Approximability of optimal social welfare in multiagent resource allocation with cardinal and ordinal preferences

Nhan-Tam Nguyen
Heinrich-Heine Universität Düsseldorf

This is in part joint work with Dorothea Baumeister , Jörg Rothe, Sylvain Bouveret, Jérôme Lang, Trung Thanh Nguyen, and Abdallah Saffidine.

Meeting of COST Action IC 1205

Sibiu, October 20–22, 2014

Outline

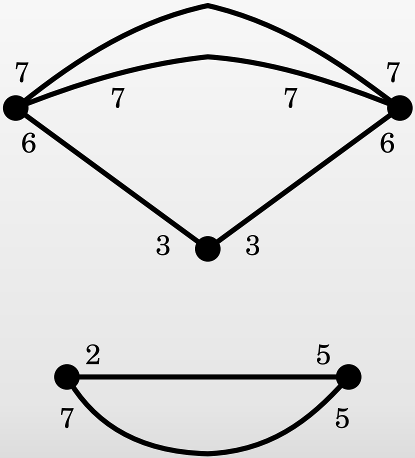
Resource allocation under...

1. Cardinal preferences
2. Ordinal preferences & Restricted model

Part One

Cardinal preferences

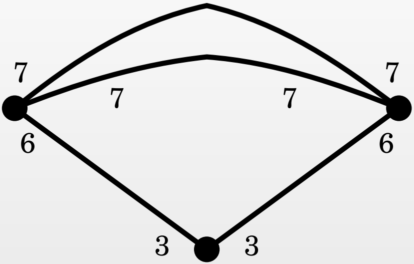
Example



Undirected graph with

- parallel edges
- self loops
- nonuniform weights

Example

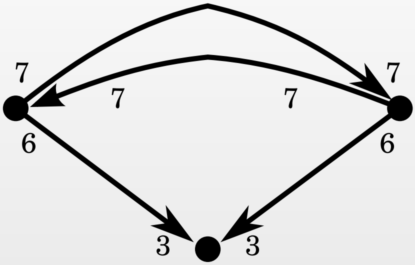


Undirected graph with

- parallel edges
- self loops
- nonuniform weights

Goal: Orient the edges so that we **maximize the minimum sum of incoming weights**.

Example

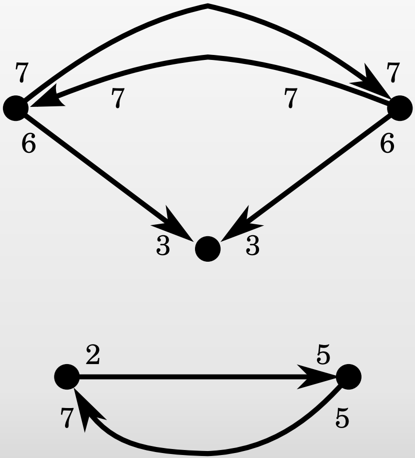


Undirected graph with

- parallel edges
- self loops
- nonuniform weights

Goal: Orient the edges so that we **maximize the minimum sum of incoming weights**.

Example



Undirected graph with

- parallel edges
- self loops
- nonuniform weights

Goal: Orient the edges so that we **maximize the minimum sum of incoming weights**.

- vertex = agent
- edge = object
- sum of incoming weights = utility

Model

What kind of resource allocation problem do we deal with?

- indivisible and non-shareable goods
- centralized
- no payments
- non-strategic agents

Model

What kind of resource allocation problem do we deal with?

- indivisible and non-shareable goods
- centralized
- no payments
- non-strategic agents

Formal model:

- a finite set of **objects** $\mathcal{O} = \{o_1, \ldots, o_m\}$
- a finite set of **agents** $\mathcal{A} = \{1, \ldots, n\}$
- each agent $i \in \mathcal{A}$ has utility function $u_i : 2^{\mathcal{O}} \to \mathbb{Q}$

Representations of utility functions

Problem: Utility functions have an exponential-size domain.

Each u_i is represented ...

Representations of utility functions

Problem: Utility functions have an exponential-size domain.

Each u_i is represented ...

- in the **bundle form** by
  a list of pairs $(S, u_i(S))$ with $u_i(S) \neq 0$,

Representations of utility functions

Problem: Utility functions have an exponential-size domain.

Each u_i is represented ...

- in the **bundle form** by
  a list of pairs $(S, u_i(S))$ with $u_i(S) \neq 0$,
- in the ***k*-additive form** by
  coefficients α_i^S for each bundle $S \subseteq \mathcal{O}$ with $\|S\| \leq k$ such that

$$u_i(T) = \sum_{S \subseteq T, \|S\| \leq k} \alpha_i^S.$$

Representations of utility functions

Problem: Utility functions have an exponential-size domain.

Each u_i is represented ...

- in the **bundle form** by
  a list of pairs $(S, u_i(S))$ with $u_i(S) \neq 0$,
- in the ***k*-additive form** by
  coefficients α_i^S for each bundle $S \subseteq \mathcal{O}$ with $\|S\| \leq k$ such that

$$u_i(T) = \sum_{S \subseteq T, \|S\| \leq k} \alpha_i^S.$$

Other representations are e.g., **straight-line programs**, **bidding languages**, **weighted goals**.

Restrictions on utility functions

Usually we have assumptions such as...

- no **externalities**
- **monotonicity** (free disposal)

$$S \subseteq T \implies u(S) \leq u(T),$$

- **normalization**

$$u(\emptyset) = 0.$$

Restrictions on utility functions

Usually we have assumptions such as…

- no **externalities**
- **monotonicity** (free disposal)

$$S \subseteq T \implies u(S) \leq u(T),$$

- **normalization**

$$u(\emptyset) = 0.$$

We may put additional restrictions on utility functions:

- **additivity**

$$u(S) = \sum_{o \in S} u(\{o\})$$

Restrictions on utility functions

Usually we have assumptions such as...

- no **externalities**
- **monotonicity** (free disposal)

$$S \subseteq T \implies u(S) \leq u(T),$$

- **normalization**

$$u(\emptyset) = 0.$$

We may put additional restrictions on utility functions:

- **additivity**

$$u(S) = \sum_{o \in S} u(\{o\})$$

- **submodularity**

$$u(S \cup T) + u(S \cap T) \leq u(S) + u(T)$$

Restrictions on utility functions

Usually we have assumptions such as...

- no **externalities**
- **monotonicity** (free disposal)

$$S \subseteq T \implies u(S) \leq u(T),$$

- **normalization**

$$u(\emptyset) = 0.$$

We may put additional restrictions on utility functions:

- **additivity**

$$u(S) = \sum_{o \in S} u(\{o\})$$

- **submodularity**

$$u(S \cup T) + u(S \cap T) \leq u(S) + u(T)$$

- **subadditivity**

$$u(S \cup T) \leq u(S) + u(T)$$

Solutions

A **solution** in this model is a partition of $\mathcal{O}$ into n disjoint subsets.

Question: How to assess the quality of a solution?

Solutions

A **solution** in this model is a partition of $\mathcal{O}$ into n disjoint subsets.

Question: How to assess the quality of a solution?

We can ask questions such as

Solutions

A **solution** in this model is a partition of $\mathcal{O}$ into n disjoint subsets.

Question: How to assess the quality of a solution?

We can ask questions such as

- Which **(fairness) properties** are satisfied?

Solutions

A **solution** in this model is a partition of $\mathcal{O}$ into n disjoint subsets.

Question: How to assess the quality of a solution?

We can ask questions such as

- Which **(fairness) properties** are satisfied?
- What is the **social welfare**?

Social welfare

We can aggregate utility values with a **collective utility function**.

- **Utilitarian social welfare**

$$sw_u(\pi) = \sum_{i \in \mathcal{A}} u_i(\pi)$$

- **Egalitarian (Rawlsian) social welfare**

$$sw_e(\pi) = \min_{i \in \mathcal{A}} u_i(\pi)$$

- **Nash product**

$$sw_N(\pi) = \prod_{i \in \mathcal{A}} u_i(\pi)$$

Social welfare

We can aggregate utility values with a **collective utility function**.

- **Utilitarian social welfare**

$$sw_u(\pi) = \sum_{i \in \mathcal{A}} u_i(\pi)$$

- **Egalitarian (Rawlsian) social welfare**

$$sw_e(\pi) = \min_{i \in \mathcal{A}} u_i(\pi)$$

- **Nash product**

$$sw_N(\pi) = \prod_{i \in \mathcal{A}} u_i(\pi)$$

We also consider the lexicographic minimum (leximin) over $(u_i(\pi))_{i \in \mathcal{A}}$.

Social welfare

We can aggregate utility values with a **collective utility function**.

- **Utilitarian social welfare**

$$sw_u(\pi) = \sum_{i \in \mathcal{A}} u_i(\pi)$$

- **Egalitarian (Rawlsian) social welfare**

$$sw_e(\pi) = \min_{i \in \mathcal{A}} u_i(\pi)$$

- **Nash product**

$$sw_N(\pi) = \prod_{i \in \mathcal{A}} u_i(\pi)$$

We also consider the lexicographic minimum (leximin) over $(u_i(\pi))_{i \in \mathcal{A}}$.

There are also approaches using inequality indices, e.g., Gini index.

Example

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Example

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Utilitarian social welfare:

$\pi = \langle \{o_1\}, \{o_2, o_3\} \rangle \rightarrow sw_u(\pi) = 5 + (4 + 6) = 15$

Example

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Utilitarian social welfare:

$$\pi = \langle\{o_1\},\{o_2,o_3\}\rangle \rightarrow sw_u(\pi) = 5 + (4+6) = 15$$
$$\pi' = \langle\{o_1,o_2\},\{o_3\}\rangle \rightarrow sw_u(\pi') = (5+1) + 6 = 12$$

Example

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Utilitarian social welfare:

$$\pi = \langle \{o_1\}, \{o_2, o_3\} \rangle \rightarrow sw_u(\pi) = 5 + (4 + 6) = 15$$

$$\pi' = \langle \{o_1, o_2\}, \{o_3\} \rangle \rightarrow sw_u(\pi') = (5 + 1) + 6 = 12$$

Egalitarian social welfare:

$$\pi = \langle \{o_1\}, \{o_2, o_3\} \rangle \rightarrow sw_e(\pi) = \min(5, 4 + 6) = 5$$

Example

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Utilitarian social welfare:

$$\pi = \langle \{o_1\}, \{o_2, o_3\} \rangle \rightarrow sw_u(\pi) = 5 + (4 + 6) = 15$$
$$\pi' = \langle \{o_1, o_2\}, \{o_3\} \rangle \rightarrow sw_u(\pi') = (5 + 1) + 6 = 12$$

Egalitarian social welfare:

$$\pi = \langle \{o_1\}, \{o_2, o_3\} \rangle \rightarrow sw_e(\pi) = \min(5, 4 + 6) = 5$$
$$\pi' = \langle \{o_1, o_2\}, \{o_3\} \rangle \rightarrow sw_e(\pi') = \min(5 + 1, 6) = 6$$

Example

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Utilitarian social welfare:

$\pi = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow sw_u(\pi) = 5 + (4 + 6) = 15$

$\pi' = \langle\{o_1, o_2\}, \{o_3\}\rangle \rightarrow sw_u(\pi') = (5 + 1) + 6 = 12$

Egalitarian social welfare:

$\pi = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow sw_e(\pi) = \min(5, 4 + 6) = 5$

$\pi' = \langle\{o_1, o_2\}, \{o_3\}\rangle \rightarrow sw_e(\pi') = \min(5 + 1, 6) = 6$

Nash product:

$\pi = \langle\{o_1\}, \{o_2, o_3\}\rangle \rightarrow sw_N(\pi) = 5 \times (4 + 6) = 50$

Example

	o_1	o_2	o_3
agent 1	5	1	5
agent 2	1	4	6

Utilitarian social welfare:

$$\pi = \langle\{o_1\}, \{o_2, o_3\}\rangle \to sw_u(\pi) = 5 + (4+6) = 15$$
$$\pi' = \langle\{o_1, o_2\}, \{o_3\}\rangle \to sw_u(\pi') = (5+1) + 6 = 12$$

Egalitarian social welfare:

$$\pi = \langle\{o_1\}, \{o_2, o_3\}\rangle \to sw_e(\pi) = \min(5, 4+6) = 5$$
$$\pi' = \langle\{o_1, o_2\}, \{o_3\}\rangle \to sw_e(\pi') = \min(5+1, 6) = 6$$

Nash product:

$$\pi = \langle\{o_1\}, \{o_2, o_3\}\rangle \to sw_N(\pi) = 5 \times (4+6) = 50$$
$$\pi' = \langle\{o_1, o_2\}, \{o_3\}\rangle \to sw_N(\pi') = (5+1) \times 6 = 36$$

Complexity

How hard is it to compute allocations of optimal social welfare?

Complexity

How hard is it to compute allocations of optimal social welfare?

Bundle form:

Utilitarian, egalitarian, and Nash product social welfare optimization are **hard**.

Complexity

How hard is it to compute allocations of optimal social welfare?

Bundle form:

Utilitarian, egalitarian, and Nash product social welfare optimization are **hard**.

k-additive form, $k \geq 1$:

Utilitarian (except for $k = 1$), egalitarian, and Nash product social welfare optimization are **hard**.

But for $m = n$ it is easy.

What to relax?

Since most social welfare optimization problems are **hard**, we have to relax our requirements.

What to relax?

Since most social welfare optimization problems are **hard**, we have to relax our requirements.

- Approximate notions of properties

What to relax?

Since most social welfare optimization problems are **hard**, we have to relax our requirements.

- Approximate notions of properties
- Suboptimal social welfare

What to relax?

Since most social welfare optimization problems are **hard**, we have to relax our requirements.

- Approximate notions of properties
- Suboptimal social welfare
- Restricted model of computation (Part two)

Utilitarian social welfare

Bundle form	Approximability	Reference
general	NP-hard in factor $m^{\varepsilon-1/2}$	[LOS99]
submodular	$1-(1/e)$ NP-hard in factor $1-(1/e)+\varepsilon$	[FGMS06],[CCPV07],[Von08] [KLMM08]
subadditive	hard in factor $1/m^{1/4}$ $1/m^{1/2}$	[DS06] [DNS10]

Egalitarian social welfare

Bundle form	Approximability	Reference
general	NP-hard in any factor	[NRR13]
submodular	$1/(m-n+1)$	[Gol05]
	$1/(m^{1/2}n^{1/4}\log m \log^{3/2} n)$	[GHIM09]
subadditive	$1/(2n-1)$	[KP07]

k-additive form	Approximability	Reference
1-additive	NP-hard in factor $1/2+\varepsilon$ $1/m^{\varepsilon}, \epsilon \in \mathcal{O}(1)$	[BD05] [CCK09]
1-additive, Santa Claus	$\mathcal{O}(1)$	[Fei08],[HSS11]
k-additive, $k \geq 2$	NP-hard in any factor	[NRR13]

Nash product

Restriction	Approximability	Reference
Bundle form	NP-hard in any factor	[NRR13]
1-additive	$1/(m-n+1)^n$	[NR13]
2-additive	NP-hard in factor $21/22+\varepsilon$	[NRR13]
k-additive, $k \geq 3$	NP-hard in any factor	[NRR13]

Part Two

Ordinal preferences & Restricted model

Our approach

Problem: What to do if no clear numerical scale (*e.g* money) is involved...?

Our approach

Problem: What to do if no clear numerical scale (*e.g* money) is involved...?

Idea: *Cardinalize ordinal preferences with the help of scoring vectors.*

About scoring vectors

Here we take inspiration from **voting theory**.

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \dots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring vectors:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \dots, s_m)$ (with $s_1 \geq \dots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring vectors:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1
Lexicographic	32	16	8	4	2	1

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring vectors:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1
Lexicographic	32	16	8	4	2	1
Quasi-Indifference	$1+{}^{s_1}/_M$	$1+{}^{s_2}/_M$	$1+{}^{s_3}/_M$	$1+{}^{s_4}/_M$	$1+{}^{s_5}/_M$	$1+{}^{s_6}/_M$

About scoring vectors

Here we take inspiration from **voting theory**.

We assume that:

1. **Ranking:** Each agent i has a **ranking** $\succ_i$ over $\mathcal{O}$ (ex: $o_6 \succ o_1 \succ o_4 \succ o_5 \succ o_2 \succ o_3$)
2. **Scoring:** We have a common **scoring vector** $s = (s_1, \ldots, s_m)$ (with $s_1 \geq \cdots \geq s_m$) mapping each rank to a utility.
3. **Additivity:** These utilities are **additive**.

4 natural scoring vectors:

$\succ_i$	o_6	o_1	o_4	o_5	o_2	o_3
Borda	6	5	4	3	2	1
Lexicographic	32	16	8	4	2	1
Quasi-Indifference	$1+{s_1}/M$	$1+{s_2}/M$	$1+{s_3}/M$	$1+{s_4}/M$	$1+{s_5}/M$	$1+{s_6}/M$
k-Approval	1	1	0	0	0	0

Example

Example

5 objects, 3 agents...

- 1 : $o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- 2 : $o_4 \succ o_2 \succ o_5 \succ o_1 \succ o_3$
- 3 : $o_1 \succ o_3 \succ o_5 \succ o_4 \succ o_2$

Example

Example

5 objects, 3 agents...

- 1 : $o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- 2 : $o_4 \succ o_2 \succ o_5 \succ o_1 \succ o_3$
- 3 : $o_1 \succ o_3 \succ o_5 \succ o_4 \succ o_2$

Let's consider allocation $\pi = \langle \{o_1\}, \{o_4, o_2\}, \{o_3, o_5\} \rangle$.

Example

Example

5 objects, 3 agents...

- $1 : o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- $2 : o_4 \succ o_2 \succ o_5 \succ o_1 \succ o_3$
- $3 : o_1 \succ o_3 \succ o_5 \succ o_4 \succ o_2$

Let's consider allocation $\pi = \langle \{o_1\}, \{o_4, o_2\}, \{o_3, o_5\} \rangle$.

- **Borda:** $u_1(\pi) = 5$; $u_2(\pi) = 5 + 4 = 9$; $u_3(\pi) = 4 + 3 = 7$.
- **Lexicographic:** $u_1(\pi) = 16$; $u_2(\pi) = 24$; $u_3(\pi) = 12$.
- **s-QI:** $u_1(\pi) = 1 + s_1/M$; $u_2(\pi) = 2 + s_1+s_2/M$; $u_3(\pi) = 2 + s_2+s_3/M$.
- **2-approval:** $u_1(\pi) = 1$; $u_2(\pi) = 2$; $u_3(\pi) = 1$.

Scoring allocation rules

Back to our resource allocation problem...

Scoring allocation rules

Back to our resource allocation problem...

Interpretation:
Borda SF
Lexicographic SF
Quasi-Indifference SF
k-Approval SF

Scoring allocation rules

Back to our resource allocation problem...

Maximize:	$\sum_i u_i(\pi)$
Interpretation:	
Borda SF	
Lexicographic SF	
Quasi-Indifference SF	
k-Approval SF	

Scoring allocation rules

Back to our resource allocation problem...

Maximize: / Interpretation:	$\sum_i u_i(\pi)$	$\min_i u_i(\pi)$
Borda SF		
Lexicographic SF		
Quasi-Indifference SF		
k-Approval SF		

Scoring allocation rules

Back to our resource allocation problem...

Maximize: / Interpretation:	$\sum_i u_i(\pi)$	$\min_i u_i(\pi)$	$\mathrm{leximin}(u_1(\pi), \dots, u_n(\pi))$
Borda SF			
Lexicographic SF			
Quasi-Indifference SF			
k-Approval SF			

Scoring allocation rules

Back to our resource allocation problem...

Maximize: Interpretation:	$\sum_i u_i(\pi)$	$\min_i u_i(\pi)$	$\text{leximin}(u_1(\pi), \ldots, u_n(\pi))$
Borda SF			
Lexicographic SF			
Quasi-Indifference SF			
k-Approval SF			

$\rightsquigarrow$ 12 **positional scoring allocation rules**

How to interpret scores?

Scores are not necessarily agents' utilities.

Question: What are we actually optimizing in the end?

How to interpret scores?

Scores are not necessarily agents' utilities.

Question: What are we actually optimizing in the end?

Two interpretations of scores:

1. Compromise between all agents (domain knowledge, learned, . . .)

How to interpret scores?

Scores are not necessarily agents' utilities.

Question: What are we actually optimizing in the end?

Two interpretations of scores:

1. Compromise between all agents (domain knowledge, learned, ...)
2. Perception of the center

Properties

1. **Separability:** Violated by almost all our rules
2. **Monotonicity:** Satisfied by all our rules
3. **Global monotonicity:** Violated by almost all our rules
   with **strictly decreasing scoring vector**
4. **Pos. object monotonicity:** Satisfied by $+$ for $n = 2$, but **violated**
   for $n \geq 3$ and **strictly decreasing scoring vector**

Baumeister, D., Bouveret, S., Lang, J., Nguyen, N., Nguyen, T., and Rothe, J. (2014).
Scoring rules for the allocation of indivisible goods.
In *Proceedings of ECAI'14*, pages 75–80. IOS Press.

Complexity

What is the precise complexity of these allocation rules?

Complexity

What is the precise complexity of these allocation rules?

For each pair (scoring vector, social criterion), what is the complexity of...

1. **Optimal Allocation Value (OAV):** is it possible to find an allocation of social welfare $\geq K$?
2. **Optimal Allocation (OA):** does π belong to the set of optimal allocations?
3. **Find Optimal Allocation (FOA):** find an optimal allocation.

Complexity

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!

Idea: give each item to the agent that ranks it the best.

Complexity

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!

Idea: give each item to the agent that ranks it the best.

For $\min_i u_i(\pi)$ (egalitarianism):

- **Bad news:** **hard** (NP-complete, coNP-complete, NP-hard for OAV, OA, FOA resp.) for Borda, lexicographic and QI scoring vectors.

(all by reduction from [X3C])

Complexity

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!

Idea: give each item to the agent that ranks it the best.

For $\min_i u_i(\pi)$ (egalitarianism):

- **Bad news:** **hard** (NP-complete, coNP-complete, NP-hard for OAV, OA, FOA resp.) for Borda, lexicographic and QI scoring vectors.

  (all by reduction from [X3C])

- **Good news:** **easy** (polynomial)
  - if the number of objects is fixed (obvious);
  - if the number of agents is fixed (dynamic programming);
  - for k-approval ("known" problem).

Complexity

For $\sum_i u_i(\pi)$ (classical utilitarianism), everything is polynomial!
Idea: give each item to the agent that ranks it the best.

For $\min_i u_i(\pi)$ (egalitarianism):

- **Bad news:** **hard** (NP-complete, coNP-complete, NP-hard for OAV, OA, FOA resp.) for Borda, lexicographic and QI scoring vectors.
  (all by reduction from [X3C])
- **Good news:** **easy** (polynomial)
  - if the number of objects is fixed (obvious);
  - if the number of agents is fixed (dynamic programming);
  - for k-approval ("known" problem).

Most results for min carry over to leximin.

Approximation

Most cases are **hard**...

Question: *Is it possible to efficiently compute* ***good*** *(but potentially suboptimal) allocations?*

Approximation

Most cases are **hard**...

Question: *Is it possible to efficiently compute* ***good*** *(but potentially suboptimal) allocations?*

Our approach: Instead of giving general approximation results, we:

- focus on a **simple** allocation protocol [Bouveret and Lang, 2011];
- and try to analyze how good the allocations it gives are.

Bouveret, S. and Lang, J. (2011).
A general elicitation-free protocol for allocating indivisible goods.
In *Proceedings of IJCAI'11*, pages 73–78. IJCAI.

An elicitation-free protocol...

Ask the agents to pick in turn their most preferred object among the remaining ones, according to some **predefined sequence** σ.

Example

3 agents 1, 2, 3 / 6 objects / sequence 123321 $\rightarrow$ 1 chooses first (and takes her preferred object), then 2, then 3, then 3 again...

An elicitation-free protocol...

Ask the agents to pick in turn their most preferred object among the remaining ones, according to some **predefined sequence** σ.

Example

3 agents 1, 2, 3 / 6 objects / sequence 123321 $\rightarrow$ 1 chooses first (and takes her preferred object), then 2, then 3, then 3 again...

Arguably a very simple (and natural) protocol! **No elicitation required**!

An elicitation-free protocol...

Ask the agents to pick in turn their most preferred object among the remaining ones, according to some **predefined sequence** σ.

Example

3 agents 1, 2, 3 / 6 objects / sequence 123321 $\rightarrow$ 1 chooses first (and takes her preferred object), then 2, then 3, then 3 again...

Arguably a very simple (and natural) protocol! **No elicitation required**!

Here we focus on **regular sequences** σ of the kind $(1 \dots n)^*$ and **Borda**.

Price of elicitation-freeness

Arguably a very simple (and natural) protocol...

Price of elicitation-freeness

Arguably a very simple (and natural) protocol...

...but obviously suboptimal!

Price of elicitation-freeness

Arguably a very simple (and natural) protocol...

...but obviously suboptimal!

Question: *What is the loss in social welfare we incur by using such a simple protocol instead of computing the optimal allocation?*

Price of elicitation-freeness

Arguably a very simple (and natural) protocol...

...but obviously suboptimal!

Question: *What is the loss in social welfare we incur by using such a simple protocol instead of computing the optimal allocation?*

Multiplicative Price of Elicitation-Freeness:

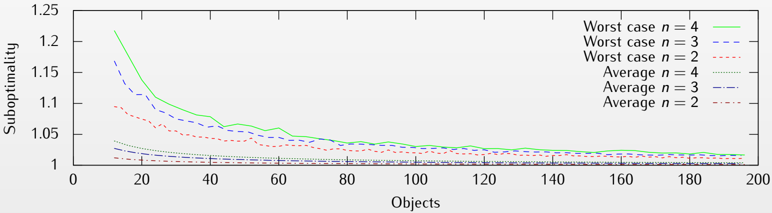
worst case ratio $sw^{\text{opt}}/sw(\sigma)$, for a sequence σ

Additive Price of Elicitation-Freeness:

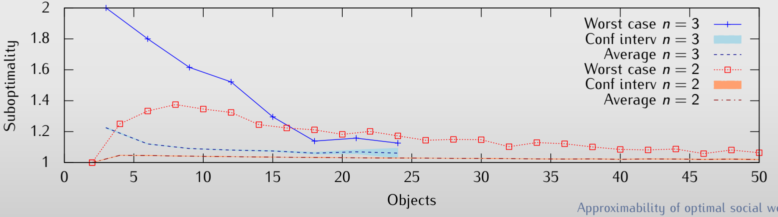
worst case difference $sw^{\text{opt}} - sw(\sigma)$, for a sequence σ

Experimental results

For **classical utilitarianism** ($\sum_i u_i(\pi)$):



For **egalitarianism** ($\min_i u_i(\pi)$):



Classical utilitarianism

For $m = kn$ objects,

$$MPEF_{+} \geq 1 + \frac{mn - m - n^2 + n}{m^2 + mn}$$

Why is this true?

Classical utilitarianism

For $m = kn$ objects,

$$MPEF_+ \geq 1 + \frac{mn - m - n^2 + n}{m^2 + mn}$$

Why is this true?

- $1 : o_6 \succ o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- $2 : o_5 \succ o_6 \succ o_1 \succ o_2 \succ o_3 \succ o_4$
- $3 : o_4 \succ o_5 \succ o_6 \succ o_1 \succ o_2 \succ o_3$

Classical utilitarianism

For $m = kn$ objects,

$$MPEF_+ \geq 1 + \frac{mn - m - n^2 + n}{m^2 + mn}$$

Why is this true?

- $1 : o_6 \succ o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- $2 : o_5 \succ o_6 \succ o_1 \succ o_2 \succ o_3 \succ o_4$
- $3 : o_4 \succ o_5 \succ o_6 \succ o_1 \succ o_2 \succ o_3$

Classical utilitarianism

For $m = kn$ objects,

$$MPEF_{+} \geq 1 + \frac{mn - m - n^2 + n}{m^2 + mn}$$

Why is this true?

- $1 : o_6 \succ o_1 \succ o_2 \succ o_3 \succ o_4 \succ o_5$
- $2 : o_5 \succ o_6 \succ o_1 \succ o_2 \succ o_3 \succ o_4$
- $3 : o_4 \succ o_5 \succ o_6 \succ o_1 \succ o_2 \succ o_3$

Classical utilitarianism

For $m = kn$ objects,

$$MPEF_+ \leq 2 - \frac{m-n}{mn+n}$$

Classical utilitarianism

For $m = kn$ objects,

$$MPEF_+ \leq 2 - \frac{m-n}{mn+n}$$

Why is this true?

If at a time step agent j gets object g_{ni+j} we learn

- a lower bound on agent j's value for this object

Classical utilitarianism

For $m = kn$ objects,

$$MPEF_+ \leq 2 - \frac{m-n}{mn+n}$$

Why is this true?

If at a time step agent j gets object g_{ni+j} we learn

- a lower bound on agent j's value for this object
- an upper bound on all agents' values for this object

Classical utilitarianism

For $m = kn$ objects,

$$MPEF_{+} \leq 2 - \frac{m-n}{mn+n}$$

Why is this true?

If at a time step agent j gets object g_{ni+j} we learn

- a lower bound on agent j's value for this object
- an upper bound on all agents' values for this object

Corollary:
If $n = 2$ and $m = 2k$,

$$1 + \frac{m-2}{m(m+2)} \leq MPEF_{+} \leq \frac{3}{2} + \frac{3}{2m+2}$$

Egalitarianism

For $m = kn$ objects,

$$MPEF_{\min} \leq \frac{2mn - m + n}{mn + 2n - n^2}$$

Egalitarianism

For $m = kn$ objects,

$$MPEF_{\min} \leq \frac{2mn - m + n}{mn + 2n - n^2}$$

Why is this true?

- Upper bound $MPEF_{\min}$ using best and worst case profile

Egalitarianism

For $m = kn$ objects,

$$MPEF_{\text{min}} \leq \frac{2mn - m + n}{mn + 2n - n^2}$$

Why is this true?

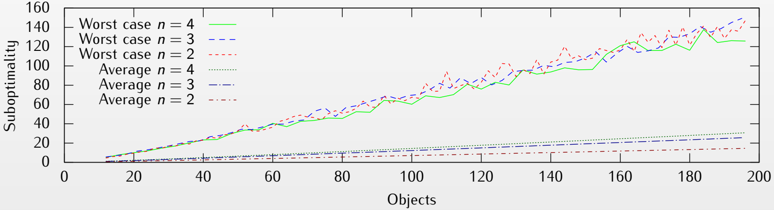
- Upper bound $MPEF_{\text{min}}$ using best and worst case profile

Corollary:
If $n = 2$ and $m = 2k$,

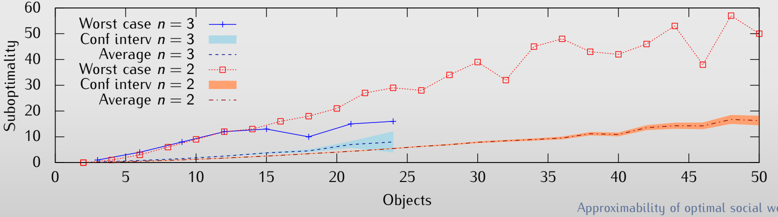
$$MPEF_{\text{min}} \leq \frac{3}{2} + \frac{5}{m + 4}$$

Experimental results

For **classical utilitarianism** ($\sum_i u_i(\pi)$):



For **egalitarianism** ($\min_i u_i(\pi)$):



Formal bounds

For $m = kn$ objects,

$$\frac{(n-1)(m-n)}{2} \leq APEF_{+} \leq \frac{(m-n)(mn-m+n^2+n)}{2n}$$

$$APEF_{\min} \leq \frac{m^2n - mn - m^2 + mn^2}{2n^2}$$

Future Work

- Closing gaps (upper and lower bounds)
- Relationships (rank weighted utilitarianism, inequality indices)
- Exact characterizations, manipulation (scoring allocation rules)

Discussion

On finding rankings with approximately optimal Kemeny score:

> *An approximation algorithm for a voting rule is, in effect, a different voting rule; and in real-world elections, voters may feel deceived if a different voting rule is used than the one that was promised to them.*
>
> – Conitzer, Davenport, and Kalagnanam (2006)

Discussion

On finding rankings with approximately optimal Kemeny score:

> *An approximation algorithm for a voting rule is, in effect, a different voting rule; and in real-world elections, voters may feel deceived if a different voting rule is used than the one that was promised to them.*
>
> – Conitzer, Davenport, and Kalagnanam (2006)

Does this argument hold in the resource allocation setting as well?

Discussion

On finding rankings with approximately optimal Kemeny score:

> *An approximation algorithm for a voting rule is, in effect, a different voting rule; and in real-world elections, voters may feel deceived if a different voting rule is used than the one that was promised to them.*
>
> – Conitzer, Davenport, and Kalagnanam (2006)

Does this argument hold in the resource allocation setting as well?

References

BBLNNR14 D. Baumeister, S. Bouveret, J. Lang, N. Nguyen, T. Nguyen, and J. Rothe. Scoring Rules for the Allocation of Indivisible Goods. In *Proceedings of ECAI'14*, pages 75–89. IOS Press, 2014.

BD05 I. Bezáková and V. Dani. Allocating indivisible goods. *SIGecom Exchanges*, 5(3):11–18, 2005.

BL11 S. Bouveret and J. Lang. A General Elicitation-Free Protocol for Allocating Indivisible Goods. In *Proceedings of IJCAI'11*, pages 73–78. IJCAI, 2011.

BL14 S. Bouveret and M. Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of critieria. In *Proceedings of AAMAS'14*, pages 1321–1328. IFAAMAS, 2014.

CCPV07 G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In *Proceedings of the 12th International Integer Programming and Combinatorial Optimization Conference*, LNCS, vol. 4512, pages 182–196. Springer, 2007.

CCK09 D. Chakrabarty, J. Chuzhoy, S. Khanna. On allocating goods to maximize fairness. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer science*, pages 107–116. IEEE Computer Society Press, 2009.

CDK06 V. Conitzer, A. Davenport, and J. Kalagnanam. Improved Bounds for Computing Kemeny Rankings. In *Proceedings of AAAI'06*, vol. 6, pages 620–626. AAAI, 2006.

DS06 S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 1064–1073. ACM Press, 2006.

DNS10 S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *Math. Oper.Res.*, 35(1):1–13, 2010.

Fei08 U. Feige. On allocations that maximize fairness. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms*, pages 287–293. Society for Industrial and Applied Mathematics, 2008.

FGMS06 L. Fleischer, M. Goemans, V. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 611–620. ACM Press, 2006.

Gol05 D. Golovin. Max-min fair allocation of indivisible goods. Technical Report CMU-CS-05-144, School of Computer Science, Carnegie Mellon University, June 2005.

References

GHIM09 M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*, pages 535–544. Society for Industrial and Applied Mathematics, 2009.

HSS11 B. Haeupler, B. Saha, and A. Srinivasan. New constructive aspects of the Lovász Local Lemma. *Journal of the ACM*, 58(6), December 2011.

LOS99 D. Lehmann, L. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 96–102. ACM Press, 1999.

NR13 T. Nguyen and J. Rothe. Envy-Ratio and Average-Nash Social Welfare Optimization in Multiagent Resource Allocation (Extended Abstract). In *Proceedings of AAMAS'13*, pages 1139–1340. IFAAMAS, 2013.

NRR13 T. Nguyen, M. Roos, and J. Rothe. A Survey of Approximability and Inapproximability Results for Social Welfare Optimization in Multiagent Resource Allocation. *Annals of Mathematics and Artificial Intelligence*, 68:65-90, 2013.

KLMM08 S. Khot, R. Lipton, E. Markakis, and A. Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. *Algorithmica*, 52(1):3–18, 2008.

KP07 S. Khot and A. Ponnuswami. Approximation algorithms for the max-min allocation problem. In *Proceedings of Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. 10th International Workshop APPROX 2007 and 11th International Workshop RANDOM 2007*, LNCS, vol. 4627, pages 204–217. Springer, 2007.

Von08 J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 67–74. ACM Press, July 2008.