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.

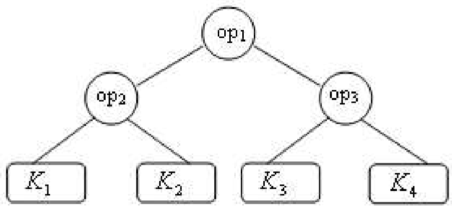

Fig.1 Representation of a multiple kernel

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.

<table>
<tr><td rowspan="2">op_1</td><td rowspan="2">op_2</td><td rowspan="2">op_3</td><td rowspan="2">t_1</td><td>d_1</td><td>r_1</td><td rowspan="2">...</td></tr>
<tr><td colspan="2">γ_1</td></tr>
</table>

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*