Cooperation in Multi-Organization Matching

Fanny Pascual, LIP6

Joint work with Laurent Gourvès and Jérôme Monnot (Lamsade)

Gamecomp, 22 mai 2008
Outline

1. Presentation of the problem
   • Definition
   • Applications
   • Advantages and limits of the cooperation

2. Complexity

3. Approximation

4. Generalization: relaxing the selfishness
Presentation of the problem

Given:
- a set of $k$ organizations $O_1, \ldots, O_k$ (agencies)
- a bipartite graph $G=(V_1,V_2,E)$ where each vertex belongs to an organization.

Cost of edge $e$: $w(e)$ (price the buyer can pay)

Aim of an organization: maximize its gain.
(percentage on the amount of transactions done)

- $p_1$ and $p_2$ s.t. $0 \leq p_1 \leq p_2 \leq 1$ and $p_1 + p_2 = 1$.

Profit of $O_i$ in a matching $M$:
$Gain(O_i,M) = p_1 \sum_{e \in (M \cap V_1) \cap O_i} w(e) + p_2 \sum_{e \in (M \cap V_2) \cap O_i} w(e)$

Example: $Gain(O_2,M) = w(e_2) + p_1 w(e_3) + p_2 w(e_1)$
Let $\text{GainAlone}(O_i)$ be the max. weight of a matching induced by the edges of $O_i$.

The multi-organization assignment problem (MOA):
Find a maximum weight matching $M$ such that for each $O_i$:

$$\text{Gain}(O_i,M) \geq \text{GainAlone}(O_i).$$

Notation: $\text{Gain}(M) = \bigcup_{1 \leq i \leq k} \text{Gain}(O_i,M)$
Presentation of the problem

Example:
(all the weights are equal to 1, \( p_1 = p_2 = 0.5 \))

\[
\begin{align*}
\text{Gain Alone}(O_1) &= 2 \\
\text{Gain Alone}(O_2) &= 1 \\
\text{Gain Alone}(O_3) &= 2
\end{align*}
\]

\[
\begin{align*}
\text{Gain}(O_1, M) &= 2 \\
\text{Gain}(O_2, M) &= 2.5 \\
\text{Gain}(O_3, M) &= 2.5
\end{align*}
\]
A scheduling example

Each organization owns:
- Machines (which may be different)
- Users: each user wants to execute a unit task on a machine, and gives its preferences (integer between 1 and B).

Aim of each organization: maximize the average satisfaction of its users.

Global aim: maximize the average total satisfaction.

MOA problem with $p_1 = 1$ and $p_2 = 0$. 

Tasks

(Machine, time slot)

$T_{1A}$ $M_1,[0,1]$ $T_{1B}$ $M_1,[1,2]$ $T_{1C}$

$T_{2A}$ $M_2,[0,1]$ $T_{2B}$ $M_2,[2,3]$ $M_2,[3,4]$ $T_{3A}$ $M_3,[0,2]$ $M_4,[0,1]$
Cooperation can help a lot

Cooperation allows much better solutions.

Without cooperation:
Gain = 2ε << 1

With cooperation:
Gain = 1
Limits of cooperation

- **Non-cooperating game:**
  - **Players** = organizations
  - **Strategies** = \{accept the proposed solution; compute its own maximum matching\}

- **Price of stability =**
  \[
  \max_{\text{instances}} \frac{\text{Gain in the best Nash equilibrium}}{\text{Gain in the best solution}}
  \]
  \[
  \max_{\text{instances}} \frac{\text{Gain}(\text{MOAopt})}{\text{Gain}(M^*)} \quad (M^* \text{ is a max. weight matching of } G)
  \]
Limits of cooperation

- Proposition: The price of stability is $p_1$.

\[
\text{Price of stability } \geq \frac{\text{Gain(MOAopt)}}{\text{Gain(M*)}} = \frac{1}{1/p_1 - \varepsilon} = \frac{p_1}{1 - (p_1 \varepsilon)}
\]
Outline

1. Presentation of the problem
   • Definition
   • Applications
   • Advantages and limits of the cooperation

2. Complexity
   • A polynomial case
   • General case

3. Approximation

4. Generalization: relaxing the selfishness
A polynomial case

- **Proposition:** If the graph is unweighted, then the MOA problem is polynomial time solvable.

**Algorithm:**
- Compute a maximum weight matching for each organization.
- Increase the size of the matching of $G$ by augmenting its paths while it is possible.
A polynomial case

• This algorithm returns an optimal solution:
  - Improving the matching via an augmenting an alternating path does not diminish the gain of any organization.
  - The resulting matching is feasible and of max. cardinality since no more augmenting path exists.

Gain = 1  \quad \text{Gain} = p1 + p2 = 1
General case

• Proposition: The MOA problem is NP-hard if $k \geq 2$.

  Proof: Reduction from the Partition problem.

• Proposition: The MOA problem is strongly NP-hard if $k$ is not fixed.

  Proof: Reduction from the 3-Partition problem.
Outline

1. Presentation of the problem
2. Complexity
3. Approximation
   • Approximate algorithm
   • Inapproximability
4. Generalization: relaxing the selfishness
Approximation algorithms

- An algorithm $A$ is $x$-approximate if
  \[ \max_{\text{instances}} \frac{\text{Gain in the solution returned by } A}{\text{Gain in the best solution}} \geq x \]

$\Rightarrow$ Algorithm $\frac{1}{2}$-approximate: returns a matching whose gain is at least $\frac{1}{2}$ the gain of an optimal solution of the MOA problem.
Approximate algorithm (APPROX)

- Construct from $G$ a graph $G'$ with the same vertices and edges and s.t. for each $e$ of $E$:
  - if $e$ is shared between 2 organizations: $w(e') = p_1 w(e)$;
  - otherwise $w'(e) = w(e)$.

- Return a maximum weight matching in $G'$.

$p_1 = p_2 = \frac{1}{2}$:
Approximate algorithm

Proposition: APPROX is a $p_1$-approximate algorithm for the MOA problem.

Proof: ($p_1=p_2=\frac{1}{2}$):

- APPROX is $\frac{1}{2}$-approximate:
  
  Gain of the returned solution $\geq \text{Gain}(M^*(G'))$
  
  In $G'$, the weight of each edge has been at most divided by 2.
  
  Thus $\text{Gain}(M^*(G')) \geq \frac{1}{2} \text{Gain}(M^*(G)) \geq \frac{1}{2} \text{Gain}(\text{MOAopt})$
Approximate algorithm

• $M$, the solution returned by APPROX is feasible:
  - $\text{Mint}(i)$: edges of $M$ whose both endpoints belong to $O_i$
  - $\text{Mshared}(i)$: edges of $M$ whose 1 endpoint belongs to $O_i$

$M$ is a max. weight matching of $G'$

$\Rightarrow \text{Gain}'(\text{Mint}(i)) + \text{Gain}'(\text{Mshared}(i)) \geq \text{GainAlone}(i)$

$\text{Gain}(O_i) = \text{Gain}(\text{Mint}(i)) + \frac{1}{2} \text{Gain}(\text{Mshared}(i))$

$\geq \text{Gain}'(\text{Mint}(i)) + \text{Gain}'(\text{Mshared}(i))$

$\geq \text{GainAlone}(i)$
Inapproximation

Proposition: If $k \geq 3$, for all $\varepsilon > 0$, 
$(p_1 + \varepsilon)$-approximation is NP-hard.

Proof: We can map a Partition instance into a MOA instance s.t. there are two possible optimal solutions $A$ and $A/p_1$ and so that 
$\exists$ Partition $\iff$ OPT = $A$

$(p_1 + \varepsilon)$-approximate algorithm can distinguish between the instances with OPT = $A$ from instances with OPT = $A/p_1$. 
Outline

1. Presentation of the problem
2. Complexity
3. Approximation
4. Generalization: relaxing the selfishness
Relaxing the selfishness of the organizations

- Each organization accepts to divide its gain by $\alpha \geq 1$.

- **MOA($\alpha$) problem**: find a max. weight matching s.t. \( \text{Gain}(O_i,M) \geq \text{GainAlone}(O_i) / \alpha \).

- If $\alpha = 1$: this is the MOA problem.
- If $\alpha \geq 1/p_1$: a max. weight matching (without taking into account the constraints of the organizations) is feasible.
- What happens when $1 > \alpha > 1/p_1$?
Relaxing the selfishness of the organizations

• **Complexity:**
  For all $1 > \alpha > 1/p_1$, the MOA($\alpha$) problem is strongly NP-hard if $k \geq 3$.

• **Approximate algorithm:**
  We slightly modify APPROX: the cost of each shared edge is multiplied in $G'$ by $(\alpha p_1)$. This is a $(\alpha p_1)$-approximate algorithm.
Conclusion

• A study of the incentive to make agents cooperate at the algorithmic level, for the assignment problem.

• A problem polynomial in the unweighted case; NP-hard, $p_1$-approximable (and not $(p_1+\varepsilon)$-approximable when $k>2$) otherwise. It remains hard when we relax the selfishness of the organizations.
Perspectives

• Is the MOA problem strongly NP-hard when $k=2$? [related to the exact perfect matching problem]

• When we relax the selfishness: is the MOA($\alpha$) problem inapproximable?

• Experimental results

• Fairness issues