Centre for
Data Analytics

Insight

Three-Dimensional Matching Instances are Rich in Stable Matchings

Guillaume Escamocher, Barry O'Sullivan

June 27, 2018

DCU NUI Galway OÉ Gaillimh UCC University College Cork, Ireland Coláiste na hOllscoile Corcaigh UCD DUBLIN sfi

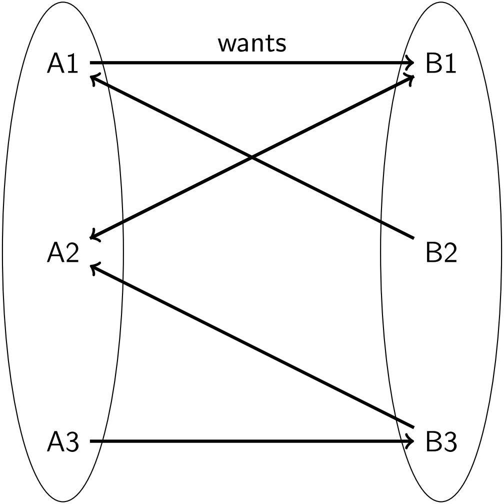
Stable Matching

- Sets of agents (individuals, organizations, programs...).
- Each agent has preferences over the other agents.

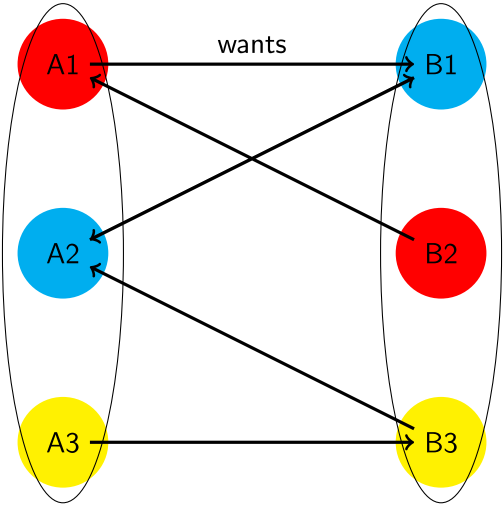
Goal

Establishing groups of agents such that there is no incentive for the agents to change the grouping.

Example



Example



Many Matching Problems

- Agents: number of sets.
- Preferences: incomplete, ties, dishonesty…
- Matching size: one-to-one, one-to-many, many-to-many…
- Stability criterion: classical, strong, exchange based…

Many Matching Problems

- Agents: number of sets.
- Preferences: incomplete, ties, dishonesty…
- Matching size: one-to-one, one-to-many, many-to-many…
- Stability criterion: classical, strong, exchange based…

Some more popular than others

Most studies look at 1 agent set (Stable Roommates) or 2 agent sets (Stable Marriage).

Three Agent Sets

Many natural applications

- Market strategy:
  1. suppliers
  2. firms
  3. buyers
- Computer networking:
  1. data sources
  2. servers
  3. end users
- Kidney exchange (reduction from stable roommates)

3 is harder than 2

Two sets

- Always a stable matching.
- Can be found in $O(n^2)$ (Gale-Shapley algorithm).

Three sets

For many matching problems with 3 agent sets:

- Not always a stable matching.
- NP-Complete to determine whether there is a stable matching.

Three-Dimensional Stable Matching

3DSM

- Three **agent sets** $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$ and $C = \{c_1, c_2, \ldots, c_n\}$.
- Agents from A **rank** agents from B, agents from B **rank** agents from C and agents from C **rank** agents from A.
- A **matching** is a set of n triples containing one agent from each agent set.
- A matching is **stable** if no triple of agents (not in the matching) would rather be together than with their matching partners.

Example

a_1 :	b_4	b_2	$\mathbf{b_3}$	b_1
a_2 :	b_3	$\mathbf{b_4}$	b_1	b_2
a_3 :	b_3	b_2	$\mathbf{b_1}$	b_4
a_4 :	b_1	$\mathbf{b_2}$	b_3	b_4

b_1 :	c_2	c_1	$\mathbf{c_4}$	c_3
b_2 :	$\mathbf{c_2}$	c_3	c_1	c_4
b_3 :	c_1	$\mathbf{c_3}$	c_2	c_4
b_4 :	c_3	$\mathbf{c_1}$	c_2	c_4

c_1 :	a_3	a_4	a_1	$\mathbf{a_2}$
c_2 :	a_2	a_1	a_3	$\mathbf{a_4}$
c_3 :	a_4	a_3	a_2	$\mathbf{a_1}$
c_4 :	a_4	a_2	$\mathbf{a_3}$	a_1

Example

a_1 :	b_4	b_2	b_3	b_1
a_2 :	b_3	b_4	b_1	b_2
a_3 :	b_3	b_2	b_1	b_4
a_4 :	b_1	b_2	b_3	b_4

b_1 :	c_2	c_1	c_4	c_3
b_2 :	c_2	c_3	c_1	c_4
b_3 :	c_1	c_3	c_2	c_4
b_4 :	c_3	c_1	c_2	c_4

c_1 :	a_3	a_4	a_1	a_2
c_2 :	a_2	a_1	a_3	a_4
c_3 :	a_4	a_3	a_2	a_1
c_4 :	a_4	a_2	a_3	a_1

Not a stable matching: $\langle a_4, b_1, c_1 \rangle$ is blocking.

Example

a_1 :	b_4	b_2	b_3	**b_1**
a_2 :	**b_3**	b_4	b_1	b_2
a_3 :	b_3	b_2	b_1	**b_4**
a_4 :	b_1	**b_2**	b_3	b_4

b_1 :	**c_2**	c_1	c_4	c_3
b_2 :	c_2	**c_3**	c_1	c_4
b_3 :	c_1	c_3	c_2	**c_4**
b_4 :	c_3	**c_1**	c_2	c_4

c_1 :	**a_3**	a_4	a_1	a_2
c_2 :	a_2	**a_1**	a_3	a_4
c_3 :	**a_4**	a_3	a_2	a_1
c_4 :	a_4	**a_2**	a_3	a_1

This matching is stable.

3DSM hardness

Difficulty of finding a stable matching

Open question.

Is there always a stable matching?

Open question. Problem 11 on Knuth's 1976 list of 12 matching problems.

Results so far

Always a stable matching for $n \leq 4$. Conjectured true for all sizes.

Thesis of the Presentation

3DSM is worth studying because it is rich in stable matchings.

How to show this? With **constraints**.

Outline of the argument

1. 3DSM instances with master preference lists have many stable matchings.
2. 3DSM instances with master preference lists are some of the most constrained 3DSM instances.

Master Preference Lists

Definition All agents within the same set share the same preferences.

Motivation

- In real life, agent preferences are rarely independent.
- Has been used to assign students to dormitories: students were ranked according to academic and socio-economic criteria.

Convention

The preferences of agents from A are $b_1 > b_2 > \cdots > b_n$. Same for the other two agent sets.

With master preference lists

- Agents are in competition with everyone else in their set.
- Cannot satisfy everyone.

$$\Downarrow$$

- Instances are very constrained.

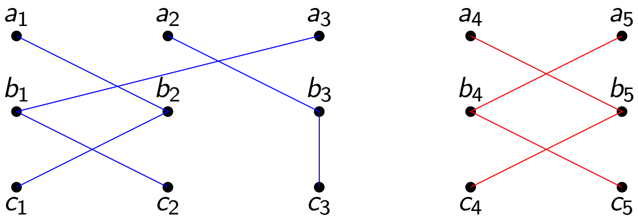
Theorem

Let f such that $f(1) = 1$, $f(2) = 4$ and
$f(n) = 2f(n-2) + 2f(n-1)$.
3DSM instances with master preference lists of size n have exactly $f(n)$ stable matchings.

What Do These Stable Matchings Look Like?

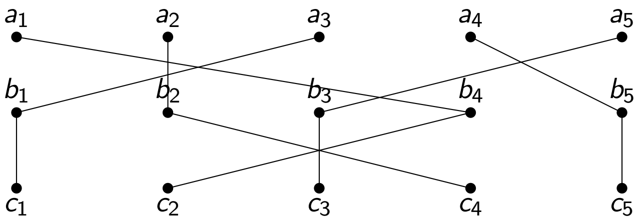
Divisible matching

Can be partitioned into sets of tuples that do not intersect.



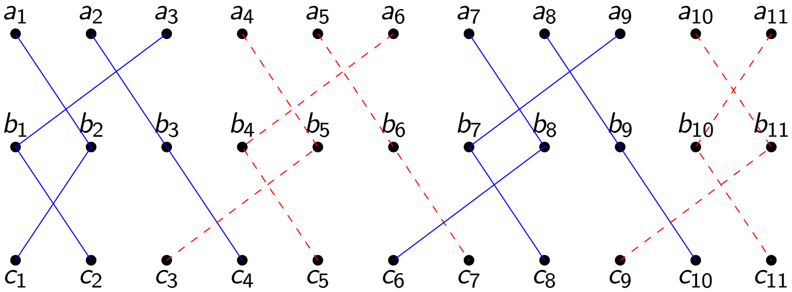
Indivisible matching

Is not a divisible matching.



Stable matchings for 3DSM with master preference lists

- Combinations of indivisible parts.
- Each part repeats the same gadget.



A Lot of Stable Matchings

$f(n) > 2f(n-1) \Rightarrow$ number of stable matchings is exponential

n	$f(n)$	g(n)
3	10	3
4	28	10
5	76	16
6	208	48
7	568	71
8	1,552	268
9	4,240	330
10	11,584	1,000

$g(n)$: Maximum number of stable matchings found for stable marriage instances of size n.

Other Matching Problems

Instances with master preference lists of any size from...

- 2DSM (stable marriage)
- 3DSM with strong stability
- Lexicographically cyclic 3DSM
- Lexicographically acyclic 3DSM

...have exactly 1 stable matching.

Quantifying Balance

Master preference lists

- For each pair of agents in the same set, one of them is always favored against the other.
- Preferences are absolute.

Perfectly split instances

- For each pair of agents in the same set, both are equally favored.
- Preferences are balanced.

Balancing the preferences How to go from one pole to the other?

ML-Step

a_1 :	b_4	b_2	b_3	b_1	b_1 :	c_2	c_1	c_4	c_3
a_2 :	b_3	b_4	b_1	b_2	b_2 :	c_2	c_3	c_1	c_4
a_3 :	b_3	b_2	b_1	b_4	b_3 :	c_1	c_3	c_2	c_4
a_4 :	b_1	b_2	b_3	b_4	b_4 :	c_3	c_1	c_2	c_4

c_1 :	a_3	a_4	a_1	a_2
c_2 :	a_2	a_1	a_3	a_4
c_3 :	a_4	a_3	a_2	a_1
c_4 :	a_4	a_2	a_3	a_1

ML-Step

a_1 :	b_4	b_2	b_3	b_1	b_1 :	c_2	c_1	c_4	c_3
a_2 :	b_3	b_4	b_1	b_2	b_2 :	c_2	c_3	c_1	c_4
a_3 :	b_3	b_2	b_1	b_4	b_3 :	c_1	c_3	c_2	c_4
a_4 :	b_1	b_2	b_3	b_4	b_4 :	c_3	c_1	c_2	c_4

c_1 :	a_3	a_4	a_1	a_2
c_2 :	a_2	a_1	a_3	a_4
c_3 :	a_4	a_1	a_2	a_3
c_4 :	a_4	a_2	a_3	a_1

Experiments

Methodology

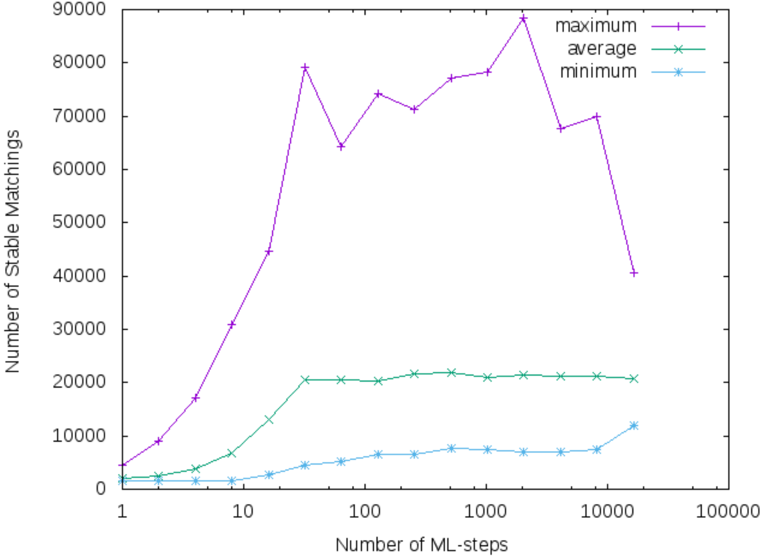
1. Start from an instance with master preference lists.
2. Add ML-steps at random.
3. Stop when the instance is perfectly split.

Size 8 instances, 100 runs.

Ideal Behavior

1. Most constrained instances, **lowest number of stable matchings**.
2. Relaxation of the constraints, increase in the number of stable matchings.
3. Least constrained instances, highest number of stable matchings.

Results for 3DSM



Actual Behavior

1. Most constrained instances, **lowest number of stable matchings**. ✓
2. Relaxation of the constraints, increase in the number of stable matchings. ✓
3. Least constrained instances, highest number of stable matchings. ×

What we have shown

1. 3DSM instances with master preference lists have an exponential number of stable matchings.
2. 3DSM instances with master preference lists are some of the 3DSM instances with the fewest stable matchings.

Conclusion

3DSM is rich in stable matchings $\Rightarrow$ it is worth studying.

The End

Thank you for your attention!

Questions are welcome!