Cooperation in Multi-Organization Matching

Fanny Pascual, LIP6

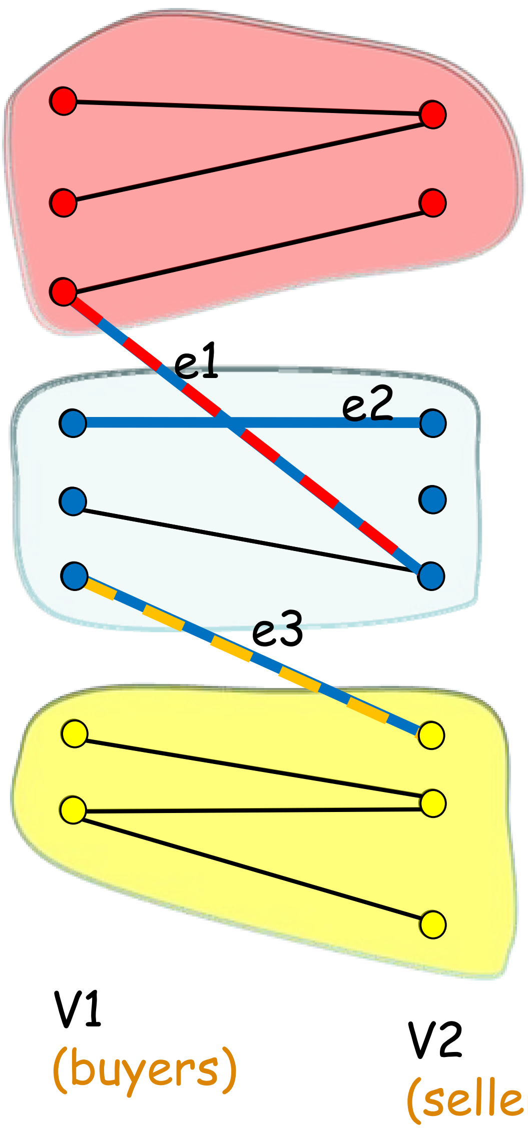
Joint work with Laurent Gourvès and Jérôme Monnot (Lamsade)

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 $G=(V1,V2,E)$ 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)

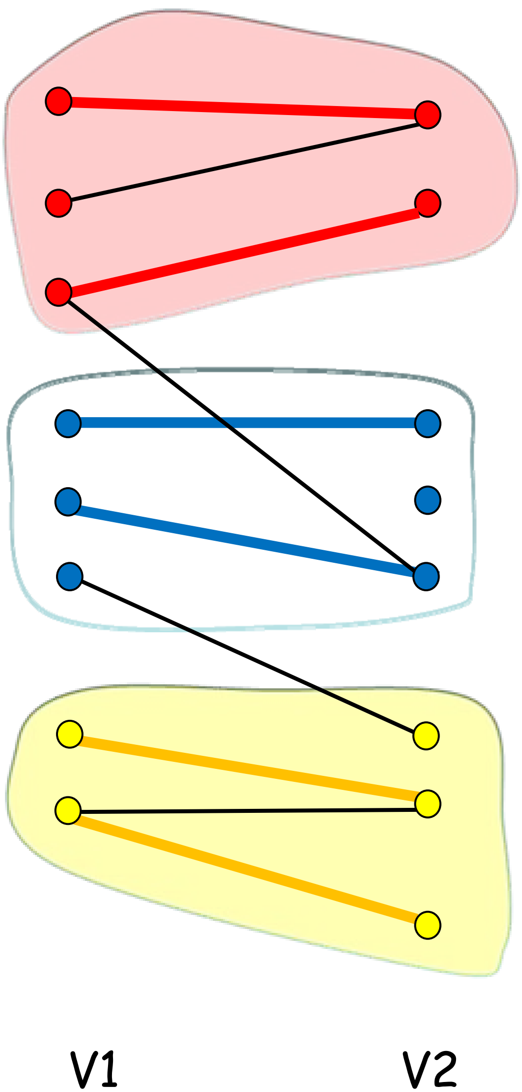
- p_1 and p_2 s.t. $0 \leq p_1 \leq p_2 \leq 1$ and $p_1 + p_2 = 1$.

Profit of O_i in a matching M:

$$\mathrm{Gain}(O_i, M) = p_1 \sum_{e \in (M \cap V1) \cap Oi} w(e) + p_2 \sum_{e \in (M \cap V2) \cap Oi} w(e)$$

Example: $\mathrm{Gain}(O_2,M) = w(e2) + p_1\, w(e3) + p_2\, w(e1)$

Presentation of the problem



Let GainAlone(Oi) be the max. weight of a matching induced by the edges of Oi.

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

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

Notation: $\text{Gain}(M)=\bigcup_{1\leq i\leq k} \text{Gain}(Oi,M)$

Presentation of the problem

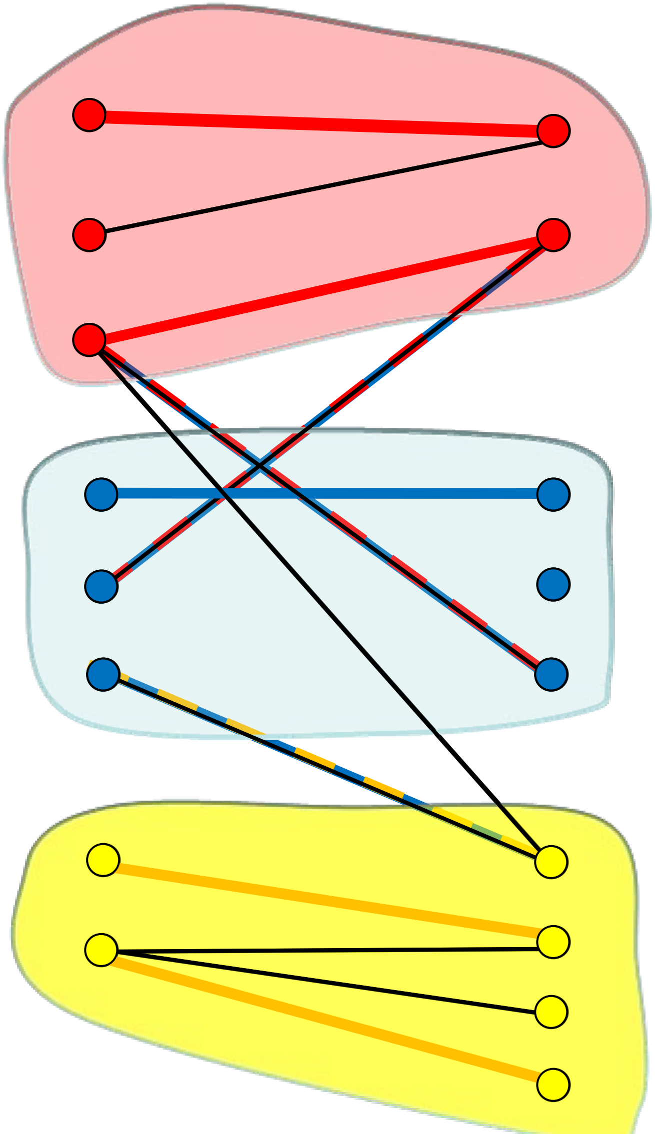
Example:

(all the weights are equal to 1, p_1=p_2=0,5)

GainAlone(O_1) = 2

GainAlone(O_2) = 1

GainAlone(O_3) = 2



Gain(O_1,M) =2

Gain(O_2,M) =2,5

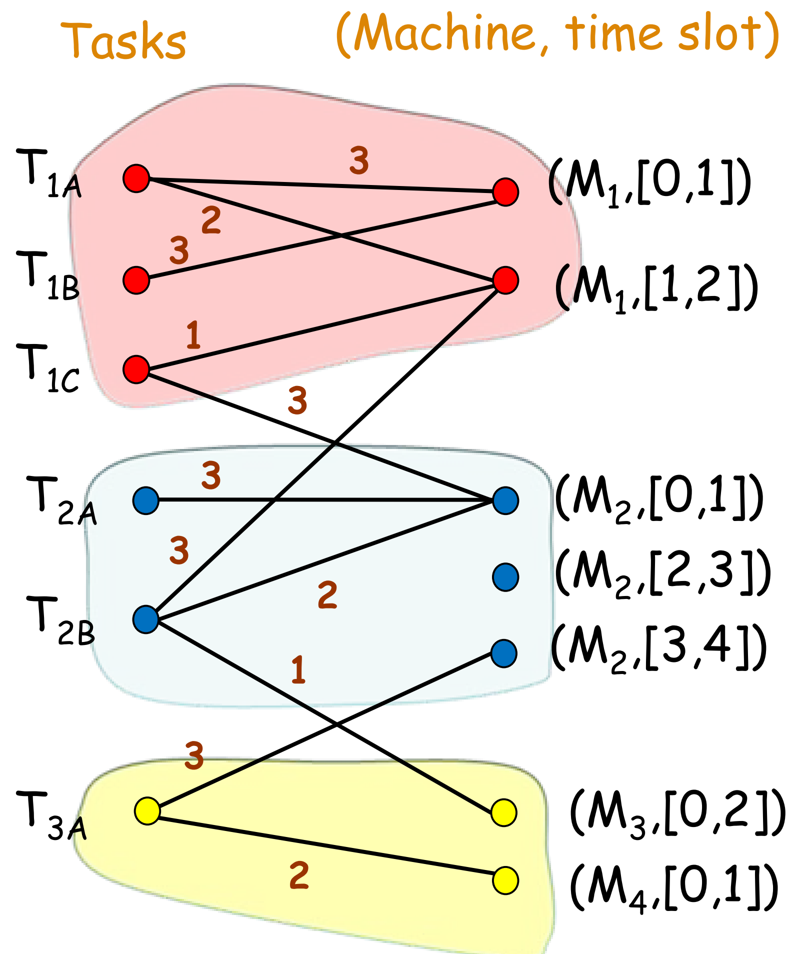
Gain(O_3,M) =2,5

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.

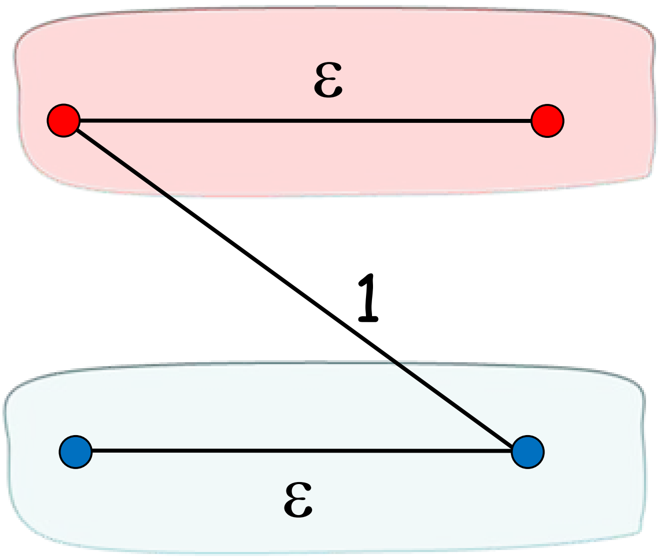
Global aim : maximize the average total satisfaction.



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

Cooperation can help a lot

Cooperation allows much better solutions.



Without cooperation :
Gain = $2\varepsilon \ll 1$

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 =

$$\max_{\text{instances}} \frac{\text{Gain in the best Nash equilibrium}}{\text{Gain in the best solution}}$$

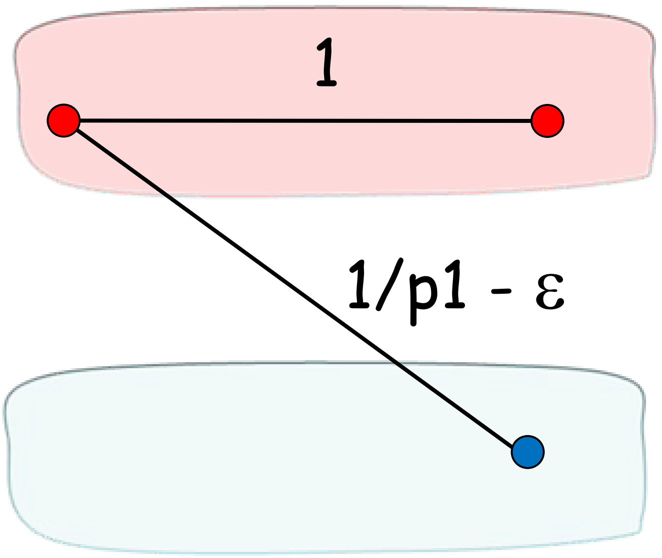
$$\max_{\text{instances}} \frac{\text{Gain(MOAopt)}}{\text{Gain}(M^*)}$$

(M^* is a max. weight matching of G)

Limits of cooperation

- Proposition : The price of stability is p1.

p1>0:



GainAlone(O_1) = 1

$$\text{Price of stability} \geq \frac{\text{Gain(MOAopt)}}{\text{Gain}(M^*)} = \frac{1}{1/p_1 - \varepsilon} = \frac{p_1}{1-(p_1\,\varepsilon)}$$

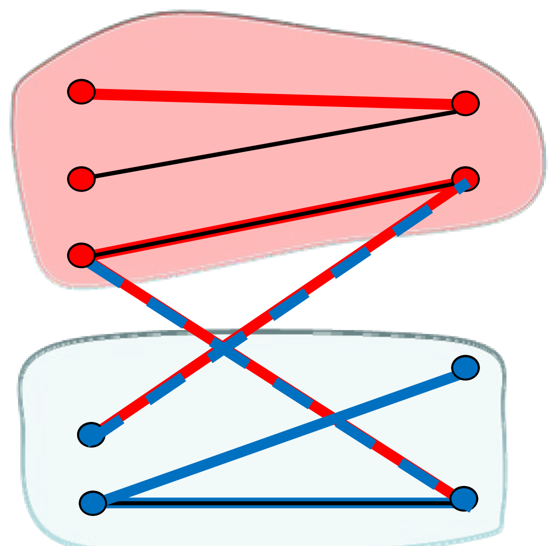
Outline

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

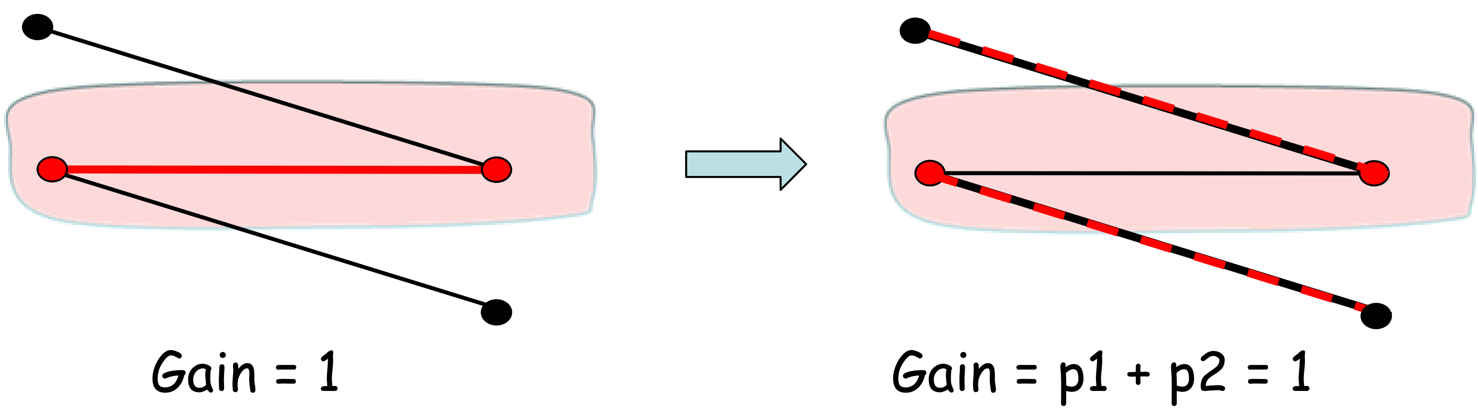
- Proposition: If the graph is unweighted, then the MOA problem is polynomial time solvable.

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 diminuate the gain of any organization.



  - The resulting matching is feasible and of max. cardinality since no more augmenting path exists.

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

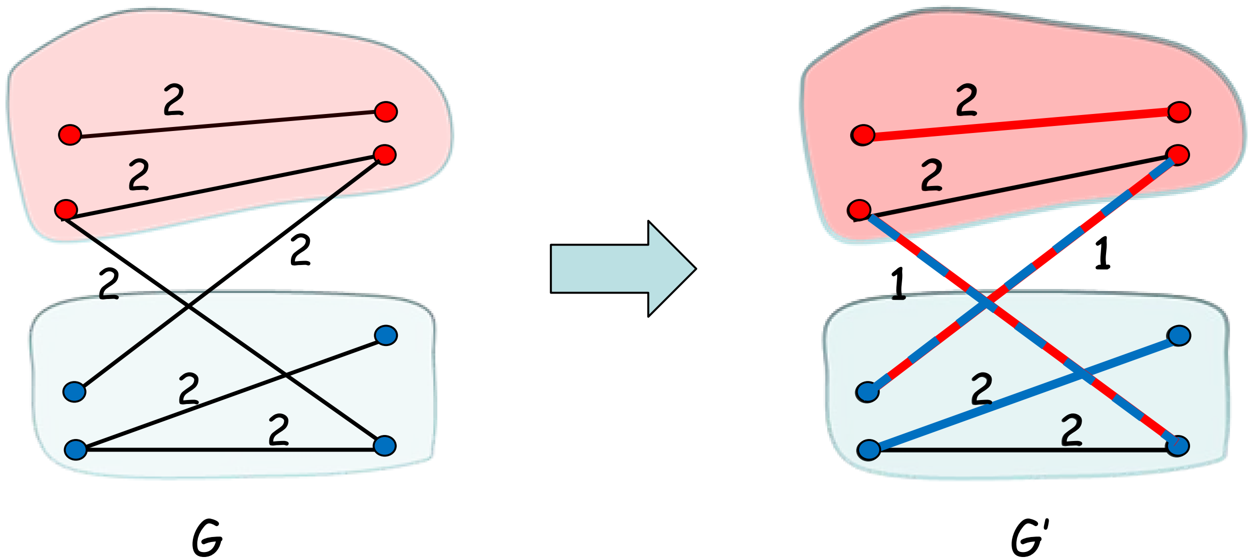
$$\max_{\text{instances}} \frac{\text{Gain in the solution returned by } A}{\text{Gain in the best solution}} \geq x$$

=> Algorithm ½-approximate : returns a matching whose gain is at least ½ 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 ≥ Gain($M^*(G')$)

In G', the weight of each edge has been at most divided by 2.

Thus Gain($M^*(G')$)≥ $\frac{1}{2}$ Gain($M^*(G)$) ≥ $\frac{1}{2}$ Gain(MOAopt)

Approximate algorithm

- M, the solution returned by APPROX is feasible:
  - Mint(i) : edges of M whose both endpoints belong to Oi
  - Mshared(i) : edges of M whose 1 endpoint belongs to Oi

M is a max. weight matching of G'
=> Gain'(Mint(i)) + Gain'(Mshared(i)) ≥ GainAlone(i)

$$\begin{aligned} \text{Gain}(Oi) &= \text{Gain}(\text{Mint}(i)) + \tfrac{1}{2}\,\text{Gain}(\text{Mshared}(i)) \\ &\geq \text{Gain}'(\text{Mint}(i)) + \text{Gain}'(\text{Mshared}(i)) \\ &\geq \text{GainAlone}(i) \end{aligned}$$

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 $\Longleftrightarrow$ 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(α) problem: find a max. weight matching s.t. Gain(Oi,M)$\geq$ GainAlone(Oi) / α .

- 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(α) problem is strongly NP-hard if $k \geq 3$.

- Approximate algorithm:
  We slighty modify APPROX: the cost of each shared edge is multiplied in G' by (α p_1).
  This is a (α 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(α) problem inapproximable?
- Experimental results
- Fairness issues