Oliver Tonnesen
V00885732

Project 2

MATH 423
October 13, 2019

1 Matchings

1.1 Matchings in graphs

A matching in a graph is a set of edges such that any two share no vertex. If every vertex in the graph is the endpoint of an edge in the matching, then the matching is called **perfect**. If the graph is bipartite, and if for every vertex in the graph, we are given an ordering of preferences for each element in the other partite set, then a **stable matching** is a perfect matching where there do not exist elements x and y, one from each partite set, such that x prefers y over the element to which it is already matched, and y also prefers x over the element to which it is already matched.

1.2 Stable matchings

The Stable Matching Problem (commonly called the Stable Marriage Problem, as it is often stated in terms of men and women selecting partners to marry) originated in part when David Gale and Lloyd Shapley wondered if they could design a self-enforcing job application process [2]. That is, given a list of job candidates with an ordering of their preferred places of work, and a list of job openings with an ordering of their most suitable applicants, can one assign applicants to jobs in such a way that no job candidate would rather work somewhere that would also rather employ them? This question eventually lead to the Gale-Shapley algorithm.

1.3 Applications

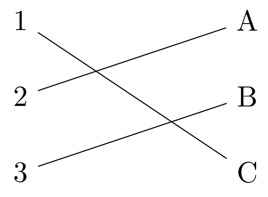
Some potential applications of the Stable Matching Problem are load balancing clients to low-latency servers, and servers to clients that cost the least to serve [1], assigning graduating medical students to hospitals, or helping transplant patients whose loved ones' organs are incompatible find suitable donors [3].

1.4 Examples

Given two sets: $\{1, 2, 3\}$ and $\{A, B, C\}$ and the following preference lists:

$$\begin{array}{ll} 1 : C, A, B & A : 1, 2, 3 \\ 2 : B, A, C & B : 3, 2, 1 \\ 3 : C, B, A & C : 1, 3, 2, \end{array}$$

a stable matching is:



2 Finding a stable matching

2.1 Gale-Shapley algorithm

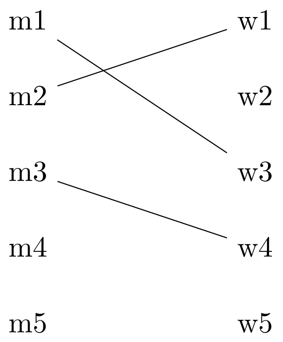
For convenience, We explain the Gale-Shapley algorithm in terms of men and women marrying. The algorithm works by allowing women to change their mind, instead of being forced to marry the first man who proposes to her despite a more optimal partner proposing later on. The algorithm operates as follows:
Repeat the following until all men and women are engaged:
(1) Find an unpaired man m.
(2) Have m propose to the woman w highest on his preference list and to whom he has not already proposed.
(3) If w prefers m to her current partner (if any) then w dumps her current partner, and m and w become engaged.

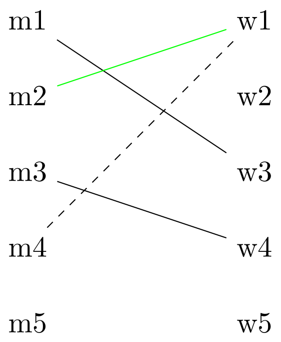
2.2 Examples

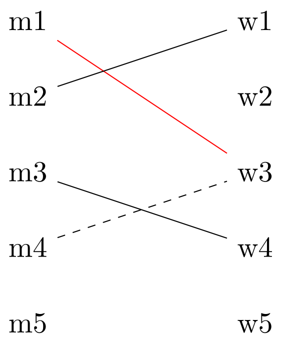
Given two sets: $\{m1, m2, m3, m4, m5\}$ and $\{w1, w2, w3, w4, w5\}$ and the following preference lists:

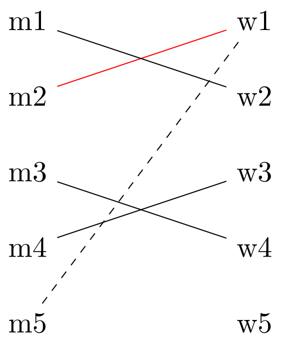
$$\begin{array}{ll}
m1 : w3, w2, w5, w1, w4 & w1 : m3, m5, m2, m1, m4 \\
m2 : w1, w2, w5, w3, w4 & w2 : m5, m2, m1, m4, m3 \\
m3 : w4, w3, w2, w1, w5 & w3 : m4, m3, m5, m1, m2 \\
m4 : w1, w3, w4, w2, w5 & w4 : m1, m2, m3, m4, m5 \\
m5 : w1, w2, w4, w5, w3 & w5 : m2, m3, m4, m1, m5
\end{array}$$

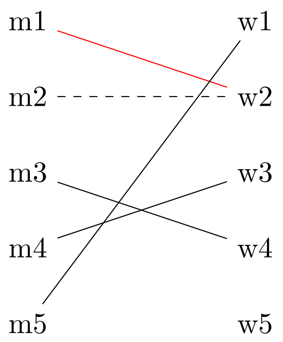
We apply the Gale-Shapley algorithm. Between any two figures below we add as many edges as we can until we reach a conflict. In such a conflicting scenario, a dashed line represents the new proposed partnership, green indicates that the current partnership remains, and red indicates that the standing partnership is broken in favour of the new one.

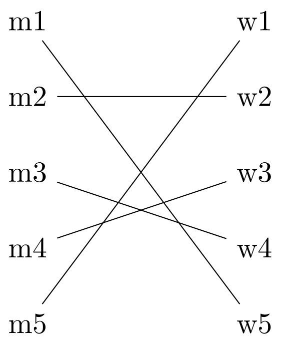












2.3 Correctness of the algorithm

We want to show that by the end of the algorithm's execution, we're left with a stable matching.

We first show that the algorithm produces a matching:

At any point during the execution of the algorithm, if m and w are matched, then m was previously unmatched, and w was either previously not matched, or she dumped her previous partner to be with m. In either case, m is matched only with w, and w is matched only with m, so what we're left with is indeed a matching.

We now show that the matching the algorithm produces is perfect:

Suppose that there exists a man m who is unmatched by the end of the algorithm's execution. Then, since there are an equal number of men and women, and since everyone is matched with at most one other person, there must exist a woman w who is also unmatched. This is not possible, since at some point, m would have proposed to w, and w would have accepted, since she was unmatched at the time. Thus the matching produced must also be perfect.

Finally, we show that the perfect matching produced by the algorithm is stable:

Assumpe that the matching the algorithm produces is not stable. Then there exists a pair (m, w) such that m and w are not married, but m prefers w over his wife, and w prefers m over her husband. Let w' and m' be m's and w's spouses, respectively. Since w' is lower on m's preference list than w, m must have proposed to w earlier in the algorithm's execution, but m' is lower on w's preference list than m, so w would have either not accepted m''s proposal, or would have dropped m' in favour of m at the time of m's proposal. Thus our assumption was false, and so no such pair (m, w) exists, and the matching is therefore stable. [2]

3 Further remarks

A particularly interesting application of Gale and Shapley's algorithm, described in [3], is helping transplant patients receive kidneys. If a transplant patient has a friend or family member who is willing to give their kidney, but for some reason is incompatible with the recipient, then the Gale-Shapley algorithm can be used to find a stable matching between kidney donors and recipients such that every transplant patient receives a kidney with which they are compatible. During the first year this program was run, it increased the rate of kidney transplants by around twenty times, and its creator, Alvin Roth, received a Nobel Prize in Economics.

References

[1] Brian Dean. Lecture notes in theoretical computer science, March 2002. `http://www-math.mit.edu/~steng/18.996/lect7.ps`.

[2] Jon Kleinberg and Éva Tardos. *Algorithm Design.* The MIT Press, 3rd edition, 2009.

[3] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119:457–488, 2004.