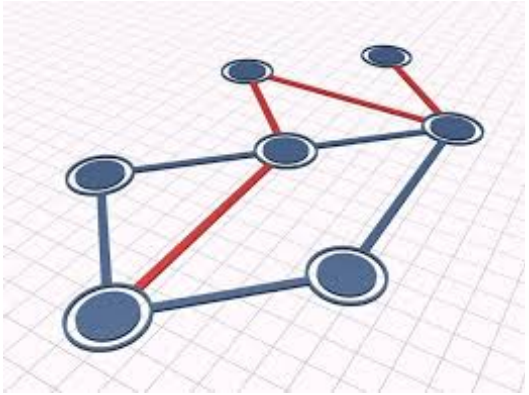
A Game-Theoretic View of Randomized Fair Multi-Agent Optimization

Hugo Gilbert and Olivier Spanjaard
olivier.spanjaard@lip6.fr
Sorbonne Université

UTC talk - april 2018

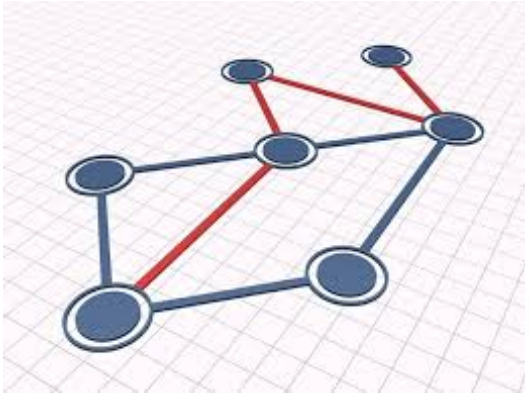
2018/04/12

Multi-Agent Optimization



– $\mathcal{P}$ is a multiagent optimization problem involving n agents.

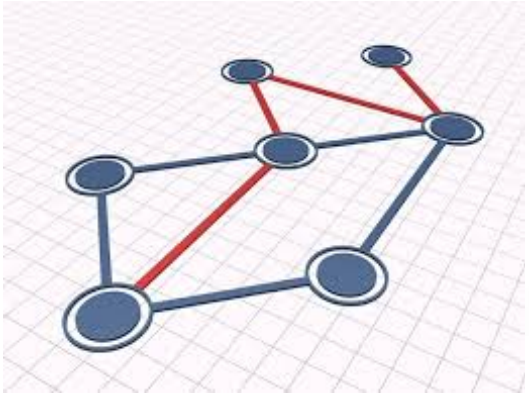
Multi-Agent Optimization



– $\mathcal{P}$ is a multiagent optimization problem involving n agents.

– $\mathcal{X} \subseteq \{0,1\}^p = \{\mathbf{x} \in \{0,1\}^p : A\mathbf{x} = \mathbf{b}\}$ is the set of feasible solutions.

Multi-Agent Optimization

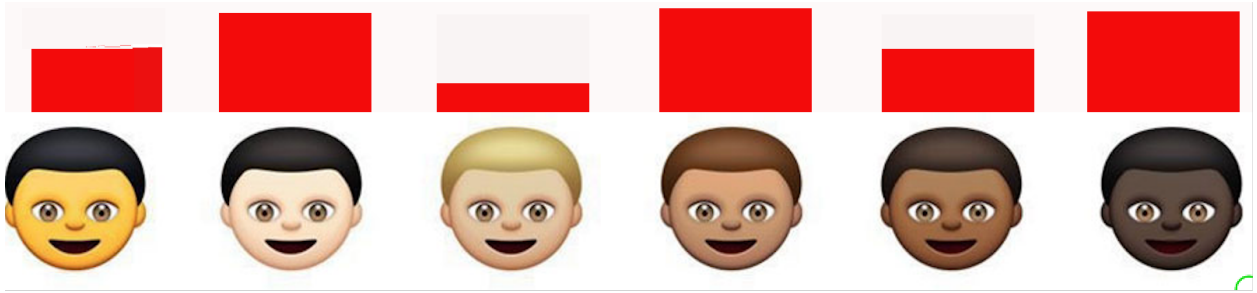


– $\mathcal{P}$ is a multiagent optimization problem involving n agents.

– $\mathcal{X} \subseteq \{0,1\}^p = \{\mathbf{x} \in \{0,1\}^p : A\mathbf{x} = \mathbf{b}\}$ is the set of feasible solutions.

– Each agent has a (dis)utility function $u_i : \mathcal{X} \to \mathbb{R}^+$ to optimize where $u_i(\mathbf{x}) = \sum_{j=1}^{p} u_{ij} x_j$.

Every solution $\mathbf{x}$ thus induces a (dis)utility vector $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}) \cdots u_n(\mathbf{x}))$.



The OWA Operator

A well known criterion to identify fair solutions is the *Ordered Weighted Average* (OWA) operator **with decreasing weights** [Yager, 1988].

The OWA operator

Let $\mathbf{w} = (w_1, \ldots, w_n)$ be a vector of weights. The $\mathtt{OWA}_{\mathbf{w}}(\cdot)$ operator induced by $\mathbf{w}$ is defined by:

$\forall$ utility vector $\mathbf{u}, \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}) = \sum_{i=1}^{n} w_i \vec{\mathbf{u}}_i$

where $\vec{\mathbf{u}}$ is vector $\mathbf{u}$ sorted in nondecreasing order: $\vec{\mathbf{u}}_1 \leq \ldots \leq \vec{\mathbf{u}}_n$

Example

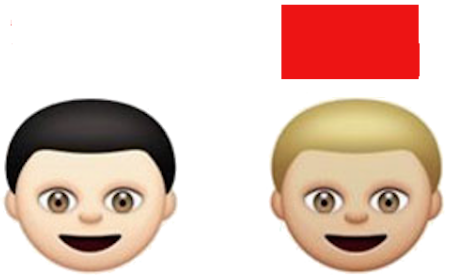
$w = (0.4, 0.3, 0.2)$

- $\mathtt{OWA}_{\mathbf{w}}(8, 11, 7) = 7 * 0.4 + 8 * 0.3 + 11 * 0.2 = 8.5$
- $\mathtt{OWA}_{\mathbf{w}}(5, 15, 5) = 5 * 0.4 + 5 * 0.3 + 15 * 0.2 = 8$
- $\mathtt{OWA}_{\mathbf{w}}(4, 9, 7) = 4 * 0.4 + 7 * 0.3 + 9 * 0.2 = 6.4$
- $\mathtt{OWA}_{\mathbf{w}}(7, 14, 6) = 6 * 0.4 + 7 * 0.3 + 14 * 0.2 = 8.7$

Why Randomization?

Machina's Mom

A mother with two children has one indivisible treat.
She can give it to either one of her children but not both.
Neither $\mathbf{x^1} = (1, 0)$ nor $\mathbf{x^2} = (0, 1)$ seem satisfying.



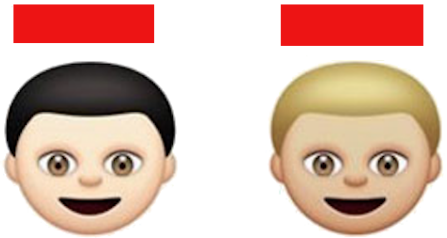
Why Randomization?

Machina's Mom

A mother with two children has one indivisible treat.
She can give it to either one of her children but not both.
Neither $\mathbf{x^1} = (1, 0)$ nor $\mathbf{x^2} = (0, 1)$ seem satisfying.

$$\text{tossing a coin} \succsim \text{choosing herself}$$

– Both children have equal chances of having the treat.



Randomized Setting

A mixed solution is a lottery over solutions and is denoted by $P_{\mathcal{X}}$.

The set of all possible mixed solutions is denoted by $\Delta_{\mathcal{X}}$.

The definition of a utility vector is extended by linearity to mixed solutions. More formally, $u_i(P_{\mathcal{X}}) = \sum_{\mathbf{x} \in \mathcal{X}} P_{\mathcal{X}}(\mathbf{x}) u_i(\mathbf{x})$.

Machina's Mom Continued

Machina's Mom Continued

Consider pure solutions $\mathbf{x^1}, \mathbf{x^2}$ and the mixed solution $P_{\mathcal{X}}$ defined by $P_{\mathcal{X}}(\mathbf{x^i}) = 0.5$ for $i \in \{1, 2\}$ such that:

$$\mathbf{u}(P_{\mathcal{X}}) = 0.5\mathbf{u}(\mathbf{x^1}) + 0.5\mathbf{u}(\mathbf{x^2}) = (0.5, 0.5)$$

If $w_1 = 3/4$ and $w_2 = 1/4$, we have:

$$\begin{aligned}
\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x^1})) &= \mathtt{OWA}_{\mathbf{w}}((1, 0)) = 0.75 \cdot 0 + 0.25 \cdot 1 = 0.25 \\
\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x^2})) &= \mathtt{OWA}_{\mathbf{w}}((0, 1)) = 0.75 \cdot 0 + 0.25 \cdot 1 = 0.25 \\
\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) &= \mathtt{OWA}_{\mathbf{w}}((0.5, 0.5)) = 0.75 \cdot 0.5 + 0.25 \cdot 0.5 = 0.5
\end{aligned}$$

The mixed solution $P_{\mathcal{X}}$ is therefore preferred to both pure solutions.

Randomized Fair Multi-Agent Assignment Problem

Data:

$\mathbf{w} = (0.6, 0.3, 0.1)$

	🎾	🎸	📚
👨🏻	6	2	10
👨🏾	6	4	8
👨🏼	8	3	9

Sol. $\mathbf{x} \in \mathcal{X}$:

	🎾	🎸	📚
👨🏻	0	1	0
👨🏾	0	0	1
👨🏼	1	0	0

Value: $\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x}))$

$2 * 0.6 + 8 * 0.3 + 10 * 0.1$

$= 4.6$

Sol. $\tilde{\mathbf{x}} \in \mathrm{Conv}(\mathcal{X})$:

	🎾	🎸	📚
👨🏻	0.8	0	0.2
👨🏾	0.2	0.3	0.5
👨🏼	0	0.7	0.3

Value: $\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\tilde{\mathbf{x}}))$

$4.8 * 0.6 + 5.2 * 0.3 + 6.4 * 0.1$

$= 5.08$

0.3	🎾	🎸	📚
👨🏻	1	0	0
👨🏾	0	1	0
👨🏼	0	0	1

0.5	🎾	🎸	📚
👨🏻	1	0	0
👨🏾	0	0	1
👨🏼	0	1	0

0.2	🎾	🎸	📚
👨🏻	0	0	1
👨🏾	1	0	0
👨🏼	0	1	0

Important Remark

Note that:

- Determining a solution $\mathbf{x}$ maximizing $\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x}))$ is NP-hard [Lesca, Minoux & Perny, *Algorithmica 2018*]
- $\exists$ a compact MIP formulation for determining such a solution
- The relaxation of this MIP yields a solution $\tilde{\mathbf{x}}$ maximizing $\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x}))$

Why not solving this relaxation and computing $P_{\mathcal{X}}$ from $\tilde{\mathbf{x}}$, all this in polynomial time ?

Actually we could! But we provide:

- A general polynomial complexity result that holds for many fair randomized optimization problems (in particular, no need for a compact MIP formulation)
- A procedure that runs faster in practice

Our contribution relies on a game theoretic view of randomized multi-agent optimization problems which enables to entail theoretical results and computational methods.

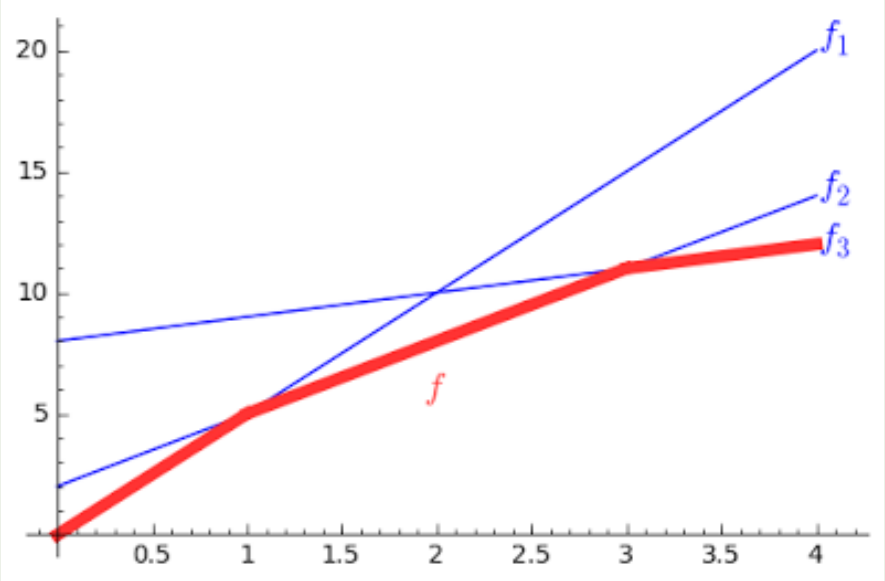
Game-Theoretic View of the Problem

Game Theoretic View

If weights are decreasing, $\mathtt{OWA}_{\mathbf{w}}$ is concave piecewise-linear and therefore:

$$\forall P_{\mathcal{X}} \in \Delta_{\mathcal{X}}, \quad \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) = \min_{\sigma \in \Sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}}))$$

where Σ is the set of permutations of $\{1, \ldots, n\}$ and $\mathbf{w}_\sigma = (w_{\sigma(1)}, \ldots, w_{\sigma(n)})$.



Game-Theoretic View of the Problem

Game Theoretic View

If weights are decreasing, $\mathtt{OWA}_{\mathbf{w}}$ is concave piecewise-linear and therefore:

$$\forall P_{\mathcal{X}} \in \Delta_{\mathcal{X}}, \quad \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) = \min_{\sigma \in \Sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}}))$$

where Σ is the set of permutations of $\{1, \ldots, n\}$ and $\mathbf{w}_\sigma = (w_{\sigma(1)}, \ldots, w_{\sigma(n)})$.

$$\max_{P_{\mathcal{X}}} \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) \Rightarrow \max_{P_{\mathcal{X}} \in \Delta_{\mathcal{X}}} \min_{\sigma \in \Sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}}))$$

Game-Theoretic View of the Problem

Game Theoretic View

If weights are decreasing, $\mathtt{OWA}_{\mathbf{w}}$ is concave piecewise-linear and therefore:

$$\forall P_{\mathcal{X}} \in \Delta_{\mathcal{X}}, \quad \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) = \min_{\sigma \in \Sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}}))$$

where Σ is the set of permutations of $\{1, \ldots, n\}$ and $\mathbf{w}_\sigma = (w_{\sigma(1)}, \ldots, w_{\sigma(n)})$.

$$\max_{P_{\mathcal{X}}} \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) \Rightarrow \max_{P_{\mathcal{X}} \in \Delta_{\mathcal{X}}} \min_{\sigma \in \Sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}}))$$

Two-player zero-sum game:

– the set of pure strategies for player 1 is the set of feasible solutions $\mathcal{X}$;

– the set of pure strategies for player 2 is the set of permutations Σ;

– the payoffs are given by values $\mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(\mathbf{x}))$ for $\mathbf{x} \in \mathcal{X}$ and $\sigma \in \Sigma$.

Illustration On The Fair Assignment Problem

Strategies of the **x**-player:

	🎾	🎸	📖	
👨🏻	6	2	10	0.3
👨🏿	6	4	8	0.6
👨🏼	8	3	9	0.1
	🎾	🎸	📖	
👨🏻	6	2	10	0.3
👨🏿	6	4	8	0.1
👨🏼	8	3	9	0.6
- ...

Strategies of the σ-player:

- $\mathbf{w}_\sigma = (0.3, 0.6, 0.1)$
- $\mathbf{w}_\sigma = (0.3, 0.1, 0.6)$
- ...

The solution $P_{\mathcal{X}}$ played by the **x**-player in an NE is an optimal mixed solution for OWA.

Payoff: $\mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(\mathbf{x}))$

A mixed **Nash Equilibrium** (NE) is a couple $(P_{\mathcal{X}}, P_\Sigma)$ s.t.:

$$\max_{P_{\mathcal{X}}} \min_{\sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}})) = \min_{P_\Sigma} \max_{\mathbf{x}} \mathtt{WA}_{\mathbf{w}_{P_\Sigma}}(\mathbf{u}(\mathbf{x}))$$

where $\mathtt{WA}_{\mathbf{w}_{P_\Sigma}}(\mathbf{u}(P_{\mathcal{X}})) = \sum_\sigma \sum_{\mathbf{x}} P_\Sigma(\sigma) P_{\mathcal{X}}(\mathbf{x}) \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{x})$

Finding the Nash Equilibrium P_{FO}

Naive solution: let's try to find a Nash Equilibrium by linear programming:

$$\mathcal{P}_{FO} \begin{cases} \max\limits_{v, p_{\mathbf{x}}: \mathbf{x} \in \mathcal{X}} v \\ v \leq \sum\limits_{i=1}^{n} w_{\sigma(i)} \Big(\sum\limits_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} u_i(\mathbf{x}) \Big) \quad \forall \sigma \in \Sigma \\ \sum\limits_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} = 1 \\ v \in \mathbb{R} \quad p_{\mathbf{x}} \geq 0 \quad \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

Unfortunately the number of variables is $|\mathcal{X}| + 1$ and the number of constraints is $|\Sigma| + 1$. The game is **too large** to be solved directly!

Finding the Nash Equilibrium P_{FO}

Naive solution: let's try to find a Nash Equilibrium by linear programming:

$$\mathcal{P}_{FO} \begin{cases} \max\limits_{v, p_{\mathbf{x}} : \mathbf{x} \in \mathcal{X}} v \\ v \leq \sum\limits_{i=1}^{n} w_{\sigma(i)} \Big(\sum\limits_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} u_i(\mathbf{x}) \Big) \quad \forall \sigma \in \Sigma \\ \sum\limits_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} = 1 \\ v \in \mathbb{R} \quad p_{\mathbf{x}} \geq 0 \quad \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

Unfortunately the number of variables is $|\mathcal{X}| + 1$ and the number of constraints is $|\Sigma| + 1$. The game is **too large** to be solved directly!

We explored two solution procedures to alleviate this difficulty.

Cutting Planes After Reformulation of P_{FO}

Two phases: 1) looking for a solution $\tilde{\mathbf{x}}$, then 2) computing $P_{\mathcal{X}}$ from $\tilde{\mathbf{x}}$

$$\max_{v, p_{\mathbf{x}}: \mathbf{x} \in \mathcal{X}} v$$

$$\underbrace{\begin{aligned} & v \leq \sum_{i=1}^{n} w_{\sigma(i)} \Big(\sum_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} u_i(\mathbf{x}) \Big) \quad \forall \sigma \in \Sigma \\ & \sum_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} = 1 \\ & v \in \mathbb{R} \quad p_{\mathbf{x}} \geq 0 \quad \forall \mathbf{x} \in \mathcal{X} \end{aligned}}_{\mathcal{P}_{FO}}$$

$|\mathcal{X}| + 1$ variables

$$\max_{v, \tilde{x}_1, \ldots, \tilde{x}_p} v$$

$$\underbrace{\begin{aligned} & v \leq \sum_{i=1}^{n} w_{\sigma(i)} u_i(\tilde{\mathbf{x}}) \qquad \forall \sigma \in \Sigma \\ & \tilde{\mathbf{x}} \in \mathrm{Conv}(\mathcal{X}) \end{aligned}}_{\widetilde{\mathcal{P}}_{FO}}$$

$\boxed{p + 1 \text{ variables}}$

We use a cutting planes method to handle the exponential number of constraints in Σ and/or $\mathrm{Conv}(\mathcal{X})$.

A General Polynomial Complexity Result

If you can optimize over $\mathcal{X}$ in polynomial time:

1) solving $\widetilde{\mathcal{P}}_{FO}$ is of polynomial complexity.
(by polynomial time equivalence between *optimization* and *separation* [Grotschel et al., 1981])

2) computing $P_{\mathcal{X}}$ from $\tilde{\mathbf{x}}$ is of polynomial complexity.
(by solving an LP [Mastin et al., 2015])

Theorem

Any randomized fair optimization problem such that optimizing over $\mathcal{X}$ is polynomial in p is polynomially solvable in p and n.

Double oracle [McMahan et al., 2003]

Main idea

Dynamically builds a **sub-game** in which a **mixed** NE is also a NE of the whole game, by using two **oracle procedures** (one for each player).

Oracle procedure

Given a mixed strategy of the opponent, an oracle procedure efficiently computes the **best pure strategy** (**in the whole game**) to play against this mixed strategy.

Double oracle [McMahan et al., 2003]

Description of the algorithm

1. Initialize with a small subset of strategies for each player.
2. Compute a mixed NE of this sub-game.
3. Using the oracles, for each player, compute a best response to the current optimal mixed strategy of the opponent.
4. If at least a new strategy has been generated, go to 2, otherwise exit.

- Convergence is achieved when the best-response oracles generate pure strategies that are already in the sub-game.
- At convergence the NE of the sub-game is an NE of the entire game.

Best response procedures

Best response oracle for the **x**-player

Amounts to solving problem $\mathcal{P}$ according to a WA criterion with weights $\tilde{\sigma}$ defined by $\tilde{\sigma}_i = \sum_{\sigma \in \Sigma} P_\Sigma(\sigma) w_{\sigma(i)}$.

	🎾	🎸	📚	0.5	0.5	
👨🏻	6	2	10	0.3	0.1	0.2
👨🏾	6	4	8	0.1	0.3	0.2
👨🏼	8	3	9	0.6	0.6	0.6

$\longrightarrow$

	🎾	🎸	📚
👨🏻	1.2	0.4	**2**
👨🏾	1.2	**0.8**	1.6
👨🏼	**4.8**	1.8	4.5

Best response oracle for the σ-player

Amounts to sorting vector $\mathbf{u}(P_\mathcal{X})$ (independent of $\mathcal{P}$).

	🎾	🎸	📚	
👨🏻	6	2	10	0.3
👨🏾	6	4	8	0.1
👨🏼	8	3	9	0.6

Experiments

Assignment problem with an increasing number of agents.

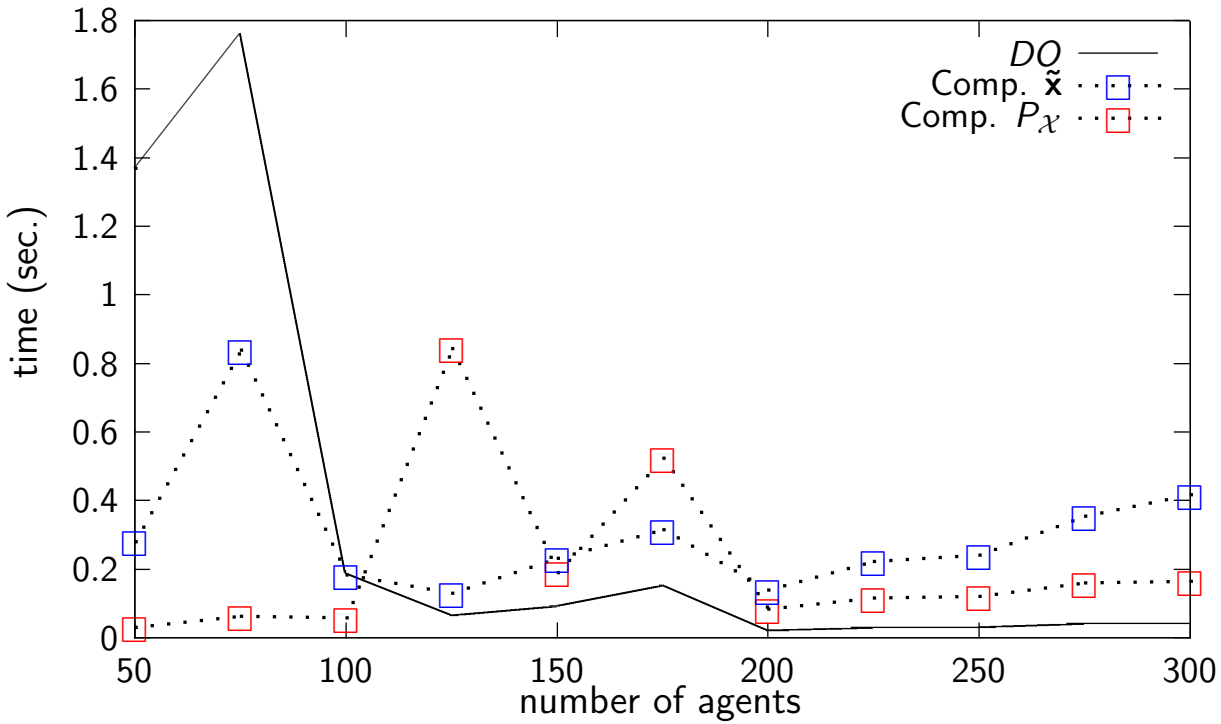


Figure: Costs are generated in $[1, 20]$.

There is a phase transition phenomenon.

Summary

- **Randomized fair multi-agent optimization** is **easy** if the standard variant of the problem is easy
- To solve a randomized fair multi-agent optimization problem, a **double oracle** procedure (which is not polynomial time) **performs better** in practice than a cutting-plane method.
- This result has been successfully applied to the **stable marriage problem**.

THANK YOU!

M. Grötschel and L. Lovász and A. Schrijver,
The ellipsoid method and its consequences in combinatorial optimization.
Combinatorica, 1, 169–197, 1981.

A. Mastin and P. Jaillet and S. Chin,
Randomized minmax regret for combinatorial optimization under uncertainty,
Int. Symp. on Algorithms and Computation, 491–501, 2015.

H. B. McMahan and G. J. Gordon and A. Blum,
Planning in the presence of cost functions controlled by an adversary,
ICML, 536–543, 2003.

H. Moulin,
Axioms of cooperative decision making.
Cambridge University Press,15,1991.

W. Ogryczak and T. Śliwiński,
On solving linear programs with the ordered weighted averaging objective.
European Journal of Operational Research, 148, 80–91, 2003.

R. R. Yager,
On ordered weighted averaging aggregation operators in multicriteria decisionmaking.
IEEE Transactions on systems, Man, and Cybernetics,1988.