in two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ using 2-means, marked out with $y_i' = 1$ and $y_i' = -1$ respectively.

Step 6. Apply *algorithm SVM1* for
$\mathcal{S}' = \{(x_i, y_i') \mid x_i \in \mathbf{R}^d,\ y_i' \in \{-1, 1\},\ i = \overline{1,N}\}$
and obtain the parameters for optimal separating hyperplane: $w_1^*, b_1^*, \rho(w_1^*, b_1^*)$
compute nr_err2 - the numbers of examples incorrect classified by $2-means$
compute $err2$ - error classification after $2-means$

Output: $w^*, b^*, \rho(w^*, b^*)$, nr_err1, $err1$, $w_1^*, b_1^*, \rho(w_1^*, b_1^*)$, nr_err2, $err2$.

5 Comparative analysis and experimental results

The experimental analysis is based on a long series of tests performed on linear/non-linear separable simulated data of different volumes. The analysis aims to derive conclusions concerning:

1. The statistical properties (the empirical means, covariance matrices, eigenvalues, eigenvectors) of the clusters computed by the 2-means algorithm as compared to their counterparts corresponding to the true distributions they come from.
2. The comparison of the performances corresponding to the linear classifier resulted as a combination of SVM and the 2-means algorithm described in section 4 and $SVM2$ in terms of the empirical error.
3. The analysis concerning the influences of the samples sizes on the performance of the procedure described in section 4.
4. The quality of cluster characterization in terms of the principal directions given by a system of unit orthogonal eigenvectors of the sample covariance and empirical covariance matrices of the computed clusters. The analysis aimed to derive conclusions concerning the contributions of each principal direction, and for this reason, some tests were performed on data whose first principal component is strongly dominant, and when the principal directions are of the same importance respectively.

The tests were performed on data generated from normal two-dimensional distributions $\mathcal{N}(\mu_i, \Sigma_i)$, $i = 1, 2$ of volumes N_1 and N_2. The sample covariance matrices are denoted by $\widehat{\mu}_i, \widehat{\Sigma}_i$, $i = 1, 2$. The centers and the empirical covariance matrices corresponding to the clusters computed by the 2-means algorithm are denoted by $\overline{\mu}_i, \overline{\Sigma}_i$, $i = 1, 2$. We denote by Z_i , $\widehat{Z}_i$, $\overline{Z}_i$, $i = 1, 2$ orthogonal matrices having as columns unit eigenvector of Σ_i , $\widehat{\Sigma}_i$, $\overline{\Sigma}_i$, $i = 1, 2$ respectively.

Test 1:
$N_1 = N_2 = 50$, $\mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$.
The matrices Z_1 , Z_2 and their eigenvalues are

$\lambda_1^{(1)} = 0.25$, $\lambda_2^{(1)} = 1$, $Z_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\lambda_1^{(2)} = 0.5$, $\lambda_2^{(2)} = 0.5$, $Z_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

The set is non-linear separable and it is represented in figure $2i)a$. In this case we get

$\widehat{\mu}_1 = \begin{pmatrix} 0.92 \\ 1.00 \end{pmatrix}$, $\widehat{\Sigma}_1 = \begin{pmatrix} 0.85 & 0.086 \\ 0.08 & 0.25 \end{pmatrix}$, $\widehat{\mu}_2 = \begin{pmatrix} 1.98 \\ 2.87 \end{pmatrix}$, $\widehat{\Sigma}_2 = \begin{pmatrix} 0.44 & 0.09 \\ 0.09 & 0.63 \end{pmatrix}$.

the matrices $\widehat{Z}_1$, $\widehat{Z}_2$ and their eigenvalues being

$\widehat{\lambda}_1^{(1)}=0.24\,,\quad \widehat{\lambda}_2^{(1)}=0.86,\ \widehat{Z}_1=\begin{pmatrix} 0.14 & -0.98 \\ -0.98 & -0.14 \end{pmatrix},\quad \widehat{\lambda}_1^{(2)}=0.40\,,\quad \widehat{\lambda}_2^{(2)}=0.67,\ \widehat{Z}_2=\begin{pmatrix} -0.92 & 0.38 \\ 0.38 & 0.92 \end{pmatrix}.$

Using the $SVM2$ with $c = 70$ we get the classification error $class\_error = 14.70$, the number of misclassified samples $n\_errors = 13$ and the width of separating area is $\rho = 0.61$. The value of the error coefficient defined as the ratio of the number of misclassified samples and total volume of the data is $c\_error = 0.13\%$. The soft margin line d_1 is represented in figure $2i)b$.

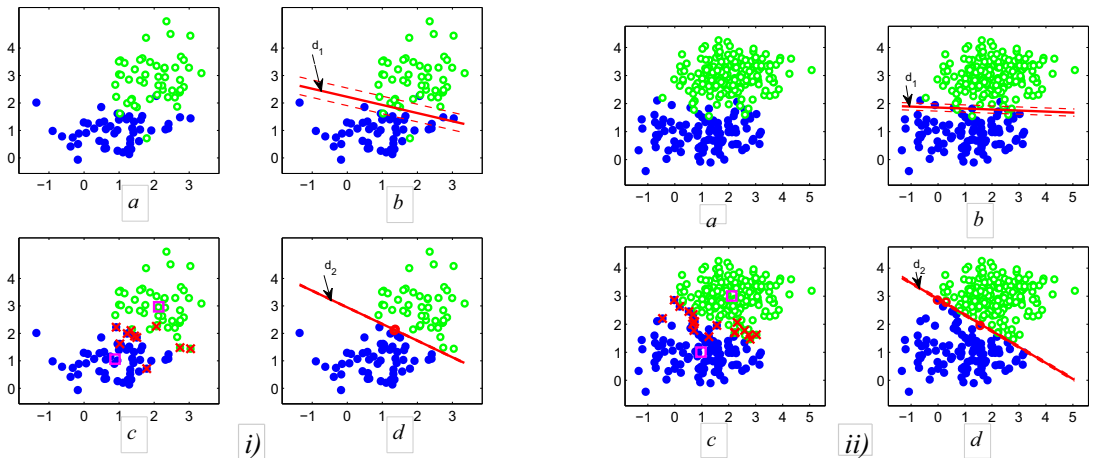


Figure 2: *i) The classification of the data set in test 1; ii) The classification of the data set in test 2.*

By applying the 2-means algorithm we get clusters whose empirical means and covariances are

$\overline{\mu}_1=\begin{pmatrix} 0.88 \\ 1.06 \end{pmatrix},\quad \overline{\Sigma}_1=\begin{pmatrix} 0.64 & 0.05 \\ 0.05 & 0.30 \end{pmatrix},\quad \overline{\mu}_2=\begin{pmatrix} 2.13 \\ 2.96 \end{pmatrix},\quad \overline{\Sigma}_2=\begin{pmatrix} 0.41 & -0.06 \\ -0.06 & 0.56 \end{pmatrix}.$

The matrices $\overline{Z}_1$, $\overline{Z}_2$ and their eigenvalues are

$\overline{\lambda}_1^{(1)}=0.29\,,\quad \overline{\lambda}_2^{(1)}=0.65,\ \overline{Z}_1=\begin{pmatrix} 0.14 & -0.98 \\ -0.98 & -0.14 \end{pmatrix},\quad \overline{\lambda}_1^{(2)}=0.39\,,\ \overline{\lambda}_2^{(2)}=0.58,\ \overline{Z}_2=\begin{pmatrix} -0.92 & -0.37 \\ -0.37 & 0.92 \end{pmatrix},$

the number of misclassified samples is 10 and the clusters are represented in figure $2i)c$.

Note that the computed centers and clusters are not influenced by the initial centers. In figure $2i)c$ are represented the clusters computed by the 2-means algorithm for randomly selected initial centers. The separating line d_2 resulted by applying the $SVM1$ algorithm to the data represented by the clusters computed by the 2-means algorithm is represented in figure $2i)d$.

Test 2:

$N_1=100\,,\quad N_2=200,\quad \mu_1=\begin{pmatrix} 1 \\ 1 \end{pmatrix},\quad \Sigma_1=\begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix},\quad \mu_2=\begin{pmatrix} 2 \\ 3 \end{pmatrix},\quad \Sigma_2=\begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix},$

$\lambda_1^{(1)}=0.25\,,\quad \lambda_2^{(1)}=1,\quad Z_1=\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\quad \lambda_1^{(2)}=0.25\,,\quad \lambda_2^{(2)}=1,\quad Z_2=\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$

$\widehat{\mu}_1=\begin{pmatrix} 1.12 \\ 0.92 \end{pmatrix},\quad \widehat{\Sigma}_1=\begin{pmatrix} 1.35 & 0.04 \\ 0.04 & 0.26 \end{pmatrix},\quad \widehat{\mu}_2=\begin{pmatrix} 2.01 \\ 3.00 \end{pmatrix},\quad \widehat{\Sigma}_2=\begin{pmatrix} 0.86 & 0.05 \\ 0.05 & 0.25 \end{pmatrix},$

$\widehat{\lambda}_1^{(1)}=0.26\,,\ \widehat{\lambda}_2^{(1)}=1.35,\ \widehat{Z}_1=\begin{pmatrix} 0.03 & -0.99 \\ -0.99 & -0.03 \end{pmatrix},\quad \widehat{\lambda}_1^{(2)}=0.25\,,\ \widehat{\lambda}_2^{(2)}=0.87,\ \widehat{Z}_2=\begin{pmatrix} 0.09 & -0.99 \\ -0.99 & -0.09 \end{pmatrix}.$

The data set is non-linear separable and it is represented in figure $2ii)a$. Applying the $SVM2$ for $c = 5$ we obtain the soft margin line d_1 represented in figure $2ii)b$ and $class\_error = 19.12$, $n\_errors = 13$, $\rho = 0.25$, $c\_error = 0.043\%$.

The clusters computed by the 2-means algorithm are represented in figure $2ii)c$ and their statistical characteristics are

$$\overline{\mu}_1=\begin{pmatrix}0.96\\1.00\end{pmatrix},\quad \overline{\Sigma}_1=\begin{pmatrix}1.19 & -0.10\\-0.10 & 0.38\end{pmatrix},\quad \overline{\mu}_2=\begin{pmatrix}2.10\\3.00\end{pmatrix},\quad \overline{\Sigma}_2=\begin{pmatrix}0.76 & -0.02\\-0.02 & 0.28\end{pmatrix},$$

$$\overline{\lambda}_1^{(1)}=0.37,\ \overline{\lambda}_2^{(1)}=1.20,\ \overline{Z}_1=\begin{pmatrix}-0.12 & -0.99\\-0.99 & 0.12\end{pmatrix},\quad \overline{\lambda}_1^{(2)}=0.27,\ \overline{\lambda}_2^{(2)}=0.76,\ \overline{Z}_2=\begin{pmatrix}-0.05 & -0.99\\-0.99 & 0.05\end{pmatrix}.$$

In this case the number of misclassified samples is 18. Note that the initial choice of the centers does not influence significantly the computed centers and clusters. For instance in figure $2ii)c$ are represented the resulted clusters in case of randomly selected initial centers.

The separating line d_2 computed by the algorithm $SVM1$ applied to the data represented by these clusters is represented in figure $2ii)d$.

Test 3:

$$N_1=N_2=50,\quad \mu_1=\begin{pmatrix}1\\1\end{pmatrix},\quad \Sigma_1=\begin{pmatrix}1 & 0\\0 & 0.25\end{pmatrix},\quad \mu_2=\begin{pmatrix}3\\4\end{pmatrix},\quad \Sigma_2=\begin{pmatrix}0.5 & 0\\0 & 0.5\end{pmatrix},$$

$$\lambda_1^{(1)}=0.25,\quad \lambda_2^{(1)}=1,\quad Z_1=\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix},\quad \lambda_1^{(2)}=0.5,\quad \lambda_2^{(2)}=0.5,\quad Z_2=\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix},$$

$$\widehat{\mu}_1=\begin{pmatrix}0.76\\1.00\end{pmatrix},\quad \widehat{\Sigma}_1=\begin{pmatrix}1.17 & -0.06\\-0.06 & 0.21\end{pmatrix},\quad \widehat{\mu}_2=\begin{pmatrix}2.87\\4.03\end{pmatrix},\quad \widehat{\Sigma}_2=\begin{pmatrix}0.56 & 0.00\\0.00 & 0.31\end{pmatrix},$$

$$\widehat{\lambda}_1^{(1)}=0.21,\ \widehat{\lambda}_2^{(1)}=1.18,\ \widehat{Z}_1=\begin{pmatrix}-0.07 & -0.99\\-0.99 & 0.07\end{pmatrix},\quad \widehat{\lambda}_1^{(2)}=0.31,\ \widehat{\lambda}_2^{(2)}=0.56,\ \widehat{Z}_2=\begin{pmatrix}0.03 & -0.99\\-0.99 & -0.03\end{pmatrix}.$$

The data set is linear separable and it is represented in figure $3i)a$. Applying the $SVM1$ we obtain the soft margin line d_1 represented in $3i)b$ and $\rho=1.19$.

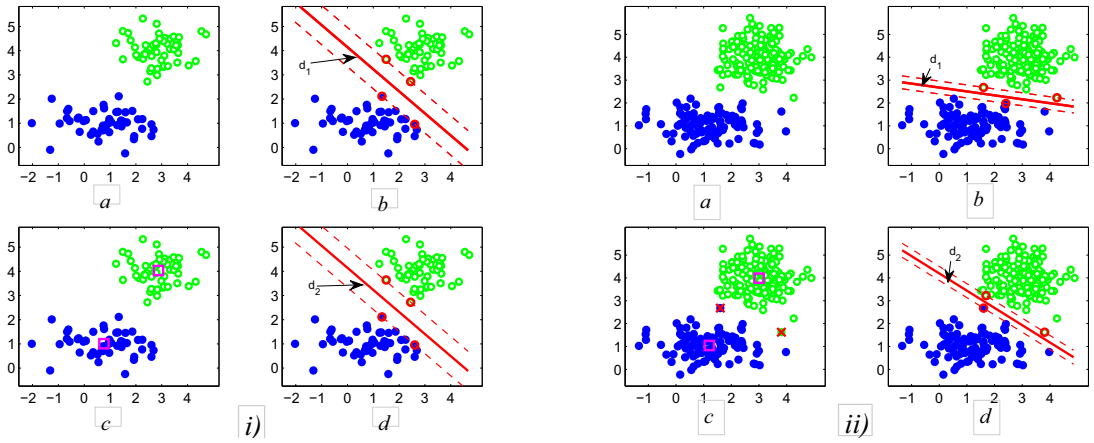


Figure 3: *i) The classification of the data set in test 3; ii) The classification of the data set in test 4.*

The clusters computed by the 2-means algorithm are represented in figure $3i)c$ and they are the same as in initial data set whatever the initial choice of the centers is. So, the statistical characteristics are

$$\overline{\mu}_1=\widehat{\mu}_1,\ \overline{\Sigma}_1=\widehat{\Sigma}_1,\ \overline{\mu}_2=\widehat{\mu}_2,\ ,\ \overline{\Sigma}_2=\widehat{\Sigma}_2,$$

$$\overline{\lambda}_1^{(1)}=\widehat{\lambda}_1^{(1)},\ \overline{\lambda}_2^{(1)}=\widehat{\lambda}_2^{(1)},\ \overline{Z}_1=\widehat{Z}_1,\ \overline{\lambda}_1^{(2)}=\widehat{\lambda}_1^{(2)},\ \overline{\lambda}_2^{(2)}=\widehat{\lambda}_2^{(2)},\ \overline{Z}_2=\widehat{Z}_2,$$

and the separating line d_2 computed by the algorithm $SVM1$ and represented in figure $3i)d$ coincides with d_1.

Test 4:

$N_1 = 100\,, \quad N_2 = 150, \quad \mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}.$

$\lambda_1^{(1)} = 0.25\,, \quad \lambda_2^{(1)} = 1, \quad Z_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \lambda_1^{(2)} = 0.5\,, \quad \lambda_2^{(2)} = 0.5, \quad Z_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$

$\widehat{\mu}_1 = \begin{pmatrix} 1.22 \\ 1.03 \end{pmatrix}, \quad \widehat{\Sigma}_1 = \begin{pmatrix} 1.04 & -0.03 \\ -0.03 & 0.24 \end{pmatrix}, \quad \widehat{\mu}_2 = \begin{pmatrix} 2.98 \\ 3.99 \end{pmatrix}, \quad \widehat{\Sigma}_2 = \begin{pmatrix} 0.48 & -0.01 \\ -0.01 & 0.43 \end{pmatrix}.$

$\widehat{\lambda}_1^{(1)} = 0.24\,, \ \widehat{\lambda}_2^{(1)} = 1.04, \ \widehat{Z}_1 = \begin{pmatrix} -0.04 & -0.99 \\ -0.99 & 0.04 \end{pmatrix}, \quad \widehat{\lambda}_1^{(2)} = 0.42\,, \ \widehat{\lambda}_2^{(2)} = 0.49, \ \widehat{Z}_2 = \begin{pmatrix} -0.27 & -0.9 \\ -0.96 & 0.27 \end{pmatrix}.$

The data set is linear separable and it is represented in figure $3ii)a$. Applying the $SVM1$ we obtain the soft margin line d_1 represented in $3ii)b$ and $\rho = 0.55$.

The clusters computed by the 2-means algorithm are represented in figure $3ii)c$ and their statistical characteristics are

$\overline{\mu}_1 = \begin{pmatrix} 1.20 \\ 1.04 \end{pmatrix}, \quad \overline{\Sigma}_1 = \begin{pmatrix} 0.98 & -0.04 \\ -0.04 & 0.26 \end{pmatrix}, \quad \overline{\mu}_2 = \begin{pmatrix} 3.00 \\ 3.98 \end{pmatrix}, \quad \overline{\Sigma}_2 = \begin{pmatrix} 0.48 & -0.04 \\ -0.04 & 0.45 \end{pmatrix},$

$\overline{\lambda}_1^{(1)} = 0.26\,, \ \overline{\lambda}_2^{(1)} = 0.98, \ \overline{Z}_1 = \begin{pmatrix} -0.05 & -0.99 \\ -0.99 & 0.05 \end{pmatrix}, \quad \overline{\lambda}_1^{(2)} = 0.42\,, \ \overline{\lambda}_2^{(2)} = 0.51, \ \overline{Z}_2 = \begin{pmatrix} -0.60 & -0.79 \\ -0.79 & 0.60 \end{pmatrix}.$

In this case the number of misclassified samples is 2 and the initial centers are randomly selected. The separating line d_2 computed by the algorithm $SVM1$ applied to the data represented by these clusters is represented in figure $3ii)d$.

References

[1] Abe, S., *Support Vector Machines for Pattern Classification*, Springer-Verlag, 2005.

[2] Burges, C.J.C., *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery, **2** (1998), 121-167.

[3] Cortes, C., Vapnik, V., *Support-vector networks*, Machine Learning, 20(3), 273-297, 1995.

[4] Cristianini, N., Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[5] Gan, G., Ma C., Wu, J., *Data Clustering: Theory, Algorithms and Applications*, SIAM, 2007.

[6] Gunn, S.R., *Support Vector Machines for Classification and Regression*, University of Southampton, Technical Report, 1998.

[7] State, L., Paraschiv-Munteanu, I., *Introducere in teoria statistică a recunoaşterii formelor*, Editura Universitaţii din Piteşti, 2009.

[8] MacQueen, J.B., *Some Methods for classification and Analysis of Multivariate Observations*, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 281-297, 1967.

[9] Paraschiv-Munteanu, I., *Support Vector Machine in solving pattern recognition tasks*, Scientific Bulletin, No.16/2009, University of Pitesti, ISSN 1453-116x.

[10] Vapnik, V.N., *The Nature of Statistical Learning Theory*, New York, Springer Verlag, 1995.

[11] Vapnik, V.N., *Statistical Learning Theory*, New York, Wiley-Interscience, 1998.

[12] Xu, R., Wunsch, D.C.II, *Clustering*, Wiley&Sons, 2009.

Luminiţa State
University of Piteşti
Faculty of Mathematics and Computer Science
1 Târgu din Vale St., Piteşti 110040
ROMANIA
E-mail: *lstate@clicknet.ro*

Iuliana Paraschiv-Munteanu
University of Bucharest
Faculty of Mathematics and Computer Science
14 Academiei St., Bucharest 010014
ROMANIA
E-mail: *pmiulia@fmi.unibuc.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Building a decision support system for long-term production planning using a distributed genetic algorithm

Florin Stoica

Abstract

This paper represents an approach of using a distributed genetic algorithm for generating an aggregative production plan to maximize the total profit of the firm. The described methodology provides a tool which assists the company management in finding an optimal long-term production plan. The entire system is composed by a business information system - a front-end for the database of the ERP system, a simulation model and a distributed genetic algorithm. The purpose of the integrated system is to help operative management personnel to take decisions with respect to long-term production planning. In order to improve the response time, the entire system is distributed across several network machines.

1 Introduction

Most real life optimization and scheduling problems are too complex to be solved completely. The complexity of real life problems often exceeds the ability of classic methods. In such cases decision-makers prepare and execute a set of scenarios on the simulation model and hope that at least one scenario will be good enough to be used as a production plan [15].

A long time goal for scheduling optimization research has been to find an approach that will lead to qualitative solutions in a relatively short computational time. The development of decision-making methodologies is currently headed in the direction of integration of simulation and search algorithms. This leads to a new approach, which successfully joins simulation and optimization. The proposed approach supports man-machine interaction in operational and long-term production planning [16].

A group of widely known meta-heuristic search algorithms are genetic algorithms (GA). Evolutionary algorithms are a very effective tool that enables solving complicated practical optimization problems [2]. An important characteristic of evolutionary algorithms is their simplicity and versatility. The principal advantage of GAs is their inherent ability to intelligently explore the solution space from many different points simultaneously enabling higher probability for locating global optimum without having to analyze all possible solutions available and without requiring derivatives (or numerical approximations) or other auxiliary knowledge. Their main drawback is a long calculation time. A solution to this problem is parallelizing genetic algorithms.

With computer imitation of simplified and idealized evolution, an individual solution-chromosome represents a possible solution to our problem. Chromosome fitness is calculated with a fitness function. After being evaluated with a fitness function, each chromosome in population receives its fitness value.

The optimization is based on a genetic algorithm which uses a new co-mutation operator called LR-Mijn, capable of operating on a set of adjacent bits in one single step.

In present, there is a major interest in design of powerful mutation operators, in order to solve practical problems which can not be efficiently resolved using standard genetic operators. These new operators are called co-mutation operators. In [7] was presented a co-mutation operator called Mijn, capable of operating on a set of adjacent bits in one single step. In [12] we introduced and studied a new co-mutation operator which we denoted by LR-Mijn and we proved that it offers superior performances than Mijn operator.

The paper is organized as follows. In Section 2 we make a brief presentation of the architecture of our decision-support system, called DGA-SIM. We also present the basic idea of evolutionary method adopted for optimization process. In section 3 is introduced the co-mutation operator LR-Mijn. The evolutionary algorithm based on the LR-Mijn operator, used within the decision-support system is presented in section 4. Section 5 contains the description of the simulation model and its integration in the DGA-SIM system. Our solution for parallelizing the genetic algorithm is also presented in section 5. Conclusions and further directions of study can be found in section 6.

2 The architecture of the decision-support system

Companies need to be flexible to compete for the market share and adapt to market demands by offering competitive prices and quality. This calls for a wide assortment of products or product types, small production costs, high productivity etc. Simulation is a strong interactive tool that helps decision-makers improves the efficiency of enterprise actions. The ability of simulation to show a real process on the computer with the consideration of uncertainty is a big advantage when analyzing system behavior in complex situations [15].

Our system is composed by a business information system - a front-end for the database of the ERP system, a simulation model and a distributed genetic algorithm, and is called in the following the DGA-SIM system.

A simulation model will be used for fitness function computation of genetic algorithm results, as well as for visual representation of qualitative evaluation of a chosen production plan following genetic algorithm optimization.

The value of the fitness function is computed by the simulation model, which uses the respective chromosome as input data. The value returned from the simulation model is used to evaluate that chromosome, which encodes a possible production plan.

By applying the genetic operations on the members of a population of decisional rules at the moment t, it results a new population of rules that shall be used at the moment t+1. The population from the initial moment, $t = 0$ is generated randomly and the genetic operations are applied iteratively until the moment T (at which the condition for stopping the algorithm is accomplished) [16].

The above iterative process may be interpreted economically as it follows. The evolution process has as objective the finding of the successful individuals. The binary strings of these individuals (chromosomes) with high fitness values (a high profit) will be the basis for building a new generation (population). The strings with lower fitness values, which represent decisions to produce with a low profit, find few successors (or none) in the following generation.

The purpose of the integrated system is to aid operative management personnel in production planning and marketing strategies (incorporated in simulation model). The main advantage of the presented system is to enhance man-machine interaction in production planning, since the computer is able to produce several acceptable solutions using the given data and a set of criteria [16].

After completion of the optimization process, the most suitable production plans are simulated on the visual model of the system. Using chosen parameters and according to defined criteria, the decision-maker is motivated to search for results which will have the most advantageous influence on the whole production process. The result, which is selected after simulation on the visual simulation model, becomes the proposed production plan. Then the user modifies it, if necessary.

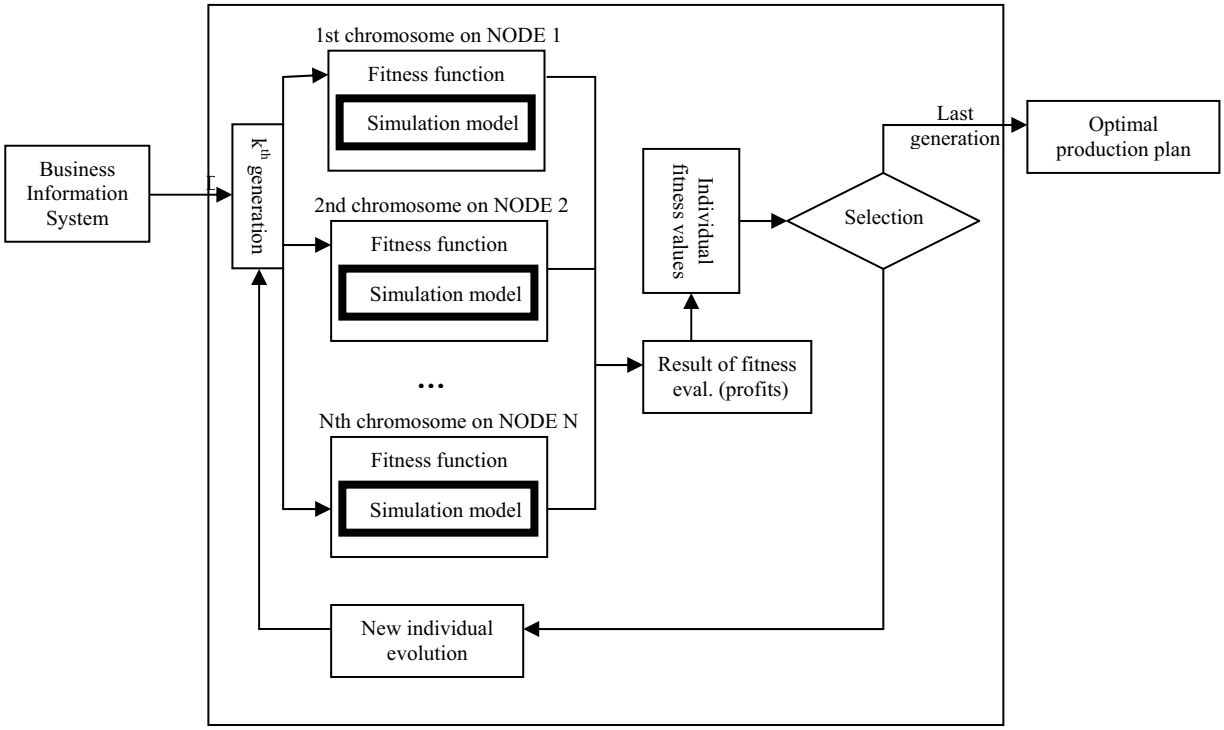


Fig. 1 The architecture of the decision-support system

Every chromosome codes a possible production plan. The quality of a chromosome is represented by the amount of profit which result from the simulation model if it receives as input data the production plan coded in that chromosome.

Parallelizing genetic algorithm is easy to be implemented by distributing one chromosome to one computer such that simulation model with different possible production plans can be run simultaneously at different computers.

The user interface of the simulation model of the DGA-SIM system is based on Microsoft EXCEL. The calculations for a specific production plan coded within a chromosome are performed in EXCEL, and result is returned in the GA as the evaluation of the respective production plan (the fitness value).

3 The LR-M_{ijn} operator

In this section we define the co-mutation operator called LR-M_{ijn}. Our LR-M_{ijn} operator finds the longest sequence of σ_p elements, situated *in the left or in the right* of the position p. If the longest sequence is in the left of p, the LR-M_{ijn} behaves as M_{ijn}, otherwise the LR-M_{ijn} will operate on the set of bits starting from p and going to the *right*.

Let us consider a generic alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_s\}$ composed by s ≥2 different symbols. The set of all sequences of length *l* over the alphabet $\mathcal{A}$ will be denoted with $\Sigma = \mathcal{A}^l$.

In the following we shall denote with σ a generic string, and $\sigma = \sigma_{l-1}\ldots\sigma_0 \in \Sigma = \mathcal{A}^l$, where $\sigma_q \in \mathcal{A}$ $\forall$ q ∈ {0, …, *l*-1}. Through σ(q,i) we denote that on position q within the sequence σ there is the symbol a_i of the alphabet $\mathcal{A}$., $\sigma^z_{p,j}$ denotes the presence of z symbols a_j within the sequence σ, starting from the position p and going left and $\underset{left,m}{\overset{right,n}{\sigma(p,i)}}$ specify the presence of symbol a_i on position p within

the sequence σ, between *right* symbols a_n on the right and *left* symbols a_m on the left. We suppose that $\sigma = \sigma(l\text{-}1)\ldots \sigma(p+left+1,m)\ \overset{right,i}{\underset{left,i}{\sigma(p,i)}}\ \sigma(p\text{-}right\text{-}1,n)\ldots \sigma(0)$.

The M_{ijn} operator is the mutation operator defined in [7]:

$M_{ijn} : \sigma \in \Sigma, p \in \{0, \ldots, l\text{-}1\} \rightarrow \sigma' \in \Sigma' \subset \Sigma$, where p is randomly chosen

$$\sigma = \sigma_{l-1}\ldots\sigma_{p+n}\sigma_{p+n-1,i}\,\sigma_{p,j}^{n-1}\,\sigma_{p-1}\ldots\sigma_0 \xrightarrow{M_{ijn}} \sigma' = \sigma_{l-1}\ldots\sigma_{p+n}\sigma_{p+n-1,j}\,\sigma_{p,i}^{n-1}\,\sigma_{p-1}\ldots\sigma_0, \quad \text{(i)}$$

for $n < l - p + 1$ and

$$(\sigma = \sigma_{p,j}^{l-p}\,\sigma_{p-1}\ldots\sigma_0 \xrightarrow{M_{ijn}} \sigma' = \sigma_{p,k}^{l-p}\,\sigma_{p-1}\ldots\sigma_0, \quad \text{(ii)}$$

for $n = l - p + 1$, with $a_k \neq a_j$ randomly chosen in $\mathcal{A}$.

In [12], we introduced and study the properties of LR-M_{ijn} co-mutation operator.

Definition 1 *Formally, the LR-M_{ijn} operator is defined as follows*:

(i) If $p \neq right$ and $p \neq l - left - 1$,

$$\text{LR-Mijn}(\sigma) = \begin{cases} \sigma_{left} = \sigma(l-1)\ldots\sigma(p+left+1,i)\ \overset{right,i}{\underset{left,m}{\sigma(p,m)}} \\ \qquad \sigma(p-right-1,n)\ldots\sigma(0) \quad for\ left > right \\ \sigma_{right} = \sigma(l-1)\ldots\sigma(p+left+1,m)\ \overset{right,n}{\underset{left,i}{\sigma(p,n)}} \\ \qquad \sigma(p-right-1,i)\ldots\sigma(0) \quad for\ left < right \\ \sigma_{right}\ or\ \sigma_{left}\ \ for\ left = right,\ with\ probability\ 0.5 \end{cases}$$

(ii) If $p = right$ and $p \neq l - left - 1$,

$\sigma = \sigma(l\text{-}1)\ldots \sigma(p+left+1,m)\ \overset{right,i}{\underset{left,i}{\sigma(p,i)}}$ and $\text{LR-Mijn}(\sigma) = \sigma(l\text{-}1)\ldots \sigma(p+left+1,m)\ \overset{right,k}{\underset{left,i}{\sigma(p,k)}}$,

where $k \neq i$ (randomly chosen).

(iii) If $p \neq right$ and $p = l - left - 1$,

$\sigma = \overset{right,i}{\underset{left,i}{\sigma(p,i)}}\ \sigma(p\text{-}right\text{-}1,n)\ldots \sigma(0)$ and

$\text{LR-Mijn}(\sigma) = \overset{right,i}{\underset{left,k}{\sigma(p,k)}}\ \sigma(p\text{-}right\text{-}1,n)\ldots \sigma(0)$, where $k \neq i$ (randomly chosen).

(iv) If $p = right$ and $p = l - left - 1$, $\sigma = \overset{right,i}{\underset{left,i}{\sigma(p,i)}}$

$$\text{LR-Mijn}(\sigma) = \begin{cases} \sigma_{left} = \overset{right,i}{\underset{left,k}{\sigma(p,k)}},\ for\ left > right,\ where\ \ k \neq i \\ \sigma_{right} = \overset{right,k}{\underset{left,i}{\sigma(p,k)}},\ for\ left < right,\ where\ \ k \neq i \\ \sigma_{right}\ or\ \sigma_{left}\ \ for\ left = right,\ with\ probability\ 0.5 \end{cases}$$

As an example, let us consider the binary case, the string σ = 11110000 and the randomly chosen application point p = 2. In this case, $\sigma_2 = 0$, so we have to find the longest sequence of 0 within string σ, starting from position p. This sequence goes to the right, and because we have reached the end of

the string, and no occurrence of 1 has been met, the new string obtained after the application of LR-M_{ijn} is 11110111.

The commutation operator LR-M_{ijn} allows long jumps [8], thus the search can reach very far points from where the search currently is. We proved in [12] that the LR-M_{ijn} operator performs more long jumps than M_{ijn}, which leads to a better convergence of an evolutionary algorithm based on the LR-M_{ijn} in comparison with an algorithm based on the M_{ijn} operator.

In the following implementation, we will consider that $\mathcal{A}$ is the binary alphabet, $\mathcal{A} = \{0, 1\}$.

4 The evolutionary algorithm based on LR-M_{ijn} operator

The basic scheme for our algorithm, called in the following LR-MEA, is described as follows:

```
Procedure LR-MEA
 begin
  t = 0
  Initialize randomly population P(t) with P elements;
  Evaluate P (t) by using fitness function;
  while not Terminated
    for j = 1 to P-1 do
      - select randomly one element among the best T% from P(t);
      - mutate it using LR-Mijn;
      - evaluate the obtained offspring;
      - insert it into P'(t).
    end for
    Choose the best element from P(t) and Insert it into P'(t)
    P(t+1) = P'(t)
    t = t + 1
  end while
 end
```

5 The simulation model

The simulation model is implemented as an Excel application. It is based on real data provided by the Business Information System but also on forecasted data (e.g. sales quantities and values for the following months). The model take account of many dates and variables: sales (values and quantities), raw material costs, discounts, packaging costs, direct & indirect labor costs, energy cost, depreciation from the rate of exchange, warehousing and logistic costs, transport costs, advertising & promotions, etc. The main workbook contains a few sheets, a VBA module, and results are synthesized in the pivot table described in Figure 2.

The role of the simulation model in the genetic algorithm is presented in the Figure 3.

In fact, the simulation model represents the implementation of the fitness function of the genetic algorithm, needed in the procedure LR-MEA to evaluate each member (chromosome) of the population.

The probability to select a certain chromosome (possible production plan) in the next generation is related with its performance in simulated conditions (the profit provided by the simulation model). That profit determines the fitness value of the respective possible solution of our optimization problem.

Because the DGA-SIM system is implemented in Java as main language, was necessary a bridge between Java code and the simulation model, implemented in Excel. Our choice for this purpose was JExcelAPI (http://jexcelapi.sourceforge.net/), a mature, open source java API enabling developers to read, write, and modify Excel spreadsheets dynamically [14].

Parallelizing genetic algorithm was implemented by distributing one chromosome to one computer such that simulation model with different possible production plans can be run simultaneously at different computers.

	A	B
1		Grand Total
2		
3	Values	
4	Kg	12,567,833
5	Sku	140,244,000
6	Sales	328,992,355
7	Discounts %	10.50%
8	Turnover	190,458,000
9	Total Costs	135,200,278
10	Direct Labour	3,189,330
11	...	...
12	Directs Costs	82,000,388
13	Energy gas and water	2,000,477
14	Indirect Labour	6,300,212
15	Depreciation	1,903,235
16	Warehousing	1,589,345
17	Logistic platforms	560,233
18	Advertising & Promotions	9,280,023
19	Advertising & Promotions %	5.05%
20	Overheads	14,934,770
21	Overheads %	8.93%
22	Profit	8,950,168

Fig. 2 A pivot table from the simulation model [16]

The implementation of the distributed genetic algorithm is based on Java Remote Method Invocation (RMI). Java RMI enables the programmer to create distributed Java technology-based to Java technology-based applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, hosted by different computers. RMI provides a standard remote communication between programs written in the Java programming language [17].

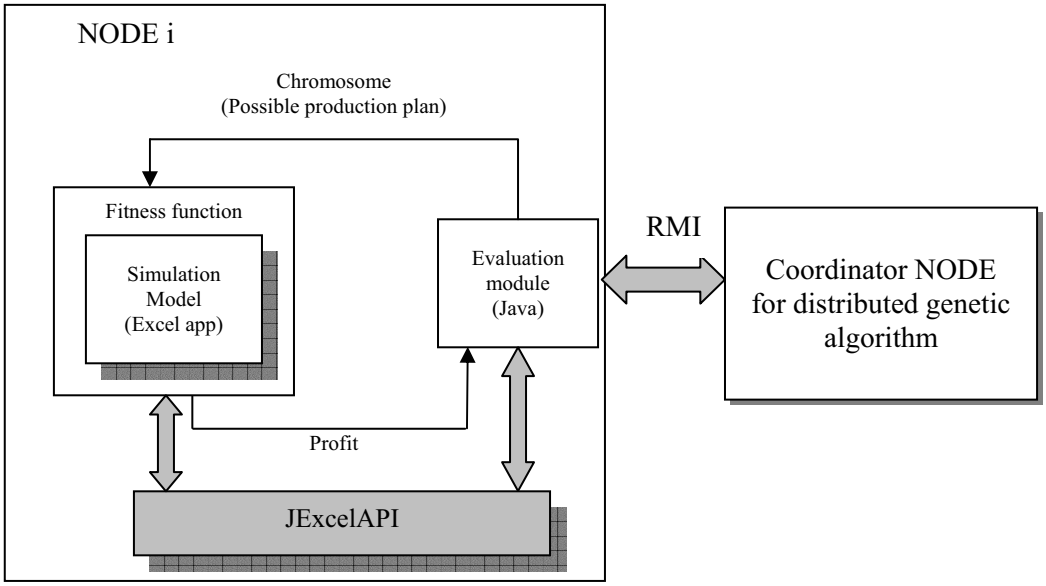


Fig. 3 Implementation of the fitness function through the simulation model

The parallel code runs in a network of 20 computers to demonstrate the efficiency of the distributed genetic algorithm to our problem. Therefore we are using 20 chromosomes (because of 20 network computers) for 20 production plans simulated/evaluated in parallel in one generation.

The role of the coordinator node is to centralize the results (profits) provided by each individual node, in order to evaluate in the following step the whole population. Then, the best production plans (chromosomes) are selected for the next generation.

6 Conclusions and further directions of study

The purpose of the DGA-SIM integrated system is to aid operative management personnel in production planning and marketing strategies (incorporated in simulation model). The main advantage of the presented system is to enhance man-machine interaction in production planning, since the computer is able to produce several acceptable schedules using the given data and a set of criteria.

The DGA-SIM system is currently under evaluation in a big company from Sibiu, Romania, which was interested in its acquisition.

As a further direction of study we want to compare the results obtained by using different genetic operators and to evaluate real codifications of variables, instead of current binary one.

Another improvement of our study is represented by designing of appropriate simulation models for multi-criteria optimization with genetic algorithms.

Acknowledgement: This paper is founded from the ULBS research project 164/2009.

References

[1] Jaber A. Q, Hidehiko Y., F., Ramli R., Machine Learning in Production Systems Design Using Genetic Algorithms, *International Journal of Computational Intelligence*, No. 4, 2008, pp. 72-79.

[2] Kofjač D., Kljajić M., Application of Genetic Algorithms and Visual Simulation in a Real-Case Production Optimization, *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Issue 12, Volume 3, December 2008, pp. 992-1001.

[3] Radhakrishnan P., Prasad V. M., Gopalan M.R., Optimizing Inventory Using Genetic Algorithm for Efficient Supply Chain Management, *Journal of Computer Science 5 (3)*, 2009, pp. 233 - 241.

[4] Lo Chih-Yao, Advance of Dynamic Production-Inventory Strategy for Multiple Policies Using Genetic Algorithm, *Information Technology Journal 7 (4)*, 2008, pp. 647-653.

[5] De Falco I., An introduction to Evolutionary Algorithms and their application to the Aerofoil Design Problem – *Part I: the Algorithms*, *von Karman Lecture Series on Fluid Dynamics*, Bruxelles, April 1997

[6] De Falco I., Del Balio R, Della Cioppa A., Tarantino E., A Comparative Analysis of Evolutionary Algorithms for Function Optimisation, *Research Institute on Parallel Information Systems*, National Research Council of Italy, 1998

[7] De Falco I, A. Iazzetta, A. Della Cioppa, Tarantino E., The Effectiveness of Co-mutation in Evolutionary Algorithms: the M_{ijn} operator, *Research Institute on Parallel Information Systems*, National Research Council of Italy, 2000

[8] De Falco I., Iazzetta A., Della Cioppa A., Tarantino E., M_{ijn} Mutation Operator for Aerofoil Design Optimisation, *Research Institute on Parallel Information Systems*, National Research Council of Italy, 2001

[9] Chen J., Using Genetic Algorithms to Solve a Production-Inventory Model, *International Journal of Business and Management*, Vol. 2, No. 2, 2007, pp. 38-41.

[10] Krishnakumar K., Goldberg D., Control system optimization using genetic algorithm, *Journal of Guidance, Control, and Dynamics*, no. 15(3), 1992, pp. 735-740.

[11] Zhu X., Huang Y., Doyle J., Genetic algorithms and simulated annealing for robustness analysis, *Proceedings of the American Control Conference*, Albuquerque, New Mexico, 1997, pp. 3756-3760.

[12] Stoica F., Simian D., Simian C., *A new co-mutation genetic operator*, Proceedings of the 9th WSEAS International Conference on Evolutionary Computing, Sofia, Bulgaria, May 2008, pp. 76-81.

[13] Vapnik V., *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

[14] Java Excel API, http://jexcelapi.sourceforge.net

[15] Kljajić M., Bernik I., Škraba A., Leskovar R., Integral simulation approach to decision assessment in enterprises, *Shaping future with simulation: proceedings of the 4th International Eurosim 2001 Congress*, Delft University of Technology, 2001.

[16]Stoica F., Cacovean L., Using genetic algorithms and simulation as decision support in marketing strategies and long-term production planning, *Proceedings of the 9th WSEAS International Conference on SIMULATION, MODELLING and OPTIMIZATION*, Budapest, Hungary, September 3-5, 2009
[17]Remote Method Invocation Home, http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp

Florin Stoica
University "Lucian Blaga" of Sibiu
Faculty of Sciences
Computer Science Department
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
E-mail: *florin.stoica@ulbsibiu.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

A Formal Approach for OOD Metrics Definition

Camelia Şerban

Abstract

The lack of standard formalism for defining software metrics has led to ambiguity in their definitions which hampers their applicability, comparison and implementation. As a result in this direction, this paper proposes a conceptual framework for object oriented design metrics definition, based on sets and relations theories and presents some concrete examples.

1 Introduction

Software metrics are important in many areas of software engineering, for example assessing software quality or estimating the cost and effort of developing software. Many metrics have been proposed and new metrics continue to appear in the literature regularly.

In spite of the existence of many metrics, problems often arise from the formality degree used to define them. When metrics are informally expressed, using natural language, people using metrics can interpret them in several ways. Two distinct teams can obtain completely different results when applying a particular metric to the same system. For example, Li and Henry [1] define their metric NOM (number of methods) as "the number of local methods" of a class. Unfortunately they do not define the term "local method". From the context it can be guessed that inherited methods are not counted. But what about class methods? Redefined methods? Is the method visibility (public etc.) considered? Many questions are left unanswered, nevertheless the authors validate their metric as a predictor for maintenance effort.

On the other extreme, when metrics are defined using some kind of formalism (e.g. OCL language), the majority of software designers may not have the required background to understand the complex expressions that are used.

In order to overcome the above mentioned problems, the current paper proposes a conceptual framework for object oriented design (OOD) metrics definition, based on sets and relations theories. The metrics that can be expressed using our framework have definitions that are *unambiguous*, *simple* and *language independent*. They are unambiguous, as their definition relies on the accurate formalism of sets and relations, knowledge considered known since the first stages of our studies. They are simple, as their computation is mainly achieved using lists traversal techniques.

We discuss the proposed approach as follows. Section 2 reviews related works in the area of OOD metrics definition. Section 3 presents our approach within the corresponding context (defined in terms of set and relations) that defines a meta-model where the entities of interest, their properties and their interrelationships are formally specified. As a proof of concept, Section 4 defines two OOD metrics, Coupling Between Objects (CBO) and Weighted Methods per Class (WMC). Finally, Section 5 summarizes the contributions of this work and outlines directions for further research.

2 Related work

Several authors have attempted to address the problem of imprecise metric definitions. Briand et al. propose two extensive frameworks for software measurement, one for measuring coupling and the other for measuring cohesion in object-oriented systems [8, 9]. While this framework allows for the unambiguous definition of coupling and cohesion metrics, new frameworks must be developed for other types of metrics. Therefore their scalability is hampered.

Another approach put forward by Reiβing involved the proposal of a formal model on which to base metric definitions [7]. This model is called ODEM (Object-oriented DEsign Model) and consists of an abstraction layer built upon the UML meta-model. However, this model can only be used for the definition of design metrics and does not solve the ambiguity problem as the abstraction layer consists of natural language expressions.

Baroni et al. propose the use of the OCL and the UML meta-model as a mechanism for defining UML-based metrics [5]. They have built a library called FLAME (Formal Library for Aiding Metrics Extraction) [6] which is a library of metric definitions formulated as OCL expressions over the UML 1.3 meta-model. We believe that this approach provides a useful mechanism for the precise definition but the majority of software designers may not have the required background to understand the OCL formalism.

In this paper, we start from the approach proposed by Briand et al. We extend this approach and integrate it within a corresponding meta-model for object oriented design. The design entities, their properties and relations are formally specify, based on sets and relations theories, knowledge considered known since the first stages of our studies. The metrics that can be expressed using our meta-model have definitions that are *unambiguous*, *simple* and *language independent*. They are unambiguous, as their definition relies on the accurate formalism of sets and relations. They are simple, as their computation is mainly achieved using lists traversal techniques.

3 A conceptual framework for OOD metrics definition. Formal approach

Before embarking in any measurement activity, we need to define the domain of our measurement; this means its constituent elements, their properties and the relationships that exist between them. These components define a meta-model of the analyzed system [3], meta-model that provides our conceptual framework for OOD metrics definitions.

Definition 1 ***(A meta-model for object-oriented design)***
The 3-tuple $D = (E, Prop(E), Rel(E))$ is called a metamodel for object oriented design corresponding to a software system S, where

- *E represents the set of design entities;*
- *$Prop(E)$ defines the properties of the elements from E;*
- *$Rel(E)$ represents the relations between the design entities.*

The components $E, Prop(E), Rel(E)$ will be specified in the following, using terms of sets and relations.

3.1 Design entities

Let $E = \{e_1, e_2, ..., e_{noE}\}$ be the set of *design entities* of the software system S, where $e_i (1 \leq i \leq noE)$ can be a *class*, a *method* from a class, an *attribute* from a class, a *parameter* from a method or a *local variable* declared in the implementation of a method. We also consider that:

- $Class(E) = \{C_1, C_2, ..., C_{noC}\}$ is a set of entities that are classes, $Class(E) \subset E$, $noC = |Class(E)|$, noC-number of classes;

- each class has a set of methods and attributes, therefore $(\forall)i, 1 \leq i \leq noC$:
  $Meth(C_i) = \left\{m_{i1}, m_{i2}, ..., m_{i(noM_{C_i})}\right\}$ is the set of methods of class C_i, $1 \leq noM_{C_i} \leq noC$, $noM_{C_i} = |Meth(C_i)|$, noM_{C_i}-number of methods of class C_i;

  $Attr(C_i) = \left\{a_{i1}, a_{i2}, ..., a_{i(noA_{C_i})}\right\}$ is the set of attributes of class C_i, $1 \leq noA_{C_i} \leq noC$, $noA_{C_i} = |Attr(C_i)|$, noM_{C_i}-number of attributes of class C_i ;

- $AllMeth(E) = \bigcup_{i=1}^{noC} Meth(C_i)$ is a set of methods from all classes of the software system S, $AllMeth(E) \subset E$, $noM = |AllMeth(E)|$, noM-the number of all methods;

- $AllAttr(E) = \bigcup_{i=1}^{noC} Attr(C_i)$ is the set of attributes from all classes of the software system S, $AllAttr(E) \subset E$, $noA = |AllAttr(E)|$, noM-the number of all attributes;

- each method has a set of parameters and local variables, $(\forall)i, 1 \leq i \leq noC$, $(\forall)j, 1 \leq j \leq noM_{C_i}$:

  $$Param(m_{ij}) = \{p_{ij1}, p_{ij2}, ..., p_{ij(noP_{m_{ij}})}\}$$

  is the set of parameters of method m_{ij}, $noP_{m_{ij}} = |Param(m_{ij})|$, $noP_{m_{ij}}$-the number of parameters of method m_{ij} and

  $$LocVar(m_{ij}) = \{lv_{ij1}, lv_{ij2}, ..., lv_{ij(noLV_{m_{ij}})}\}$$

  is the set of local variables of method m_{ij}, $noLV_{m_{ij}} = |LocVar(m_{ij})|$, $noLV_{m_{ij}}$-the number of local variables of method m_{ij};

- $AllParam(E) = \bigcup_{i=1}^{noC} \bigcup_{j=1}^{noM_{C_i}} Param(m_{ij})$, $Param(E) \subset E$, $noP = |AllParam(E)|$, noP-the number of all parameters;

- $AllLocVar(E) = \bigcup_{i=1}^{l} \bigcup_{j=1}^{noM_{C_i}} LocVar(m_{ij})$, $LocVar(E) \subset E$, $noLV = |AllLocVar(E)|$, $noLV$-the number of all local variables;;

Based on the above notations, the design entities set is defined as in equation 1:

$$E = Class(E) \bigcup AllMeth(E) \bigcup AllAttr(E) \bigcup AllParam(E) \bigcup AllLocVar(E) \tag{1}$$

3.2 Properties of design entities

As we have mentioned before, the second element of our meta-model is the set of properties of the design entities, denoted by $Prop(E)$. Because, in this approach we will refer to five types of design entities (classes, methods, attributes, parameters, local variables), each type having its own set of properties, we define a model in order to specify the properties of entities of a generic type T. Then, we apply this model for our concrete types of design entities enumerated above.

3.2.1 Properties of generic type entities. Formal specification.

- Let us consider a set of entities $A = \{a_1, a_2, ..., a_k\}$ of a generic type T and a set of properties defined on this type, $Prop_T = \{P_1, P_2, ...P_{noP_T}\}$;
- Each property P_i from $Prop(T)$, $1 \leq i \leq noP_T$, has a set of values $P_i = \{v_{i1}, v_{i2}, ...v_{iNoV_i}\}$.
- Each entity a_i from A, $1 \leq i \leq k$, will be assigned to a value from the cartesian product, $P_1 \times P_2 \times ... \times P_{noP_T}$, assignment that will be fomally expressed as follows:

$$PropVal_T : A \rightarrow P_1 \times P_2 \times ... \times P_{noP_T}$$

With the above mentioned notations and remarks, we will introduce the following definitions:

Definition 2 *A component v_i of the noP_T- dimensional vector*

$$PropVal_T(a_j) = (v_1, v_2, ..., v_{noP_T})$$

is called the value of property P_i corresponding to entity a_j. In this approach this component will be refered as $v_i = a_j.P_i$.

Definition 3 *The set $PropVal_T(A) = \{PropVal_T(a_j)|(\forall)i, 1 \leq i \leq k\}$ is called the values of properties corresponding to a set of entities A of type T.*

Definition 4 *The 4-tuple $Prop_{T,A} = [T, A, Prop_T, PropVal_T(A)]$ is called properties specification corresponding to a set of entities A of type T.*

3.2.2 Properties of design entities. Formal specification.

In the following, we apply the pattern defined in Section 3.2.1 in order to specify the properties for each type of design entities from our meta-model: class, method, attribute, parameter, localVariable. In Table 1 we describe the abbreviations used for the five entity type mentioned above. The following definitions will use these abbreviations.

Entity type	class	method	attribute	parameter	local variable
Abbreviation	C	M	A	P	LV

Table 1: Abbreviation of design entitites type

Definition 5 ***(Specification of properties for entities of type "class")***
The 4-tuple

$$Prop_{C,Class(E)} = [C, Class(E), Prop_C, PropVal_C(Class(E))]$$

is called specification of properties for entities of type"class", where

- $Prop_C = \{Abstraction, Visibility, Reuse\}$
- $Abstraction = \{concrete, abstract, interface\}$,
- $Visibility = \{normal, inner, public(Java)\}$,
- $Reusability = \{user - defined, user - extended, library\}$

Definition 6 ***(Specification of properties for entities of type "method")***
The 4-tuple

$$Prop_{M,AllMethod(E)} = [M, AllMethod(E), Prop_M, PropVal_M(AllMethod(E))]$$

is called specification of properties for entities of type"method", where

- $Prop_M = \{Abstraction, Visibility, Reuse, Kind, Instantion, Binding\}$
- $Abstraction = \{concrete, abstract\}$,
- $Visibility = \{private, protected, public\}$,
- $Reuse = \{defined, overrriden, inherited, library\}$
- $Binding = \{static, dynamic(virtual)\}$

- $Kind = \{constructor, destructor, normal, accesor\}$
- $Instantiation = \{class(static), object(instance)\}$

Definition 7 *(**Specification of properties for entities of type "attribute"**)*
The 4-tuple

$$Prop_{A,AllAttrib(E)} = [A, AllAttrib(E), Prop_A, PropVal_A(AllAttrib(E))]$$

is called specification of properties for entities of type"attribute", where

- $Prop_A = \{Type, Agregation, Visibility\}$
- $Type = \{built-in(predefined), user-defined, library\}$,
- $Agregation = \{simple, array\}$,
- $Visibility = \{private, protected, public\}$,

Definition 8 *(**Specification of properties for entities of type "parameter"**)*
The 4-tuple

$$Prop_{P,AllParam(E)} = [P, AllParam(E), Prop_P, PropVal_P(AllParam(E))]$$

is called specification of properties for entities of type"parameter", where

- $Prop_P = \{Type, Agregation\}$
- $Type = \{built-in(predefined), user-defined, library\}$,
- $Agregation = \{simple, array\}$,

Definition 9 *(**Specification of properties for entities of type "local variable"**)*
The 4-tuple

$$Prop_{LV,AllLocVar(E)} = [LV, AllLocVar(E), Prop_{LV}, PropVal_{LV}(AllLocVar(E))]$$

is called specification of properties for entities of type"local variable", where

- $Prop_{LV} = \{Type, Agregation\}$
- $Type = \{built-in(predefined), user-defined, library\}$,
- $Agregation = \{simple, array\}$,

Definition 10 *(**Formal specification of properties of design entities**)*
The 5-tuple

$$Prop(E) = [Prop_{C,Class(E)}, Prop_{M,AllMethod(E)}, Prop_{P,AllParam(E)},$$
$$Prop_{A,AllAttrib(E)}, Prop_{LV,AllLocVar(E)}]$$

is called formal specification of properties of design entities.

3.3 Relations between design entities

In this section we summarize the type of relations that exist between the entities from the meta-model. We mention here that for each entity, we consider only those relations in which it directly interacts with other entities.

3.3.1 Inheritance relations between classes

Inheritance defines a relation among classes in which a class shares its structure and behavior with one or more classes. Regarding the inheritance concept, there are two types of direct relations among classes: a class is either a specialization of another class or an implementation of an interface class.

Definition 11 ***(Inheritance relation)***
Consider $a, b \in Class(E)$. There are two types of direct relations among classes:

- ***a extends b**, if class a is a specialization of class b (class a inherits the structure and behavior of class b);*
- ***a implements b**, if class a is an implementation of the interface class b (class a implements the behavior of the interface class b).*

Definition 12 ***(Inheritance relations set)***

- $ExtendsSet = \{(a, b) \in Class(E) | a \ extends \ b\} \subseteq Class(E)^2$;
- $ImplementsSet = \{(a, b) \in Class(E) | a \ implements \ b\} \subseteq Class(E)^2$,

3.3.2 Method invocation relations

In order to define certain metrics for a class c, it is necessary to know the set of methods that are called by any method $m \in Meth(E)$ and the set of variables referenced by any method of the class c. The definition of method call relation was defined by Briand in [8]. We have adapted it to our framework as follows:

Definition 13 ***(Method call)***
Let $c \in Class(E)$, $m_1 \in Meth(c)$ such that $m_1.Reuse \in \{defined, overriden\}$, and $m_2 \in Meth(E)$. We say that m_1 call $m_2 \Leftrightarrow \exists d \in Class(E)$ such that $m_2 \in Meth(d)$ and the body of m_1 has a method invocation where m_2 is invoked for an object of type class d, or m_2 is a class method of type class d.

Definition 14 ***(Methods calls set)***
Let $MCall(E)$ be the set of all methods invocations, where

$$MCall(E) = \{(m_1, m_2) | m_1, m_2 \in Meth(E), m_1 \ call \ m_2\}$$

3.3.3 Attributes references relations

Methods may reference attributes. It is sufficient to consider the static type of the object for which an attribute is referenced because attribute references are not determined dynamically. For the discussion of measures later, it must be possible to express for a method, m, the set of attributes referenced by the method:

Definition 15 ***(Attribute references** [8]) For each method $m \in AllMeth(E)$ let $AttrRef(m)$ be the set of attributes referenced by method m.*

4 Metrics definition

As a proof of concept regarding the proposed approach, we define two of the Chidamber and Kemerer [4] metrics suite: Weighted Methods per Class (WMC) and Coupling Between Objects (CBO).

The WMC metric could be used in reverse engineering for detecting the central control classes in a system, based on the assumption that these classes are more complex than the others (model capture). Regarding the definition of this metric, we have to extend our model in order to offer a formal definition for Cyclomatic complexity metric [2]. This extension will be one of the objectives of our future work.

The CBO metric measures some aspects of coupling in an object oriented design. Coupling is an important criterion when evaluating a system because it captures a very desirable characteristic: a change to one part of the system should have a minimal impact on the other parts. An excessive coupling plays a negative role on many external quality attributes like reusability, maintainability and testability.

Metric Name	**Coupling Between Objects (CBO)** [4]
Informal definition	*The number of other classes that are coupled to the current one. Two Classes are coupled when methods declared in one Class use Methods or instance variables defined by the otherClass.*
`Formal definition`	$CBO(c) = \|d \in Class(E) - \{c\} \| \exists m_1 \in Meth(c), \exists m_2 \in Meth(d) :$ (m_1 *call* m_2) *or* (m_2 *call* m_1) *or* ($AttrRef(m_1) \cap Attr(m_2) \neq \emptyset$) *or* ($AttrRef(m_1) \cap Attr(m_1) \neq \emptyset$)$\|$, $c \in Class(E)$
Comments	

Table 2: CBO Metric Definition

Metric Name	**Weighted Methods per Class (WMC)** [4]
Informal definition	*The sum of complexities of the Methods in the current Class. If all method complexities are considered to be unique, WMC is equal to the number of Methods.*
`Formal definition`	$$WMC(c) = \sum_{k=1}^{noM_c} Complexity(m_k)$$ where $m_k \in Meth(c), noM_c = \|Meth(c)\|, c \in Class(E)$
Comments	$Complexity(m_k)$ is the Cyclomatic complexity of method m_k[2]

Table 3: WMC Metric Definition

5 Conclusions and Future Work

We have presented in this paper a new approach that address the issue of formal definition of OOD metrics, approach that allow us to define metrics in a general, flexible and extensible way. The main advantage of our framework is its scalability, new design entities can be added, e.g package, together with their properties and relations. Further work can be done in the following directions:

- to extend the approach in order to define any OOD metric;
- to define a library for OOD metrics definitions.

6 Acknowledgement

This research has been supported by the Romanian CNCSIS through the PNII-IDEI research grant ID_550/2007.

References

[1] W. Li, S. Henry, Object-oriented metrics that predict maintainability. *Journal of Systems and Software*, 23(2), 111–122, 1993.

[2] T.J. McCabe, A Complexity Measure. *IEEE Transactions on Software Engineering*, 2(4), 308–320, 1976.

[3] R. Marinescu, *Measurement and quality in object-oriented design.*, Ph.D. thesis in the Faculty of Automatics and Computer Science of the Politehnica University of Timisoara, 2003.

[4] S. Chidamber and C. Kemerer, A metric suite for object- oriented design. *IEEE Transactions on Software Engineering*, 20(6), 476–493, 1994.

[5] A. Baroni, S. Braz and F. Brito e Abreu , Using OCL to formalize object-oriented design metrics defnitions. *Proceedings of ECOOP Workshop on Quantative Approaches in Object-Oriented Software Engineering, Spain*, 2003.

[6] A. Baroni, S. Braz and F. Brito e Abreu , A formal library for aiding metrics extraction. *Proceedings of ECOOP Workshop on Object-Oriented Re-Engineering, Darmstadt, Germany* , 2003.

[7] R. Reiβing, Towards a model for object-oriented design measurement. *Proceedings of ECOOP Workshop on Quantative Approaches in Object-Oriented Software Engineering*, 2001.

[8] L. Briand, J. Daly and J. Wust, A Unified Framework for Coupling Measurement in Object-Oriented Systems. *IEEE Transactions on Softw. Engineering*, 25(1), 91-121, 1999.

[9] J. Wust, L. Briand, J. Daly, A Unified Framework for Cohesion Measurement in Object-Oriented Systems. *Empirical Software Engineering: An International Journal*, 3(2), 65-117, 1998.

Camelia Şerban
Babeş-Bolyai University, Department of Computer Science
Kogalniceanu 1, 400084, ClujNapoca
ROMANIA
E-mail: *camelia@cs.ubbcluj.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Tracking object's type changes with fuzzy based fusion rule

Albena Tchamova, Jean Dezert and Florentin Smarandache

Abstract

In this paper the behavior of three combinational rules for temporal/sequential attribute data fusion for target type estimation are analyzed. The comparative analysis is based on: Dempster's fusion rule proposed in Dempster-Shafer Theory; Proportional Conflict Redistribution rule no. 5 (PCR5), proposed in Dezert-Smarandache Theory and one alternative class fusion rule, connecting the combination rules for information fusion with particular fuzzy operators, focusing on the t-norm based Conjunctive rule as an analog of the ordinary conjunctive rule and t-conorm based Disjunctive rule as an analog of the ordinary disjunctive rule. The way how different t-conorms and t-norms functions within TCN fusion rule influence over target type estimation performance is studied and estimated.

1 Introduction

Target type estimates can be used during different target tracking process' stages for improving data to track association and for the quality evaluation of complicated situations characterized with closely spaced or/and crossing targets. It supports the process of identification, (e.g. friendly aircraft against hostile ones, fighter against cargo), helping that way the operator of the tracking system to initiate relevant measures. The process of identification could use many types of different attribute information, for instance, discrete information associated with ESM (Electronic Support Measures) data, IRST (Infra-Red Search and Track) measurements and direct target type observations. It requires also efficient fusion rules and criteria to estimate the correct associations. Our motivation for attribute fusion is inspired from the necessity to ascertain the targets' types, information, that in consequence has an important implication to enhance the tracking performance. In the literature we can find many fusion methods derived from different underlying frameworks, ranging from Bayesian probability theory ([1]), Dempster-Shafer evidence theory ([2], [3], [4]) to fuzzy sets ([5], [6], [7]) etc. In temporal multiple target tracking the main requirements we have to deal with relate especially to the way of adequate conflict processing/redistribution, the simplicity of implementation, satisfaction of the impact of neutrality of Vacuous Belief Assignment (VBA), reflection of majority opinion, etc. The choice of method ([15]) depends on the richness of abstraction and diversity of sensor data.

The most used until now Dempster-Shafer Theory (DST) proposes a suitable mathematical framework for representation of uncertainty. It is one of the widely applied framework in the area of target

tracking when one wants to deal with uncertain information and take into account attribute data and/or human-based information. DST, thanks to belief functions, is well suited for combining the information obtained by independent sources (bodies of evidence), especially in case of low conflicts between them. Although very appealing, DST presents some weaknesses and limitations and can give rise to some paradoxes/anomalies ([8]), [13], [14]), failed to provide the correct solution for some specific association problems, when the conflict increases and becomes very high (close to 1). To overcome the drawbacks of Dempster's fusion rule and in the meantime to extend the domain of application of the belief functions, recently a new mathematical framework, called Dezert-Smarandache Theory (DSmT) ([8], [9], [10]) was proposed for solving fusion/association problems, overcoming the practical limitations of DST, coming essentially from its inherent constraints, which are closely related with the acceptance of the law of the third excluded middle and can be interpreted as a general and direct extension of probability theory and the DST. The basic idea of DSmT is to work on Dedekind's lattice (called Hyper-Power Set) rather than on the classical power set of the frame as proposed in DST and, when needed, DSmT can also take into account the integrity constraints on elements of the frame, constraints which can also sometimes change over time with new knowledge. Hence DSmT deals with uncertain, imprecise and high conflicting information for static and dynamic fusion as well. Recently there is defined a new set of combination rules within DSmT, among them the Proportional Conflict Redistribution no. 5 ([9], [11]) which proposes a sophisticated and efficient solution for information fusion even when Shafer's model is considered and the conflict between the sources of information increase. In this paper we focus our attention on the ability of one alternative class fusion rule, connecting the combination rules for information fusion with particular fuzzy operators, focusing on the t-norm based conjunctive rule as an analog of the ordinary conjunctive rule of combination and on t-conorm based disjunctive rule as an analog of the ordinary disjunctive rule of combination, the so called T-Conorm-Norm (TCN) fusion rule ([12]). It is defined within DSmT and based on PCR5 fusion rule. The goal of fuzzy set/logic, which the TCN rule is based on, is to extend the classical binary logic into interval valued logic, thinking about something between false and truth. In this paper we concentrate our attention on the particular logical operator: *and*, implemented by the class of operators called t-norms and on the particular logical operator: *or*, implemented by the class of operators called t-conorms. We will study the way how different t-conorms and t-norms operators within TCN fusion rule influence over target type estimation performance. In the next sections we present briefly the basics of DSmT based PCR5 and TCN fusion rules, general properties and different types of t-conorm and t-norm functions. Then we describe the problem of object type estimation and identification. We apply the proposed TCN fusion rule for solving the problem and will evaluate how different types of t-norm functions influence over the estimation process, applied on a simple example. Then a comparative analysis of object type identification, obtained by using: *(1) Dempster's ([10]), PCR5 ([10]) and TCN fusion rules; (2) different types of t-conorm and t-norm functions within TCN fusion rule*; is provided on the base of estimated belief assignments. Concluding remarks are given in section 6.

2 Dezert-Smarandache theory based Proportional Conflict Redistribution rule no.5

Instead of distributing equally the total conflicting mass onto elements of power set as within Dempster's rule through the normalization step, or transferring the partial conflicts onto partial

uncertainties as within DSm hybrid rule, the idea behind the Proportional Conflict Redistribution rules is to transfer conflicting masses (total or partial) proportionally to non-empty sets involved in the model according to all integrity constraints. The general principle is to :

- calculate the conjunctive rule of the belief masses of sources;
- calculate the total or partial conflicting masses ;
- redistribute the conflicting mass (total or partial) proportionally on non-empty sets involved in the model according to all integrity constraints.

These rules work both in DST and DSmT frameworks and for static or dynamical fusion problematic, for any degree of conflict in [0, 1], for any DSm models (Shafer's model, free DSm model or any hybrid DSm model). The most sophisticated rule among them is PCR5. For only two sources of information it is defined by:

$$m_{PCR5}(\varnothing) = 0$$

And for $\forall X \in G^{\Theta} \setminus \{\varnothing\}$,

$$m_{PCR5}(X) = m_{12}(X) + \sum_{\substack{Y \in G^{\Theta} \setminus \{X\} \\ c(X \cap Y) = \varnothing}} \left[\frac{m_1(X)^2 m_2(Y)}{m_1(X) + m_2(Y)} + \frac{m_2(X)^2 m_1(Y)}{m_2(X) + m_1(Y)} \right] \quad (1)$$

Where G^{Θ} is the DSmT hyper-power set; $m_{12}(X)$ corresponds to the conjunctive consensus on X between the two sources and where all denominators are different from zero and $c(X)$ is the canonical form of X ([9], [11]).

No matter how big or small is the conflicting mass, PCR5 mathematically does a better redistribution of the conflicting mass than Dempster's rule and other rules since PCR5 goes backwards on the tracks of the conjunctive rule and redistributes the partial conflicting masses only to the sets involved in the conflict and proportionally to their masses put in the conflict, considering the conjunctive normal form of the partial conflict. PCR5 is quasi-associative and preserves the neutral impact of the vacuous belief assignment.

3 Fusion Rule based on Fuzzy T-Conorm-Norm Operators

The T-Conorm-Norm rule of combination represents a new class of fusion rules based on specified fuzzy t-Conorm, t-Norm operators. Triangular norms (t-norms) and Triangular conorms (t-conorms) are the most general families of binary functions that satisfy the requirements of the conjunction and disjunction operators, respectively. They are twoplace functions that map the unit square into the unit

interval, i.e. $T-norm(x,y):[0,1]\times[0,1]\to[0,1]$ and $T-conorm(x,y):[0,1]\times[0,1]\to[0,1]$. They are monotonic, commutative and associative. TCN rule takes its source from the t-norm and t-conorm operators in fuzzy logics, where the *and* logic operator corresponds in information fusion to the conjunctive rule and the *or* logic operator corresponds to the disjunctive rule. In this work we propose to interpret the fusion/association between the sources of information as a vague relation, characterized with the following two characteristics:

- *The way of association between the possible propositions, built on the base of the frame of discernment.* It is based on union and intersection operations, and their combinations. These sets' operations correspond to logic operations conjunction and disjunction and their combinations.

- *The degree of association between the propositions.* It is obtained as a t-norm (for conjunction) or t-conorm (for disjunction) operators applied over the probability masses of corresponding focal elements. While the logic operators deal with degrees of truth and false, the fusion rules deal with degrees of belief of hypotheses. We will study the way how different t-norms functions within TCN fusion rule influence over target type estimation performance.

TCN fusion rule does not belong to the general Weighted Operator Class. The base principle of this rule developed in [12] consists in the following steps:

Step 1: Defining the T-norm conjunctive consensus:

The t-norm conjunctive consensus is based on the particular t-norm function. In general it is a function defined in fuzzy set/logic theory in order to represent the intersection between two particular fuzzy sets. If one extends t-norm to the data fusion theory, it will be a substitute for the conjunctive rule.

The t-norm has to satisfy the following conditions:

- Associativity:
$$T_{norm}(T_{norm}(x,y),z)=T_{norm}(x,T_{norm}(y,z)) \quad (2)$$
- Commutativity:
$$T_{norm}(x,y)=T_{norm}(y,x) \quad (3)$$
- Monotonicity:
$$if\ (x\le a)\ \&\ (y\le b)\ then\ T_{norm}(x,y)\le T_{norm}(a,b) \quad (4)$$
- Boundary Conditions:
$$T_{norm}(0,0)=0;\ T_{norm(x,1)}=x \quad (5)$$

There are many functions which satisfy these t-norm conditions:

- Zadeh's (default) min operator:

$$m(X) = min\{m_1(X_i), m_2(X_j)\} \tag{6}$$

- Algebraic product operator:

$$m(X) = m_1(X_i).m_2(X_j) \tag{7}$$

- Bounded product operator:

$$m(X) = \max\{0, [m_1(X_i) + m_2(X_j) - 1]\} \tag{8}$$

The way of association between the focal elements of the given two sources of information, $m_1(.)$ and $m_2(.)$, is defined as $X = \theta_i \cap \theta_j$ and the degree of association is as follows:

$$\widetilde{m}_{12}(X) = T_{norm}\{m_1(\theta_i), m_2(\theta_j)\} \tag{9}$$

where $\widetilde{m}_{12}(X)$ represents the basic belief assignments (bba) after the fusion, associated with the given proposition X by using particular t-norm based conjunctive rule.

Step 2: Distribution of the mass, assigned to the conflict.

In some degree it follows the distribution of conflicting mass in the most sophisticated DSmT based Proportional Conflict Redistribution rule number 5, but the procedure here relies on fuzzy operators. The total conflicting mass is distributed to all non-empty sets proportionally with respect to the *Maximum (Sum)* between the elements of corresponding mass matrix's columns, associated with the given element X of the power set. It means the bigger mass is redistributed towards the element, involved in the conflict and contributing to the conflict with the maximum specified probability mass. The general procedure for fuzzy based conflict redistribution is as follows:

- Calculate all partial conflict masses separately;
- If $\theta_i \cap \theta_j = \varnothing$, then θ_i and θ_j are involved in the conflict; redistribute the corresponding masses $m_{12}(\theta_i \cap \theta_j) > 0$, involved in the particular partial conflicts to the non-empty sets θ_i and θ_j with respect to $max\{m_1(\theta_i), m_2(\theta_j)\}$ and with respect to $max\{m_1(\theta_j), m_2(\theta_i)\}$;
- Finally, for the given above two sources, the T_{norm} conjunctive consensus yields:

$$\begin{aligned}
\widetilde{m}_{12}(\theta_i) = {} & T_{norm}(m_1(\theta_i), m_2(\theta_j)) \\
& + T_{norm}(m_1(\theta_i), m_2(\theta_i \cup \theta_j)) \\
& + T_{norm}(m_1(\theta_i \cup \theta_j), m_2(\theta_i))
\end{aligned}
\qquad
\begin{aligned}
\widetilde{m}_{12}(\theta_j) = {} & T_{norm}(m_1(\theta_j), m_2(\theta_i)) \\
& + T_{norm}(m_1(\theta_j), m_2(\theta_i \cup \theta_j)) \\
& + T_{norm}(m_1(\theta_i \cup \theta_j), m_2(\theta_j))
\end{aligned} \tag{10}$$

$$\widetilde{m}_{12}(\theta_i \cup \theta_j) = T_{norm}(m_1(\theta_i \cup \theta_j), m_2(\theta_i \cup \theta_j))$$

Step 3: The basic belief assignment, obtained as a result of the applied TCN rule becomes:

$$\widetilde{m}_{TCN}(\theta_i) = \widetilde{m}_{12}(\theta_i) + m_1(\theta_i) \times \frac{T_{norm}\left(m_1(\theta_i), m_2(\theta_j)\right)}{T_{conorm}\left(m_1(\theta_i), m_2(\theta_j)\right)} + m_2(\theta_i) \times \frac{T_{norm}\left(m_1(\theta_j), m_2(\theta_i)\right)}{T_{conorm}\left(m_1(\theta_j), m_2(\theta_i)\right)};$$

$$\widetilde{m}_{TCN}(\theta_j) = \widetilde{m}_{12}(\theta_j) + m_2(\theta_j) \times \frac{T_{norm}\left(m_1(\theta_i), m_2(\theta_j)\right)}{T_{conorm}\left(m_1(\theta_i), m_2(\theta_j)\right)} + m_1(\theta_i) \times \frac{T_{norm}\left(m_1(\theta_j), m_2(\theta_i)\right)}{T_{conorm}\left(m_1(\theta_j), m_2(\theta_i)\right)} \tag{11}$$

Step 4: Normalization of the result.

The final step of the TCN fusion rule concerns the normalization procedure:

$$\widetilde{m}_{TCN}(\theta) = \frac{\widetilde{m}_{TCN}(\theta)}{\sum_{\substack{\theta \neq \varnothing \\ \theta \in 2^{\Theta}}} \widetilde{m}_{TCN}(\theta)} \tag{12}$$

4 Object's type tracking issue

4.1 Formulation of the problem

The Target Type Tracking Problem can be simply stated as follows:

- Let $k = 1,...,k_{max}$ be the time index and consider M possible target types: $T_i \in \Theta = \{\theta_1,...,\theta_M\}$, for example $\Theta = \{Fighter, Carg\,o\}$ with $T_1 \Rightarrow Fighter$, $T_2 \Rightarrow C\arg o$, or $\Theta = \{Friend, Hostile\}$, etc.
- At each instant k, a target of true type $T(k) \in \Theta$ (not necessarily the same target) is observed by an attribute-sensor (we assume a perfect target detection probability here).
- the attribute measurement of the sensor (for example noisy Radar Cross Section) is then processed through a classifier which provides a decision $T_d(k)$ on the type of the observed target at each instant k.
- the sensor is in general not totally reliable and is characterized by a $M \times M$ confusion matrix:

$$C = \left[c_{ij} = P\left(T_d = T_j \,/\, TrueTarg\,etType = T_i\right)\right] \tag{13}$$

Our goal is to estimate $T(k) \in \Theta$ from the sequence of declarations done by the unreliable classifier up to time k, i.e. to build an estimator:

$$\hat{T}(k = f(T_d(1), T_d(2), ..., T_d(k))) \text{ of } T(k). \quad (14)$$

4.2 Estimator's principle

In general the principle of our estimator is based on:

- a sequential temporal combination of the current basic belief assignment (drawn from classifier decision, i.e. our *measurements*) with the prior bba estimated up to current time from all past classifier declarations;
- Shafer's model for the frame of target types;
- One and the same information (vacuous belief assignment) as prior belief and same sequence of measurements (classifier declarations).

All the above assumptions are to get a fair comparative analysis according to: *(i)* performance evaluation of estimators, build on Dempster's rule of combination, Dezert-Smarandache Theory based PCR5 fusion rules and fuzzy-based fusion rule; *(ii*) different types of applied t-conorm and t-norm functions within TCN fusion rule. The algorithm's steps are:

1. Initialization at $k = 0$:

   - select the frame $\Theta = \{\theta_1, ..., \theta_M\}$

   - set the a priori $m^-(\theta_1 \cup ... \cup \theta_M) = 1$. (Full ignorance)

2. Generation of the current measurement from the classifier declaration $T_d(k)$:

$$m_{obs}(T_d(k)) = c_{T_d(k)T_d(k)} \quad (15)$$

The unassigned mass is committed to the total ignorance $m_{obs}(\theta_1 \cup ... \cup \theta_M) = 1 - m_{obs}(T_d(k))$.

3. Obtain the updated (estimated) bba $m(.) = [m^- \oplus m_{obs}](.)$ on the base of particular fusion rules: DST, DSmT-PCR5, TCN with different t-conorm *(sum, max)* and t-norm functions, described in equations (6), (7), (8).

4. Estimation of True Target Type by taking the hypothesis having the maximum of belief (or eventually the maximum Pignistic Probability [8]).

5. Propagation to next time step $k+1$

5 Simulation results

In order to evaluate the performances of all considered estimators, based on: Dempster's, DSmT based PCR5, TCN fusion rules with different types of t-conorm and t-norm functions applied, and in order to have a fair comparative analysis, Monte-Carlo simulations on a simple scenario for a 2D Target Type frame, i.e. $\Theta = \{Fighter, C\,arg\,o\}$ is done for a classifier with the following confusion matrix:

$$C = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Here we assume there are two closely spaced targets: Cargo and Fighter. Due to circumstances, attribute measurements received are predominately from one or another and both targets generates actually one single (unresolved kinematics) track. To simulate such scenario, a true Target Type sequence over 100 scans was generated according figure 1 below. The sequence starts with the observation of a Cargo type and then the observation of the target type switches two times onto Fighter type during different time duration. At each time step k the decision $T_d(k)$ is randomly generated according to the corresponding row of the confusion matrix of the classifier given the true target type (known in simulations). Then the algorithm presented in the previous section is applied. The simulation consists of 10000 Monte-Carlo runs. The computed averaged performances are shown on the figures 2 and 3 on the base of estimated belief masses obtained by the trackers based on Demspter's rule, DSmT based PCR5 rule, and TCN fusion rule with the following t-conorm and t-norm functions: ***(1) TCN with t-corm=max and t-norm=bounded product; (2)TCN with t-corm=max and t-norm=min; (3) TCN with t-corm=sum and t-norm=min***. It is evident that the target tracker based on Dempster's rule ([10]) is unable to track properly and quickly the changes of target type. Such a very long latency delay is due to the too long integration time necessary to the Demspter's rule for recovering the true belief estimation. PCR5 rule can quickly detect the type changes ([10]) and properly re-estimates the belief masses contrariwise to Dempster's rule. Our goal here is to evaluate the performance of TCN fusion rule with the analyzed above combinations between applied t-conorm and t-norm functions. In general it is obvious that TCN fusion rule shows a stable, quite proper and effective behavior, following the performance of PCR5 rule. TCN fusion rule with *t-corm=max* and *t-norm=bounded product* reacts and adopts better than TCN with t-*corm=sum* and *t-norm=min*, followed by TCN with *t-corm=max and t-norm=min*. As a whole, TCN rule is more cautious in the process of re-estimating the belief masses in comparison to PCR5. It provides symmetric type estimation contrariwise to Dempster's.

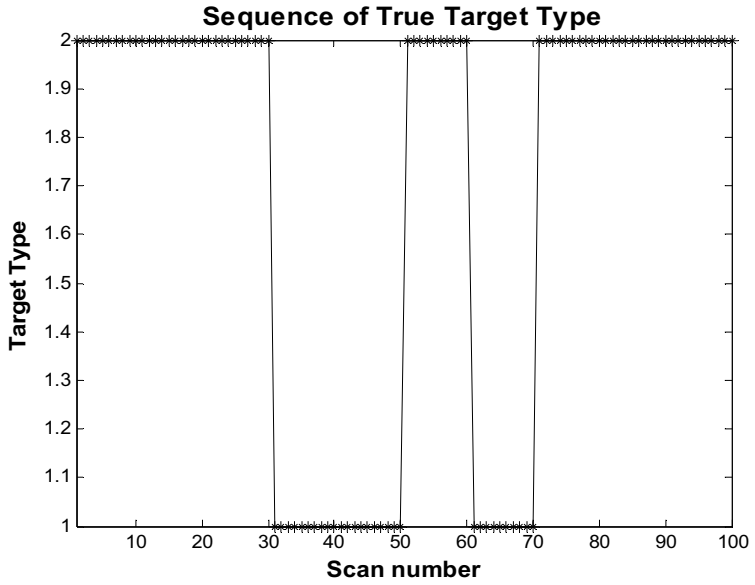


Fig.1 Sequence of true target type (Ground truth)

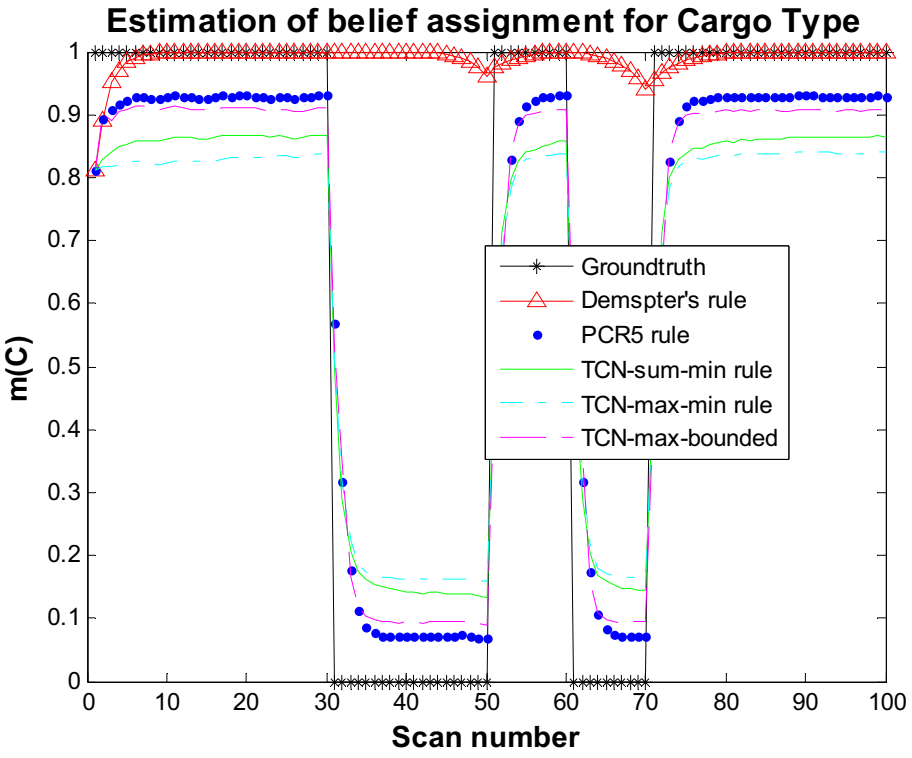


Fig.2 Estimation of belief assignment for Cargo type

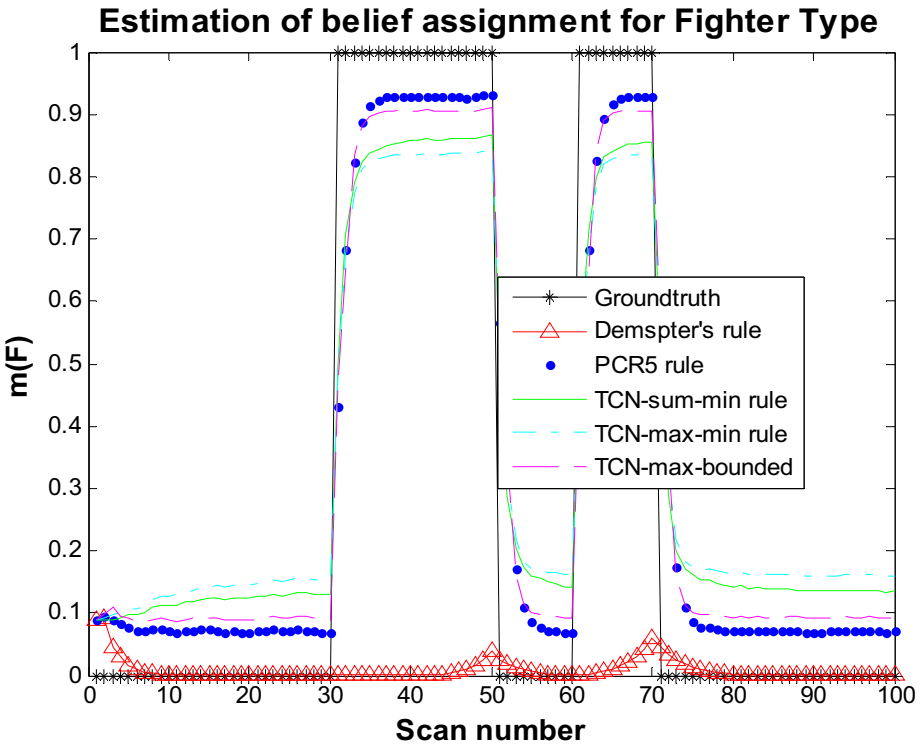


Fig.3 Estimation of belief assignment for Fighter type

6 Conclusions

The behavior of three combinational rules for temporal/sequential attribute data fusion for target type estimation has been estimated and analyzed. The comparative analysis is based on: Dempster's fusion rule proposed in Dempster-Shafer Theory; Proportional Conflict Redistribution rule no. 5 (PCR5), proposed in Dezert-Smarandache Theory and on the T-Conorm-Norm alternative class fusion rule, connecting the combination rules for information fusion with particular fuzzy operators, focusing on the t-norm based conjunctive rule as an analog of the ordinary conjunctive rule and t-conorm based disjunctive rule as an analog of the ordinary disjunctive rule. The way how different t-conorms and t-norms functions within TCN fusion rule influence over the target type estimation performance were studied and estimated on the base of simple scenario and Monte-Carlo simulations. In general the different t-conorm, t-norms, available in fuzzy set/logic theory provide us with richness of possible choices to be used in applied here TCN fusion rule. In engineering applications however, the most suitable t-norms are the product or the minimum, because of simplicity of computation and in order to preserve the principle of cause and effect. The attractive features of the new rule could be defined as: very easy to implement, satisfying the impact of neutral Vacuous Belief Assignment; commutative, convergent to idempotence, reflects majority opinion, assures adequate data processing in case of partial and total conflict between the information granules. It is appropriate for the needs of temporal fusion. The general drawback of this rule is related to the lack of associativity, which is not a main issue in temporal data fusion. The target type tracker built on TCN fusion rule shows a stable, quite proper and effective behavior, following the performance of PCR5 rule and overcoming the limitations and drawbacks of DST based tracker.

Acknowledgement: This work is founded and supported by the Bulgarian National Science Fund-grant MI-1506/05.

References

[1] Bayes T., *Essay toward solving a problem in the doctrine of changes*, Philosophical transactions on the royal society, London, 1763.
[2] Dempster A.P., *A Generalization of Bayesian Inference*, Journal of the Royal Statistical Society, Series B, 30: 205-247, 1968.
[3] Shafer G*., A mathematical theory of evidence*, Princeton, NJ: Princeton Univ. Press, 1976.
[4] Blackman S., Popoli R*., Design and Analysis of Modern Tracking Systems*, Artech House, 1999
[5] Zadeh L., *A Simple View of the Dempster-Shafer Theory of Evidence and its implication for the Rule of Combination*, AI Magazine 7, pp. 85–90, 1986.
[6] Zadeh L., *Outline of a new approach to the analysis of complex systems and decision processes*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, 1973.
[7] Mendel J., *Fuzzy Logic Systems for Engineering: A Tutorial*, Proc. of the IEEE, pp. 345-377, March 1995.
[8] Smarandache F. and J. Dezert (Eds), *Advances and application of DSmT for Information Fusion*", *Vol.1*, *Rehoboth*, *USA,* American Research Press, 2004.

[9] F. Smarandache and J. Dezert (Eds), *Advances and application of DSmT for Information Fusion*", *Vol.2*, *Rehoboth*, *USA,* American Research Press, 2006.
[10] F. Smarandache and J. Dezert (Eds), *Advances and application of DSmT for Information Fusion*", *Vol.3*, *Rehoboth*, *USA,* American Research Press, 2009.
[11] Smarandache F., Dezert J., *Information Fusion Based on New Proportional Conflict Redistribution Rules*, Proc. of Fusion 2005, Philadelphia, July 26-29, 2005.
[12] A. Tchamova, J. Dezert and F. Smarandache , *A new class of fusion rules based on T-Conorm and T-Norm fuzzy operator*", Information&Security, An International Journal, Vol. 20, 2006.
[13] Zadeh L., *On the Validity of Dempster's Rule of Combination of Evidence*, Memo M 79/24, Univ. California, Berkeley, 1979.
[14] Florea M., *Dempster-Shafer evidence theory through the years : limitations, practical examples ,variants under conflict and a new adaptive combination rule,* NATO - Advanced Study Institutes "Multisensor Data and Information Processing for Rapid and Robust Situation and Threat Assessment", Albena-Bulgaria . May 2005.
[15] Smarandache F*., Unification of Fusion Theories*, NATO - Advanced Study Institute "Multisensor Data and Information Processing for Rapid and Robust Situation and Threat Assessment", Albena-Bulgaria . May 2005.

Albena Tchamova
Bulgarian Academy of Sciences,
Institute for Parallel Processing,
Acad. G. Bonchev' Str., Bl. 25-A,
1113 Sofia,
BULGARIA
E-mail: : tchamova@bas.bg

Jean Dezert
ONERA,
29 Av. de la Division Leclerc,
92320 Chatillon,
FRANCE
E-mail: Jean.Dezert@onera.fr

Florentin Smarandache
University of New Mexico,
Chair of Department of Mathematics,
Gallup, NM 87301,
USA
E-mail: smarand@unm.edu

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Fuzzy Agent in a Medical Diagnosis Support System

Maria Tosa-Abrudan, Diana Man, Vlad Zdrenghea

Abstract

This paper presents artificial intelligence techniques in a fuzzy logic-based model for the medical diagnostic process. The medical diagnostic system is an intelligent agent that interacts with the resident doctors while they are examining a patient to coordinate she/him with the questions they have to ask for an accurate diagnostic. The system computes the most appropriate disease and the next information needed for a clearer separation between possibilities at each step and in the end will give suggestions for the diagnostic.

Such an intelligent decision support system constitutes a class of computer-based information systems including knowledge-based systems that support decision-making activities. Ideally, it should behave like a human consultant, supporting decision makers in better understanding their problems.

The knowledge base is a database with information regarding diseases and diseases tree, possible symptoms, possible values for a symptom and disease symptom values – (symptom value indicating the disease) and the relation between disease symptom values, since there are cases when not all symptoms are mandatory. This system contains data from a psychiatry domain.

1. Introduction

This paper presents the prototype of a medical diagnostic system acting as an intelligent agent that interacts with the resident doctors while they are examining a patient to coordinate she/him with the questions they have to ask for an accurate diagnostic establishment. Considering the scope of the system we can name it as a decision support system for the medical stuff.

2. Theoretical backgrounds

2.1 Intelligent agent approach

An intelligent agent is an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals [1]. Agents and multi-agent systems are the actual terms and concepts for Distributed Artificial Intelligence - DAI. Intelligent agents may learn and use knowledge to achieve their goals. The diagnosis system represents a knowledge-based agent that interacts with a medical operator in the diagnostic establishment process supported by a diseases and symptoms database behind.

Software implemented intelligent agents are often described as an abstract functional system, but Russell and Norvig's definition [1] consider the goal-directed behaviour as the essence of rationality and so prefer the term rational agent.

In Medical Diagnosis Systems, the intelligent agent indicates the user that the program still needs some more data – other medical characteristics that have to be given a value in order to make a more accurate diagnostic. The nature of intelligent agents can be heterogeneous. For example, in a Decision Support System (DSS) the interaction takes place between a human and an artificial problem solver – this is the case of the Medical Diagnosis System.

2.2 Decision Support System

A Decision Support System (DSS) has evolved during time, it was considered to be "a computer based system to aid decision making" in the 70s (Sol[2]), definition that describes well the type of system to be presented. Later definitions contained a larger domain for DSS: “an interactive computer-based system which help decision-makers utilize data bases and models to solve ill-structured problems” in the 80s (Sol[2]). Decision Support System systems may be developed in any knowledge domain. Examples include a bank loan officer verifying the credit of a loan applicant or other applications used in business and management. Executive dashboard and other business performance software allow faster decision making, identification of negative trends, and better allocation of business resources. A growing area of DSS application, concepts, principles, and techniques is in agricultural production and marketing.

2.3 Fuzzy logic definitions

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. In contrast with binary sets having binary logic, also known as crisp logic, the fuzzy logic variables may have a membership value of not only 0 or 1.

Just as in fuzzy set theory with fuzzy logic the set membership values can range (inclusively) between 0 and 1, in fuzzy logic the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values {true (1), false (0)} as in classic propositional logic [3].

Paradoxically, one of the principal contributions of fuzzy logic is its high power of precisiation of what is imprecise. This capability of fuzzy logic suggests, as was noted earlier, that it may find important applications in the realms of economics, linguistics, law and other human-centric fields.

In mathematical logic, there are several formal systems of "fuzzy logic"; most of them belong among so-called t-norm fuzzy logics.

The notions of a "decidable subset" and "recursively enumerable subset" are basic ones for classical mathematics and classical logic. Then, the question of a suitable extension of such concepts to fuzzy set theory arises. A first proposal in such a direction was made by E.S. Santos by the notions of fuzzy Turing machine, Markov normal fuzzy algorithm and fuzzy program (see Santos 1970). Successively, L. Biacino and G. Gerla showed that such a definition is not adequate and therefore proposed the other one, like.

U denotes the set of rational numbers in [0,1]. A fuzzy subset s : S ->[0,1] of a set S is recursively enumerable if a recursive map h : S×N ->U exists such that, for every x in S, the function h(x,n) is increasing with respect to n and s(x) = lim h(x,n). We say that s is decidable if both s and its complement –s are recursively enumerable. An extension of such a theory to the general case of the L-subsets is proposed in Gerla 2006. The proposed definitions are well related with fuzzy logic. It is an open question to give supports for a Church thesis for fuzzy logic claiming that the proposed notion of recursive enumerability for fuzzy subsets is the adequate one [4].

2.3.1. Linguistic variables and fuzzy if–then rules. Computing with words.

The concept of “computing with words”(CW) is rooted in several papers, starting with Zadeh paper in 1973 – “Outline of a New Approach to the Analysis of complex Systems and Decision Processes”, where the concepts of linguistic variable and granulation were introduced[9].

Computing with words evolved in a distinct methodology during time and it reflects many advantages of fuzzy logic and soft computing, advantages that took place within the past few years. A key aspect is that it involves a fusion of natural languages and computation with fuzzy variables[9].

The machinery of linguistic variables and fuzzy if–then rules is unique to fuzzy logic. This machinery has played and is continuing to play a pivotal role in the conception and design of control systems and consumer products.

Rules are usually expressed in the form:

IF variable IS property THEN action

For example, an extremely simple temperature regulator that uses a fan might look like this:

IF temperature IS very cold THEN stop fan

IF temperature IS cold THEN turn down fan

IF temperature IS normal THEN maintain level

IF temperature IS hot THEN speed up fan

The AND, OR, and NOT operators of boolean logic exist in fuzzy logic, usually defined as the minimum, maximum, and complement; when they are defined this way, they are called the Zadeh operators, because they were first defined as such in Zadeh's original papers. So for the fuzzy variables x and y:

NOT x = (1 - truth(x))

x AND y = minimum(truth(x), truth(y))

x OR y = maximum(truth(x), truth(y))

There are also other operators, more linguistic in nature, called hedges that can be applied. These are generally adverbs such as "very", or "somewhat", which modify the meaning of a set using a mathematical formula.

In fuzzy logic everything is or is allowed to be granulated, with a granule being a clump of attribute-values drawn together by indistinguishability, similarity, proximity or functionality. Graduated granulation, or equivalently fuzzy granulation, is a unique feature of fuzzy logic. Graduated granulation is inspired by the way in which humans deal with complexity and imprecision.

An important concept which is related to the concept of a linguistic variable is the concept of a granular value. A granular variable is a variable which takes granular values. In this sense, a linguistic variable is a granular variable which carries linguistic labels. It should be noted that a granular value of Age is not restricted to young, middle-aged or old. For example, ‘‘not very young” is an admissible granular value of age [79,81].

Fuzzy logic has many facets. Mathematically, the logical facet and the fuzzy-set-theoretic facet are the basic facets of fuzzy logic. A facet which plays a pivotal role in almost all applications of fuzzy logic is the relational facet – a facet which is focused on linguistic variables and fuzzy if–then rules [5].

3. Presentation of the Medical Diagnosis System

The Medical Diagnostics System is intended to be a software application mainly destined to orientate the resident doctors in the diagnostic process for patients' examinations. The system will be implemented for psychiatry diseases, but the architecture will try to make it flexible for further modules covering other medical areas.

3.1 System functionalities

The medical diagnostic system will expose the following functionalities

1. Orientate the doctor with the questions to be addressed to the patient based on the information from the previews answered question and the medical file data. Because there is a large amount of data and some symptom values automatic exclude certain diseases an intelligent system using artificial intelligence techniques will guide the user (resident doctor) throw the process of the questions to be answered.
2. Expert knowledge part to include the psychiatry diagnostics information about symptoms and diseases filled in by the medical stuff, in a later stage imported form an existing database while the format is proper for import.
3. The system should work in distributed environment in order to be accessible to people working in different places; the medical stuff that should fill in the medical data and the computer science researchers. This requirement determine us to use the internet interface to make it accessible in internet, we could also use web services to expose further functions of our system to other informatics systems.
4. Maintaining a "Examination File"/(or Medical File) which to include the main characteristics of the patient
5. Making reports/ extracting results from the database regarding disease characteristics and disease correlations.
6. The system should be protected; only authorized personnel should have access to fill in the steps for establishing the diagnostic or to complete the diagnostics information and parameters.
7. The system should support collaboration from international workgroups and English will be the first language of the user interface, but will be considered a extension to Multilanguage support.

3.2. Knowledge Database model

The knowledge base of the system contains a few main entities: symptom with associated symptom values, disease and disease symptom value.

3.2.1. Symptom

A symptom in our model is not equivalent with a medical symptom; it is an independent characteristic, like temperature, pulse, etc. The symptom has associated a name, a description - attribute that will be used to formulate the questions, and a list of two or more symptom values.

3.2.2. Symptom value

The symptom value in conjunction with a symptom on the other hand is similar with a medical symptom e.g. temperature of 38 degrees, pulse of 80, etc. For symptom "temperature" the

symptom values would be a list of values: “36 degrees”, ”37 degrees”, etc. Some of the symptom may have only the yes/no values.

3.2.3. Disease symptom value

A disease symptom value represents a symptom value associated to a disease. It contains a symptom and a symptom value associated plus some other characteristics like frequency (haw often the symptom appears), duration (haw long is since the first appearance of the symptom), if the symptom value is or is not mandatory (absolutely necessary to diagnose a person with that disease) and the weight of the symptom in the disease establishment on a scale from 1 to 1000. This is the weight given by the medical specialist. Other weight, called by us the Fuzzy Weight it is computed inside the system according to a bunch of other variables. A disease symptom value has associated also a code.

3.2.4. Disease

The diseases are stored under a hierarchical form following the model already defined in medicine. Important attributes are a list of disease symptom values for that disease. A patient should have most if not all the disease symptom values associated with a disease (at least all the mandatory ones) to be diagnosed with that disease. The second important attribute of the disease is the validation formula.

3.2.5. Knowledge Database Model Diagram

The relations between the entities of the knowledge database, and their entities can be seen in the following database diagram. (The database has been created using Ms. SQL Server)

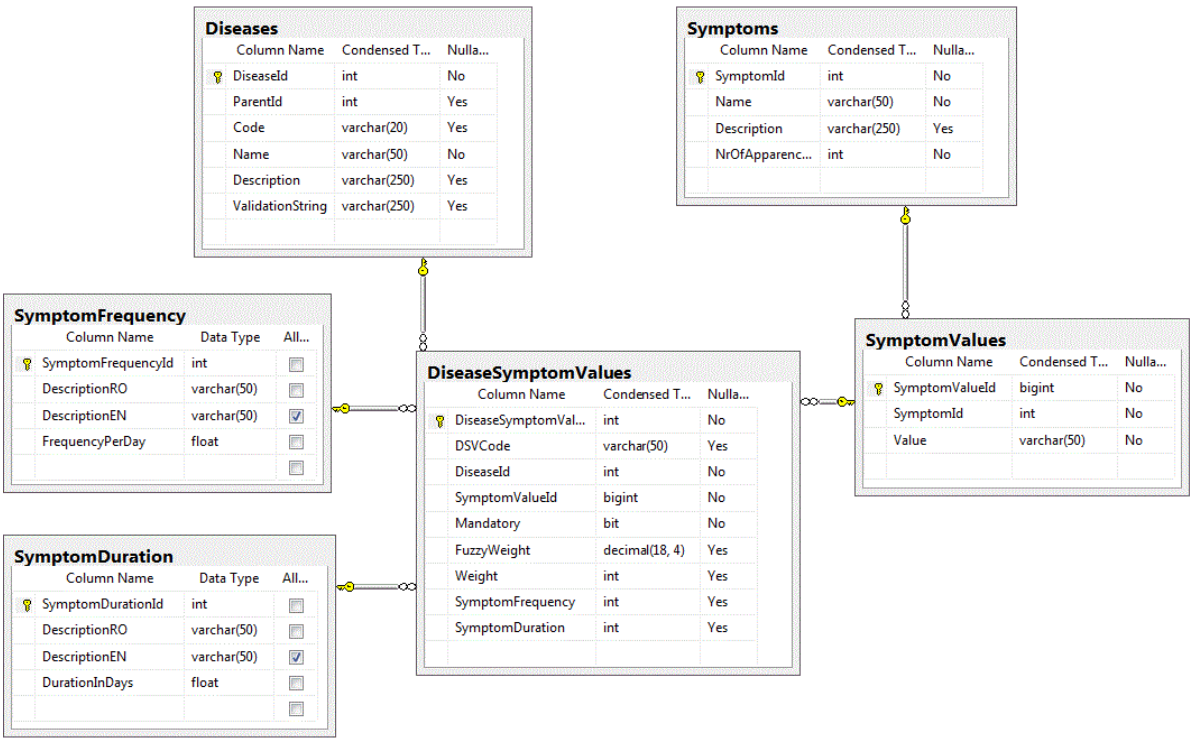


Figure 1: The Medical Disease System Data Base Diagram

3.3. Validation Formula

Disease establishment in the medical literature is defined as follows "Patient has disease X if it has at least three of the following symptoms S1, S2, … , S5 and if he/she also presents the symptoms S6 and S7". The disease validation string or validation formula contains this kind of constraints in order to validate a disease. To write it we use the disease symptom value codes and the following operators:

Operator	Descriptive word	Example
+	And	dsv1+dsv2+dsv3 stands for "For having the current disease the patient should present disease symptom value dsv1, disease symptom value dsv2 and disease symptom value dsv3"
|	Or	Dsv1 + (dsv2|dsv3) stands for "For having the current disease the patient should present disease symptom value dsv1 and disease symptom value dsv2 or disease symptom value dsv3"
()	Group of symptoms	-
min X {s1, …, sn}	Minimum X from a list	Min2{dsv1,dsv2,dsv3,dsv4}) stands for "For having the current disease the patient should present 2 of the following symptom value dsv1 and disease symptom value dsv2 or disease symptom values dsv1,dsv2, dsv3, dsv4"
max X {s1, …, sn}	Maximum X from a list	-

Table 1 : Operators of the validation expression

4. Fuzzy logic approach for medical diagnostics system

4.1 Notations

We are considering the following notations for the Medical Diagnosis System:

d – the disease, D – the set of diseases D=[d1, d2, ..., dn]

f_d – Validation formula for disease d, there is a different validation formula for each disease d, but the sub-components of the f_d functions are similar.

F – all the f_d validation formulas, F= [f_{d1}, f_{d2}, ..., f_{dn}], each disease d from D has an associated validation formula f_d from F.

s – the symptom, S – the set of symptoms in the system S=[s1,s2,..., sm]

sv –the symptom value, SV – the set of symptom values in the system,

SV=[sv1, sv2, ..., svl], l >= m (there are more symptom values than symptoms)

S_kV – the set of symptom values associated with the symptom s_k., S_KV =[sv1,....svr]

S_kV is a subset of SK. [sv1,....svr] is included in [sv1, sv2, ..., svl] = SV, r<l

dsv – disease symptom value, DSV =the set of all disease symptom values possible

f_d = function(dsv_1, dsv_2, ..., dsv_p) – the validation formula f_d contains a function whose parameters are disease symptom values dsv;

X – Examination information

X_T – examination information at moment T,

X_T = [dsv_1, dsv_2, ..., dsv_k] – examination information at moment T contains a list of disease symptom values [dsv_1, dsv_2, ..., dsv_k] that are know at moment T to be true for the patient condition

X_{T+1} = [dsv_1, dsv_2, ..., dsv_k, dsv_{k+1}], at each iteration there will be a new disease symptom value added.

4.2. Main step with IF ... THEN rules

In a diagnostic process we have a main IF-THEN rule for each disease:

IF f_d (X_T)=true THEN Patient diagnostic is d

At a moment T in the examination we have to evaluate each f_d in F, that means to make an evaluation for each d in D

EVAL[f_{d1} (X_T)]

EVAL[f_{d2} (X_T)]

.....

EVAL[f_{dn} (X_T)]

And to check the IF...THEN Rules

IF f_{d1} (X_T)=true THEN Patient diagnostic is d1

IF f_{d2} (X_T)=true THEN Patient diagnostic is d2

.....

IF f_{dn} (X_T)=true THEN Patient diagnostic is dn

Because especially in the earlier T moments in the evaluation none of the f_d functions will be true, we will have to compute a "degree of truthiness", to evaluate in which degree f_d is true

Also at each moment T in the examination we have a set of disease symptom values we know for the patient: X_T = [dsv_1, dsv_2, ..., dsv_k].

4.2.1. The actual evaluation

The EVAL function, EVAL[f_d (X_T)] is an evaluation of the f_d function and is using fuzzy logic operators AND, OR, NOT and also reunion, intersection operators to compute the "degree" in which the patient has disease d at an intermediate time T. The fuzzy operators will be used over the more "linguistic" operators presented in section 3.3 Validation Formula. The EVAL function should take in consideration that not all parameters are known at the current moment T, we know only dsv_1, dsv_2, ..., dsv_k, not dsv_1, dsv_2, ..., dsv_p, p>k.

EVAL[f_{d2} (X_T)] = EVAL[f_{d2} (dsv_1, dsv_2, ..., dsv_k)]

Further testing and analysis of the EVAL function is under work.

4.2.2. Optimization

In order to optimize the computation we will make the evaluation of the f_d functions and check the IF ... THEN rules only for the f_d functions that have as parameter(s) one or more of the disease symptom values at moment T.

That means that at moment T, having

$X_T = [dsv_1, dsv_2, ..., dsv_k]$, $D=[d1,..., dn]$, $F=[f_{d1}, ..., f_{dn}]$, $f_d = function(dsv_1, dsv_2, ..., dsv_p)$

We will compute

$EVAL(f_d)$, for each $f_d = function(dsv_1, dsv_2, ..., dsv_p)$ with the property that exists a dsv_i in $(dsv_1, dsv_2, ..., dsv_p)$ that belongs also to $[dsv_1, dsv_2, ..., dsv_k]$, $= X_T$.

4.3. Finding the next iteration

The role of the intelligent agent is to determine the next symptom to be interrogated for its value in order to obtain the next disease symptom value dsv to be added in the examination list at moment T+1.

The rules to choose the next symptom are the following:

Search for the disease d with the best scores for the $EVAL(f_d)$,

- $f_d = function(dsv_1, dsv_2, ..., dsv_p)$ and for which not all $(dsv_1, dsv_2, ..., dsv_p)$ are included in $X_T = [dsv_1, dsv_2, ..., dsv_k]$; exists at last one dsv that is not included.
  If there are more such diseases choose randomly one of them
- Choose the next disease symptom value dsv_{T+1} needed to determine the truth of this disease with the following conditions:
  - dsv_{T+1} belongs to $(dsv_1, dsv_2, ..., dsv_p)$, parameters of $f_d = function(dsv_1, dsv_2, ..., dsv_p)$ for the chosen disease d
  - dsv_{T+1} does not belong to $X_T = [dsv_1, dsv_2, ..., dsv_k]$;
  - if there are more disease symptom values candidates which fulfil the above conditions choose the one that are mandatory for disease d, and if there is none mandatory choose the one that has higher weight
  - if applying the above rule none of the disease symptom values has been chosen choose one randomly
- The next symptom to be questioned will be the symptom of the chosen disease symptom value dsv_{T+1}

5. Other Approaches in CDSS

Common example of the DSS is the Clinical Decision Support System for medical diagnosis. Clinical decision support systems (CDSS) are interactive computer programs, which are designed to assist physicians and other health professionals with decision making tasks.

5.1 Iliad

Iliad was developed at the University of Utah School of Medicine, Dept. of Medical Informatics, it is an expert system and has been under development for several years. It uses Bayesian reasoning to calculate the posterior probabilities of various diagnoses under consideration, given the findings present in a case. Iliad was developed primarily for diagnosis in Internal Medicine and now covers about 1500 diagnoses in this domain, based on several thousand findings. The Iliad shell has also been used to develop knowledge bases for diagnosis in other domains.

5.2. TheraDoc

TheraDoc is a suite of real-time monitoring products as well as Infection prevention, mediation, and correction. TheraDoc uses a variety of standards compliant messaging taxonomies to work with currently existing software in the clinical setting to guide decision making as well as report trends in problems your institution is encountering (http://www.theradoc.com/products/). An Expert System Platform is used as a unique knowledge execution platform that concurrently utilizes TheraDoc Knowledge Modules to intelligently alert clinicians and deliver patient- and disease-specific information and treatment recommendations to the point-of-care. The TheraDoc Expert System Platform is based on a scalable n-tier architecture based on robust third party technologies.

5.3. Lifecom -CHAMP

Lifecom has developed the CHAMP Project(Computerized Healthcare Assistant and Management Project) which is an entirely new form of artificial intelligence technology and a supporting suite of knowledge development tools. The Lifecom technology portfolio includes novel knowledge extraction and management tools, the artificial intelligence engine , and a highly graphical user interface.(http://www.ohsu.edu/champ/ch_champ.html)[7]. The A. I. Engine - Integral Diagnostic Engine (patent pending) assesses all entered data in support of optimal decision-making.

5.4. VisualDx

VisualDx is a point-of-care diagnostic resource that allows clinicians to build patient-centric visual differential diagnoses based on the patient's signs, symptoms, medical history, and more. Clinicians have immediate access to more than 17,000 images as well as expert-reviewed information for nearly 1,000 visually identifiable diseases, drug reactions, and infections represented in all age ranges and skin types [8]. The system is a JAVA-based decision-support program developed by Logical Images to be used in clinical care to develop differential diagnoses based upon morphologic finding- and patient finding-driven searching. It consists of several modules, many of which are very relevant to infectious diseases specialists.

5.5. Auctoritas

Auctoritasis psychiatric expert system is a global database managing the newly-developed advisory system; it is appropriate for managing a complete hospital network system for the continuing individual long-distance observation of patients. It consists of four parts: administration, diagnostic decision support system, activities concerning treatment, and statistics. The diagnoses relying on the up-to-date psychiatric diagnostic systems manuals - DSM-IV and ICD-X (based on classical logic). For the construction authors used fuzzy logic and backward chaining.

6. Conclusions and further work

This paper presents a fuzzy CDSS - an intelligent agent using fuzzy logic. We study the possibilities of using fuzzy logic in building agent software assuming the role of an experienced medical person, which benefits of a vast medical knowledge regarding symptoms and diseases and has the role to orientate the young resident doctors in the process of diagnosis establishment.

The action this agent system should take according to the functional requirements is to generate at each iteration the next more appropriate question - whose answer will bring the diagnosis process closer to its end: the diagnostic of the patient. The question is closely related to the symptom (in DB model the question is an attribute of the symptom entity and we intend to use fuzzy set theory elements and fuzzy logic to find the more appropriate symptom that needs a value.

Further work will consist in finding possible linguistic values for the evaluation function that computes the degree of truth in a validation formula at a time T, adjustments of the fuzzy weights that are initially given and possible optimization of the next iteration choosing algorithm. We intend to make also a model based on fuzzy clustering to compare the results with the existing one or to use it for system improvement. Further research will also be done in the distributed artificial intelligence domain, specifically regarding the intelligent agents models and in indentifying the appropriate technologies to be used in such a distributed environment.

References

[1] Russell Stuart J., Norvig Peter - *Artificial Intelligence: A Modern Approach* 2002(2nd ed.), Prentice Hall, ISBN: 0-13-103805-2

[2] Henk G. Sol et al. *Expert systems and artificial intelligence in decision support systems*: proceedings of the Second Mini Euroconference, Lunteren, The Netherlands, 17-20 November, 1985. Springer, 1987. ISBN 9027724377. p.1-2.

[3] Novák, V., Perfilieva, I. and Mockor, J. *Mathematical principles of fuzzy logic* Dodrecht: Kluwer Academic. ISBN 0-7923-8595-0, 1999

[4] Passino, K. M.; Yurkovich, S. *Fuzzy control*. Boston: Addison-Wesley,1998

[5] Zadeh, L.A. *Is there a need for fuzzy logic?,* Information Sciences 178, 2008, 2751–2779

[6] The International Benchmark Sites Network for Agrotechnology Transfer, *Decision Support for Agrotechnology Transfer* http://www.icasa.net/dssat/

[7] Warner HR Jr, Bouhaddou O. *Innovation review: Iliad--a medical diagnostic support program*, 1994, Applied Informatics, Salt Lake City, UT. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&dopt=Abstract&list_uids=10134761&query_hl=2&itool=pubmed_docsum

[8] Tleyjeh IM, Nada H, Baddour LM *VisualDx: decision-support software for the diagnosis and management of dermatologic disorders*, Department of Medicine, Mayo Clinic College of Medicine, Mayo Clinic, Rochester, MN, USA. http://www.ncbi.nlm.nih.gov/pubmed/17029139?log$=activity

[9] Zadeh L.A., From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions, Int. J. Appl. Math. Comput. Sci., 2002, Vol 12, No 3, 307-324.

[10] R.A. Miller. *Medical diagnostic decision support systems--past, present, and future: a threaded bibliography and brief commentary*. J Am Med Inform Assoc 1994 Mar-Apr;1(2

[11] Eta S. Berner, *Clinical Decision Support Systems*: *Theory and Practice,* New York, NY: Springer, 2007.

[12] Kovács M, Juranovics J., *"Auctoritas" psychiatric expert system shell*, Medinfo. MEDINFO, 1995

Maria Tosa-Abrudan
Faculty of Mathematics and Computer Science
Babes-Bolyai University
400084 Cluj-Napoca
ROMANIA
maria@cs.ubbcluj.ro

Diana Man
Faculty of Mathematics and Computer Science
Babes-Bolyai University
400084 Cluj-Napoca
ROMANIA
man.diana@cs.ubbcluj.ro

Vlad Zdrenghea
Department of Psychiatry
Iuliu Hategan University of Medicine and Pharmacy
Cluj-Napoca
ROMANIA
vlad_zdrenghea@yahoo.com

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Guided Maximum Entropy Method

Milan Tuba

Abstract

The maximum entropy method (MEM) has been successfully applied in many different areas for solving under-determined systems. It favors uniform distribution and tends to make variables as equal as possible, satisfying constraints. The problem is that for the initial adjustment of some problem for the MEM applications variables have to be selected in such a way that the solution is feasible, but may be far from desirable. This paper presents an improvement of the MEM model by introduction of new variables, constraints and weight factors that shift solution from feasibility to optimality. This modified method exploits the property of the MEM that it can smoothly move from cases where constraints can be satisfied to cases where constraints become desirable goals that are satisfied as much as possible. A software system was developed which includes all the mentioned features.

1 Introduction

The maximum entropy method was recently used with great success in many different areas where under-determined systems are involved. It is most frequently used in chemistry ([1]), but also in many other very diverse areas: computer network design ([2]), character recognition ([3]), data analysis ([4]), image processing ([5], [6]), economy ([7]). Theoretical developments also continue ([8]).

The basic idea is to get a unique solution from the under-determined system by introducing the additional constraint that the entropy function should be maximized. The other methods that were used for solving under-determined systems use the same technique: they introduce additional, artificial constraints that make the number of constraints equal to the number of unknowns. The difference is that the maximum entropy method introduces the most natural additional constraint: one that does not introduce any new, arbitrary and unwarranted information. It uses only the information that is given and makes no assumptions about missing information.

Before going to the formal definition of the maximum entropy principle, it is interesting to mention that, besides very pragmatic uses, there are very extensive and still open philosophical discussions about the real meaning of this principle. The predecessor of the maximum entropy principle is the principle of insufficient reason (James Bernoulli: *Ars Conjectandi,* 1713). It states that in the absence of any information (knowledge), all outcomes should be considered equally possible. This principle was involved in the discussions about prior probabilities (probabilities of one event, state of the knowledge) and relative frequencies. Relative frequencies become predominant and some useful works from Laplace and Bayes were criticized. Shannon's works on information theory opened a new opportunity for revitalization of the principle of insufficient reason, this time as a more sophisticated maximum entropy principle which was introduced by Jaynes. Philosophical discussions about the real meaning of the maximum entropy method are interesting, but since method was successfully applied in many areas, for any new area the most important criterion is not how well the relation between the MEM and that area can be explained, but how useful are the results of the application of the method.

2 The MEM Formalism

General model of the MEM calls for random variables and probabilities, but for most problems more suitable description is:

A system of k equations with n variables v_i, $k < n$ represents constraints. Maximum entropy solution which respects constraints and makes variables as equal as possible is looked for. It is interesting to notice that the same goal can be attained by using some other function that has maximum when all variables are equal. One very simple such function is the product of all variables. The product function expression seems simpler then the entropy function expression which involves logarithms, but the fact that partial derivatives are needed points out that entropy function is better since it separates variables.

$$\begin{aligned} x_{1,1}v_1 + x_{1,2}v_2 + ... + x_{1,n}v_n &= l_1 \\ x_{2,1}v_1 + x_{2,2}v_2 + ... + x_{2,n}v_n &= l_2 \\ &\vdots \\ x_{k,1}v_1 + x_{k,2}v_2 + ... + x_{k,n}v_n &= l_k \end{aligned} \tag{1}$$

Variables v_i are converted to probabilities by normalization: $p_i = v_i / \sum_{j=1}^{n} v_j$ and $m_i = l_i / \sum_{j=1}^{n} v_j$. The system (1) then becomes

$$\sum_{i=1}^{n} x_{r,i} p_i = m_r \ , \qquad r = 1, 2, ..., k \tag{2}$$

This is equivalent to the classical definition of the MEM where it is assumed that for a discrete random variable X the values $x_1,\ x_2, ...,\ x_n$ that it can take are known, but the corresponding probabilities $p_1,\ p_2, ...,\ p_n$ are not known. The expected values for $k < n-1$ functions of X (for example, the first k moments) are also known and represent constraints:

$$E[\ f_r(X)] = m_r \qquad r = 1, 2, ..., k. \tag{3}$$

Equation (2) (or (3)) gives (together with $\sum p_i = 1$) $k+1 < n$ constraints for n unknown variables $p_1,\ p_2,\ ...,\ p_n$. This system is under-determined and has an infinite number of solutions. The unique solution that maximizes the entropy of the system is looked for. That is the best solution in the sense that it uses only the information given. It is neutral to the missing information (it does not introduce any hidden assumptions). This additional constraint can be expressed as:

Maximize the entropy function

$$H(p_1,\ p_2,\ ...,\ p_n) = -K \sum_{i=1}^{n} p_i \ ln(p_i). \tag{4}$$

For $K = 1$, entropy will be expressed in natural units (rather than in bits).

2.1 Solution

The method of Lagrange multipliers is used. This will not guarantee that probabilities are non-negative. The substitution $p_i = e^{-q_i}$ is introduced, but this gives a stronger constraint than the one required: all probabilities are now positive definite (none of them can be zero). The problem now is to maximize

$$H(q_1,\ q_2, ...,\ q_n) \ = \ \sum_{i=1}^{n} q_i\ e^{-q_i} \tag{5}$$

under the conditions

$$\sum_{i=1}^{n} e^{-q_i} \ = \ 1 \tag{6}$$

$$\sum_{i=1}^{n} e^{-q_i} f_r(x_i) \ = \ m_r\ , \qquad r = 1, 2, ..., k \tag{7}$$

Lagrange multipliers $\lambda,\ \mu_1,\ \mu_2, ...,\ \mu_k$ are introduced with the function:

$$F(q_1,\ q_2, ...,\ q_n) \ = \ \sum_{i=1}^{n} q_i\ e^{-q_i} \ + \ \lambda \sum_{i=1}^{n} e^{-q_i} \ + \ \sum_{r=1}^{k} \mu_r \sum_{i=1}^{n} e^{-q_i}\ f_r(x_i) \tag{8}$$

All partial derivatives should be zero:

$$\frac{\delta F}{\delta q_i} \ = \ e^{-q_i}\ [1 - q_i - \lambda - \sum_{r=1}^{k} \mu_r\ f_r(x_i)] \ = \ 0\ , \qquad i = 1, 2, ..., n \tag{9}$$

Since e^{-q_i} is never zero

$$q_i \ = \ 1 - \lambda - \sum_{r=1}^{k} \mu_r\ f_r(x_i)\ , \qquad i = 1, 2, ..., n \tag{10}$$

The problem is now solved: Equations (6), (7), and (10) give $n+k+1$ equations for $n+k+1$ unknown variables $p_1,\ p_2, ...,\ p_n,\ \mu_1,\ \mu_2, ...,\ \mu_k,\ \lambda$. The system should have unique solution, but it is not linear and some numerical method has to be used.

To make the calculations easier, the partition function is introduced:

$$Z(\mu_1,\ \mu_2, ...,\ \mu_k) \ = \ \sum_{i=1}^{n} p_i\ e^{-\lambda} \ = \ \sum_{i=1}^{n} e^{-\lambda - q_i}$$

$$Z(\mu_1,\ \mu_2, ...,\ \mu_k) \ = \ \frac{1}{e} \sum_{i=1}^{n} e^{\sum_{r=1}^{k}\ \mu_r\ f_r(x_i)} \tag{11}$$

It is easy to see that

$$\lambda \ = \ - \ln Z(\mu_1,\ \mu_2, ...,\ \mu_k) \tag{12}$$

$$m_r \ = \ \frac{\delta}{\delta \mu_r} \ln Z(\mu_1,\ \mu_2, ...,\ \mu_k) \tag{13}$$

or

$$m_r \ = \ \sum_{i=1}^{n} [m_r - f_r(x_i)] e^{\sum_{j=1}^{k}\ \mu_j\ f_j(x_i)} = 0\ , \qquad r = 1, 2, ..., k \tag{14}$$

Equation (14) represents k equations for k unknown variables $\mu_1,\ \mu_2, ...,\ \mu_k$. When it is solved, from Equation (12) λ is calculated, and then from Equation (10) $q_1,\ q_2, ...,\ q_n$ are determined , and finally, from $p_i = e^{-q_i}$ the probabilities $p_1,\ p_2, ...,\ p_n$ are calculated.

Substitution $t_j = e^{\mu_j},\ \ j = 1, 2, ...,\ k$ can be introduced. Then Equations (12) and (14) become:

$$\lambda \ = \ 1 - ln[\sum_{i=1}^{n} \Pi_{j=1}^{k} t_j^{f_j(x_i)}] \tag{15}$$

$$\sum_{i=1}^{n} [m_r - f_r(x_i)] \Pi_{j=1}^{k} t_j^{f_j(x_i)} \ = \ 0, \qquad r = 1, 2, ..., k \tag{16}$$

There is an algorithm to solve this system. However, the function that is to be minimized is not convex even in the simplest case when there is only one constraint: expected value. The standard Newton-Rapson procedure will not work. But the Jacobian matrix for this system is symmetric and positive definite. This gives a scalar potential function which is strictly convex and whose minimum is easy to find. The use of the second order Taylor expansion is recommended. However, after much experience with the algorithm, our impression is that it is not even worth trying to find the exact value for α that determines how far to go along a certain direction, let alone inverting the Jacobian matrix every time. For our software system we developed a heuristic that performs well.

2.2 Selection Principle

The previous model has constraints $p_i \ > \ 0,\ i = 1, 2, ..., n$. This may be too strong since the probabilities need only to be nonnegative. To make $p_i \ \geq \ 0$, $p_i \ = \ q_i^2$ can be introduced instead of $p_i \ = \ e^{-q_i}$, which was used before. In that case, the problem becomes to maximize

$$H(q_1,\ q_2, ...,\ q_n) \ = \ -2 \sum_{i=1}^{n} q_i^2 ln(q_i) \tag{17}$$

under the conditions

$$\sum_{i=1}^{n} q_i^2 \ = \ 1 \tag{18}$$

$$\sum_{i=1}^{n} q_i^2 \ f_r(x_i) \ = \ m_r, \qquad r = 1, 2, ..., k \tag{19}$$

Lagrange multipliers are introduced:

$$F(q_1,\ q_2, ...,\ q_n) \ = \ -2 \sum_{i=1}^{n} q_i^2 \ ln(q_i) \ + \ \lambda \sum_{i=1}^{n} q_i^2 \ + \ \sum_{r=1}^{k} \mu_r \sum_{i=1}^{n} q_i^2 \ f_r(x_i) \tag{20}$$

Partial derivatives should be zero:

$$\frac{\delta F}{\delta q_i} \ = \ -2q_i[2ln(q_i) + 1 - \lambda - \sum_{r=1}^{k} \mu_r f_r(x_i)] \ = \ 0, \qquad i = 1, 2, ..., n \tag{21}$$

Now, the selection has to be made: any q_i can be zero.

$$q_i = 0 \quad or \quad q_i \ = \ e^{(\ -1+\lambda+\sum_{r=1}^{k} \ \mu_r \ f_r(x_i)\ )^{0.5}}, \qquad i = 1, 2, ..., n \tag{22}$$

When it is decided which q_i are to be zero, the remaining equations will give as many equations as there are unknown variables. The partition function is equal as in the previous model, and the whole discussion repeats. The only difference is that summations are not carried for all $i = 1$ to n, but only for those i for which $q_i \neq 0$.

This new model is used only to show how the case p_i=0 for some i can be included. In practice, we have to decide which p_i will be zero. We can do it in advance and consider a model that has only $n-m$ probabilities (if m probabilities are selected to be zero). If we select too many probabilities to be zero, the system may become over-determined.

3 The Guided MEM

For many problems initial adjustment for the MEM application requires that variables of the system be determined in such a way that a feasible solution is obtained. This may not be a desirable solution for the optimization, but constraints have to be satisfied first.

It is possible to modify the MEM model and include a mechanism to guide the process of optimization. Once the necessary constrains are satisfied, artificial variables can be introduced that will guide the optimization process in the desirable direction.

MEM guidance will be demonstrated on an example, similar to Brandeis Dice Problem.

A die, possibly irregular, is considered. The number of spots that shows up when the die is tossed defines a random variable with possible outcomes and corresponding probabilities.

$$X = [1, 2, 3, 4, 5, 6]$$

$$P_{(6)} = [p_1, p_2, p_3, p_4, p_5, p_6]$$

The constraint that the sum of the probabilities is 1 is always present and in usual terminology not counted as an additional constraint. Without any (additional) constraints the expected value $E(X)$ is 3.5 and the solution for the probabilities is an uniform distribution: $p_i = 0.167, \quad i = 1, 2, ...6$.

For a single constraint EX=4.4 there is one (additional) constraint:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 = 4.4$$

and the MEM solution is:

$$P_{(6)} = [0.063, 0.087, 0.121, 0.169, 0.234, 0.325]$$

As expected, the probabilities density is shifted towards larger outcomes since expected value shifted in that direction.

If the elementary probabilities were not the goal of equalization but some coarser variables, additional constraint can be included. If, for example, the goal is to to make $p_x = p_1+p_2+p_3$ equal to $p_y = p_4+p_5+p_6$, a system of two constraints can be used:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 = 4.4$$

$$1p_1 + 1p_2 + 1p_3 - 1p_4 - 1p_5 - 1p_6 = 0$$

In this case it is possible to have a solution that will satisfy both constraints:

$$P_{(6)} = [0.004, 0.042, 0.454, 0.004, 0.042, 0.454] \tag{23}$$

The problem with this approach is that it limited to cases when the guidance goal (in this case the total equalization of p_x and p_y) is possible. However, the main advantage of the MEM method is its ability to push towards the guidance goal even when exact goal satisfaction is not possible.

This can be illustrated on the previous example, but with changed requirement that $E(X) = 4.6$. It is easy to see that the constraint

$$p_1 + p_2 + p_3 = p_4 + p_5 + p_6$$

can not be satisfied. The maximum value for $E(X)$ is reached when probabilities density is pushed toward higher values:

$$P_{(6)} = [0, 0, 0.5, 0, 0, 0.5]$$

The value for $E(X)$ is in that case equal to 4.5. For any higher value of $E(X)$ exact equalization (which is the second constraint) is not possible.

To make the sums $p_1 + p_2 + p_3$ and $p_4 + p_5 + p_6$ as equal as possible, new variables are introduced: $p_6 = p_x = p_1 + p_2 + p_3$ and $p_7 = p_y = p_4 + p_5 + p_6$. Two new constraints that define these new probabilities are added. The fact that new variables are mentioned as constraints will make them participate in the equalization process.

Care mast be taken about normalization. New probabilities (p_7 and p_8) are not independent from the old ones and the sum of all probabilities becomes 2. Considering that the sum of all probabilities has to be 1 and that the sum of old probabilities (only old probabilities participate in the first constraint) is only 0.5, the first constraint has to be redefined.

Three constraints now become:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 + 0p_7 + 0p_8 = 2.3$$

$$1p_1 + 1p_2 + 1p_3 + 0p_4 + 0p_5 + 0p_6 - 1p_7 + 0p_8 = 0$$

$$0p_1 + 0p_2 + 0p_3 + 1p_4 + 1p_5 + 1p_6 + 0p_7 - 1p_8 = 0$$

and the corresponding MEM solution is:

$$P_{(8)} = [0.020, 0.039, 0.076, 0.055, 0.106, 0.205, 0.135, 0.365]$$

or, when only $P_{(6)}$ is denormalized:

$$P_{(6)} = [0.040, 0.078, 0.152, 0.109, 0.211, 0.409]$$

This solution represents smooth extrapolation of the previous case. All constraints are satisfied. Expected value is 4.6. However, p_7 and p_8 are not equal since that was not the requirement any more. These variables were mentioned in the system of constraints so they participate in the process of equalization, but only to some extent. In this case (after denormalization), $p_7 = 0.270$ and $p_8 = 0.730$. This is far from being equal, the ratio p_8/p_7 is 2.7. We can make them closer to being equal by forcing them to contribute more significantly in the optimization process. This can be accomplished by redefining them in such a way that the larger mass of the probability is concentrated in them. If the constraints $p_6 = p_1 + p_2 + p_3$ and $p_7 = p_4 + p_5 + p_6$ are replaced with $p_6 = 9p_1 + 9p_2 + 9p_3$ and $p_7 = 9p_4 + 9p_5 + 9p_6$ only the 10% of the probability mass will remain in the old probabilities and 90% will be concentrated in the new probabilities. This will make new probabilities more significant in the equalization process, but the first constraint has to be redefined to reflect the fact that old probabilities, that define it, now contribute 10 times less. The new set of constraint is:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 + 0p_7 + 0p_8 = 0.46$$

$$9p_1 + 9p_2 + 9p_3 + 0p_4 + 0p_5 + 0p_6 - 1p_7 + 0p_8 = 0$$

$$0p_1 + 0p_2 + 0p_3 + 9p_4 + 9p_5 + 9p_6 + 0p_7 - 1p_8 = 0$$

The corresponding MEM solution is:

$$P_{(8)} = [0.001, 0.007, 0.031, 0.002, 0.010, 0.049, 0.348, 0.552]$$

or, when only $P_{(6)}$ is denormalized:

$$P_{(6)} = [0.014, 0.066, 0.307, 0.022, 0.104, 0.487]$$

New probabilities p_7 and p_8 are now closer to being equal since ratio p_8/p_7 is 1.6.

We can push this process further in that direction by making old probabilities contain only 2% of the probability mass, which is equivalent of making new probabiities 50 times more important.

The new set of constraints is now:

$$1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + 6p_6 + 0p_7 + 0p_8 = 0.092$$

$$49p_1 + 49p_2 + 49p_3 + 0p_4 + 0p_5 + 0p_6 - 1p_7 + 0p_8 = 0$$

$$0p_1 + 0p_2 + 0p_3 + 49p_4 + 49p_5 + 49p_6 + 0p_7 - 1p_8 = 0$$

The corresponding MEM solution is:

$$P_{(8)} = [0.000, 0.000, 0.009, 0.000, 0.000, 0.010, 0.444, 0.536]$$

or, when only $P_{(6)}$ is denormalized:

$$P_{(6)} = [0.001, 0.019, 0.434, 0.001, 0.023, 0.523]$$

New probabilities p_7 and p_8 are now even closer to being equal since ratio p_8/p_7 improved to 1.2.

For significance of new probabilities equal to 100, the corresponding probabilities are $P_{(8)} = [0.000000, 0.000038, 0.004603, 0.000000, 0.000044, 0.005314, 0.459509, 0.530491]$, $P_{(6)} = [0.0000, 0.0038, 0.4603, 0.0000, 0.0044, 0.5314]$ and ratio $p_8/p_7 = 1.15$.

The process that is described shows that it is possible to adjust MEM for some constrained optimization problem and then guide it in the desired direction, but there is no universal way how to do it, each problem has to be investigated separately.

4 Conclusion

The maximum entropy method can successfully be used for optimization with constraints that are represented by under-determined systems. A software system is developed that includes standard MEM solution with some improvements which include a heuristics for speeding up the calculations. For each particular application a problem has to be transferred into the form usable for the MEM. This often leads to MEM solution that is only feasible. Introduction of artificial variables and appropriate coefficients allows to guide optimization process in the desired direction. The system was tested on the computer network design problem where good quality initial topology and routing were obtained. This software system represents a tool that is universal for all MEM applications, each particular problem, however, requires very careful adjustments and that part can not be automatized.

Acknowledgment: This paper is founded from the research Project No. 144007, Ministry of Science, Republic of Serbia.

References

[1] Ding YS, Zhang TL, Gu Q, et al., Using Maximum Entropy Model to Predict Protein Secondary Structure with Single Sequence *Protein and Peptide Letters* Volume 16, Issue 5, pp. 552-560, 2009

[2] Tuba, Milan: Maximum Entropy Method and Underdetermined Systems Applied to Computer Network Topology and Routing, Plenary Lecture, 9th WSEAS International Conference on APPLIED INFORMATICS and COMMUNICATIONS (AIC '09) Moscow, Russia, August 20 - 22, 2009, chapter in Recent Advances in Applied Informatics and Communications, A Series of Reference Books and Textbooks, WSEAS Press 2009, pp. 18 and 127-132.

[3] Xuan Wang, Lu Li, Lin Yao, Anwar, W., A Maximum Entropy Approach to Chinese Pin Yin-To-Character Conversion. *2006 IEEE International Conference on Systems, Man, and Cybernetics*, 2006, Taipei, Taiwan

[4] Teh, Chee Siong Lim, Chee Peng, A probabilistic SOM-KMER model for intelligent data analysis, *WSEAS Transactions on Systems* Vol. 5, no. 4, pp. 825-832. Apr. 2006

[5] Zhengmao Ye, Habib Mohamadian1, Yongmao Ye, Practical Approaches on Enhancement and Segmentation of Trimulus Color Image with Information Theory Based Quantitative Measuring, *WSEAS Transactions on Signal Processing*, Issue 1, Volume 4, January 2008, pp. 12-20

[6] Heric, Dusan; Zazula, Damjan, Reconstruction of Object Contours Using Directional Wavelet Transform, *WSEAS Transactions on Computers*. Vol. 4, no. 10, pp. 1305-1312. Oct. 2005

[7] Ciavolino E, Dahlgaard JJ, Simultaneous Equation Model based on the generalized maximum entropy for studying the effect of management factors on enterprise performance, *Journal of Applied Statistics* Volume 36, Issue 7, pp. 801-815, 2009

[8] Aladdin Shamilov, Generalized entropy optimization problems and the existence of their solutions, *Physica A: Statistical Mechanics and its Applications*, Volume 382, Issue 2, August 2007, pp. 465-472

[9] Tuba, Milan: Cost Function for Communication Links in Computer Networks, *Bulletins for Applied Mathematics (BAM)*, LXXIII-1028/94, pp. 115-122, Budapest, 1994

[10] Tuba, Milan: A Mathematical Model for Routing Comparison in Computer Networks, *Bulletins for Applied Mathematics (BAM)*, LXXXVI-A 1565/98, Arad, July 1998, pp. 493-503

[11] Tuba, Milan: Parameters for the Internet Optimization on the Local Level, *Applied & Computing Mathematics*, Vol. II, pp. 139-142, Kosice, 1997

[12] Tuba, Milan: Relation between Static and Dynamic Optimization in Computer Network Routing, *Recent Advances in Artificial Intelligence, Knowledge Engineering and Data Bases*, WSEAS Press 2009, pp. 484-489

[13] Tuba, Milan: Computer Network Routing Based on Imprecise Routing Tables, *WSEAS Transactions on Communications*, Issue 4, Volume 8, April 2009, pp. 384-393

[14] Abd-El-Barr M: Topological network design: A survey, *Journal of Network and Computer Applications*, Volume 32, Issue 3, pp. 501-509, 2009

[15] Karavetsios, P., Economides, A.: Performance Comparison of Distributed Routing Algorithms in Ad Hoc Mobile Networks, *WSEAS Transactions on Communications*, Vol. 3, Issue 1, 2004, pp. 317-321

[16] Sokullu, R., Karaca, O.: Comparative Performance Study of ADMR and ODMPR in the Context of Mobile Ad Hoc Networks and Wireless Sensor Networks, *International Journal of Communications*, Issue 1, Volume 2, 2008, pp. 45-53

[17] Kumar, D., Bhuvaneswaran, R.: ALRP: Scalability Study of Ant Based Local Repair Routing Protocol for Mobile Ad Hoc Networks, *WSEAS Transactions on Computer Research*, Vol. 3, Issue 4, Apr 2008, pp. 224-233

Milan Tuba
Megatrend University Belgrade
Faculty of Computer Science
Bulevar umetnosti 29
SERBIA
E-mail: *tubamilan@ptt.rs*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Service Oriented Architecture Considerations for Distributed Intelligent Control of Modern Manufacturing Systems

Gabriela Varvara, Carlos Pascal, Doru Pănescu

Abstract

The paper presents some architectural issues on the implementation of software systems designed to support the control of modern manufacturing processes. In the introduction some challenging aspects of modern manufacturing and market problems are pointed out. Then the concepts of an architectural manufacturing paradigm, based on dynamic recursive organization of holons for reconfigurable systems, are synthetically presented. In order to implement control systems that support the holonic recursive decomposition, the authors add to classical multi-agent distributed implementation the Service Oriented Architecture (SOA) characteristics. This development is justified by the necessity to sustain the dynamic cooperation among holons identified during the modeling decomposition phase, as exemplified in the paper, and, also, must be adapted to the characteristics of manufacturing systems as illustrated by the proposed architecture.

1 Introduction

1.1 The Modern Manufacturing Systems – problems, challenges, scenarios

The 21st century market problems required significant changes within the manufacturing systems and the development of associated emerging technologies. Some critical business aspects that need to be addressed are worthwhile to be mentioned [3]: the regular reactivity to "rush orders" and new arrived specification from the customer, an active balance between volume and variety of the production within a single shop floor, the demand for short delivery times for customer-specific products and the need to tightly integrate all the supply chains to hold minimal reserve stocks. All these problems result from everyday plant scenarios and imply a specific response from the manufacturing system as detailed in the next paragraph. The "rush order" reactivity scenario has some characteristics: a rush order can be introduced at any time, has electronic financial credit to buy manufacturing services, is able to negotiate with the existing orders to gain access to services as soon as possible and behaves in order to maximize its profit limits. A mass customization scenario involve flexible choice to meet together different orders that reflect changing of customer requirements with minimizing the stocks (warehouse and out). Another usual scenario refers to the existence of tightly integrated supply chains; the manufacturing system behaves in a proactive manner to find the optimal solution to outsourcing some production by maximal utilization of available resources through cooperation for real-time reaction to changing demands. The above mentioned scenario is tightly coupled with another one

devoted to a high degree of agility and flexible reconfiguration in order to face a great variety of customer demands and/or resources breakdown. This means the capacity to coordinate different resources with their various facilities, multiple and changing ways to make a product, both low volume high variety and high volume low variety manufacturing in the same environment and regular operations of adding, removing or reconfiguration of production machine resources.

In order to deal with these new complexity challenges, the well known centralized solutions to control factories have major drawbacks namely slow reactivity, generation of operational bottlenecks and finally reaching of critical failure points. Consequently, the domain research efforts were focused to develop flexible, fault-tolerant control mechanisms. The result was a new technology that combines both real-time control strategies and deliberative distributed information processing. It brings a new vision on manufacturing process founded on a complete decentralization of the control. As a result of its popularity among the domain researchers it becomes a paradigm of agility named "Holonic Manufacturing Systems (HMS)".

1.2 Holonic Manufacturing Systems – architectural paradigm for modelling of modern production processes

The HMS concepts address all the critical challenges of modern manufacturing and its declared purpose is to develop architectures and technologies for open, distributed, intelligent, autonomous and cooperative systems. As stated by the HMS consortium, there are a few concepts that govern the holonic vision, namely [9]:

- The holon, as an autonomous and cooperative building block of manufacturing systems for transforming, transporting, storing and/or validating information and physical objects. The holon has always an information processing part and often a physical processing part. It is a recursive structure that can be modeled as a tree structure; the nodes from intermediate positions are holons that contains other holons until no more decomposition is possible for the holons from the leaf nodes.
- The autonomy defined as a capability of a holon to create and execute its own plans and to maintain its functionality.
- The cooperation viewed as a process of the execution by a set of holons of mutually acceptable plans.
- Self-organization as the ability to make dynamic configurations of holons addressing a specific production goal.
- Holarchy – the form of holon organization founded on basic rules of cooperation that act in the direction of achieving a goal or objective. This entity defines the limits of holon autonomy.

As a consequence, the holon structure provides a decentralized "bottom-up" approach to develop complex manufacturing systems that ensure a high responsiveness and an agile behavior to changing production environment. Moreover, the holonic concept covers to a certain degree an open model, intelligent and prepared to participate in negotiation processes.

In order to apply the above abstract concepts, it was adopted one of the most used generic architecture PROSA [2] that proposes three different types of holons: order (OH), product (PH) and resource (RH).

The order holons manage the production requirements originating from direct customer demands or other external bodies such as a company/department situated upstream in the supply chain, or even anticipated production needs. They assign priorities; give the options to achieve the current demands by loading the local manufacturing resource holons or outsourcing partially or totally the job.

Resource holons provide all the generic active entities from a manufacturing system such as machines, transportation/storage devices, visual inspection stations or even independent companies.

On the other hand product holons provide knowledge on how to achieve manufacturing objectives, production best recipe/processing guides from multiple options and can disseminate information among other holons; they can offer sometimes and to some degree expert advice.

For complex expertise an expert holon type (EH) with an appropriate architecture can be defined too. As a particular type, a configuration holon (CH) can be created [9] for systems capable of changing their configuration to adapt for external order changes or to internal disturbances. It selects, based on the process plan received from the PH, from more candidate resource configurations capable to accomplish the order, the most efficient one. It is an iterative process based on reconfiguration costs. As soon as PH will be informed by CH about the current configuration, it will send appropriate process information to the RHs. For manufacturing contexts defined by rapid order changes and/or frequent resources breakdown, the RH state will change dramatically. The CHs will be informed about these new states and reconfiguration plans will be started. The optimal resource configuration founded will be sent to OH that, based on the updated product information received from PH, will transmit a new order to all the RHs from the current configuration.

2 Decentralized control and cooperation in HMS

HMS vision proposes [10] for the entire manufacturing system the holarchy model obtained by dynamic aggregation of individual cooperation domain. For example, the cooperation domain of an order holon requests tasks to product and possible resource holons. The entities that are interested to cooperate for achieving the assumed task interact inside a new established cooperation domain. The process continues until no manufacturing tasks can be identified. In order to illustrate this aggregation concept the next assembling scenario it is considered: the goal refers to production of two stacks of pallets, the first containing 3 pallets and the second with four pallets. The same pallet type can appear in different stacks. The following diagram depicts as a tree the dynamic recursive cooperation domains for the holarchy of the proposed manufacturing process (see Fig. 1). The higher level of cooperation for the proposed holarchy [8] is made as a consequence of receiving a manufacturing order from the customer and it is the result of the associated order holon named OH1. The holon entities that accepted to cooperate inside this domain are the product holons for each stack and the additional resource holons RH_1, RH_2 involved in stack transportation from/to loading/storage positions. At the second level of cooperation, each stack product holon will try to identify and aggregate the resources necessary for building the stacks according to their structure (stack I has three pallets of type A, B, C put in a predefined order, etc.). Product holons for pallets A, B and C will agree to cooperate for accomplishing this task. Then the raw part fill-in process for the pallets, according to assembling rules and known recipes, will impose the creation of the third cooperation level domain. It can be noticed that these are the most interior cooperation levels and are based on the participation of only resource holons as the active manufacturing entities of the system.

The configuration holons CH_1, CH_2 and CH_3 will be generated for each level of the holarchy: CH_1 is necessary to establish the resource configuration for assembling the different stacks of the product, CH_2 will be involved in stack formation and CH_3 chooses the configuration for the pallet filling process. The achievement of the global goal will generate configuration holons at different levels of the holarchy. They are dismissed when a current mission finishes and vanish when a reconfiguration process starts.

The cooperation among all the holons inside the holarchy imposes a limitation of the holon autonomy; only the mutually acceptable plans will be started in order to accomplish the global goal. All the cooperation domains along the established holarcy are virtual and dynamic. The connection arrows are dynamically created. They will be implemented using service invocation

mechanism considered to be an appropriate approach. Any task alteration or introduction of new resources will impose real-time reorganization of production resources in different cooperation domains in order to fulfill the customer request.

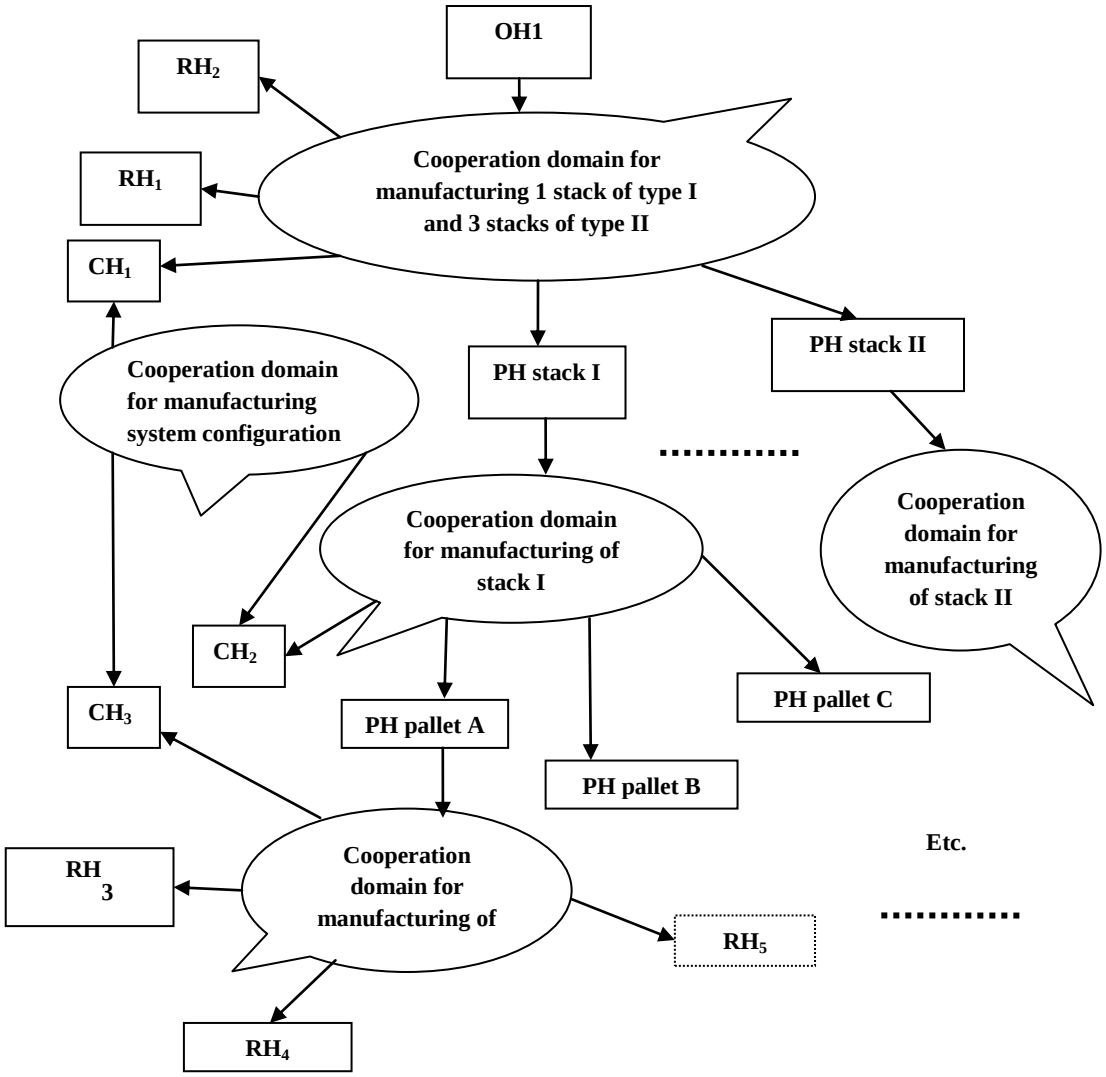


Fig. 1 The Cooperation Domains inside the Holarchy

The decentralization proposed by HMS is based on dynamic creation, breakdown and reconfiguration of virtual cooperation domains. This must be sustained by simultaneous decoupling of control and information processing. The key aspect of such an approach is to separate the planning, controlling, coordination, execution and communication mechanisms for each cooperation domain of the holarchy.

As illustrated in Fig. 2, dynamic cooperation domains will be created along the holarchy for each specific activity. They result as a collaboration agreement among heterarchically organized holons PH, and RH_1 to RH_n. Applying this model to the example from Fig. 1, "PH pallet A" knows the strategy to fill in with parts the pallet of type A. Its goal is to initiate a manufacturing process according to the above mentioned strategy. Consequently, it will initiate a collaboration domain with all the available resource holons able to assume palletizing goals. For instance, at a given moment the pallet assembling plan will impose the attainment of the sub-goal of

transferring of a raw/processed part. The PH will consult all the RHs available for this task. On the other hand the appropriate resource holons will start a planning phase devoted to the selection of an own appropriate plan to realize the transfer operation. It will often imply collaboration with other resource holons that agree to help for reaching the system goal by activating the visual inspection system, other manufacturing cells, resource/part assignment for the assumed operation and the evaluation of the operational costs. All the resource offers will be analyzed by the PH during its own planning phase [8] and finally the most efficient resource holons will be charged to execute the transfer and an execution phase will start.

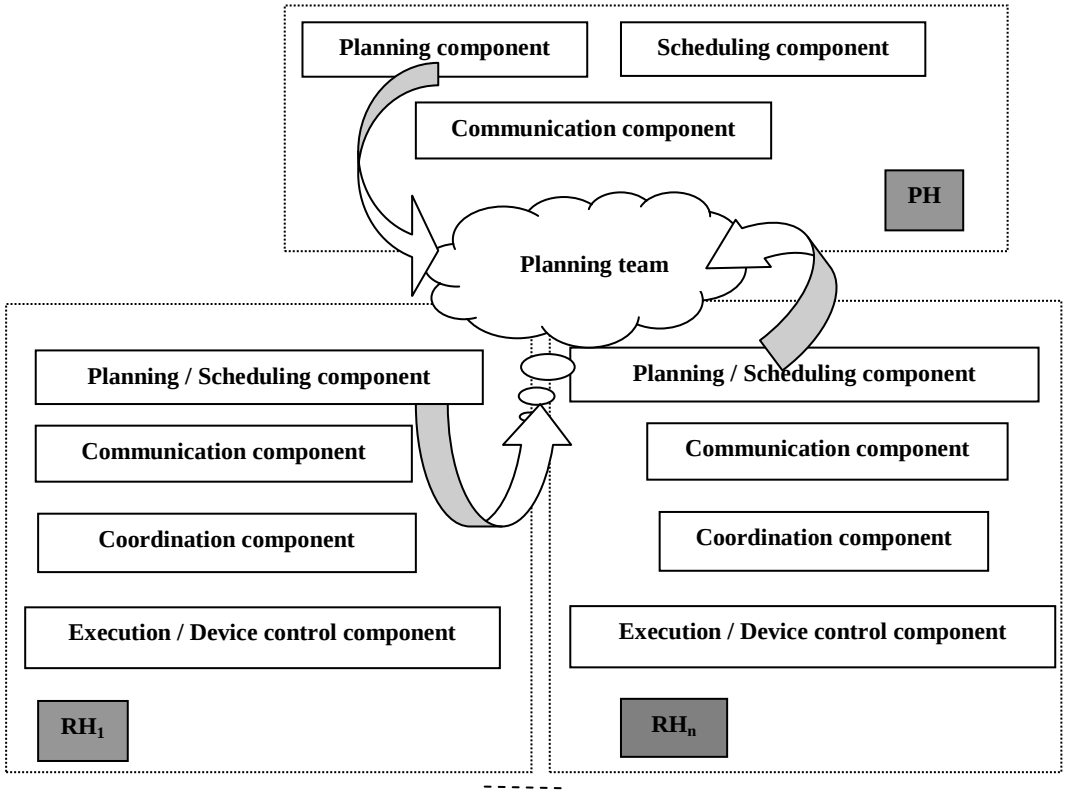


Fig. 2 HMS decentralization

Following the operation planning model, the other activities can determine a dynamic formation of collaborative teams that will assume both their own goals and the goals of the ongoing organization they belong to.

As it can be easily noticed, the holonic organization of the manufacturing modern processes involves decentralization mechanisms along with dynamic collaborative domains formation. Both characteristics define the system flexibility that will be the subject of the following paragraphs. As already sketched, in order to increase the system flexibility two aspects were taken into consideration: the decentralization of holons activities and the operational standardized aggregation of the resources in new defined cooperation domains.

3 Considering SOA for the HMS decentralized control

The HMS philosophy imposes a new architectural control design with a strong orientation to ad-hoc applications made up of loosely coupled functionality units, that are intrinsically

unassociated and have no calls to each other embedded in them. The already identified chunks of functionality will be implemented under a distributed multi-agent environment using a new software paradigm, namely the service oriented architecture (SOA) [1], [7]. The agent major characteristics will be the autonomy and self determination. The agent mobility is not so important for a manufacturing set-up, but sometimes it can be also taken into consideration. Each identified holon type of activity can be designed as a service and the overall control system will be implemented using a multi-agent wrapper model FIPA compliant for inter-agent communication at holon level and SOA guidelines for low-level middleware communication. This development model is able to offer appropriate rules for grouping of pieces of functionality named manufacturing services from the top of management level to the bottom production and distribution tasks along with supporting activities. The Open Group offers the following definition [9]: SOA is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform and standardized manner to offer, discover, interact and use capabilities to produce desired effects consistent with measurable preconditions and assumed expectations. In the opinion of the enterprise architects SOA will help business to adapt in a cost-effective manner to changing market conditions in the sense of promoting the reuse of services (functions at macro level) rather that the reuse of classes (functions of micro level). It will further simplify the interconnection with existing information technology legacy assets. Moreover, SOA promotes the separation of consumers from the service implementations that can be run on various different platforms and be accessed across networks.

The main goals of service oriented architecture and development is to construct complex software-intensive systems from a set of universally interconnected and interdependent blocks named services [1],[7] with the increasing of:

- The interoperability as information exchange and reusability;
- The federation as application of unifying mechanism while maintaining the autonomy and the self-governance;
- The alignment of business and technology domain.

The basic entity of any SOA is the service that comprises a stand-alone unit of functionality available only via a formally defined interface and that is implemented according to the principles of abstraction, autonomy, composability, discoverability, formal contract, loose coupling, reusability and statelessness. All these principles come in agreement with the concepts of holonic manufacturing architectures and the service-orientation seems to be the appropriate development strategy for HMS.

4 SOA implementing guidelines for HMS

Generally speaking, the holon should be viewed as a service able to appropriately combine both the SOA and HMS concepts. Thus, the holon is an entity which offers services and may use other holon services to accomplish its demanded goals. Making use of the SOA architecture, the Fig 3 sketches the grouping of the different holon types (order, product and resource holons) as an ad-hoc collection of holon services. The main propose of such a collection is to group stand-alone holon services into a service space where services are used/reused to carry out the proposed goals. The entire architecture is supported by the CORBA [5] standard which is a cross-platform approach with non-proprietary direction. This standard is used in HMS to resolve the integration problems of the physical devices in resource holons. It is adopted to the whole architecture in order to maintain a homogeneous approach. Moreover, the Foundation for Intelligent Physical Agents (FIPA) offers standards for the interoperation of heterogeneous software agent platforms, but still does not provide the means to integrate several physical resources in a common software structure. The technique to implement services using this standard is relied on the object invocation method through the *Bus-based Architecture* which includes the Internet Inter-Orb

Protocol (IIOP) communication layer. CORBA comes with a set of services that improves the communication layer and put forward the means for the holon service composition. Some CORBA services come near the holon services into Fig. 3, to supplement the scheme. Thus, the *Naming Service* manages a binding list between names and object references. It is used by each holon to register its services under friendly unique names and to locate the services by using the known names. To enable holons locate services based on the functionality and the quality of required services without knowing their names, the *Trader Service* is employed. Another useful service is the *Event Service,* which offers a mechanism for bidirectional service calls from the consumer holon to the service holon and in the reverse way. Besides these ones, others CORBA services [4] may be used, in order to solve specific issues from the HMS (e.g. service life cycle, concurrence control, relationships, etc.).

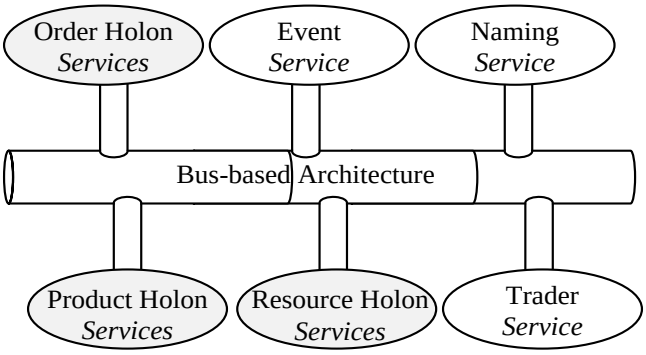


Fig. 3 SOA concept for HMS

From the implementation point of view, it is efficient that every holon should be designed to expose a single service, enabling the receiving of goal's description (Fig. 4). The service contains a remote object that encloses a single method to receive the content of the goal that describes the activity/activities that should perform by the service holon, and further methods (useful to administrate holon services). An Interface Description Language (IDL) file presents the scheme of the holon service that is the basis of the remote object interface; in CORBA, this object is referred as skeleton interface. The consumer of the holon service obtains the necessary object (known as the stub), for making a dynamic link with the reference of the remote object, from IDL description. The dynamic link between skeleton and stub can be established, if consumer service has the reference of the remote object (skeleton). Because the holon offers a service and may invoke other services, it knows the definition of both the skeleton and stub interfaces. The remote object is obtained from the implementation of the skeleton interface. Since there is a single service, this should deliver to the trader service at the activation time a list with the types of goals that can be carried out by the respective holon.

For example, in the case of a product holon the list can contain goal types like: stack or pallet products, or both product types. These lists are obtained at the time when holons publish the location and the reference of the remote objects and are used by the *Trader Service* to discover the appropriate holons for a specific goal. It maintains a repository with these remote objects and their associate proprieties that can be exposed if there is a request from other holons(Fig. 4). As an example, the order holon OH1 from Fig. 1 can request the addresses of the product holons PH Stack I or PH Stack II that can handle the stack product goal. If the order holon has received at least one remote address, it means there is a holon in the holonic system that may perform the stack assembling goal. Using these returned addresses, the holon can bind the stub objects to the object references. Through the service remote method invocation (the remote methods of the objects are invoked), the body of the goal message is transferred between holons. This communication model is made in a synchronous approach, which means that the client uses the

remote method and waits the server to respond. The use of the synchronous mode for SOA and HMS schemes is not enough, and the asynchronous mode is needed, too.

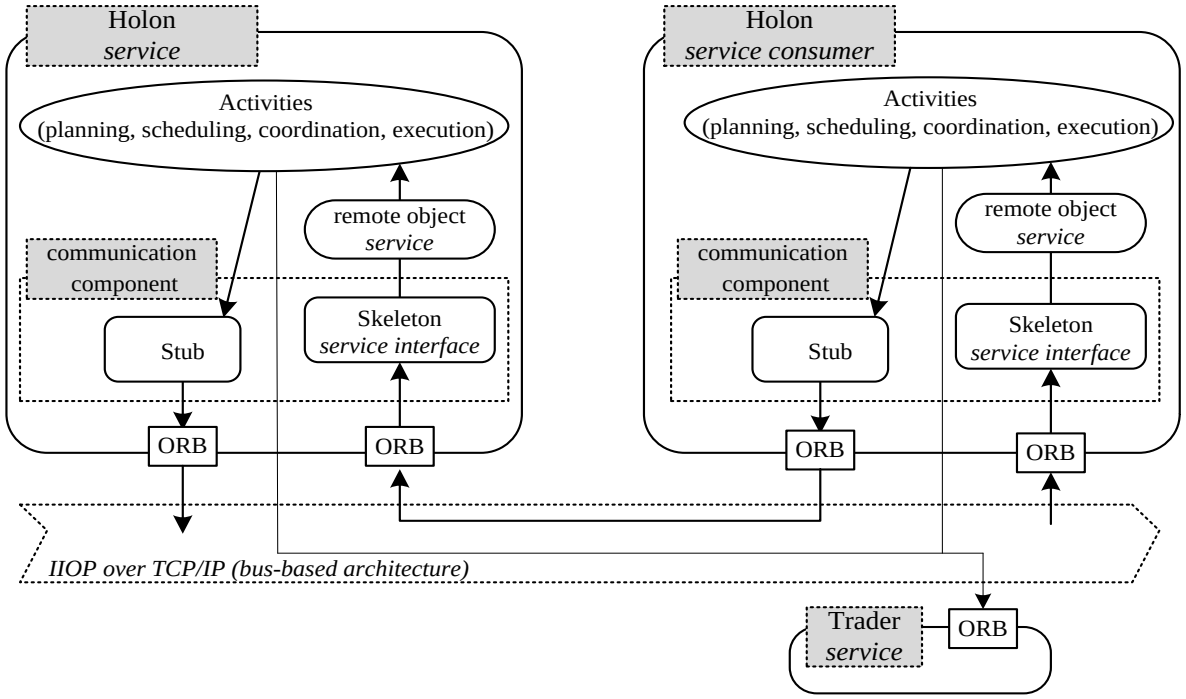


Fig. 4 The holon service model

For the asynchronous approach two mechanisms were used: through the remote method the message goal is transferred together with a remote reference object, which is used by the holon service to return an answer to the respective call; the other possibility is to use the *Event Service* to facilitate the bidirectional message exchange. In the first mechanism, the service' consumer needs to expose a remote object, able to handle an asynchronous response. Before the consumer forwards the goal, it activates the remote object which can be used by the holon service. Thus, the service knows the interface of the remote object (stub), which is identical for all the holons in the answering case. After receiving the goal message and an address, the service responses immediately and the customer should wait the appropriate answer in its active remote object at the given address. The second applicable mechanism that was used is more dynamic, and it is guided by the concept of subscribing and notification or the so called "change notification" [6]. Both holons (service and consumer) are using an event channel, which is a middleware between them, providing asynchronous communication of event data. For the communication process, three schemes (push-style, pull-style and mixed style schemes) can be used to involve the event channel [6]. As a service, the holon publishes a goal through the event channel, and this notifies all the consumer holons, which subscribed in the channel. In this way, the task to find holons that can fulfill a certain goal type is accomplished by the Event Service. Moreover, the consumer holons can pull the services through the channel to return an answer for the received event. Internally, when the holon receives a goal message through its remote object, the message is forwarded to the deliberative part to select a specific plan [8]. The chosen plan can either determines to carry out alone the received goal, or it becomes a consumer if the goal is divided in sub-goals. In the last case, the *Trader Service* is again used to discover holons for solving the sub-goals. The dynamic goal decomposition selects and uses the needed holon services from the service space. The solution and the performance of the completed goals depend on the dimension of the service space – the number of holons.

5 Conclusions

The purpose of the presented work is to sketch the decentralized control mechanisms existing in modern manufacturing systems and to identify software solutions in order to implement them. Starting from the holon entity, as recursive self organized unit and on the basis of hetararchical organization principles (the replacement of master-slave hierarchic control with collaboration relationships established after a negotiation process), the possible dynamic cooperation domains among different holon types in order to achieve a mutual goal/task were identified. In order to ensure the required flexible behavior of the control system, based on negotiation and real-time regrouping of active entities, the classical multi-agent implementation was improved by the service oriented architectural design vision. The paper proposes and gives some details on the correlation between SOA and HMS organizing concepts and proposes practical portable solutions. The authors have already obtained valuable results of the proposed architecture using the JACK ® Multi Agent Platform coupled with a CORBA based services implementation that will be the subject of further reports.

Acknowledgement: This paper is founded from the research project SOFHICOR, Contract no. 11-042/2007.

References

[1] M. Bell, *Introduction to Service-Oriented Modeling. Service-Oriented Modeling: Service Analysis, Design, and Architecture*. Wiley & Sons, 2008.
[2] V. Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters, Reference architecture for holonic manufacturing systems: PROSA, *Computers in Industry*, vol. 37, pp. 255 – 274, 1998.
[3] M. Fletcher, T. Neligwa, chapter : An HMS Operational Model, from the volume *Agent Based Manufacturing,* editor S.M. Deen, Springer-Verlag, New York Heidelberg Berlin, pp. 163 – 192, ISBN:3-540-44069-0, 2003.
[4] T. Emmerich , *Engineering Distributed Object*, John Wiley&Sons, Baffins Lane, pp. 73-98, 190-200, 2000.
[5] OMG, *Common Object Request Broker Architecture Specification v3.1 part 1*, Technical report, Object Management Group, Inc, Needham, pp 13 – 28, 2008.
[6] OMG, *Event Service Specification v.1.2*, Technical report, Object Management Group, Inc, Needham, 2004.
[7] T. Erl, *Service-oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River: Prentice Hall PTR, 2005.
[8] G. Varvara, Multi-agent Control System development for Manufacturing Processes based on Holonic Scenarios, *Proceedings of the RAAD 2009 18th International Workshop on Robotics in Alpe – Adria - Danube Region*, Brasov, Romania, May 2009.
[9] L.Wang, K. Chen, and Y.S. Ong (Eds.). An Agent-Based Holonic Architecture for Reconfigurable Manufacturing Systems. *ICNC 2005, LNCS 3612*, Springer-Verlag Berlin Heidelberg, pp. 622-627, 2005.
[10] A. Koestler, *The ghost in the machine*, in: Arkana Books, 1971.

GABRIELA VARVARA
"Gheorghe Asachi" Technical University
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Applied Informatics
Mangeron 53A, 700050, Iaşi
ROMANIA
E-mail: gvarvara@ac.tuiasi.ro

CARLOS PASCAL
"Gheorghe Asachi" Technical University
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Applied Informatics
Mangeron 53A, 700050, Iaşi
ROMANIA
E-mail: cpascal@ac.tuiasi.ro

DORU PĂNESCU
"Gheorghe Asachi" Technical University
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Applied Informatics
Mangeron 53A, 700050, Iaşi
ROMANIA
E-mail: dorup@ac.tuiasi.ro

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Algebraic model for the synchronous D flip-flop behaviour

Anca Vasilescu

Abstract

The complex systems like modern computers are designed using multiagent concepts. Agent interactions, namely communication and synchronization, fit very well the corresponding hardware components interactions. The agent-based approach is applied here to cover the digital logic circuits design and verification.

In order to achieve the computer components behaviour modelling and the formally verification of the involved agents properties, we use in this paper the SCCS as the appropriate process algebra supporting our interest of developing an algebraic model for the computer system behaviour.

In this paper we consider the specific structure for the synchronous D flip-flop as example for modern computer elementary memory cells. We define the appropriate specification and implementation agents for algebraic modelling of the given circuits behaviour and we formally prove the corresponding bisimilarities between the target agents. As application, we modify the obtained model in order to prepare an appropriate solution for algebraic modelling of a general synchronous logic circuit based on a D flip-flops matrix arrangement.

1 Introduction

The complex systems like modern computer systems are designed using multi-agent concepts. Agent interactions, namely communication and synchronization, fit very well the corresponding hardware components interactions. A component-based style allows components to be specified and verified individually. Larger combinations of trusted components can then be verified more easily. In this paper we use a correctness meaning based on the bisimilarity relation between agents.

There are two modern approaches for system verification, namely model checking and theorem proving. In this paper we apply a theorem proving approach based on the SCCS process algebra for modelling and verifying the appropriate agents provided by the D flip-flops bahaviour. Process algebra are well known as mathematical tools for describing and analysing the concurrent and communication systems like the result of interconnecting hardware components, each hardware component being modelled as an agent. Considering the digital logic level of the computer architecture description, the agent-based approach is applied here to cover both the digital logic circuits design and verification.

Having an algebraic-based model for a multi-agent system has the main advantage of ensuring the reliability and correctness of the core processes of the computer operation following the accurate results provided by a formal methods based verification approach.

The present results follow the contributions of the author already obtained in [4], [6], [7] for modelling the behaviour of different hardware components involved in a modern computer system architecture.

The final outcomes of this paper consist in defining the specification and implementation agents for modelling the synchronous D flip-flop behaviour, proving the bisimulation equivalence between the corresponding agents and preparing the appropriate agents for later modelling the behaviour of a complex memory component based on a matrix organization of D flip-flops.

2 Preliminaries

2.1 Flip-flops. Computer memory organization

Modern computer systems are based on digital components dealing with binary variables and with operations that assume a logical meaning. From the structural point of view, the manipulation of binary information is done by *logic circuits*, combinational or sequential. For a *combinational circuit* or CLC the outputs at any given time are entirely dependent on the inputs that are present at that time. Although every digital system is likely to have a combinational circuit, most systems encountered in practice also include storage, memory elements, which require that the system has to be described in terms of *sequential logic circuits* or SLC.

A *flip-flop* is a sequential circuit, a binary cell capable of storing one bit of information. It has two outputs, one for the normal value and one for the complement value of the bit stored in it. A flip-flop maintains a binary state until it is directed by a clock pulse to change that state. At the different levels of detailing, the theoretical flip-flop might be asynchronous, but the synchronous models are widely used in practice. The difference among various types of flip-flops is the number of inputs and the manner in which the inputs, both data inputs and the clock signal, affect the binary state. Depending on the number of data inputs, the most common types of flip-flops are: SR flip-flop, D flip-flop, JK flip-flop and T flip-flop [3]. In this paper we will consider the case of the synchronous D flip-flop structurally based on the synchronous SR flip-flop model.

Basically, a flip-flop is a one-bit memory. In order to store more data, we need to build a larger memory by linking several one-bit memories to form a cell, and then join many cells together such that one cell is on top of another and so on. In order to assure a straightforward modelling process for such a complex memory component, we prepare in this paper the detailed agents for modelling the behaviour of a specific SLC made up by an appropriate matrix of D flip-flops.

2.2 Process algebra SCCS

The process algebra SCCS, namely *Synchronous Calculus of Communicating Systems* is derived from CCS [1], [2] especially for achieving the synchronous interaction in the framework of modelling the concurrent communicating processes. In SCCS processes are built from a set of atomic actions A. Denoting the set of labels for these actions by Λ, an SCCS action is either (1) a *name* or an input on $a \in \Lambda$ denoted by a, (2) a *coname* or an output on $a \in \Lambda$ denoted by $\overline{a}$ or (3) an internal on $a \in \Lambda$ denoted by the action 1, identified with the empty product. In SCCS the *names* together with the *conames* are called the *particulate actions*, while an *action* $\alpha \in \Lambda^*$ can be expressed uniquely (up to order) as a finite product $a_1^{z_1} a_2^{z_2} ...$ (with $z_i \neq 0$) of powers of names. Note the usual convention that $a^{-n} = \overline{a}^n$.

An SCCS *process* P is defined with the syntax:

P	::=	nil	termination
	|	α:P	prefixing
	|	P+P	external choice
	|	P $\times$ P	product, synchronous composition
	|	P $\upharpoonright E$	restriction, $L \subseteq A \cup \overline{A}$ and $E = (A\text{-}L)^*$
	|	P[f]	relabelling with the morphism $f : A \cup \overline{A} \to A \cup \overline{A}$

In the restriction definition $E = (A\text{-}L)^*$ is the submonoid of A generated by the set difference $A\text{-}L$. By definition, the P $\upharpoonright E$ agent is forced to execute only the actions from the set E as the external actions.

The operational semantics for SCCS is given via inference rules that define the transition available to SCSS processes. Combining the product and the restriction, SCCS calculus defines the synchronous interaction as a multi-way synchronization among processes.

A formal approach such as the process algebra SCCS supports a way to relate two different specifications in order to show that those specifications actually describe *equivalent* concurrent systems, for some specific meaning of *equivalence*. In this section we use a concrete relation between two different specifications – a notion of refinement:

Impl *refines* Spec

where a low-level specification, namely Impl, refines a higher-level specification, namely Spec. For each of the next circuits, we construct both of these specifications as follows. The specification Spec is based on the definition of the circuit, while the specification Impl is based on the behaviour of that given circuit. As demonstration technique, we start with the specifications Spec and Impl and then we apply a set of SCCS-based algebraic laws in order to formally prove that the low-level specification, Impl, is correct with respect to the higher-level one, Spec. This correctness proof is based on the *bisimulation congruence*, the appropriate equivalence in the theory of concurrent communicating processes. Implicitly, this result of bisimilarity shows that the behaviour follows the definition of the given system and, on the other hand, it is a guarantee of using that model in other complex circuits.

3 Algebraic model for the synchronous D flip-flop behaviour

In this section we firstly define both specification and implementation agents for asynchronous SR flip-flops, synchronous SR flip-flops and synchronous D flip-flops, respectively. In the second part of this section we formally prove the corresponding agents bisimilarities.

It is helpfully for the next sections to define some generic agents, as follows.

$$\text{NODE} = in\,\overline{out} : \text{NODE}$$

$$\text{NODE1} = \sum_{i\in\{0,1\}} (in_i\overline{out}_i : \text{NODE1})$$

$$\text{NODE2} = \sum_{i\in\{0,1\}} (in_i\overline{up}_i\overline{down}_i : \text{NODE2}) \tag{1}$$

$$\text{NOT} = \sum_{i\in\{0,1\}} (in_i\overline{out}_j : \text{NOT}),\ \text{where } j = \text{NOT } i$$

$$\text{AND\_rec} = \sum_{i,j\in\{0,1\}} (andin1_i\ andin2_j\ \overline{andout}_k : \text{AND\_rec}),\ \text{where } k = i \text{ AND } j.$$

3.1 Algebraic model for the SR flip-flop behaviour

The logic diagram of an SR flip-flop consists in two NOR gates composed like in Figure 1. For the next algebraic models we will recognize the circuit lines as follows: σ for the input value S, ρ for the input value R, $\overline{\delta}$ for the output value Q and $\overline{\gamma}$ for the complemented output value $Q' = \text{NOT } Q$.

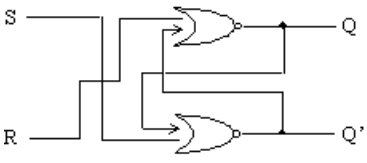


Figure 1: Asynchronous SR flip-flop

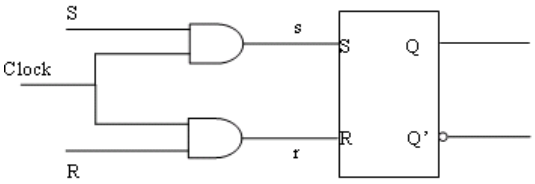


Figure 2: Synchronous SR flip-flop

The current value of a variable like σ_S shows that the circuit line σ is carrying the logic value S. If the current values on the outputs $\overline{\gamma}$ and $\overline{\delta}$ are m and n, respectively, then the specification Spec for the SR flip-flop behaviour might be [1]

$$\text{SpecSR}(m,n) = \sum_{i,j\in\{0,1\}} (\sigma_i\rho_j\overline{\gamma}_m\overline{\delta}_n : \text{SpecSR}(k,l)) \tag{2}$$

where the values k and l are defined by $k = i$ NOR n and $l = j$ NOR m.

In order to achieve the composition and to assure the fork of the output signal, we have to define two morphisms to make two appropriately relabelled copies of the NOR gate. These two morphisms are: Φ defined by the relabelling pairs $\alpha_i \mapsto \sigma_i$ and $\gamma_i \mapsto \gamma_i\alpha_i$, for $i \in \{0,1\}$ and respectively Ψ defined by the relabelling pairs $\beta_i \mapsto \rho_i$ and $\gamma_i \mapsto \beta_i\delta_i$, for $i \in \{0,1\}$.

We also define the set E of external actions, $E = \{\sigma_i, \rho_i, \gamma_i, \delta_i | i \in \{0,1\}\}$, in order to form the SCCS product between the two communicating NOR gate-agents via the *lines* $(\alpha_i, \overline{\alpha}_i)$ and $(\beta_i, \overline{\beta}_i)$. We usually abbreviate the subsets like $\{\alpha_0, \alpha_1\}$ by α.

Hence, the SCCS low-level specification (or implementation) for the SR flip-flop behaviour might be

$$\text{ImpSR}(m,n) = (\text{NOR}(m)[\Phi] \text{ x } \text{NOR}(n)[\Psi]) \upharpoonright E \tag{3}$$

This example shows how appropriate combinations of morphism, product and restriction may be used to model the channelling of data, here along the hardware wires.

Based on ideas from [1], it is also shown in [5] that the two previous specifications for the SR flip-flop behaviour are bisimulation equivalent (or bisimilar). So, the relation:

$$\text{SpecSR}(m,n) \sim \text{ImpSR}(m,n) \tag{4}$$

is established.

From the structural point of view, in order to obtain the synchronous SR flip-flop we consider the asynchronous circuit and we add an extra level of AND gates for involving the clock signal. The synchronous SR flip-flop is represented in Figure 2. We consider new indexed variables, namely CLK, for representing the clock input signal following the index values $c \in \{0, 1\}$.

We define two levels for specifying the synchronous SR flip-flop behaviour, a specification and an implementation, given by the agents, respectively:

$$\text{SpecSRs}(m,n,c) = (\text{SpecInput}(c) \times \text{SpecSR}(m,n)) \upharpoonright E\_SRmn(c) \tag{5}$$

and

$$\text{ImpSRs}(m,n,c) = (\text{ImpInput}(c) \times \text{ImpSR}(m,n)) \upharpoonright E\_SRmn(c) \tag{6}$$

where the set of external actions is $E\_SRmn(c) = \{CLK_c, \sigma, \rho, \gamma, \delta\}$ for a specific value of $c \in \{0,1\}$. Note that the parameters m and n have complemented values, by definition of the flip-flop.

It is important for the next results of this paper that we have just prove in [8] that these two agents are bisimulation equivalent based on the relation

$$\text{SpecSRs}(m,n,c) \sim \text{ImpSRs}(m,n,c) \tag{7}$$

3.2 Algebraic model for the synchronous D flip-flop behaviour

A D flip-flop is derived from an SR flip-flop by replacing the R input with an inverted version of the S input, which thereby becomes D like in Figure 3. For the synchronous D flip-flop it is essential that when the clock is reset the circuit does not operate, meaning it does not change the state, and when the clock is set the D flip-flop loads the D input.

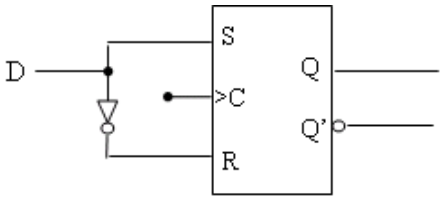


Figure 3: Synchronous D flip-flop

In this section we propose the SCCS algebraic models for both the specification and the implementation of the synchronous D flip-flop behaviour and we conclude by proving the bisimilarity of these two models.

Based on the previous definition of the synchronous D flip-flop, we consider its internal structure consisting of an entrance level of gates and an internal synchronous SR flip-flop. We define the agent SpecInD for specifying the behaviour of the entrance level of the D flip-flop as follows.

$$\text{SpecInD} = \sum_{D\in\{0,1\}} (\zeta_D \overline{\sigma}_S \overline{\rho}_R : \text{SpecInD}) \tag{8}$$

where $S = D$ and $R = \text{NOT } D$.

The appropriate specification agent for the synchronous D flip-flop behaviour is given by the relation

$$\text{SpecDs}(m,n,c) = (\text{SpecInD} \times \text{SpecSRs}(m,n,c)) \upharpoonright E\_Dmn(c) \tag{9}$$

where the set of external actions is $E\_Dmn(c) = \{CLK_c, \zeta, \gamma, \delta\}$ for each value of $c \in \{0,1\}$.

Let the agent ImpInD be the implementation agent for the entrance level of the D flip-flop. As expected, this lower-level specification is based on the operation of the intercommunicating logic gates combination. Using the generic agents already defined in (1), we have

$$\text{NODE2\_D} = \text{NODE2}[in_i \mapsto \zeta_D, up_i \mapsto \sigma_S] \text{ for each } D = i \text{ and } S = i$$

$$\text{NOT\_R} = \text{NOT}[in_i \mapsto down_i, out_j \mapsto \rho_R] \text{ for each } R = j \text{ and } j = \text{NOT } i.$$

We define the ImpInD agent as follows

$$\text{ImpInD} = (\text{NODE2\_D} \times \text{NOT\_R}) \upharpoonright E\_ImpInD \tag{10}$$

where the set of external actions is $E\_ImpInD = \{\zeta, \sigma, \rho\}$.

The appropriate implementation agent for the synchronous D flip-flop behaviour is given by the relation

$$\text{ImpDs}(m,n,c) = (\text{ImpInD} \times \text{ImpSRs}(m,n,c)) \upharpoonright E\_Dmn(c) \tag{11}$$

where the set $E\_Dmn(c)$ of external actions is the same as in the specification case.

Proposition 1 *The previous agents SpecDs(m,n,c) and ImpDs(m,n,c) for $(m,n) \in \{(0,1),(1,0)\}$ and $c \in \{0,1\}$ are bisimulation equivalent.*

Proof: The bisimulation relation '$\sim$' is a congruence over the class $\mathcal{P}$ of agents [1], [2]. Besides, it is already established in the previous relation (7) that $\text{SpecSRs}(m,n) \sim \text{ImpSRs}(m,n)$. Comparing the definitions (9) and (11) for the target agents $\text{SpecDs}(m,n,c)$ and $\text{ImpDs}(m,n,c)$ respectively, it follows that we only need to prove that:

$$\text{SpecInD} \sim \text{ImpInD}$$

We first evaluate the lower-level specification ImpInD:

$$\begin{aligned}
\text{ImpInD} &= (\text{NODE2_D} \times \text{NOT_R}) \upharpoonright E_ImpInD = \\
&= (\text{NODE2}[in_i \mapsto \zeta_D, up_i \mapsto \sigma_S] \times \text{NOT_R}[in_i \mapsto down_i, out_j \mapsto \rho_R]) \upharpoonright \\
&\upharpoonright \{\zeta, \sigma, \rho\} = \\
&= (\sum_{i\in\{0,1\}} (\zeta_D \overline{\sigma}_S \overline{down_i} : \text{NODE2_D}) \times \sum_{i\in\{0,1\}} (down_i \overline{\rho}_R : \text{NOT_R})) \upharpoonright \\
&\upharpoonright \{\zeta, \sigma, \rho\}
\end{aligned}$$

The parameter values are: $D = i$, $S = i$, $R = j$ and $j = \text{NOT } i$.
We apply the specific SCCS operational laws and the result is

$$\begin{aligned}
\text{ImpInD} &= \sum_{D\in\{0,1\}} (\zeta_D \overline{\sigma}_S \overline{\rho}_R : (\text{NODE2_D} \times \text{NOT_R}) \upharpoonright E_ImpInD)) = \\
&= \sum_{D\in\{0,1\}} (\zeta_D \overline{\sigma}_S \overline{\rho}_R : \text{ImpInD})
\end{aligned} \tag{12}$$

with $S = D$ and $R = \text{NOT } D$.
Comparing the definition (8) of the agent SpecInD with the previously proved relation (12) for the agent ImpInD, it follows that both these agents are solutions for the same equation $X = \sum_{D\in\{0,1\}}(\zeta_D\overline{\sigma}_S\overline{\rho}_R : X)$ up to a permutation of actions in the multi-particulate prefix action. Because here the variable X is guarded, the equation has a unique solution up to bisimilarity [2], so that the agents SpecInD and ImpInD are bisimilar, as required. □

We have just proved that

$$\text{SpecDs}(m, n, c) \sim \text{ImpDs}(m, n, c) \tag{13}$$

3.3 Specific synchronous D flip-flop models for SLC integration

In order to model the sequential circuits we have to apply two modifications to both the specification and the implementation of the synchronous D flip-flop agents as follows. The first modification consists in making obvious the initial state of the D flip-flop and defining separate agents for the D flip-flop operation with each of the input value $D \in \{0, 1\}$. The final agents of this step are modelling the synchronous D flip-flop operation starting from a specific initial state, with specific value on the D input and a certain clock signal. The second modification consists in unifying the previous agents by selecting the D flip-flop operation for a specific input value $D \in \{0, 1\}$, but for all the initial state combinations and all the clock signal combinations.

As the first modification we define the next agents for making obvious the initial state Q_n of the flip-flop and for selecting the D flip-flop operation for a specific Input value $D \in \{0, 1\}$:

$$\text{InDisI}(D, n) = Q_n\zeta_D\overline{\zeta}_D : \text{InDisI}$$

where $D, n \in \{0, 1\}$.

The appropriate specification agents are:

$$\text{SpecDs\_isI}(m, n, c, D) = (\text{InDisI}(D, n) \times \text{SpecDs}(m, n, c)) \upharpoonright E\_DsisI(c, D) \tag{14}$$

where the set of external actions is $E\_DsisI(c, D) = \{Q, \gamma, \delta, CLK_c, \zeta_D\}$ and the parameters are $(m, n) \in \{(0, 1), (1, 0)\}$, $c \in \{0, 1\}$ and $D \in \{0, 1\}$.

The appropriate implementation agents are:

$$\text{ImpDs\_isI}(m, n, c, D) = (\text{InDisI}(D, n) \times \text{ImpDs}(m, n, c)) \upharpoonright E\_DsisI(c, D) \tag{15}$$

with the same set of external actions as in the specification case.

Proposition 2 *For all $(m, n) \in \{(0, 1), (1, 0)\}$, $c \in \{0, 1\}$ and $D \in \{0, 1\}$ the previous agents SpecDs_isI(m, n, c, D) and ImpDs_isI(m, n, c, D) are bisimulation equivalent.*

Proof: The result is obvious considering that the bisimulation relation '$\sim$' is a congruence over the class $\mathcal{P}$ of agents [1], [2] and it is already established in the previous relation (13) that $\text{SpecDs}(m, n, c) \sim \text{ImpDs}(m, n, c)$. □

As the second modification we define the appropriate agents for unifying the previous agents by selecting the D flip-flop operation for a specific input value $D \in \{0, 1\}$, but for all the initial state combinations and all the clock signal combinations. The specification agents are:

$$\text{SpecCBBDisI}(D) = (\sum_{\substack{(m,n)\in\{(0,1),(1,0)\}\\ c\in\{0,1\}}} SpecDs\_isI(m, n, c, D))[\delta_n \mapsto Q_n^*, \gamma_m \mapsto 1] \tag{16}$$

for each value $D \in \{0, 1\}$ and the implementation agents are:

$$\text{ImpCBBDisI}(D) = (\sum_{\substack{(m,n)\in\{(0,1),(1,0)\}\\ c\in\{0,1\}}} ImpDs\_isI(m, n, c, D))[\delta_n \mapsto Q_n^*, \gamma_m \mapsto 1] \tag{17}$$

for each value $D \in \{0, 1\}$.

Proposition 3 *The previous agents SpecCBBDisI(D) and ImpCBBDisI(D) for $D \in \{0,1\}$ are bisimulation equivalent.*

Proof: The result is obvious considering that the bisimulation relation '$\sim$' is a congruence over the class $\mathcal{P}$ of agents [1], [2] and it is already established in the previous Proposition 2 that SpecDs_isI(m,n,c,D) $\sim$ ImpDs_isI(m,n,c,D) for all $(m,n) \in \{(0,1),(1,0)\}$, $c \in \{0,1\}$ and $D \in \{0,1\}$. □

It follows to use these results for modelling varied examples of sequential logic circuits behaviour, especially those involving the D flip-flops.

4 Conclusions

Based on ideas from [1], [5], in this paper we have considered the internal structure of specific memory cells, namely SR flip-flop and D flip-flop. For both the asynchronous and synchronous circuit organization, we have defined specific agents for modelling the concrete flip-flops behaviour.

The authors' scientific contributions in this paper refer mainly two directions. Firstly, we have defined appropriate SCCS agents at two different specification levels for all of the agents involved in the hardware components structure we were interested in. Secondly, we have proved all the corresponding agents bisimilarities.

An immediate extension of this paper will consist in assembling all these results in order to have an appropriate model for a computer memory component. Another direction for this work progress will be the use of these final agents into a more complex digital logic circuit like a standard read/write memory hardware component with an input/output controller unit integrated. For the interest of modelling the behaviour of such a complex memory component, we have prepared here the detailed involved agents, as application of this paper theoretical results concerning the internal D flip-flops bahaviour models.

After the modelling achievements from [4], [6], [7], these modelling attempts represent a further step in the direction of having a global algebraic model of the entire computer system behaviour. Both the properties of the SCCS calculus and the computer organization are supporting our interest of developing a scalable, open algebraic models hierarchy for hardware components.

References

[1] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25, pp. 267–310, 1983.

[2] R. Milner. *Communication and concurrency*, Prentice Hall, 1989.

[3] M.M. Mano. *Digital Logic and Computer Design*, Prentice-hall Of India Pvt Ltd, 2007.

[4] A. Vasilescu. Algebraic Model for the Behaviour of the Computer's Arithmetic-Logic Unit. *ICAM Abstracts*, pp. 50, 2006.

[5] A. Vasilescu. Formal models for non-sequential processes. *PhD Thesis, Babeş-Bolyai University of Cluj-Napoca*, 2006.

[6] A. Vasilescu. Algebraic model for the intercommunicating hardware components behaviour. *New Aspects of Computers, Proc. of the 12th WSEAS International Conference on COMPUTERS*, pp. 241-246, 2008.

[7] A. Vasilescu. Counter register. Algebraic model and applications. *WSEAS TRANSACTIONS on COMPUTERS*, Issue 10, Vol. 7, pp. 1618-1627, Oct. 2008.

[8] A. Vasilescu. Algebraic model for the synchronous SR-flip-flop behaviour. *Proc. of the Intl. Conf. on Knowledge Engineering, Principles and Techniques, KEPT2009, Special Issue of Studia Universitatis Babes-Bolyai Informatica*, Anul LIV, pp. 235-238, 2009.

Anca Vasilescu
Transilvania University of Braşov
Department of Theoretical Computer Science
Str. Iuliu Maniu nr. 50, 500091 Braşov
ROMANIA
E-mail: *vasilex@unitbv.ro*

First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Distribution of result sets cardinalities in heterogeneous random databases

Letiția Velcescu, Laurențiu Vasile

Abstract

In this paper, we present an extension of the concept of random database, in which the records are random vectors following a certain multidimensional probability distribution, to heterogeneous random databases, in which columns can have their own unidimensional distribution. We investigate the sizes of some relational operations results in these databases, focusing on difference, join and outer join. In this approach to random databases, we will show that the number of tuples in the results set is Poisson distributed in the cases of heterogeneous random tables with normal and exponential columns, or discrete and exponential columns, but this behavior depends on the choice of the approximation considered in the relational operations.

1 Introduction

Nowadays, every research, medical, economic or industrial field needs to store and manage large amounts of data. In many cases, these data are likely to be uncertain or to contain errors, but the problem is still to provide a good management and to be able to extract the needed information or to appropriately support the decision making, all based on such uncertain data. This is where the concept of random database has become important.

Our work mainly focuses on the behaviour of relational operations in random databases. As known so far, this type of database supposed a vision of the table as a set of random vectors, following a common multivariate distribution. In order to distinguish this concept from the one we propose, we name it homogenous random database. In this framework, previous research has been already done. We are interested in a generalization of this concept of random database and in studying the behaviour of relational operations in this context. The extended concept is that of heterogeneous random database, in which different columns of the tables can have different probability distributions.

Applying a "traditional" relational operator when working with databases that contain uncertain data will often result in an irrelevant, even empty, data set. Because of uncertainty, one should work with approximate rather than exact operations.

In our approach, we obtained samples of some specific unidimensional distributions, stored them in columns of relational tables and applied approximate relational operators on these random tables. As in the homogeneous case, we want to obtain an estimation of the distribution of the number of lines in the result set of an approximate relational operation. The technique used to confirm the likeliness of a probability distribution is based on the chi square goodness of fit test.

The paper is organized as follows: in the second part, we introduce the main database and random database concepts and results related to the homogeneous case; in the third part, we describe our extension and approach to this field; we will conclude with some considerations and perspectives of our future work in the fourth part of this paper.

2 Preliminaries

In order to introduce the definitions of some fundamental database notions, we will consider the finite domains $D_1, D_2, ..., D_n$, not necessarily disjoint ([6]).

Definition 1 *The cartesian product $D_1 \times D_2 \times ... \times D_n$ of the domains $D_1, D_2, ..., D_n$ is defined by the set of the tuples $(V_1, V_2, ..., V_n)$, where $V_1 \in D_1, V_2 \in D_2, ..., V_n \in D_n$. The number n defines the tuple's arrity.*

Definition 2 *A subset of the cartesian product $D_1 \times D_2 \times ... \times D_n$ defines a relation R on the sets $D_1, D_2, ..., D_n$. Consequently, a relation is a tuple set.*

There is an alternative definition of a relation, in the terms of a set of functions. Suppose that we associate to each domain D_i an attribute A_i.

Definition 3 *A relation R is a set $\{f_1, f_2, ..., f_m\}$, where $f_i : \{A_1, A_2, ..., A_n\} \rightarrow D_1 \cup D_2 \cup ... \cup D_n$ and $fi(Aj) \in Dj$ for each values of i and j.*

We can easily remark that both definitions of a relation refer to sets which are varying over time. In these sets, elements can be inserted, deleted or updated. Obviously, not the content of this set characterizes a relation, but a time-invariant element. Such an element is the relational schema, which is actually the relation's structure.

Definition 4 *The relational schema of a relation R is defined by the set of the attributes' names which correspond to the relation R. We denote the relational schema by $R(A_1, A_2, ..., A_n)$.*

The representation of a relation can be done by a table in which each line corresponds to a tuple and each column corresponds to an attribute. In other words, a column corresponds to a domain. The relational databases are perceived by the users as a set of tables.

Definition 5 *The degree of a relation is represented by the number of its attributes. The cardinality of a relation is given by the number of its tuples.*

Generally, a table is a representation of a relation, but it is important to mention that there is an important difference between these notions: a table is a sequence of records, contrary to a relation, which is a set of records. This means that the tuples of a relation must be distinct, whereas those of a table can be not.

2.1 Relational operations

The relational operations are performed by the operators of the relational model, which includes the relational algebra. The operators of the relational algebra are either the usual set operators (*union*, *intersect*, *product*, *difference*) or some specific relational operators (*project*, *select*, *join*, *division*). The usual set operators, except *product*, require that the operands had the same type.

The relations in the database are subjected to operations. The result of a relational operation is a new relation.

Definition 6 *The intersection of two relations R and S is the relation composed by the set of the tuples which belong both to R and S.*

Definition 7 *The difference between two relations R and S is the relation composed by the set of the tuples which belong to R but do not belong to S.*

The join operator allows the information retrieval from more correlated relations. The required condition in order to apply the join operator is that the tuples are similar.

Definition 8 *The join between two relations R and S is a binary operation whose result is a new relation in which each tuple is a combination of a tuple in the first relation and a tuple in the second one, satisfying a given join condition.*

The *join* operator composes projection, selection and cartesian product. Generally, the cartesian product is built, some tuples are eliminated by selection and some attributes are eliminated by projection. When dealing with large tables, the cartesian product is a costly operation and consequently, so is the join. In some cases, when evaluating a join operation, even if the volume of the intermediate cartesian product is significant, the result of the join is quite small. Therefore, when joining more than two tables, it is relevant to know the sizes of the joins, in order to perform these operations in the proper order. This is the reason why the work related to the estimation of the number of lines in the result set of each operation is important, in the context of database optimization, when joining multiple tables or query's result sets.

There are several types of the *join* operation, such as the *equi-join* and the *outer join*. The *equi-join* requires that the values of the specified attributes are equal.

From definition 8, it results that a *join* operation will lose a tuple which belongs to a relation if there is no tuple in the other relation such that the *join* condition be satisfied. In order to keep such tuples in the result set, we use the *outer join* operation. This one combines the tuples in the two relations for which the correlation conditions are satisfied, without losing the other tuples. This operator assigns *null* values to the attributes that exist in a tuple of one of the input relations, but does not exist in the second relation. There are three types of *outer join* operators: *left*, *right* and *full*. They keep in the result each tuple of the relation in the left, right, respectively in both relations.

2.2 Approximate relational operations in random databases

In random databases, the operations above find a homologous in the approximate operations. As stated in the introduction, one would get an irrelevant result if using the exact match in the cases when uncertainty arises.

In this section, we will describe what some of the relational operations mentioned above become in the context of approximation in the random databases. In order to define the approximate version of these operations, we consider a distance $d(x, y)$ between the elements in the domains D_A and D_B, which are the projections on the attributes in A and B of the domains D_{U_1} and D_{U_2} and which are assumed to be subsets of a metric space where the distance d is defined ([7]). An example of such a distance is the Hamming distance, given by the number of different join attributes in the two tuples. We denote by $B_\tau(x)$ the ball with the centre in x having the radius ε.

We define the following ε-operations: *ε-difference*, *ε-equi-join* and *ε-outer-join*. For two relations R and S, we denote these approximate operations by $difference_\varepsilon(R, S)$, $join_\varepsilon(R, S)$ and $outer\text{-}join_\varepsilon(R, S)$, respectively.

Definition 9 *The ε-difference between two relations R and S is a relation containing the following set of tuples:*

$$difference_\varepsilon(R, S) = \{x \in R \mid \neg\exists y \in S, x \in B_\varepsilon(y)\} \quad (1)$$

Definition 10 *The ε-join between two relations R and S is a relation containing the following set of tuples:*

$$join_\varepsilon(\mathrm{R}, \mathrm{S}) = \{(x, y) \in R \times S \mid d(x_A, y_B) \le \varepsilon\} \quad (2)$$

Definition 11 *The ε-outer-join between two relations R and S is a relation containing the following set of tuples:*

$$outer\text{-}join_{\varepsilon}(R, S) = \text{join}_{\varepsilon}(R, S) \cup \{(x, y) \in R \times S \mid \neg \exists y \in S, x \in B_{\varepsilon}(y)\} \quad (3)$$

The number of lines in the result set of each of these operations is denoted by $N_{\varepsilon}(difference_{\varepsilon}(R,S))$, $N_{\varepsilon}(join_{\varepsilon}(R,S))$, respectively $N_{\varepsilon}(outer\text{-}join_{\varepsilon}(R,S))$ ([8]). In the cases when it is clear what these values refer, we denote them by N_{ε} for simplicity.

The approximative match problems have been already studied for the *equi-join* operation, in the case of the homogeneous random databases.

Definition 12 *Two tuples* $x \in D_A$ *and* $y \in D_B$ *are ε-close, with* $\varepsilon \geq 0$, *if d(x, y)* $\leq \varepsilon$.

As one can remark, the *ε-equi-join* operation's result set contains the ε-close tuples according to the given distance. For the particular case $\varepsilon = 0$, we get the usual *equi-join* operation.

2.3 Previous work

The definition of the number N_{ε} of lines in the join's result has one other significant importance, concerning the constraints in the database, as shown in the previous research on the random databases ([7]). The keys and functional dependencies represent the constraints in a database. The concept of key generalizes to that of functional dependency, which specifies relations between two distinct sets of attributes, meaning that the values of the first set determine the values of the second set of attributes. The cardinality of the set of constraints is extremely important in the database design. A model in which this cardinality is exponential depending on the number of attributes becomes impossible to manage ([2]).

Definition 13 *A minimal set of attributes whose values uniquely identify a tuple in a relation represents a key for that relation.*

Consequently, a key of a relation *R* is a set of attributes *K*, such that ([6]):

i) for each tuples t_1, t_2 of *R*, we have $t_1(K) \neq t_2(K)$;

ii) there is no proper subset of *K* having the property i).

Extending the notion of key to the one of *ε*-key, a set of attributes *K* is an *ε*-key if there are no *ε*-close tuples $t_i(K)$, $i = 1,\ldots,m$. Such a property is denoted by $R \models_{\varepsilon} K$. For the particular case $\varepsilon = 0$, one gets the usual definition of a key.

The set of all attributes of a relation composes a key, but the keys are better as their attributes set is smaller. We have that $N_{\varepsilon}(join_{\varepsilon}(R,R))$ is the number of *ε*-close tuples in *R*, for a specified attributes join set *A*. The distribution of $N_{\varepsilon}(join_{\varepsilon}(R,R))$ defines the capacity of the set *A* to distinguish the tuples in the relation *R*. Thus, the attributes set *A* is an *ε*-key if and only if $N_{\varepsilon}(join_{\varepsilon}(R,R)) = m$.

Initially, the problem of characterization of the most relevant properties of the constraints has been addressed in the worst case ([1]). Then, the problem has been studied in the average case for a general class of probabilistic models ([7]). In this second approach, the entropy of the records distribution was used in order to explain the properties of the constraints.

A recent research direction concerning the random databases uses, as main tools, the Poisson approximation and the Rényi *ε*-entropy. This type of entropy is introduced as a generalization of the Rényi entropy for the discrete distributions.

In the stochastic models which were considered in the random databases modeling so far, it is assumed that, in a random table *T*, the tuples are random vectors with n elements, independent and identically distributed, with a common probability distribution *P*.

In the previous research, three types of random databases were considered ([8]): uniform, if *P* is a uniform multivariate distribution; normal, if *P* is a multivariate normal distribution; Bernoulli,

if the attributes A_i are independent and identically distributed, with a common univariate discrete distribution. Two particular cases of the latter database type are the standard Bernoulli database, where the attributes are uniformly distributed, and the conventional Bernoulli databases, where the domains of the attributes have the property $|D_i| = 2$.

3 Cardinality of ε-operations result sets in heterogeneous random databases

The technique used to estimate the distribution of the N_ε values is the chi square test of goodness of fit ([5]). Before that, we generated the histograms for the frequency of the number of lines in the results set. These histograms indicate the possibility that the N_ε values are Poisson distributed.

Our approach considered both cases: homogenous random tables, denoting the concept existing so far, and heterogeneous random tables, which represent an extension we propose. We recall that a heterogeneous random table is a table in which different columns can have different unidimensional distributions.

3.1 Cardinality of ε-operations on heterogeneous tables

For the heterogeneous case, we considered random tables in which we used either samples of two continuous distributions, namely the normal $N(0, 1)$ and exponential $Exp(1)$, or samples of a discrete and a continuous distribution, namely *Binomial*(200, 0.5) or *Geometric*(0.5) and exponential $Exp(1)$.

We considered two random relations $R(A_1, A_2)$ and $S(B_1, B_2)$, with at most two attributes each, differently distributed and containing 1000 lines. In Table 1 we show the types of relations we worked with.

We made the implementations of these tables and performed the queries representing the relational operations in the *Oracle 11g* DBMS. Afterwards, we realized the histograms concerning the cardinality of the ε-operations.

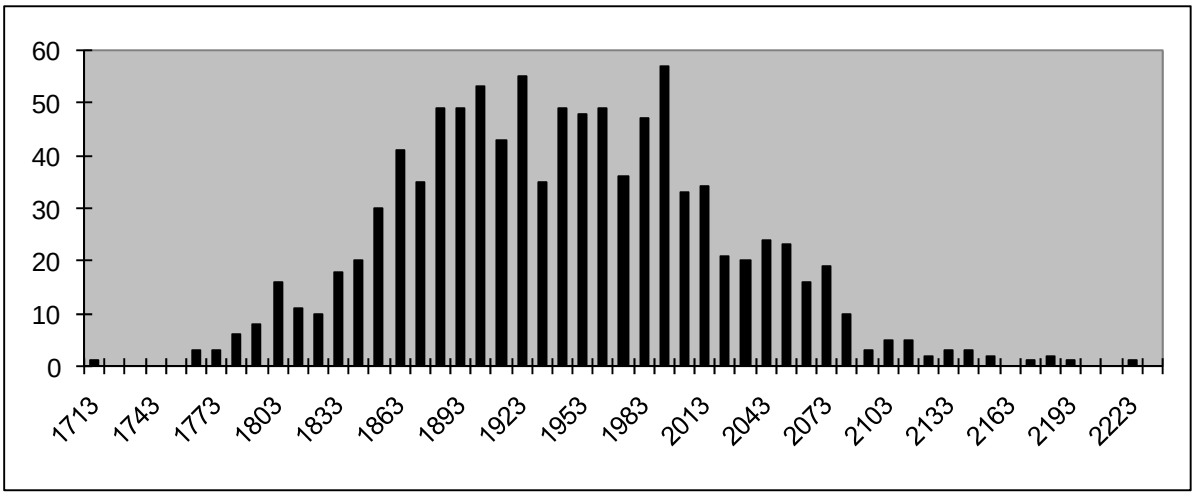


Figure 1. *Histogram for the* 0.05*-join operation between two sets of columns, each with a* *Binomial*(200, 0.5) *column and an* *Exp*(1) *one.*

As an example, Figure 1 shows the histogram for the ε-join in the heterogeneous case, for a *Binomial*(200, 0.5) column and an *Exp*(1) one, with $\varepsilon = 0.05$. In Figure 2, we have the histogram for the similar distributions, but $\varepsilon = 0.01$.

A1	A2	B1	B2
Bin(200, 0.5) or Geom(0.5)	-	Bin(200, 0.5) or Geom(0.5)	
Exp(1)	$N(0, 1)$	Exp(1)	$N(0, 1)$
Bin(200, 0.5) or Geom(0.5)	Exp(1)	Bin(200, 0.5) or Geom(0.5)	Exp(1)

Table 1 *The distribution of the columns in the random tables.*

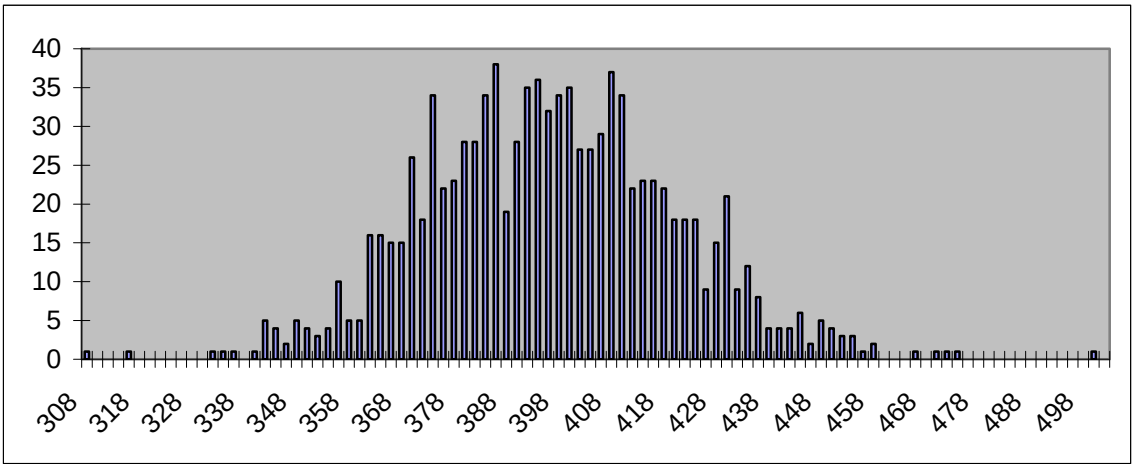


Figure 2. *Histogram for the* 0.01*-join operation between two sets of columns, each with a Binomial*(200, 0.5) *column and an Exp*(1) *one.*

From these histograms, one could observe that it is possible that the random variable N_ε follow a Poisson process on the line. This result has been already stated for homogeneous random databases in [8]. In our homogeneous cases we considered, we observed that the property that the number of lines in the *ε-join* operation has a Poisson distribution depends on the value of *ε*. This means there is a threshold up to which this distribution is followed.

3.2 The Chi square test of goodness of fit for the distribution of the result of the heterogeneous *ε*-operations

Consider N experiments, where N is the number of points on the line for which $P(N=n)=\frac{\lambda^n}{n!}\cdot e^{-\lambda}$, $\lambda>0$. The parameter λ of the corresponding Poisson distribution, also called the intensity of the Poisson process, is $E(N)$. The parameter λ is constant, which determines that the Poisson process is homogenous.

In order to estimate the parameter λ, we consider a sample $\nu_1,...\nu_N$ of the homogenous uniform Poisson process, for a given ε, and we take the following estimation of the Poisson parameter:

$$\hat{\lambda}_N=\frac{1}{N}\sum_{i=1}^{N}\nu_i \tag{4}$$

The chi square test of goodness of fit is the technique used in order to show that the values of the random variable N_ε can be estimated by a homogenous uniform Poisson process.

In this respect, we take k the number of distinct values $\nu_1,...,\nu_k$ in the sample of N_ε and $f_1,...,f_k$ the frequencies corresponding to each value. Obviously, we have that $f_1+...+f_k=N$. We compute the following theoretic probabilities:

$$p_j = P(N_\varepsilon = \nu_j) = \frac{\hat{\lambda}_N^{\nu_j}}{\nu_j!} \cdot e^{-\hat{\lambda}_N} \tag{5}$$

Based on the above values, we can we determine the following statistics:

$$\chi_c^2 = \sum_{j=1}^{k} \frac{(f_j - N \cdot p_j)^2}{N \cdot p_j} \tag{6}$$

Because the parameter λ_N has been estimated before, it implies that this statistics has the distribution χ^2 with $k-2$ degrees of freedom.

The last step of this test is to determine if there is a level of significance $\alpha \le 0.05$ such that $P(\chi_c^2 \ge \chi_{k-1,\alpha}^2) = \alpha$. In the case when the inequality $\chi_c^2 < \chi_{k-1,\alpha}^2$ is true, we can state that the values ν_i do not differ significantly from a Poisson distribution. Otherwise, the possibility that the values ν_i follow a Poisson distribution is rejected.

4 Conclusions and perspectives

We showed that it is possible to extend the concept of random tables to more general cases, in which one table can host columns of different probability or mass distributions. For the two heterogeneous random table cases above, with a binomial and an exponentially distributed column, we considered the values 0.05, 0.005 and 0.001 for ε. In this case, we remarked that the chi square test of goodness of fit does not pass for larger values of ε (e.g. 0.05), but it passes successfully for smaller values (e.g. 0.01 or 0.005). The same remarks are available for the case when the binomial distribution was replaced by a geometric one.

We also performed the test in the case of the homogeneous random tables with normal distribution or exponential distribution. Here, the chi square test of goodness of fit does not pass for a larger value of ε (e.g. 10^{-3}); for $\varepsilon = 10^{-4}$ the test fails, but very closely, and for smaller values of ε (e.g. 10^{-5}), the test passes, so we can state that the cardinality is Poisson distributed.

These conclusions remain true in the case of the difference or outer-join operations.

Concerning the perspectives of our work, one direction would be to determine accurately the threshold up to which the ε-operations cardinalities remain Poisson distributed. Another direction is to find the dependency between the value of ε, the sample size and the acceptance of the Poisson distribution. We also intend to extend the meaning of heterogeneous databases to the case in which some combinations of attributes can follow multidimensional distributions.

References

[1] S. Abiteboul, R. Hull and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] A. Gupta, Y. Sagiv, J.D. Ullman and J. Widom. Efficient and complete tests for database integrity constraint checking. *PPCP 1994*, LNCS vol. 874, Springer, 1994.

[3] N. Johnson, A. Kemp and S. Kotz. *Univariate Discrete Distributions*. 3rd edition, John Wiley & Sons, 2005.

[4] W. Martinez and A. Martinez. *Computational Statistics Handbook with Matlab*. Chapman&Hall, 2002.

[5] Gh. Mihoc and V. Craiu. *Tratat de statistică matematică*. vol. 2, Editura Academiei RPR, Bucuresti, 1977.

[6] I. Popescu and L. Velcescu. *Proiectarea bazelor de date.* Editura Universităţii din Bucureşti, 2008.

[7] O. Seleznjev and B. Thalheim. Average Case Analysis in Database Problems. *Methodology and Computing in Applied Probability*, Kluwer Academic Publishers, 2003.

[8] O. Seleznjev and B. Thalheim. Random Databases with Approximate Record Matching. *Methodol. Comput. Appl. Probab.*. Springer Verlag, 2008.
[9] I. Văduva. *Modele de simulare*. Editura Universităţii din Bucureşti, 2004.

LETIŢIA VELCESCU
University of Bucharest
Faculty of Mathematics and Informatics
Department of Informatics
14, Academiei Street, 010014 Bucharest
ROMANIA
E-mail: letitia@fmi.unibuc.ro

LAURENŢIU VASILE
University of Bucharest
Faculty of Mathematics and Informatics
Department of Informatics
14, Academiei Street, 010014 Bucharest
ROMANIA
E-mail: vsl@fmi.unibuc.ro

List of authors

OCTAVIAN AGRATINI, p. 7
Babes-Bolyai University
Faculty of Mathematics and Computer
400084
Cluj-Napoca, **ROMANIA**
E-mail: *agratini@math.ubbcluj.ro*

KIRIL ALEXIEV, p. 14
Institute for Parallel Processing – Bulgarian Academy of Sciences
Department of Mathematical Methods for Sensor Information Processing
25A Acad.G.Bonchev Str., 1113
Sofia, **BULGARIA**
E-mail: *alexiev@bas.bg*

VERA BEHAR, p. 23
Institute for Parallel Processing (IPP)
Bulgarian Academy of Sciences (BAS),
Department of Mathematical Methods for Sensor Information Processing.
Institute for Parallel Processing, BAS, "Acad. G. Bonchev",
Str., bl.25-A, 1113
Sofia, **BULGARIA**.
E-mail: *behar@bas.bg*

NICOLAE BOJA, p. 33
"Politehnica" University Timisoara
P-ta Victoriei, 2, RO- 300006
Timisoara, **ROMANIA**
e-mail: *nicolae.boja@yahoo.com*

GEORGETA BRAILOIU, p. 33
"Politehnica" University Timisoara
P-ta Victoriei, 2, RO- 300006
Timisoara, **ROMANIA**
E-mail:

LAURA FLORENTINA CACOVEAN, p. 46
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *laura.cacovean@ulbsibiu.ro*

LAURENTIU CIOVICA, p. 56, p. 61
University "Lucian Blaga" of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *ciovica_laurentiu@yahoo.com*

LIVIU CIOVICA, p. 56, p. 61
University "Lucian Blaga" of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *ciovica_liviu@yahoo.com*

DOMENICO CONSOLI, p. 68
Universita Politecnica delle Marche
Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione
Via Brecce Bianche 12, 60131
Ancona, **ITALY**
E-mail: *consoli@diiga.univpm.it*

VASILE CORNITA, p. 158
University Politehnica of Bucharest
Faculty of Electronics, Telecommunications and Information Technology
B-dul Iuliu Maniu 1-3, Sector 6
Bucharest, **ROMANIA**
E-mail: *cornita_vasile@yahoo.com*

JEAN DEZERT, p. 270
ONERA,
29 Av. de la Division Leclerc, 92320
Chatillon, **FRANCE**
E-mail: *jean.dezert@onera.fr*

STEFAN DIMITROV, p. 121
University of Sofia "St. Kliment Ohridski",
Sofia, **BULGARIA**
E-mail: *stefan@ucc.uni-sofia.bg*

GEORGI DIMKOV, p. 78
Institute of Mathematics and Informatics
Department of Complex Analysis
8, Acad. G.Bonchev str. 1113
Sofia, **BULGARIA**
E-mail: *gdimkov@math.bas.bg*

MIHAITA DRAGAN, p. 182
University of Bucharest
Technologic Department: Information Technology
Mihail Kogalniceanu Bd., no. 36-46, sector 5, cod. 70709,
Bucharest, **ROMANIA**
E-mail: *dragan_mihaita@yahoo.com*

VLADISLAV DUREV, p. 89
Department of Electronics
Technical University of Sofia
8 Kl. Ohridski blvd., 1000
Sofia, **BULGARIA**
e-mail: *v_p_durev@yahoo.com*

REMUS ENE, p. 33
"Politehnica" University Timisoara
P-ta Victoriei, 2, RO- 300006
Timisoara, **ROMANIA**
E-mail: *eneremus@gmail.com*

CRISTIAN FRASINARU, p. 96
University A.I.Cuza of Iasi
Faculty of Computer Science
General Berthelot 16, 700483
Iasi, **ROMANIA**
E-mail: *acf@info.uaic.ro*

ELISSAVETA GADJEVA, p. 89
Department of Electronics
Technical University of Sofia
8 Kl. Ohridski blvd., 1000
Sofia, **BULGARIA**
e-mail: *egadjeva@tu-sofia.bg*

IOAN DANIEL HUNYADI, p. 106
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *daniel.hunyadi@ulbsibiu.ro*

PAVLINA KONSTANTINOVA, p. 23
Institute for Parallel Processing (IPP)
Bulgarian Academy of Sciences (BAS),
Department of Mathematical Methods for Sensor Information Processing.
Institute for Parallel Processing, BAS, "Acad. G. Bonchev",
Str., bl.25-A, 1113
Sofia, **BULGARIA**.
E-mail: *pavlina@bas.bg*

JAN KRCAL, p. 115
Czech Technical University in Prague
Faculty of Transportation Sciences
Konviktska 20, Prague, 11000
CZECH REPUBLIC
E-mail: *krcal@fd.cvut.cz*

PETKO LALOV, p. 121
University of Mining and Geology "St. Ivan Rilski"
Sofia, **BULGARIA**
E-mail: *p.lalov@gmail.com*

DIANA MAN, p. 281
Faculty of Mathematics and Computer Science,
Babes-Bolyai University
Computer Science Dep.
Cluj-Napoca, **ROMANIA**
E-mail: *man.diana@nessie.cs.ubbcluj.ro*

GEORGE MANIU, p. 128
Univ. Spiru Haret Brasov
Management Faculty
Turnului No.7
Brasov**, ROMANIA**
E-mail: costelmaniu@yahoo.com

IONELA MANIU, p. 128
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *mocanionela@yahoo.com*

ANA-MARIA MIREA, p. 134
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, 200585
Craiova, **ROMANIA**
E-mail: *ammirea@acm.org*

VIORICA MURESAN, p. 143
Technical University of Cluj-Napoca
Faculty of Automation and Computer Science
Department of Mathematics
15 C. Daicoviciu Street
Cluj-Napoca, **ROMANIA**
e-mail: *vmuresan@math.utcluj.ro*

MIRCEA IOSIF NEAMTU, p. 151
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *neamtu12@yahoo.com*

MILEN NIKOLOV, p. 23
Institute for Parallel Processing (IPP)
Bulgarian Academy of Sciences (BAS),
Department of Mathematical Methods for Sensor Information Processing.
Institute for Parallel Processing, BAS, "Acad. G. Bonchev",
Sofia, **BULGARIA**.
E-mail: *milenik@bas.bg*

IVA NIKOLOVA, p. 14
Technical University of Sofia
Department of Computer Systems,
Faculty of Computer Systems and Control
8 Kl. Ohridski Blvd. bl 1, 1000
Sofia, **BULGARIA**
E-mail:*inni@tu-sofia.bg*

DORU PANESCU, p. 299
"Gheorghe Asachi" Technical University
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Applied Informatics
Mangeron 53A, 700050
Iaşi, **ROMANIA**
E-mail: *dorup@ac.tuiasi.ro*

IULIANA PARASCHIV-MUNTEANU, p. 244
University of Bucharest
Faculty of Mathematics and Computer Science
14 Academiei St., 010014
Bucharest, **ROMANIA**
E-mail: *pmiulia@fmi.unibuc.ro*

CARLOS PASCAL, p. 299
"Gheorghe Asachi" Technical University
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Applied Informatics
Mangeron 53A, 700050
Iaşi, **ROMANIA**
E-mail: *pascalcarlos@gmail.com*

CRISTINA POP, p. 158
Academy of Economic Studies of Bucharest
Faculty of Cybernetics, Statistics and Informatics
Calea Dorobantilor 15-17, Sector 1
Bucharest , **ROMANIA**
E-mail: *cristinel19@yahoo.com*

DANIEL NICOLAE POP, p. 164
Romanian-German University
Faculty of Economics and Computers
Calea Dumbravii ,Nr: 28-32
ROMANIA
E-mail: *danielnicolaepop@yahoo.com*

IOAN POP, p. 174
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *me.ioanpop@gmail.com*

MARIANA POPA, p. 182
Universitatea Bucuresti
Dep. Tehnologii
Fac. : Tehn. Informatiei
Blvd. M. Kogalniceanu
Nr. 36-46, Sector 5
Cod postal 70709
Bucuresti, **ROMANIA**
e-mail: *marpopa2002@yahoo.com*

MARIN POPA, p. 182
Universitatea Bucuresti
Dep. Tehnologii
Fac. : Tehn. Informatiei
Blvd. M. Kogalniceanu
Nr. 36-46, Sector 5
Cod postal 70709
Bucuresti, **ROMANIA**
e-mail: *marpopa2002@yahoo.com*

CRISTINA POPENTA, p. 197
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *ac_cris@yahoo.com*

DOINA PREDA, p. 134
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, 200585
Craiova, **ROMANIA**
E-mail: *dpreda@acm.org*

MIRCEA PREDA, p. 134
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, 200585
Craiova, **ROMANIA**
E-mail: *mpreda@acm.org*

ERNEST SCHEIBER, p. 203
Transilvania University of Brasov
Faculty of Mathematics and Computer Science
str. Iuliu Maniu 50
Brasov, **ROMANIA**
E-mail: *scheiber@unitbv.ro*

STANISLAW SIENIUTYCZ, p. 211
Warsaw University of Technology
Faculty of Chemical and Process Engineering
1 Warynskiego Street, 00-645
Warszawa, **POLAND**
E-mail: *sieniutycz@ichip.pw.edu.pl*

CORINA SIMIAN, p. 221
Institute of Mathematics, University of Zurich
Zurich, **SWITZERLAND**
E-mail: *corina.simian@yahoo.com*

DANA SIMIAN, p. 229
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *d_simian@yahoo.com*

FLORENTIN SMARANDACHE, p. 270
University of New Mexico
Chair of Department of Mathematics
Gallup, NM 87301,
USA
E-mail: *smarand@unm.edu*

MIHAI STANCU, p. 237
Universitatea Lucian Blaga din Sibiu
Facultatea de Stiinte
Bd-ul. Victoriei, Nr.10,
Sibiu, **ROMANIA**
E-mail: *stancu_meehigh@yahoo.com*

LUMINITA STATE, p. 244
University of Pitesti
Faculty of Mathematics and Computer Science
1 Targu din Vale St., 110040
Pitesti, **ROMANIA**
E-mail: *lstate@clicknet.ro*

FLORIN STOICA, p. 254
University "Lucian Blaga" of Sibiu
Faculty of Sciences, Computer Science Department
5-7 I. Ratiu str., 550021
Sibiu, **ROMANIA**
E-mail: *florin.stoica@ulbsibiu.ro*

CAMELIA SERBAN, p. 262
Babes-Bolyai University
Department of Computer Science
Kogalniceanu 1, 400084
ClujNapoca, **ROMANIA**
E-mail: *camelia@cs.ubbcluj.ro*

SADDIKA TARABIE, p. 7
Tishrin University
Science Faculty of Sciences
1267
Latakia, **SYRIA**
E-mail: *sadikatorbey@yahoo.com*

ALBENA TCHAMOVA, p. 270
Bulgarian Academy of Sciences
Institute for Parallel Processing
Acad. G. Bonchev' Str., Bl. 25-A, 1113
Sofia, **BULGARIA**
E-mail: *tchamova@bas.bg*

CONSTANTIN TEODORESCU-MIHAI, p. 134
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, 200585
Craiova, **ROMANIA**
E-mail: *constantintm@gmail.com*

MARIA TOSA-ABRUDAN, p. 281
Faculty of Mathematics and Computer Science,
Babes-Bolyai University
Computer Science Dep.
Cluj-Napoca, **ROMANIA**
E-mail: *maria@nessie.cs.ubbcluj.ro*

RADU TIBERIU TRIMBITAS, p. 164
Babes-Bolyai University Cluj-Napoca
Department of Applied Mathematics
Mihail Kogalniceanu street ,Nr:1
Cluj-Napoca , **ROMANIA**
E-mail: *tradu@math.ubbcluj.ro*

MILAN TUBA, p. 291
Megatrend University Belgrade
Faculty of Computer Science
Bulevar umetnosti 29
SERBIA
E-mail: *tubamilan@ptt.rs*

GABRIELA VARVARA, p. 299
"Gheorghe Asachi" Technical University
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Applied Informatics
Mangeron 53A, 700050
Iaşi, **ROMANIA**
E-mail: *gvarvara@ac.tuiasi.ro*

LAURENTIU VASILE, p. 316
University of Bucharest
Faculty of Mathematics and Informatics, Department of Informatics
14, Academiei Street, 010014
Bucharest, **ROMANIA**
E-mail: *vsl@fmi.unibuc.ro*

ANCA VASILESCU, p. 308
Transilvania University of Brasov
Department of Theoretical Computer Science
Str. Iuliu Maniu nr. 50, 500091
Brasov, **ROMANIA**
E-mail: *vasilex@unitbv.ro*

LETITIA VELCESCU, p. 316
University of Bucharest
Faculty of Mathematics and Informatics, Department of Informatics
14, Academiei Street, 010014
Bucharest, **ROMANIA**
E-mail: *letitia@fmi.unibuc.ro*

GEORGI ZAPRYANOV, p. 14
Technical University of Sofia
Department of Computer Systems,
Faculty of Computer Systems and Control
8 Kl. Ohridski Blvd. bl 1, 1000
Sofia, **BULGARIA**
E-mail: *gszap@tu-sofia.bg*

VLAD ZDRENGHEA, p. 281
University of Medicine and Pharmacy Iuliu Hategan
Psychiatry Department
Cluj-Napoca, **ROMANIA**
E-mail: *vlad_zdrenghea@yahoo.com*