Distributed Computing and Systems
Chalmers university of technology

Overlays with preferences: Approximation algorithms for matching with preference lists

Giorgos Georgiadis
Marina Papatriantafilou

Happier times in Iceland, when no volcanoes were erupting…

Overview

- How do nodes flirt?
- Matching with preferences
- Recent work on matchings
- Key question
- Satisfaction and how it works
- Distributed Matching using satisfaction
- Calculating the approximation
- Conclusions/Future work

How do nodes flirt*?

Nodes may strive for the best <enter metric here>

prefer “better” nodes/peers to connect to

Node i wants to chose the b_i “best” ones

They use preferences when matching

better → worse

2	5	11	32	14	28	7

Preference list

Distance, Connectivity

Bandwidth

Latency

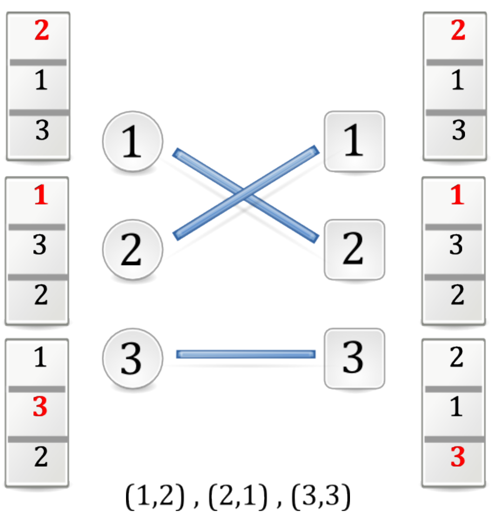
Social info, trust, etc

*Especially when they are polygamous

Matching with preferences
Nodes are tough customers

- Well studied (centralized)
- More complex than simple matching [GaleShapley62, Iwama-etal99, Manlove-etal02, Irving-etal07, …]
- Stability in focus of these studies

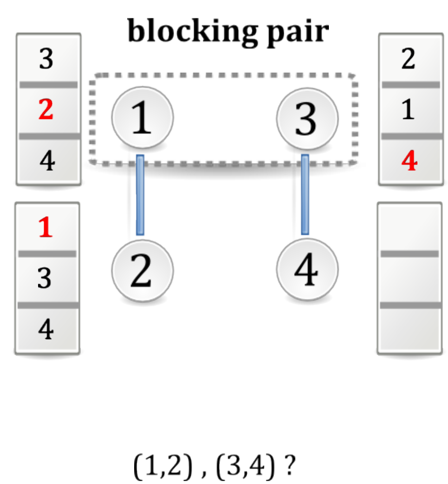
Marriages



Stable solution? Yes*

*no ties though

Roommates



Stable solution? Not always

Recent work on matchings

- [Gai-etal07, Lebedev-etal07, Mathieu08]:

  b-matching with preferences [aka stable fixtures, Irving-etal07]; stabilization in overlay construction

  1. *m-to-m matchings: proposal-refusal distributed algorithm leads to stable conf in n^2 initiatives*
  2. *acyclic preferences imply stable configurations*
  3. If stable configuration exists, *can be reached in a finite number of blocking pair resolutions*

No other guarantees

- Defined Satisfaction

max 1, subtract penalty for each "hole" in the list

$$S_i = \left(\frac{1}{b_i} - \frac{1}{b_i} \frac{R_i\left(C_1(i)\right)}{|L_i|} \right) + \ldots + \left(\frac{1}{b_i} - \frac{1}{b_i} \frac{R_i\left(C_{c(i)}(i)\right) - \left(c(i) - 1\right)}{|L_i|} \right)$$

Simulation results [Mathieu08]
Satisfaction and convergence

Problem Instance	Convergence time						Satisfaction	
	i = B (best)		*i* = R (random)		*i* = H(hybrid)			
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
• • •								
Global ordering	45.0	1.5	947.2	162.0	43.0	2.0	**0.52**	0.0
Random ordering	**N/A**						**0.77**	0.031

- Random ordered lists could not converge (!)
- Globally ordered lists converge but $\overline{S} = 0.52 < 0.77$

Key question

What is important in m-to-m matchings?

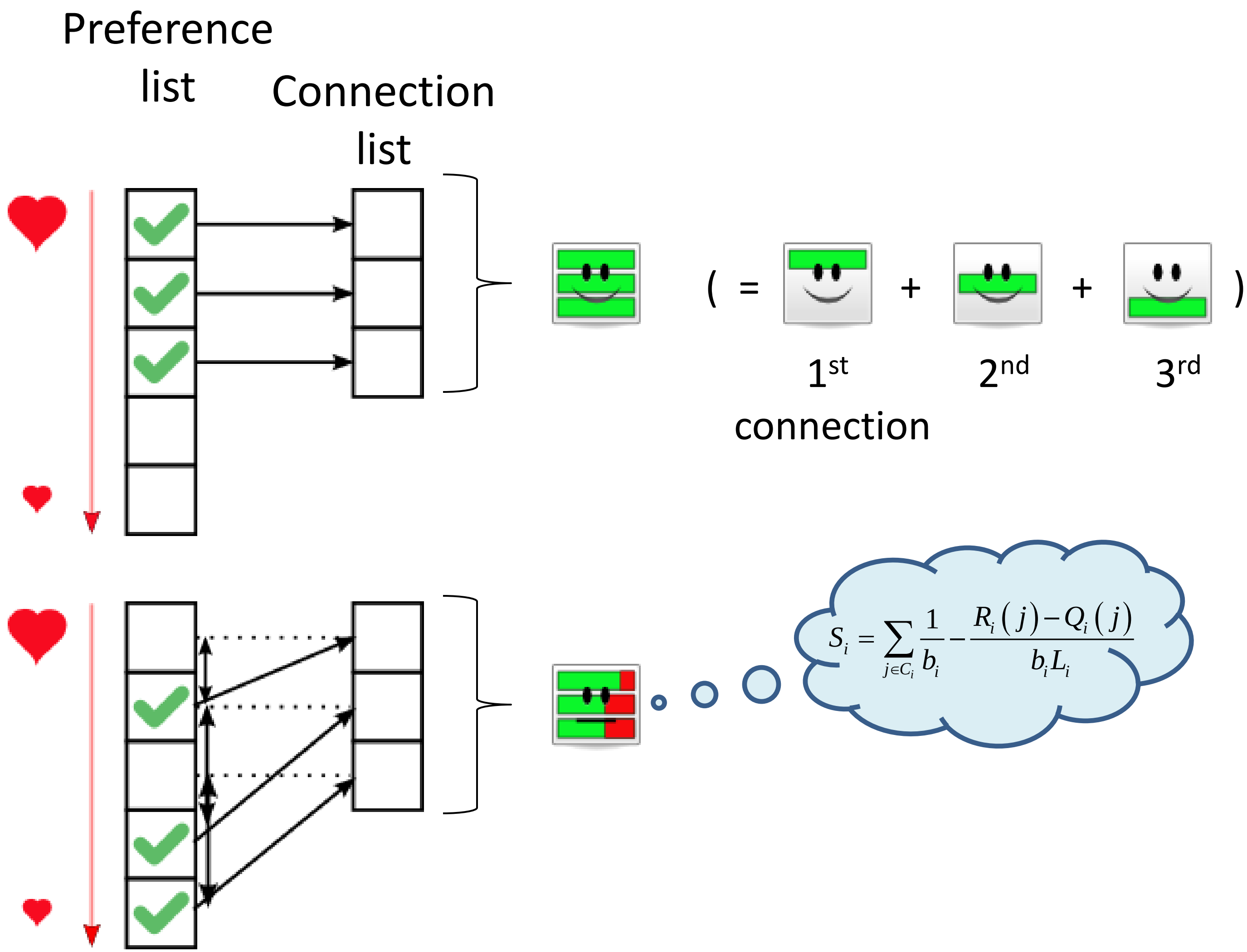
- Strict stabilization?
- Some stabilization condition?
- Something else?

What if we see it as an optimization problem?
What would that problem be?

Overview

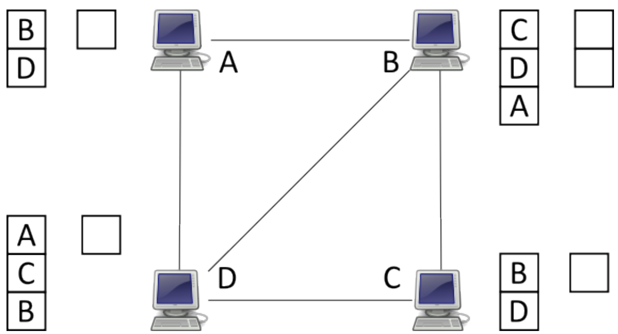
- How do nodes flirt?
- Matching with preferences
- Recent work on matchings
- Key question
- Satisfaction and how it works
- Distributed Matching using satisfaction
- Calculating the approximation
- Conclusions/Future work

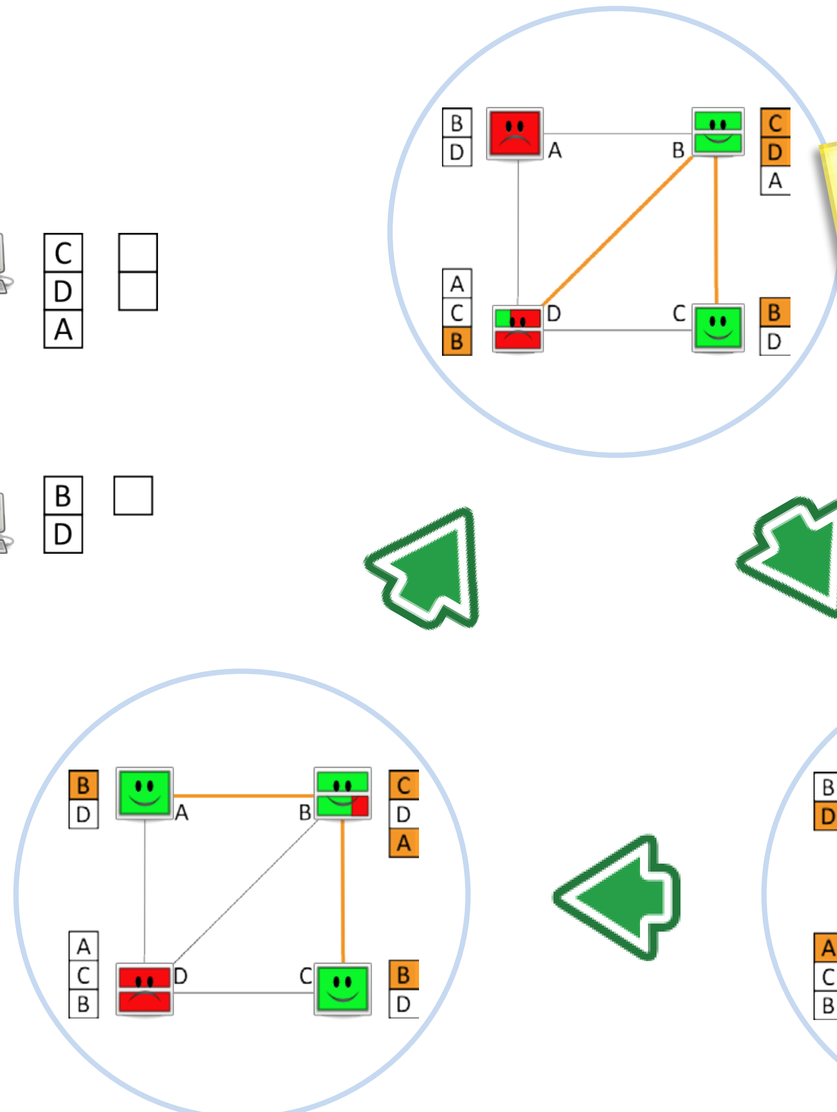
How satisfaction works



Classical stable matchings revisited

An example





Cf "The price of being near-sighted"
Kuhn, Moscibroda and Wattenhofer in SODA'06

Stable matching

\+

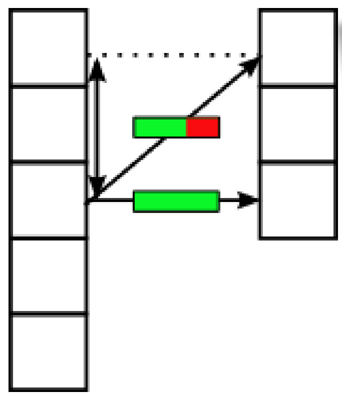
Satisfaction

=

Optimization problem

Approximating satisfaction

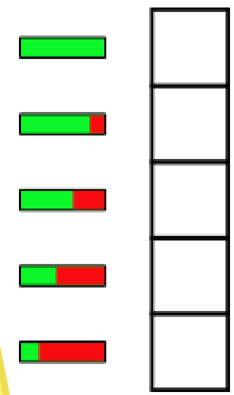
Static + Dynamic term



$$\Delta S_i^j = \frac{1}{b_i} - \frac{R_i(j) - Q_i(j)}{b_i L_i}$$

Lemma:
maximizing approx satisfaction
=
max many-to-many weighted matching
(optimization problem)

Only static term



$$\overline{\Delta S_i^j} = \frac{1}{b_i} - \frac{R_i(j)}{b_i L_i}$$

The story so far…

… and then some.

Satisfaction maximization problem

↓

Approx satisfaction maximization problem

- Satisfaction values are known locally from the beginning
- Neighbors exchange and add (approx) satisfaction values
- Weights for edges are formed

↓

Maximum m2m weighted matching

- Non-trivial to solve!

Greedy Local Distributed Matching (LID algo) using satisfaction

Greedy Distributed m2m weighted Matching

- p_i : find b_i locally heaviest edges
- Generalization of 1-1 weighted matching by [Hoepman04]
- Convergence depends on longest weight chains

Lemma: *LID algo gives ½ approximation of opt weighted many-to-many matching*

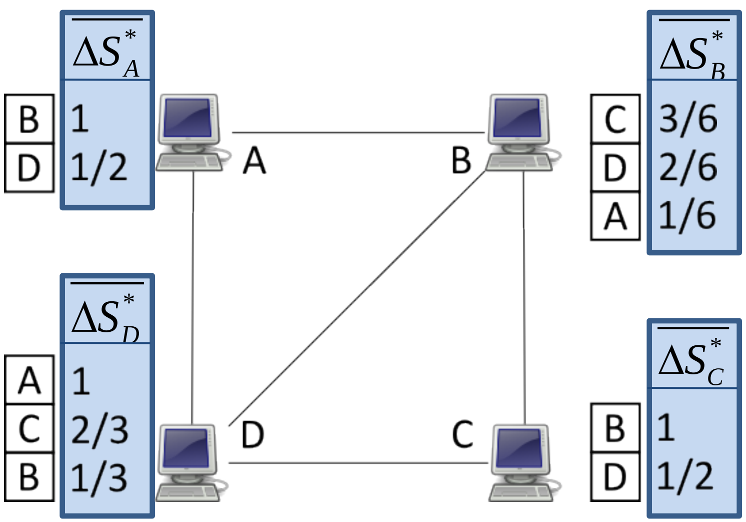
Generalization of proof in [Preis99] for centralized 1-1 matching

Thm: ... $\frac{1}{4}\left(1+\frac{1}{b_{max}}\right)$ *-approximation of optimal max satisfaction*

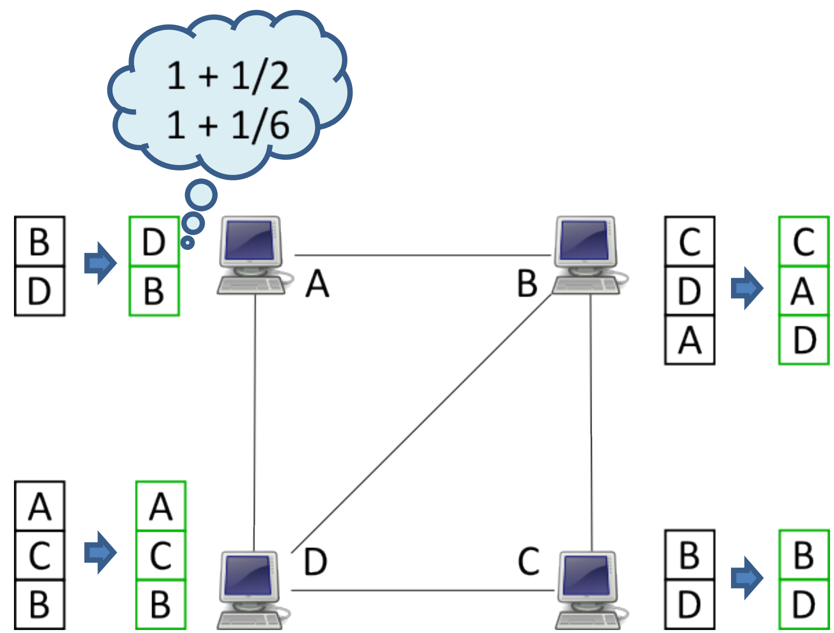
Distributed Matching using satisfaction

Initialization phase

Calculate & Send



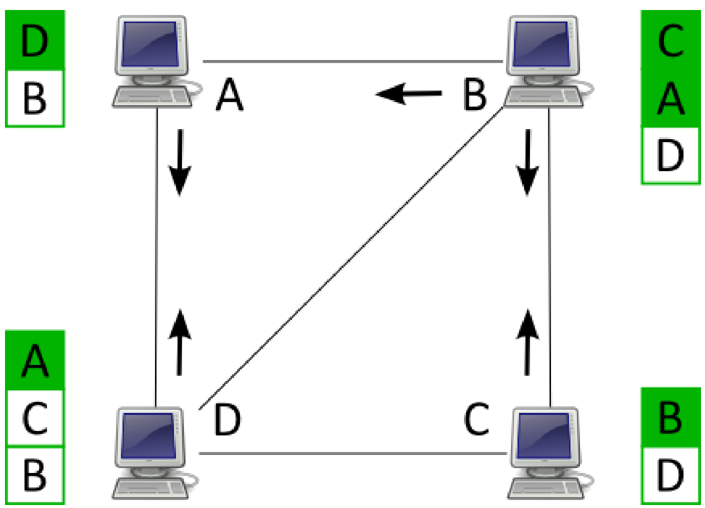
Create new list



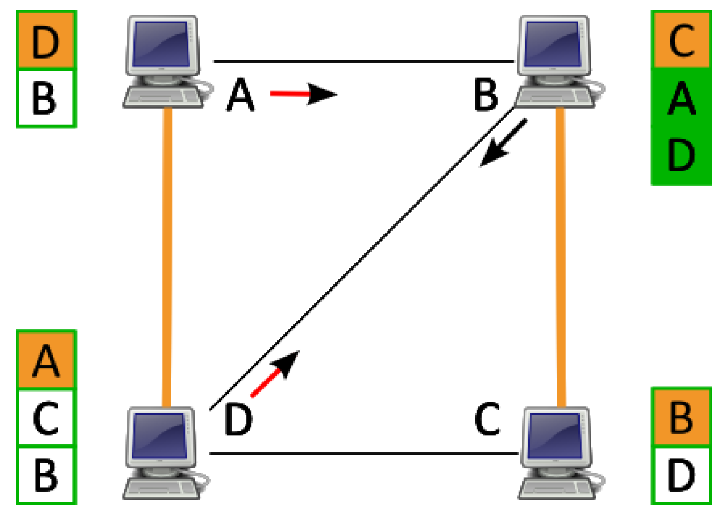
Distributed Matching using satisfaction

Matching phase

Send PROP to top b_i



Upon REJ continue



Total satisfaction (sum): 3.0

$\frac{1}{4}\left(1+\frac{1}{b_{\max}}\right)$ -approximation of optimal

Calculating the approximation

Two steps to $\frac{1}{4}\left(1+\frac{1}{b_{\max}}\right)$

① Using approx. satisfaction $\overline{\Delta S}$ instead of ΔS

$$\Rightarrow \frac{1}{2}\left(1+\frac{1}{b_{\max}}\right)$$

② Fully distributed many-to-many matching algorithm

$$\Rightarrow \frac{1}{2}$$

Calculating the approximation

① Using approx. satisfaction

- Find the proportions of S_i^{static} and $S_i^{dynamic}$ inside

$$S_i = \sum_{j \in C_i} \frac{1}{b_i} - \frac{R_i(j) - Q_i(j)}{b_i L_i}$$

> Hint: $S_i^{dynamic}$ max when b_i connections and S_i^{static} lowest when these connections are from the bottom of the list.

- Deduce: $$\frac{S_i^{static}}{S_i^{static} + S_i^{dynamic}} \geq \frac{1}{2}\left(1 + \frac{1}{b_{\max}}\right)$$

Conclusion
How to keep everybody (approx) happy*?

- Overlay construction and matching
- Seeking alternative to classical stable matchings: satisfaction
- Converted max satisfaction problem to m-to-m weighted matching
- Distributed m-to-m weighted matching algorithm (*LID*)
  - Guaranteed minimum collective satisfaction
  - Exchange of local info only (cf. also "price of being near-sighted" [Kuhn-etal06])
- Algorithm of independent interest to weighted matchings

*provided they cooperate

Future work
And now?

- Other optimization targets may be set (ie min individual satisfaction).
- Could it work to build on more sophisticated matching algos? (can get better approx.ratio/convergence?)
- Relation of convergence and churn?
- Non-collaborating actions/nodes?

Thank you for your attention!

Contact

{georgiog,ptrianta}@chalmers.se

Visit

http://www.cs.chalmers.se/~dcs/