Stable marriage with indifference.

Jagadeesh B Hooli(8244731)
comp-5703

Outline

- Introduction to stable marriage with indifference.
- Notions of stability.
- Extended Gale/Shapley algorithm and an example.
- Super Stable Matching algorithm and it’s analysis with example
- Strongly Stable Matching algorithm and it’s analysis with example

Introduction to Stable marriage

- We have set of n men M={ m1,m2,m3 ,..mn } and n women W={ w1,w2,w3 ,..wn } .
- Each man ranks the women in W in strict order of preference.
- Each woman ranks the men in M in strict order of preference.
- A Matching $M \subseteq M \times W$ is a bijection between men and women.

Blocking pair

- A matching M will have blocking pair (m,w) if m prefer w over the current partner and w prefer m over her current partner in M.

m1: w1 w2	w1: m2 m1
m2: w1 w2	w2: m1 m2

- Matching M={ (m1,w1) (m2,w2) } is blocked by the pair (m2,w1).

- A matching is said to be stable if it doesn’t have blocking pairs.

- In 1962, David Gale and Lloyd Shapley[1] provided an algorithm to find at least one stable matching between equal number of men and women in O(n2) time.

Stable marriage with indifference

- Till now we have assumed that every men and women strictly provide their preference list to solve the stable matching problem.

- However in real world a person may not be able to choose between two alternatives so there by allowing preference list to have ties. This brings out a new concept called as *indifference*.

m1: (w1 w2)	w1: m2
m2: w1 w2	w2: (m1 m2)

- Also we can note that there is possibility that a person can rule out some of his/her available options and wish to gets paired with only with restricted persons i.e partial preference list. In the above example w1 only wants m2 as a partner else would prefer to be single.

Notions of stability

Once we allow men and women to have tied preference lists then any stable marriage will have three kinds of stability.

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.

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.

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.

Notions of stability(contd...)

m1: w2 w1 w3	w1: m3 m2 m1
m2: (w3 w1) w2	w2: (m1 m2) m3
m3: w1 w2 w3	w3: m2 m3 m1

- M={ (m1,w2) (m2,w1) (m3,w3) } is weakly stable because of the pair (m3, w1) .
- M={(m1, w2) (m2, w3) (m3, w1)} is Strongly stable.
- M={(m1, w2) (m2, w3) (m3, w1)} is also Super stable.

Extended Gale/Shapley algorithm[2]

- We assume that each men and women can have tied(indifference) preference list but no incomplete preference list.
- Break the ties arbitrary.
- A man is said to be free if he is currently not engaged with any woman.
- Each free man m’ will propose the first woman w’ on his current preference list.
- 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 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.

Extended Gale/Shapley algorithm[2]

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

Example

m1: (w4 w2) w1 w3 — w1: m2 (m4 m1) m3
m2: w3 w1 w4 w2 — w2: m4 m3 m1 m2
m3: w2 w3 (w4 w1) — w3: (m4 m1) m3 m2
m4: w4 w1 w3 w2 — w4: m2 m1 m4 m3

→

m1: w2 w4 w1 w3 — w1: m2 m1 m4 m3
m2: w3 w1 w4 w2 — w2: m4 m3 m1 m2 M = {(m1,w2) }
m3: w2 w3 w1 w4 — w3: m1 m4 m3 m2
m4: w4 w1 w3 w2 — w4: m2 m1 m4 m3

→

m1: w2 w4 w1 w3 — w1: m2 m1 m4 m3
m2: w3 w1 w4 w2 — w2: m4 m3 m1 m2 M= {(m1,w2) (m2,w3) }
m3: w2 w3 w1 w4 — w3: m1 m4 m3 m2

Example

m1: w2 w4 w1 w3 w1: m2 m1 m4 m3
m2: w3 w1 w4 w2 w2: m4 m3 m1 m2
m3: w2 w3 w1 w4 w3: m1 m4 m3 m2
m4: w4 w1 w3 w2 w4: m2 m1 m4 m3

Now m3 will propose w2 so (m1 w2) is deleted from matching M.
M={(m2,w3) (m3,w2)}

m1: w2 w4 w1 w3 w1: m2 m1 m4 m3
m2: w3 w1 w4 w2 w2: m4 m3 m1 m2
m3: w2 w3 w1 w4 w3: m1 m4 m3 m2
m4: w4 w1 w3 w2 w4: m2 m1 m4 m3

M={(m2,w3) (m3,w2) (m1,w4) }

Example

m1: w2 w4 w1 w3	w1: m2 m1 m4 m3
m2: w3 w1 w4 w2	w2: m4 m3 m1 m2
m3: w2 w3 w1 w4	w3: m1 m4 m3 m2
m4: w4 w1 w3 w2	w4: m2 m1 m4 m3

Now m4 will propose w1 so (m1,w2) is deleted. Since there are no more free man

M={ (m1,w4) (m2,w3) (m3,w2) (m4,w1) } is the final stable matching

Super Stable Matching [2]

- 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,
  - She will delete all the successors of man m from her preference list, and corresponding men will delete from theirs.
  - 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.

Super Stable Matching(contd..)

- Proposal sequence will also terminate if each man is engaged to one or more woman and a multiply engaged person would be having fiancé tied in his/her list.

- Since multiple engagements never lead to stable partners all such pairs must be deleted. So all such pairs are deleted and 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.

Algorithm-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
     begin
     m proposes, and becomes engaged, to w;
       for each (strict successor m' of m on w's list) do
       begin
           if (m' is engaged) to w then
              break the engagement;
           delete the pair (m'. w)
       end
   end
   for each (woman w who is multiply engaged) do
       begin
           break all engagements involving W;
           for each (man m at the tail of w's list) do
             delete the pair (m. w)
                    end;
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
```

Example

m1: w3 (w2 w1) w4 — w1: m2 m4 m3 m1

m2: (w1 w2) w3 w4 — w2: m4 (m1 m3) m2

m3: w4 w3 w1 w2 — w3: (m1 m2) m3 m4

m4: (w1 w2 w4) w3 — w4: m3 m1 m2 m4

- Each men will propose to woman at their head list. So m1 proposes to w3.Now w3 will delete all the successors of m1 from her list. Corresponding men will delete w3 from their list However w3 will not delete m2 since w3 finds tie between m2 and m1.

m1: w3 (w2 w1) w4 — w1: m2 m4 m3 m1

m2: (w1 w2) w3 w4 — w2: m4 (m1 m3) m2

m3: w4 w3 w1 w2 — w3: (m1 m2) m3 m4

m4: (w1 w2 w4) w3 — w4: m2 m1 m2 m4

M={(m1,w3)}

Example

- Since m2 finds tie between w1 & w2 ,he will propose to both simultaneously. Now w1 w2 will accept the proposal. Start delete process.

m1: w3 (w2 w1) w4	w1: m2 m4 m3 m1
m2:(w1 w2) w3 w4	w2: m4 (m1 m3) m2
m3: w4 w3 w1 w2	w3: (m1 m2) m3 m4
m4:(w1 w2 w4) w3	w4: m3 m1 m2 m4

M={(m1,w3) (m2,w1) (m2,w2) }

- Now m3 proposes to w4. Start delete process

m1: w3 (w2 w1) w4	w1: m2 m4 m3 m1
m2:(w1 w2) w3 w4	w2: m4 (m1 m3) m2
m3: w4 w3 w1 w2	w3: (m1 m2) m3 m4
m4:(w1 w2 w4) w3	w4: m3 m1 m2 m4

M={(m1,w3) (m2,w1) (m2,w2) (m3,w4) }

Example

- m4 proposes to w2. However w2 is already engaged with the m2. So she will break her engagement with m2 and get paired with m4

m1: w3 (w2 w1) w4	w1: m2 m4 m3 m1
m2:(w1 w2) w3 w4	w2: m4 (m1 m3) m2
m3: w4 w3 w1 w2	w3: (m1 m2) m3 m4
m4:(w1 w2 w4) w3	w4: m3 m1 m2 m4

- Now no woman is multiply engaged and all the free men are engaged. There are no multiple engagements in M . So M={(m1,w3) (m2,w1) (m3,w4) (m4,w2) } is the super stable matching

Proof of Correctness

- **Lemma 1.1: 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.**
- 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 1.2: A matching generated by Algorithm SUPER is super-stable.**
- Let M be the super stable matching blocked by the pair (m,w) then by lemma 1.1 (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.

Proof of correctness

- **Lemma 1.3: No super-stable pair is ever deleted during an execution of Algorithm SUPER.**

- 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.

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 n2 engaged pairs algorithm runs in O(n2) time

Strongly Stable matching

Perfect Matching for bipartite graph[3]: It is one to one matching 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.

According to Hall's theorem[4] graph G has perfect matching iff $|N(Z)| \geq |Z|$ where $Z \subset X$ and $N(Z)$ is neighbourhood vertices of Y which are adjacent to vertices in X.

Strongly Stable matching [2]- Algo-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. Deficiency $d(Z)=|Z|-|N(Z)|$

- There exists a unique subset Z of X which is maximally deficient and contains no proper subset which is maximally deficient. Z is called critical set.

Strongly Stable matching[2]- Algo-STRONG

- In our 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.

Algorithm-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
     begin
     m proposes, and becomes engaged, to w;
       for each (strict successor m' of m on w's list) do
       begin
           if (m' is engaged) to w then
              break the engagement;
           delete the pair (m'. w)
       end
   end
if (the engagement relation does not contain a perfect matching) then
begin
   find the critical set Z of men;
   for each (woman w who is engaged to a man in Z) do
   begin
      break all engagements involving w;
        for each man m at the tail of w's list do
        delete the pair (m, w)
   end;
end;
until (some man's list is empty) or (everyone is engaged);
```

Example

m1: (w1 w2)   w1: (m1 m2)
m2: (w1 w2)   w2: m2 m1

m1: (w1 w2)   w1: (m1 m2)
m2: (w1 w2)   w2: m2 m1

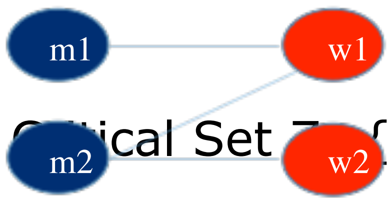
M={(m1 w1) (m1 w2) }

- Now m2 proposes both m1 and m2. So the engagement (m1 w2) is deleted

m1: (w1 w2)   w1: (m1 m2)
m2: (w1 w2)   w2: m2 m1

M={(m1 w1) (m2 w1) (m2 w2) }

Example



Critical Set Z = {m2}

- Since w2 and w1 both are engaged with m2 all the engagement involving w2 and w1 are broken. The tail of w1 contains m1 and m2 so both are deleted and correspondingly, m1 and m2 will delete w1 from their list. The pair (m2 w2) is also deleted as w2 has no option other than deleting m2 .

m1: (w1 w2)	w1: (m1 m2)
m2: (w1 w2)	w2: m2 m1

- Now we can observe that both m1 and m2 has empty preference so the given problem doesn’t have strongly stable matching.

Proof of Correctness

- **Lemma 2.1: If the pair (m, w) is deleted during an execution of Algorithm STRONG, then that pair cannot block any matching consisting of pairs that are never deleted.**
- 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 2.2: A matching generated by Algorithm STRONG is Strongly-stable.**
- Let M be the strongly stable matching blocked by the pair (m,w) then by lemma 2.1 (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.

Proof of correctness

- **Lemma 2.3: No strongly-stable pair is ever deleted during an execution of Algorithm STRONG.**
- 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

Proof of correctness

- 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\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\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.

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 this will lead to equation ***O((k+1)n3 +n2(n2-kn))***

  =***O(n4)***. Where k is total no of iterations carried out.

Applications

- Used in college admissions problem and Canadian Resident Matching Service (CaRMS) [5].

- Used in matching doctors to hospitals.

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

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

[5]. http://www.carms.ca/en/residency/match-algorithm/

Thank You!