Stable marriage with indifference

Jagadeesh B Hooli

November 7, 2015

1 Introduction

Till now, we have assumed that in a given stable marriage problem each men and women provides strict order of preference list of opposite sex in order to get a stable partner. Gayle/shapely provided an algorithm[1] which runs in O(n^2) to provide such stable matching. However in a real world context, it is very difficult for a person to provide such strict preferences since at times, a person may find two or more persons equally favourable as a partner. We will call such a tied preference as indifference. In this article we will try to analyse whether stable matching exists for tied preference, if exists, we will find the stable matching.

2 Notions of stability

Figure 1 shows an example for tied preference list.We use '()' to show the indifference. e.g in the below figure m_2 finds tie in choosing between w_3 and w_1

$$
\begin{array}{ll}
m_1\ [\ w_2\ w_1\ w_3\ ] & w_1\ [\ m_3\ m_2\ m_1\ ] \\
m_2\ [(w_3\ w_1)\ w_2] & w_2\ [(m_1\ m_2)\ m_3] \\
m_3\ [\ w_1\ w_2\ w_3\ ] & w_3\ [\ m_2\ m_3\ m_1\ ]
\end{array}
$$

Figure 1: An instance of tied preference list

Once we allow tied preference lists then we will have three definitions of stability.We will define the three notions of stability for a given matching in the below sections.

2.1 Weak Stability

A matching will be called weakly stable unless there is a couple each of whom strictly prefers the other to his/her partner in the matching. e.g in the Figure 1 matching M=$\{(m_1, w_2)\ (m_2, w_1)\ (m_3, w_3)\}$ is weakly stable because of the pair (m_3, w_1).

2.2 Strong stability

A matching is strongly stable if there is no couple x, y such that x strictly prefers y to his/her partner and y either strictly prefers x to his/her partner or is indifferent between them. e.g in the Figure 1 matching M=$\{(m_1, w_2)\ (m_2, w_3)\ (m_3, w_1)\}$ is strongly stable because there are no blocking pairs for strong stability.

2.3 Super stability

A matching is super-stable if there is no couple each of whom either strictly prefers the other to his/her partner or is indifferent between them.e.g in the Figure 1 matching M={(m_1, w_2) (m_2, w_3) (m_3, w_1)} is also super stable.

In this whole article we will assume that numbers of men and women are equal and each men and women can have tied(indifference) preference list but no incomplete preference list. Robert W. Irving has provided three algorithms[2] which finds the matching of above three kinds of stability if exists.All the three algorithms are discussed in the below sections. If the incomplete lists are allowed then every stable marriage will have many different matchings. However Gale and Sotomayor[3] has proved that for given instance of stable marriage with incomplete list problem every stable matching will have the same size and matches exactly the same set of people.

3 Extended Gale/Shapley algorithm

This is a simple extension of classical stable matching algorithm given by Gale/Shapley [1]. Preference list of men/women are reduced as the algorithm proceeds. Algorithm is stopped once each man is atleast engaged with one woman.

First break the ties in men and women's preference list arbitrary. Each free man m will propose the first woman w on his current preference list. A man is said to be free if he is currently not engaged with any woman. A woman w will immediately accept the proposal and delete all the successors of m from her current preference list. At the same time Corresponding men will delete woman w from their list. This is referred as deletion of (m, w) pair. Proposal sequence is continued until there is no free man which means everyone is engaged. Pseudo code of Extended Gale/Shapley algorithm[2] is as follows:

Algorithm 1 -WEAK

```
assign each person to be free
while (some man m is free) do
    w ← first woman on m's list
    m proposes, and becomes engaged, to w
    if (some man x is engaged to w) then
        assign x to be free
    end if
    for each (successor m′ of m on w's list) do
        delete the pair (m′, w)
    end for
end while
output the engaged pairs, which form a stable matching
```

3.1 Proof of correctness

Since each man proposes to a woman whom he strictly prefers over other woman and each woman deletes men from her preference list whom she doesn't prefer as a partner algorithm-WEAK will always produces stable matching. However the matching is weakly stable as there is no specific criterion for breaking the ties in preference list which may introduce the couple who doesn't prefer each other.

Algorithm-WEAK will terminate when all the men and women are engaged. Suppose a man m's preference list becomes empty then at this point all the women are engaged to man whom they prefer over man m. However, this is contradiction, since no man is multiply engaged and the number of men and women is equal.

3.2 Time Complexity

Locating a person in his/her preference list can be done in constant time. So total no of operations carried out during algorithm is bounded by a constant times no of deleted pairs plus no of engaged pairs. Since the proposal involves n^2 engaged pairs, algorithm runs in O(n^2) time.

If the ties as well as incomplete preference list are allowed then the weakly stable matching can be found in linear time. However, given problem can have many stable matchings of different sizes and to find maximum cardinality weakly stable matching for a given stable marriage problem becomes NP-hard problem. This has been proved by David F. Manlove[4]

4 Super Stable Matching

Each free man m will propose to all the woman in his head of preference list simultaneously. Head of man is set of one or more woman, tied in his list, whom he strictly prefers over others. When woman w receives proposal two things can happen. First, She will delete all the successors of man m from her preference list, and corresponding men will delete from theirs. Second,If she is already engaged to another man m' and if she has tie between m' and m then she will hold the proposal of both m and m'.

Proposal sequence will be stopped if a man's preference list becomes empty, in this case no super stable matching exists. Proposal sequence will also terminate if each man is engaged to one or more woman and a multiply engaged person would be having fiance tied in his/her list. Since multiple engagements never lead to stable partners all such pairs must be deleted. Woman will delete all the men from tail of her preference list. Tail of woman is set of one or more men whom she strictly discards over others. When all such pairs are deleted proposal sequence is reactivated. The whole process is repeated until it produces one to one mapped pairs or some man's list becomes empty which means there is no super stable matching possible for the given problem. Pseudo-code[2] for finding super stable matching is as follows:

Algorithm 2 -SUPER

```
assign each person to be free;
repeat
    while (some man m is free) do
        for each woman w at the head of m's list do
            m proposes, and becomes engaged, to w
            for each (strict successor m' of m on w's list) do
                if (m' is engaged to w) then
                    break the engagement;
                end if
                delete the pair (m',w)
            end for
        end for
    end while
    for each (woman w who is multiply engaged) do
        break all engagements involving w;
        for each (man m at the tail of w's list) do
            delete the pair (m, w)
        end for
    end for
until (some man's list is empty) or (everyone is engaged);
if (everyone is engaged) then
    the engagement relation is a super-stable matching
else
    no super-stable matching exists
end if
```

4.1 Proof of correctness

We will provide proof of correctness using following lemmas.

Lemma 1. *If the pair (m, w) is deleted during an execution of Algorithm SUPER, then that pair cannot block any matching consisting of pairs that are never deleted.*

Proof. Whenever woman w deletes a man m from preference list then it actually means that there is a man m' who is strictly preferred by w over m. In other words (m, w) can never block the matching. □

Lemma 2. *A matching generated by Algorithm SUPER is super-stable.*

Proof. Let M be the super stable matching blocked by the pair (m, w) then by Lemma 1 (m, w) cannot have been deleted. If (m, w) is blocking pair then w must be in the head of the m's preference list. This means that m should have proposed w and (m, w) could be the partners in the matching. This contradicts the fact that (m, w) is blocking pair of M. □

Lemma 3. *No super-stable pair is ever deleted during an execution of Algorithm-SUPER.*

Proof. Let (m, w) be the first super stable matching deleted during execution. Let M be the super stable matching in which m and w are partners

Case 1. Suppose m is deleted because of a man m' becoming engaged with w. Then w strictly prefers m' over m . Also there is no super-stable partner for m' say w' whom he strictly prefers over w then (m',w') is deleted before (m, w). Consequently M will be blocked by (m', w)

Case 2. Suppose that (m, w) is deleted as a result of w being multiply engaged. If m' is any man who was engaged to w at that point, then m' cannot have a super-stable partner w' whom he strictly prefers to w, for the same reason as before. So once again the supposed super-stable matching M is blocked by the pair (m', w), giving a contradiction □

4.2 Time Complexity

Similar to Extended Gale/Shapley algorithm the total number of operations carried out during the execution of Algorithm-SUPER is essentially bounded by a constant times the number of pairs deleted. So algorithm runs in O(n^2) time in worst case.

For a given stable marriage problem having tied and incomplete preference lists, there exists an algorithm which computes super stable matching if exists and such matching can be also found in O(n^2) . This algorithm is explained in [4]

5 Strongly Stable Matching

Before we proceed towards strongly stable matching algorithm we will learn about perfect matching.

5.1 Perfect Matching for bipartite graph

It is one to one matching[5] between two sets of disjoint vertices say X and Y so that no two edges share a common vertex in X and Y and also matching should cover all the vertices in X and Y.In the below figure 2 as depicted perfect matching is obtained by removing the edges (m_2, w_1) (m_3, w_3)

Figure 2: An instance of perfect matching

According to Hall's theorem[6] graph G has perfect matching iff $|N(Z)||Z|$ where $Z \subseteq X$ and $N(Z)$ is neighbourhood vertices of Y which are adjacent to vertices in X.

5.2 Algorithm-STRONG

The algorithm is very similar to Super stable algorithm. The proposal sequence continues until some man's list becomes empty or everyone is engaged. If everyone are engaged and corresponding bipartite graph representing engaged pairs has perfect matching then resulting matching is strongly stable. Now let us consider a case where there is no perfect matching. Then in a graph $G = (V, E)$ where $V = X \cup Y$ there exists deficient set $Z \subseteq X$ containing r vertices which are adjacent to fewer than r vertices in Y. We define Deficiency $d(Z) = |Z| - |N(Z)|$. There exists a unique subset called critical set Z of X which is maximally deficient and contains no proper subset which is maximally deficient. In the given stable marriage problem if Z is critical set of man obtained after some execution then no woman who is engaged to Z can have stable partner. So all such engaged pairs are deleted and proposal sequence is reactivated. The whole process runs till some man's preference list becomes empty in which case no strongly stable matching exists or bipartite graph of currently engaged pairs contains perfect matching then the current matching is strongly stable. Pseudo-Code[2] for finding strongly stable matching is as follows:

Algorithm 3 -STRONG

```
assign each person to be free;
repeat
    while (some man m is free) do
        for each woman w at the head of m's list do
            m proposes, and becomes engaged, to w
            for each (strict successor m' of m on w's list) do
                if (m' is engaged to w)  then
                    break the engagement;
                end if
                delete the pair (m',w)
            end for
        end for
    end while
    if (the engagement relation does not contain a perfect matching) then
        find the critical set Z of men;
        for each (woman w who is engaged to man in Z ) do
            break all engagements involving w;
            for each (man m at the tail of w's list) do
                delete the pair (m, w)
            end for
        end for
    end if
until (some man's list is empty) or (everyone is engaged);
if (everyone is engaged)  then
    any perfect matching in the engagement relation is strongly stable
else
    no strongly stable matching exists
end if
```

5.3 Proof of correctness

We will provide proof of correctness using following lemmas. It can be observed that Lemma 1 and Lemma 2 of Algorithm-SUPER also holds good for Algorithm-STRONG.

Lemma 4. *No strongly-stable pair is ever deleted during an execution of Algorithm STRONG.*

Proof. Let (m, w) be the first strongly stable matching deleted during execution. Let M be the strongly stable matching in which m and w are partners

Case 1. Suppose m is deleted because of a man m' becoming engaged with w. Then w strictly prefers m' over m . Also there is no strongly-stable partner for m' say w' whom he strictly prefers over w then (m',w') is deleted before (m, w). Consequently M will be blocked by (m', w) .

Case 2. Suppose that (m, w) is deleted as a result of w being engaged to man in critical set Z, then m must be in tail list of w.We will call this list as current list . Let Z' be the set of man in Z who are matched in M with a woman from the head of their current list. U' be set of woman who are matched in M with man from tail of their current list. Now w is in U', So U' is non-empty.Any man m' who is engaged to w must be in Z' else (m',w) will be blocking pair. Also $Z \setminus Z'$ is also nonempty because $|Z| - |U| = d(G) > 0$ where G is current engagement graph. Now there can be edge (x, y) in engagement graph with x is in $Z \setminus Z'$ and y is in U'. This means that x strictly prefers y to his partner in M and y is indifferent between x and her partner. So strongly stable matching is blocked by the pair (x, y) which is contradiction. □

5.4 Time Complexity

During iteration of Algorithm-STRONG for every engagement relation we will search for a perfect matching in bipartite graph. Such search can be accomplished by maximum cardinality matching algorithm. So Majority of work is done in finding maximum cardinality matching in engagement graph. So from Lemma 4.6 of [2] overall time complexity of Algorithm-STRONG is $O((k + 1)n^3 + n^2(n^2 - kn)) = O(n^4)$.

Recently it has been proved that for a given stable marriage problem having tied and incomplete preference lists, there exists an algorithm which computes strongly stable matching if exists and such matching can be found in O(nm) where m is total length of all preference lists. All the relevant information of the algorithm is present in [7]

6 Exercises

1. If we allow indifference in stable roommates problem, Can we apply three algorithms discussed above for the room-mates problem? if no then what modifications has to be done for the above algorithms.

2. [8] Suppose there are n students and n universities, each of which has a capacity of 1, but student's and university's preferences may contain indifference. Is there always a stable matching How could we find one?

References

[1] D. Gale and L. Shapley, 'College Admissions and the Stability of Marriage', The American Mathematical Monthly, vol. 69, no. 1, p. 9, 1962.

[2] R. Irving, 'Stable marriage and indifference', Discrete Applied Mathematics, vol. 48, no. 3, pp. 261-272, 1994.

[3] D. Gale and M. Sotomayor. Some remarks on the stable matching problem. Discrete Applied Mathematics, 11:223232, 1985.

[4] D. F. Manlove, Stable marriage with ties and unacceptable partners, University of Glasgow, Computing Science Department Research Report, TR-1999- 29, 1999.

[5] https://en.wikipedia.org/wiki/Matching_(graph_theory)#Maximum_bipartite_matchings

[6] https://en.wikipedia.org/wiki/Hall%27s_marriage_theorem

[7] T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch, Strongly stable matchings in time O(nm) and extension to the H/R problem, Proc. STACS 2004, pp. 222 233, 2004.

[8] http://www.math.ucla.edu/~radko/circles/lib/data/Handout-995-1049.pdf