CS 295 Market Design **Fall 2019**

Lecture 1: Stable Matchings

Lecturer: Vijay Vazirani *Scribe: Karthik Gajulapalli, James Liu*

1.1 Introduction

Stable Matching was introduced by Gale and Shapley in 1962 and has found many applications, such as matching students to colleges and matching resident interns to hospitals. We analyze the Deferred-Acceptance Algorithm proposed by Gale and Shapley for finding a stable-matching.

1.2 Preliminaries

We consider the complete bipartite graph (W, F) over n firms who each have one job opening and n workers who are each looking to be employed. We define a preference list of a worker to be a total ordering over all firms. If w prefers firm f to firm f', we represent this preference using $f \succ_w f'$. Likewise, we define a preference list of a firm to be a total ordering over all workers. We assume that preference lists are given in decreasing order of preference.

Definition 1.1. *A* ***perfect matching*** *on* (W, F) *is a bijection of workers to firms.*

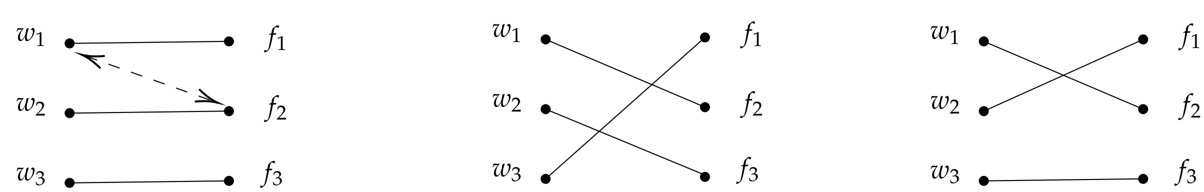
Definition 1.2. *A worker and firm form a* **blocking pair** *w.r.t. a perfect matching* μ, *if they like each other more than they like their partners in* μ.

If we have a perfect matching which allows a blocking pair, then the blocking pair is incentivized to break away from the matching. This leads us to the following notion:

Definition 1.3. ***Stable Matching****:*
A perfect matching on (W, F)*, with no blocking pairs.*

Example 1.4. *Let us look at an instance on 3 workers and firms, under the following preference lists:*

$w_1 : f_2, f_1, f_3 \qquad f_1 : w_1, w_2, w_3$
$w_2 : f_2, f_3, f_1 \qquad f_2 : w_1, w_2, w_3$
$w_3 : f_1, f_2, f_3 \qquad f_3 : w_1, w_3, w_2$



(a) Perfect Matching (unstable) (b) Stable Matching 1 (c) Stable Matching 2

Figure 1.1: Some Perfect Matchings for (W, F)

1.3 Stable Matching Algorithm

In the beginning, all firms are *uncrossed* on all the workers' preference lists. In each iteration, each worker proposes to the first uncrossed firm on its list. Each firm says "maybe" to the best proposal it receives and rejects all other proposals. If a worker got rejected from a firm it crosses that firm off its list. If each firm receives a proposal we have a perfect matching and the algorithm terminates.

DEFERRED-ACCEPTANCE ALGORITHM(DA):
Input: (W, F), preferences of each worker w and each firm f'
Output: Stable Matching μ

While $\exists$ firm that has not received a proposal, do:

1. $\forall w$, w proposes to best uncrossed firm
2. $\forall f$: says "maybe" to best proposal and "reject" to the rest
3. $\forall w$: s.t. w got "reject", cross that firm from its list

Figure 1.2: Gale-Shapley algorithm for finding a stable matching.

Key observations of DA:

1. If firm f gets a proposal in iteration i, it will get a proposal in all future iterations.
2. If there is an iteration in which a worker w proposes to its last firm then we have a perfect matching. This follows from observation 1: since w was rejected from all other firms, each of those firms must currently have a proposal.

Lemma 1.5. *DA terminates in at most n^2 iterations.*

Proof. In every iteration before the last, at least one worker will cross a firm off its list. By observation 2, only one worker can propose to the last firm on its list. So at most $n^2 - 2n + 1$ crosses are made. ■

Theorem 1.6. *DA outputs a stable matching μ.*

(a) Execution step 1 (b) Execution step 2

Figure 1.3: Execution of DA-algorithm

Proof. By contradiction. Assume there is a blocking pair (w, f'). Then w got rejected by f' in some iteration, so f' must be matched to a worker it prefers more than w, which contradicts (w, f') being a blocking pair. ∎

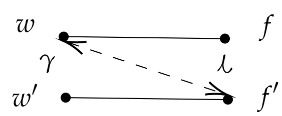


Figure 1.4: Blocking Pair (w', f)

Definition 1.7. *Let S be the set of all stable matchings over (W, F). For each worker w, the* ***realm of possibilities*** *$R(w)$ is the set $\{f | (w, f) \in \mu, \mu \in S\}$. The* ***optimal firm*** *for w is the best firm in $R(w)$ w.r.t. w's preference list, also denoted as $opt(w)$. The* ***pessimal firm*** *for w is the worst firm in $R(w)$ w.r.t. w's preference list, also denoted as $pess(w)$. The* ***realm of possibilities, optimal worker, and pessimal worker*** *are defined similarly for firms.*

Lemma 1.8. *Each worker has a unique optimal firm.*

Proof. By contradiction. Suppose two workers have the same optimal firm, i.e. $f = opt(w) = opt(w')$. f prefers one of w or w'. WLOG, let us assume that $w \succ_f w'$. Let μ be a stable matching where $(w', f), (w, f') \in \mu$. Since $f = opt(w)$, $f \succ_w f'$. Then (w, f) forms a blocking pair w.r.t. μ, which is a contradiction. ∎

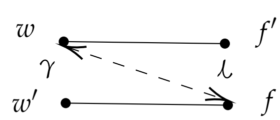


Figure 1.5: Blocking Pair (w, f)

Corollary 1.9. *Matching every worker to its optimal firm results in a perfect matching μ_W.*

Lemma 1.10. *μ_W is stable.*

Proof. By contradiction. Assume there is a blocking pair (w, f'), and that $(w, f), (w', f') \in \mu_W$.

Then $f' \succ_w f$ and $w \succ_{f'} w'$.
Since $f' = opt(w')$, there is a a stable matching μ' s.t. $(w', f'), (w, f'') \in \mu'$.

$f = opt(w)$, so $f' \succ_w f \succeq_w f''$. Then (w, f') is a blocking pair w.r.t. μ', which is a contradiction.

∎

(a) Blocking Pair $(w, f') \in \mu$

(a) Blocking Pair $(w, f') \in \mu$'

Figure 1.6: Blocking pair w.r.t. *Lemma* 1.8

We call μ the worker-optimal stable matching.

Theorem 1.11. *DA outputs the worker-optimal stable matching.*

Proof. By contradiction. Consider the first iteration in which a worker w is rejected by its optimal firm f. f sends a maybe to worker w' in this iteration. So $w' \succ_f w$. Since $opt(w') \neq opt(w)$ by Lemma 1.8, and w' couldn't have been rejected by $opt(w')$ yet, we have that $f \succ_{w'} opt(w')$. Now consider a stable matching μ where $(w, f), (w', f') \in \mu$. Then $f \succ_{w'} opt(w') \succeq_{w'} f'$. Then (w', f) is a blocking pair w.r.t. μ, which is a contradiction.

■

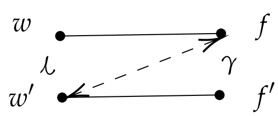


Figure 1.7: Blocking Pair (w', f)

Corollary 1.12. *The worker-optimal stable matching is also the firm-pessimal stable matching.*

Proof. By contradiction. Let μ be the worker-optimal stable matching with $w \neq pess(f)$ for some $(w, f) \in \mu$. Then let μ' be another stable matching with $(w, f'), (w', f) \in \mu'$, where $w \succ_f w'$. Since μ is worker optimal, $f \succ_w f'$. Then (w, f) forms a blocking pair w.r.t. μ', which is a contradiction. ■

1.4 More General Setting

1.4.1 Incomplete Lists

We now generalize our setting to different numbers of workers and firms, and incomplete preference lists. Consider the bipartite graph $G = (W, F, E)$ where $|W| = n$, $|F| = m$, and the preference list for each participant is defined only on its neighbors, specified by E.

Definition 1.13. *Let μ be any maximal matching over (W, F, E), then (w, f) form a* ***blocking pair*** *w.r.t. μ if either:*

> ***(Type 1)*** *w, f are matched in μ and prefer each other over their partners in μ.*
>
> ***(Type 2)*** *w unmatched and f prefers w over its partner in μ, or vice-versa.*

We modify the termination condition of DA so that the algorithm stops when in an iteration, each firm receives at most one proposal.

Lemma 1.14. *DA outputs a maximal matching on G.*

Proof. By contradiction. Suppose worker w and firm f are both unmatched and are neighbors in G. Then by DA, w must have proposed to f and got rejected. But this contradicts Observation 1. ■

Definition 1.15. *If a worker is unmatched in μ then $\mu(w) = \perp$. Likewise, if a firm is unmatched in μ then $\mu(f) = \perp$. We denote the set of workers matched in μ to be $W(\mu)$ and the set of firms matched in μ to be $F(\mu)$.*

The proof for stability from DA follows from Theorem 1.6. We note that Lemma 1.8 still holds, except that multiple workers may have $\perp$ as their optimal firm, so worker-optimality follows from Theorem 1.11.

Lemma 1.16. *For all stable matchings $c = |W(\mu)| = |F(\mu)|$ is constant, i.e. the numbers of workers/firms matched in all stable matchings is the same.*

Proof. Let μ_W be the worker-optimal matching, and μ_F be the firm-optimal matching. Then all workers who are unmatched in μ_W will be unmatched in other stable matchings by worker-optimality of μ_W.
So for any stable matching μ we get $W(\mu_W) \supseteq W(\mu) \supseteq W(\mu_F)$.
Thus $|W(\mu_W)| \geq |W(\mu)| \geq |W(\mu_F)|$. Similarly for firms we get $|F(\mu_W)| \leq |F(\mu)| \leq |F(\mu_F)|$. But $|W(\mu_W)| = |F(\mu_W)|$ and $|W(\mu_F)| = |F(\mu_F)|$, so the cardinality of all sets are equal.

■

Theorem 1.17. *For all stable matchings $\mu : W(\mu) = F(\mu)$,i.e. if a worker/firm is matched in one stable matching it is matched in all stable matchings.*

Proof. $W(\mu_W) \supseteq W(\mu) \supseteq W(\mu_F)$. By Lemma 1.16 they must all be equal and so are all the same set. ■

1.4.2 Stable Matching with Firm Capacities

We extend the previous setting by allowing firms to have multiple job openings. Consider the bipartite graph (W, F, E) where $|W| = n$, and the preference list for each participant is defined only on its neighbors. In addition we define a capacity function on all firms $c : F \to \mathbb{Z}^+$, such that $\sum_{f \in F} c(f) = m$.

Definition 1.18. *Let μ be any maximal matching over (W, F, E), then (w, f) form a* ***blocking pair*** *w.r.t. μ if:*

> ***(Type 1)*** *f is matched to capacity and w is matched to f' but there is a worker w' matched to f such that $w \succ_f w'$ and $f \succ_w f'$.*
>
> ***(Type 2a)*** *f has unmet capacity and $f \succ_w \mu(w)$.*
>
> ***(Type 2b)*** *w is unmatched but firm $w \succ_f w'$ and f is matched to w'.*

Reduction 1.19. *Stable matching with capacities can be reduced to stable matching under incomplete lists.*

Let $I = (W, F, E), c$, preference lists over all firms and workers
We construct I' by making $c(f)$ copies of f:
$W' = W$
$F' = \cup_{f \in F} \{f^{(1)}, ..., f^{(c(f))}\}$
$E' = \{(w, f^{(i)}) | (w, f) \in E\}$
We update the preference lists of all workers such that:
$\forall w$: if $f \succ_w f' \implies \forall i, j \;\; f^{(i)} \succ_w f'^{(j)}$ and if $i < j$ then $f^{(i)} \succ_w f^{(j)}$, i.e. workers prefers copies with smaller indices, but still retain their preferences over the firms.

All firms retain the same preference list in I′.

Key observations of Reduction 1.19:

(a) If $f^{(1)}...f^{c(f)}$ is matched with $k < c(f)$ workers, than $f^{(1)}...f^{(k)}$ must be matched and $f^{(k+1)}...f^{(c(f))}$ remain unmatched.

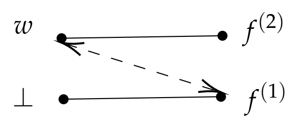


Figure 1.8: Blocking Pair $\left(w,\ f^{(1)}\right)$

(b) If $(f^{(i)}, w), (f^{(j)}, w') \in \mu$ with $i < j$, then $w \succ_f w'$

Lemma 1.20. *There is a bijection ϕ between stable matchings in I and I'.*

Proof. Let μ be a stable matching for I. $\phi(\mu)$ is created by matching each worker w to $f^{(i)}$ if w is the ith most preferred worker matched to f in μ. We prove by contradiction that $\phi(\mu)$ is stable. Assume that there is a blocking pair $(w, f^{(i)})$. By construction, w is not matched to a copy of f, and $f^{(i)}$ prefers w to its current match. Therefore, (w, f) is a blocking pair w.r.t. μ,which is a contradiction.
$\phi^{-1}(\mu')$ is given by matching w to f if w is matched to a copy of f in μ'. $\phi^{-1}(\mu')$ is stable since each f has the same preferences as its copies. ■

Theorem 1.21. ***(Rural Hospital Theorem)***

1. *For all stable matchings μ, the set of matched workers, and the number of filled positions in each firm, are the same*
2. *if $|\mu(f)| < c(f)$ then $\mu(f)$ is the same for all μ.*

Proof. Proof for 1 follows from Theorem 1.17.
We prove 2 by contradiction. Let μ_W be the worker-optimal stable matching, and μ_F be the firm-optimal stable matching. Suppose $\mu_W(f) \neq \mu_F(f)$. Then there is a worker w who is matched to f in μ_W, but not matched to f in μ. w prefers f to its partner in μ by worker-optimality. Since f is not filled to capacity in μ, (w, f) forms a blocking pair w.r.t. μ, a contradiction. ■

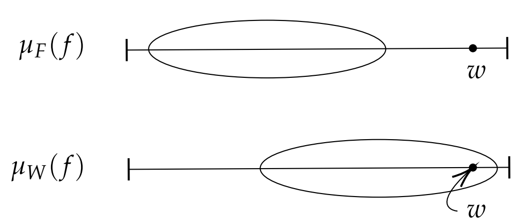


Figure 1.9: Worker w on f's preference list

Roadmap

Problem	Stability	DSIC	Rotation & Lattice	LP
Complete Lists $\lvert W\rvert = \lvert F\rvert = n$	*blocking pair*			
Incomplete Lists $\lvert W\rvert = n$, $\lvert F\rvert = m$	*Type* 1 & *Type* 2 *blocking pair*	?	?	?
Incomplete Lists *firms with capacities*	*Reduction*	*Reduction*	*Reduction*	*Reduction*

Exercises

1. Construct a stable matching instance that has 1 stable matching.
2. Construct a stable matching instance that has exponentially many stable matchings.

References

[GS 62] Gale, D. and Shapley, L. S. "College Admissions and the Stability of Marriage". American Mathematical Monthly. 69 (1): 9–14

[GI 89] D. Gusfield and R. Irving, The Stable Marriage: Structure and Algorithms, *The MIT Press* (1989)