Stable Matching

CMSC 27230: Honors Theory of Algorithms

January 4, 2023

Corresponding section(s) of Kleinberg-Tardos: 1.1

1 Stable Matching

Problem 1.1 (Stable Matching)**.** *In the stable matching problem, we have the following data:*

1. *There are two sets of n people, $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$.*
2. *Each person $a_i \in A$ has a preference order on the people $b_1, \ldots, b_n \in B$.*
3. *Each person $b_j \in B$ has a preference order on the people $a_1, \ldots, a_n \in A$.*

We are then asked to find a stable matching M between A and B, which is a matching M such that there does not exist a pair of people a_i and b_j who would both rather be with each other than with their partners in M.

More precisely, we want to find a bijective map $\pi : [n] \to [n]$ such that there do not exist $i, j \in [n]$ such that:

1. $\pi(i) \neq j$
2. *a_i prefers b_j to $b_{\pi(i)}$*
3. *b_j prefers a_i to $a_{\pi^{-1}(j)}$*

We then take $M = \{(a_i, b_{\pi(i)}) : i \in [n]\}$

Algorithm 1.2 (Gale-Shapley Algorithm for Stable Matching)**.** *In the Gale-Shapley Algorithm, the people in group A go down their preference lists and ask the people in group B if they would like to be partners. When a person b_j in group B receives such an offer from a person a_i in A, b_j accepts the offer if he/she currently has no partner or has a partner $a_{i'}$ but prefers a_i to $a_{i'}$ (in which case $a_{i'}$ becomes unmatched and must start making offers again starting from the next person down in his/her preference list). Otherwise, b_j turns the offer down.*

More precisely, the Gale-Shapley algorithm is as follows

Stored data: In the Gale-Shapley algorithm, we keep track of a map $\pi : [n] \to [n] \cup \{0\}$ which describes the current partnerships between A and B. If $\pi(i) = j$ for some $j \in [n]$ then a_i is currently partners with b_j and if $\pi(i) = 0$ then i currently has no partner. Thus, π corresponds to the partial matching $M = \{(a_i, b_{\pi(i)}) : i \in [n], \pi(i) \neq 0\}$

Inititialization: We start with $\pi(i) = 0$ for all $i \in [n]$. In other words, we start with no partnerships between A and B.

Iterative step: At each step, we choose an i such that $\pi(i) = 0$ (i.e. a_i is unmatched) and have a_i ask person b_j if he/she would like to be partners where b_j is a_i's most preferred person that a_i has not yet asked. b_j then responds as follows

1. *If $\nexists i' \in [n]\,(\pi(i') = j)$ (i.e. b_j is currently unmatched) then we set $\pi(i) = j$, making a_i and b_j partners.*
2. *If $\exists i' \in [n]\,(\pi(i') = j)$ (i.e. b_j is currently matched with $a_{i'}$) and b_j prefers a_i to $a_{i'}$ then we set $\pi(i) = j$ and set $\pi(i') = 0$. This breaks the partnership between $a_{i'}$ and b_j and makes a_i and b_j partners.*
3. *If $\exists i' \in [n]\,(\pi(i') = j)$ (i.e. b_j is currently matched with $a_{i'}$) and b_j prefers $a_{i'}$ to a_i then b_j turns down a_i's offer and we keep π as is.*

Once everyone in group A is matched or has nobody left to make an offer to, the algorithm terminates.

Theorem 1.3. *The Gale-Shapley algorithm always terminates in $O(n^2)$ steps and results in a stable matching.*

Proof. The intuition behind the Gale-Shapley algorithm is as follows. For the people in group B, their partner only gets better with time as they only accept a new partnership if it is better than their current partnership (if any). Thus, once a person a_i in group A is turned down by a person b_j in group B, either because a_i's initial offer was turned down or because b_j left a_i to partner with another person $a_{i'}$, this rejection is final; b_j will never regret turning down a_i.

This means that when the algorithm finishes, all of the people in A are as happy as they can reasonably be. There might be people in group B that they'd rather be with, but those people are unattainable. This makes the resulting matching stable.

We now turn this intuition into a proof. For this, the following definition is useful (this differs slightly from the presentation in Kleinberg-Tardos):

Definition 1.4. *We say that b_j is available to a_i if any of the following cases hold:*

1. *$\pi(i) = j$ (b_j is already matched to a_i)*
2. *$\nexists i'(\pi(i') = j)$ (b_j is unmatched)*
3. *$\exists i'(\pi(i') = j)$ but b_j prefers a_i to $a_{i'}$ (b_j is currently matched but would switch to partner with a_i if asked)*

Otherwise, we say that b_j is unavailable to a_i.

When analyzing algorithms, it is often useful to find invariants which remain true as the algorithm progresses. Here we have the following invariants:

Lemma 1.5. *As the algorithm progresses,*

1. *Every person $b_j \in B$ only gets happier. In other words, once a person $b_j \in B$ has a partner they will never be unmatched again and whenever b_j changes partners, b_j is happier with his/her new partner.*

2. *For all $i, j \in [n]$, if b_j becomes unavailable to a_i then b_j never becomes available to a_i again.*

Proof. For the first statement, observe that once b_j has a partner a_i, the only way this partnership will break is if b_j accepts an offer from another person $a_{i'} \in A$ in which case b_j remains matched and is happier with his/her new partner.

To see the second statement, assume that b_j becomes unavailable to a_i and then later becomes available to a_i. Let $a_{i'}$ be b_j's partner when b_j becomes unavailable to a_i and let $a_{i''}$ be b_j's partner when b_j becomes available again. We must have that b_j prefers $a_{i'}$ to a_i and b_j prefers a_i to $a_{i''}$ which implies that b_j prefers $a_{i'}$ to $a_{i''}$. However, by the first statement, this is impossible as b_j only gets happier as time goes on. $\square$

Remark 1.6. *This is a common strategy for proving that something can never happen. We assume that it does happen and then derive a contradiction.*

With these invariants in hand, we now prove that the Gale-Shapley algorithm always terminates in $O(n^2)$ steps and when it does, it results in a stable matching. To see that the Gale-Shapley algorithm must terminate in $O(n^2)$ steps, note that in each step, a person from group A makes an offer to a person from group B which they have not yet made an offer to. This can happen at most n^2 times, so there can be at most n^2 iterations.

Remark 1.7. *Here the number of proposals made by people in group A is a progress measure which allows us to see that the algorithm is making progress and will eventually terminate.*

To see that the Gale-Shapley algorithm results in a matching, note that the only way the Gale-Shapley algorithm can fail to give a matching is if some person a_i in group A is rejected by every person in group B. However, in order for this to happen, every person in group B must be partnered with someone who they prefer to a_i. However, this is impossible because there are n people in group B and there are only $n-1$ other people in group A.

Finally, to see that the Gale-Shapley algorithm results in a stable matching, assume that the resulting map/matching is not stable, i.e. there exist $i, j \in [n]$ such that

1. $\pi(i) \neq j$

2. a_i prefers b_j to $b_{\pi(i)}$

3. b_j prefers a_i to $a_{\pi^{-1}(j)}$

If so, then on the one hand b_j is available to a_i because b_j prefers a_i to $a_{\pi^{-1}(j)}$. However, on the other hand, since a_i prefers b_j to $b_{\pi(i)}$, a_i would have first made an offer to b_j before making an offer to $b_{\pi(i)}$. Since a_i is not with b_j, a_i must have previously made an offer to b_j and been turned down, either initially or because b_j received a better offer. At this point, b_j was unavailable to a_i so by Lemma 1.5, b_j must still be unavailable to a_i, which is a contradiction. $\square$

In fact, we can say more about the Gale-Shapley algorithm. When we gave intuition for the algorithm, we claimed that the people in group A will be as happy as they can reasonably be. This statement can be made precise and proved.

Theorem 1.8. *No matter which order we have the people in group A make their offers, the resulting map/matching π/M will be the same. Moreover, this stable matching is the best possible stable matching for the people in group A and the worst possible stable matching for the people in group B. More precisely, for any other stable map/matching π'/M',*

1. *For all $i \in [n]$, either $\pi'(i) = \pi(i)$ or a_i prefers $b_{\pi(i)}$ to $b_{\pi'(i)}$*
2. *For all $j \in [n]$, either ${\pi'}^{-1}(j) = \pi^{-1}(j)$ or b_j prefers $a_{\pi'^{-1}(j)}$ to $a_{\pi^{-1}(j)}$*

Proof.

Definition 1.9. *For each $i \in [n]$, define S_i to be the set of possible partners for a_i in a stable matching, i.e.*

$$S_i = \{j : \text{ There exists a stable map/matching } \pi'/M' \text{ such that } \pi'(i) = j\}$$

Lemma 1.10. *Whenever we run the Gale-Shapley algorithm, there is never an $i, j \in [n]$ such that*

1. *$j \in S_i$*
2. *b_j is unavailable to a_i.*

Proof. Assume that this does happen. If so, consider the first time where there is an $i, j \in [n]$ such that $j \in S_i$ but b_j is unavailable to a_i. At this point, b_j must have just accepted an offer from $a_{i'}$ for some $i' \in [n]$.

Now note that since $j \in S_i$, there must be some stable map/matching π/M such that $\pi(i) = j$. Let $j' = \pi(i')$. We now consider whether $a_{i'}$ prefers b_j or $b_{j'}$. On the one hand, if $a_{i'}$ prefers b_j to $b_{j'}$ then since b_j prefers $a_{i'}$ to a_i, π/M is not a stable matching, which is a contradiction. On the other hand, if $a_{i'}$ prefers $b_{j'}$ to b_j then before making an offer to b_j, $a_{i'}$ must have made an offer to $b_{j'}$ and been turned down or turned away. Since $j' \in S_{i'}$, this is an earlier case where $i', j' \in [n]$, $j' \in S_{i'}$, and $b_{j'}$ is unavailable to $a_{i'}$, which contradicts the definition of i and j. $\square$

Corollary 1.11. *If π/M is a stable map/matching resulting from the Gale-Shapley algorithm then for all $i \in [n]$, if $\pi(i) = j$ then*

1. *$j \in S_i$*
2. *For all $j' \in S_i$ such that $j' \neq j$, a_i prefers b_j to $b_{j'}$.*

Proof. The first statement follows immediately from the fact that the Gale-Shapley algorithm gives a stable matching. For the second statement, note that by the lemma we just proved, whenever we run the Gale-Shapley algorithm, for all $j' \in S_i$, $b_{j'}$ is always available to a_i. Thus, a_i never needs to make an offer to anyone below their top choice in S_i. $\square$

We can now prove Theorem 1.8. Let π/M is a stable map/matching resulting from the Gale-Shapley algorithm and let π'/M' be a different stable matching. Corollary 1.11 implies that M must be the matching $\{(i, a_i\text{'s most preferred partner in } S_i) : i \in [n]\}$. We just need to show that whenever ${\pi'}^{-1}(j) \neq \pi^{-1}(j)$, b_j prefers $a_{\pi'^{-1}(j)}$ to $a_{\pi^{-1}(j)}$. To see this, observe that $a_{\pi^{-1}(j)}$ must prefer b_j to $b_{\pi'\pi^{-1}(j)}$ because both j and $\pi'\pi^{-1}(j)$ are in $S_{\pi^{-1}(j)}$ and $a_{\pi^{-1}(j)}$ gets his/her most preferred partner in $S_{\pi^{-1}(j)}$ in the map/matching π/M. Thus, b_j must prefer $a_{\pi'^{-1}(j)}$ to $a_{\pi^{-1}(j)}$ because otherwise both $a_{\pi^{-1}(j)}$ and b_j would prefer each other to their current partners in M' so M' would not be stable. $\square$