An Equitable Solution to the Stable Marriage Problem

Ioannis Giannakopoulos\textsuperscript{1}, Panagiotis Karras\textsuperscript{2}, Dimitrios Tsoumakos\textsuperscript{3}, Katerina Doka\textsuperscript{1}, Nectarios Koziris\textsuperscript{1}

\textsuperscript{1}Computing Systems Laboratory, National Technical University of Athens, Greece
{ggian,katerina,nkoziris}@cslab.ece.ntua.gr
\textsuperscript{2}Skolkovo Institute of Science and Technology, Russia
karras@skoltech.ru
\textsuperscript{3}Department of Informatics, Ionian University, Greece
dtsouma@ionio.gr

ICTAI 2015, November 9-11
Presentation Overview

1. Introduction
2. Background
3. Equitable Stable Marriage Algorithm
4. Evaluation
5. Conclusions
Introduction

The *stable marriage problem* (SMP) is a typical matching problem.

Input:
- $n$ members of group A (men)
- $n$ members of group B (women)
- each member retains a preference list for each member of the opposite group

**Objective:** identify a *perfect* and *stable* matching between A and B.

**Perfect:** all members of A and B should be paired with exactly one member of the opposite group.

**Stable:** there exists no pair of couples $(a_k, b_l), (a_m, b_n)$ such that:
- $a_k$ prefers $b_n$ to $b_l$
- $b_n$ prefers $a_k$ to $a_m$

simultaneously.
Introduction

Solution: Gale-Shapley algorithm
- one group is assigned as the proposers, the other is the acceptors
- the proposers issue proposals to the other group
- the acceptors evaluate their proposals and marry their most preferable choice
- guarantee for a stable solution
- termination in $O(n^2)$ steps

Inequality: the proposers are more satisfied than the acceptors
- various real world use cases demand equal treatment to both sides
  - e.g. college admission problem
Introduction

Equitable Stable Marriage Problem (ESMP):
- perfect and stable matching between A and B
- members from both groups should be equally satisfied

ESMP: sex equality minimization is NP-hard

Propose: ESMA, a heuristic algorithm to identify equal and stable solutions.
Background

Gale-Shapley algorithm

Preparation step:
- the input groups are divided to *proposers* and *acceptors*

Step $i$:
- the *proposers*:
  - if married, they do nothing
  - if not married, they propose to their next most preferred choice
- the *acceptors*:
  - receive the proposals
  - engage to the proposer they prefer the most

Termination condition
- everybody is married
Background

Gale-Shapley algorithm: Preference Lists

Preference lists:
- ordered lists indicating the preference ordering of each agent for the opposite group members
- index $n$ indicating the current fiance OR the next most desirable member

Proposer

Acceptee
Background
Gale-Shapley algorithm: Monotonicity

Preference pointers always move to the same direction:

- **proposers** pointer starts from the first preference and moves downwards (less satisfied)
- **acceptor** pointer starts from a random position and moves upwards (more satisfied)

This monotonicity ensures that:

- the algorithm terminates
- the matching is stable

However:

- the proposers start from higher preference ranks and they are, eventually, favored
**Background**

**Swing**

**Swing**: an equitable algorithm
Both groups act as proposers and acceptors:
- group A proposes at even steps
- group B proposes at odd steps

Each proposer issues multiple proposals:
- from their first up to their $n^{th}$ choice

**Results:**
- **stability** is guaranteed
- **termination** is not

**Complexity:**
- $O(n^2)$ per iteration
Equitable Stable Marriage Algorithm

Introduction

Main idea: both groups act as *proposers* and *acceptors* in different steps. This way:

- both groups try to achieve the best for themselves
- the frequency of picking a group as proposers impacts the group’s final satisfaction

However:

- complex indexing needed for the preference list
- the *monotonicity* is lost
- the GS termination condition (everyone married) is not enough for stability
Equitable Stable Marriage Algorithm
Preference Lists and Members states

Two indices for each agent:
- \( n \), indicating the next most preferable choice
- \( m \), indicating the preference rank of the current fiance (if any)

![Preference Lists and Members states diagram]

Agent states:

<table>
<thead>
<tr>
<th>Status</th>
<th>single</th>
<th>motivated</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>( m = \infty )</td>
<td>( m &gt; n )</td>
<td>( m = n )</td>
</tr>
</tbody>
</table>

- **single/motivated** the agent proposes to their next preference
- **content** the agent stops

If an agent with rank \( r < n \) proposes, then \( m = n = r \) (\( n \) and \( m \) move upwards).
Equitable Stable Marriage Algorithm

Propose and evaluate functions

1: Function **EVALUATE**\{a, b\}
2: \textit{old} = M(a)
3: \textbf{if} \ a.m == \infty \textbf{or} \ pr_a[b] < a.m \textbf{then}
4: \quad \textit{old}.m = \infty
5: \quad a.m = pr_a[b]
6: \quad \textbf{if} \ a.n > pr_a[b] \textbf{then}
7: \quad \quad a.n \leftarrow pr_a[b] + 1
8: \quad \textbf{else}
9: \quad \textbf{return} \ false
10: \textbf{return} \ true

\textit{pr}_a[b]: \text{ the preference rank of agent } a \text{ for agent } b
\textit{ℓ}_a[a.n]: \text{ the agent that } a \text{ prefers } n^{th}

1: Function **PROPOSE**\{a\}
2: \textit{old} = M(a)
3: \textbf{if} \ a.n < a.m \textbf{then}
4: \quad b = \textit{ℓ}_a[a.n]
5: \quad \textbf{if} \ EVALUATE(b, a) \textbf{then}
6: \quad \quad \textit{old}.m = \infty
7: \quad \quad a.m = a.n
8: \quad \textbf{else}
9: \quad \quad a.n = a.n + 1
Equitable Stable Marriage Algorithm

Algorithm overview & stability

Require: $A, B$
Ensure: A stable matching $w$

1. $w = \emptyset$
2. $k = 0$
3. while not (everyone is content) do
4.     $k+ = 1$
5.     $P = \text{PICK\_PROPOSERS}(A, B, k)$
6.     for all $p \in P$ do
7.     PROPOSE($p$)
8.     for all $a \in A$ do
9.     $w = (a, M(a))$
10. return $w$

Stability theorem: if the Algorithm terminates, it finds a stable matching.
Equitable Stable Marriage Algorithm

Circular dependencies

When an agent receives a proposal from a more preferable choice than $n$, $m$ and $n$ are moved to previous positions of the preference list

- this creates *circularity*
- circularity may lead to infinite loops: a set of agents may re-iterate their preference lists endlessly

Observation:

- the pattern of picking the proposer group, affects the appearance of endless algorithm loops
- the pattern should be:
  - state agnostic but reproducible
  - aperiodic but fair
Equitable Stable Marriage Algorithm
Assigning Proposers

1: Function **PICK_PROPOSERS**\{A, B, k\}
2: if \(\sin(k^2) \geq 0\) then
3:    return A
4: else
5:    return B

“Fair” function between the opposite groups:
Equitable Stable Marriage Algorithm
Performance Optimization

No need to reiterate over the entire preference list:

- when \textit{content}, the agent keeps track of the proposals that he denies
- when \textit{single} or \textit{motivated}, the agent only re-proposes to agents that:
  - they have never received a proposal from him
  - they have proposed to him while content

Not a stability breach:

- absence of proposals means absence of interest (from the other agent)

The optimization is easy to implement and speeds up the algorithm execution.
Evaluation

Performance metrics

Egalitarian cost:

\[ c(M) = \sum_{(m,w) \in M} pr_m(w) + \sum_{(m,w) \in M} pr_w(m) \]

Sex equality cost:

\[ d(M) = \left| \sum_{(m,w) \in M} pr_m(w) - \sum_{(m,w) \in M} pr_w(m) \right| \]

\( pr_a[b] \): the preference rank of agent \( a \) for agent \( b \)
Evaluation

Data

Synthetics datasets following different distributions:

- **Uniform**: random assignment of scores to different agents
- **Gaussian**: default order of preference lists and adding *Gaussian noise* with different amplitude and resorting

**Discrete Regions**: divide each preference list in two regions (*Hot* and *Cold* region) and uniform distribution within a region

- 100 up to 2000 agents
- 5 variants per size
- Comparing **ESMA** to **Gale-Shapley** and **Swing**
- Each dataset that leads **Swing** to termination is tested
Evaluation

Results vs Dataset size - Uniform distribution
Evaluation

Results vs Dataset size - Gaussian distribution (noise 20%)

![Graphs showing execution time, egalitarian cost, and sex equality cost vs dataset size for SMA, Swing, and ESMA.](image)
Evaluation

Results vs Dataset size - Discrete Regions (hot region: 20%)

- Execution time (sec)
- Egalitarian cost
- Sex Equality cost
Evaluation

Results vs data polarity - Gaussian distribution

![Graph showing execution time vs data polarity for different methods.](image)

![Graph showing egalitarian cost vs data polarity for different methods.](image)

![Graph showing sex equality cost vs data polarity for different methods.](image)
Conclusions

In this paper:

- revisited the *Equitable Stable Marriage Problem*
- observed that the lack of *monotonicity* introduces problems with stability or termination
- identified that a non periodic pattern of assigning proposers solves the problem of termination
- proposed *ESMA*, that:
  - provides “fair” solutions in time equivalent to the non-equitable Gale-Shapley algorithm
  - works satisfactorily for all the tested data distributions
Thank you! Questions?