DS 320 Algorithms for Data Science	**Lecture #23**
Spring 2022	Prof. Kira Goldner

Stable Matching

Today we'll consider a problem faced in practice all the time—the stable matching problem. Its algorithmic solution is a real "killer app," and a way to see how much impact an elegant algorithmic solution can have in practice. Mass General Shapley and Alvin Roth won the Nobel Prize in Economics in 2012 for the Gale-Shapley algorithm and their work on applying it in practice.

One real-world application has to do with the National Resident Matching Program. Every year, doctors graduating from medical school, who seek hospitals to work at to continue their training, are matched with hospitals who seek doctors to employ. Doctors have preferences over which hospitals they want to work at, and hospitals have preferences over which doctors they hire.

Any setting matching two sides that both have preferences falls into this scenario. Other applications include: college admissions, other worker-job matchings (especially in online labor markets), the *school choice* problem where public school students apply to various high schools in the city and those schools also have preferences over which students they admit, and even web content delivery, where one side requests web content and the other is servers with preferences over which customers they serve.

For now, we'll simplify the problem to pretend that each hospital hires exactly one doctor—we're looking for a *perfect matching*—but it's easy to modify the scenario to handle hospitals hiring multiple doctors, schools admitting multiple students, etc.

Consider a set of n doctors $D = \{d_1, \ldots, d_n\}$, and a set of n hospitals $H = \{h_1, \ldots, h_n\}$. Let $D \times H$ denote the set of all possible ordered pairs of the form (d, h), where $d \in D$ and $h \in H$. A matching M is a set of ordered pairs, each from $D \times H$, with that property that each member of D and each member of H appears in at most one pair in M. A perfect matching M' is a matching with the property that each member of D and each member of H appears in exactly one pair in M'.

Now we can add the notion of preferences to this setting. Each doctor $d \in D$ ranks all the hospitals; we will say that d prefers h to h' if d ranks h higher than h'. We will refer to the ordered ranking of d as their preference list. We will not allow ties in the ranking. Each hospital, analogously, ranks all the doctors.

The problem that used to happen with the NRMP Match is that they would release all of the results—this doctor is matched with this hospital—and then some doctor and hospital would realize that they would prefer to contract with each other than their assigned matches. They would call each other up and make alternative arrangements, throwing away the proposed matching. Formally, some pairs $(d, h), (d', h')$ would be proposed in the matching, but

$$h' \succ_d h \qquad \text{and} \qquad d \succ_{h'} d',$$

that is, doctor d prefers h' to h, and hospital h' prefers d to d', so doctor d and hospital h' would prefer to deviate from the assignment and form a pair together as (d, h'). We call such a pair that prefers one another to their assigned partners a *blocking pair*.

A matching M is *stable* if and only if it has no blocking pairs. We will to come up with an algorithm to find a stable matching.

Basic ideas:

- Initially, everyone is unmatched. Suppose an unmatched hospital h chooses the doctor d who ranks highest on their preference list and "proposes" to them, offering them a job. Can we declare immediately that (d, h) will be one of the pairs in our final stable matching? Not necessarily: at some point in the future, a hospital h' whom d prefers may propose to them. On the other hand, it would be dangerous for d to reject h right away; they may never receive an offer from a hospital they rank as highly as h. So a natural idea would be to have d tentatively accept. Mirroring marriage terms, we say the pair (d, h) enters an engagement.

- Suppose we are now at a state in which some doctors and hospitals are free—not engaged—and some are engaged. The next step could look like this. An arbitrary free hospital h chooses the highest-ranked doctor d to whom they have not yet proposed, and they proposes to d. If d is also free, then h and d become engaged. Otherwise, d is already engaged to some other hospital h'. In this case, d determines which of h or h' ranks higher on their preference list; this hospital becomes engaged to d and the other becomes free.

- Finally, the algorithm will terminate when no one is free; at this moment, all engagements are declared final, and the resulting perfect matching is returned.

The hospital-proposing Gale-Shapley (Deferred Acceptance) algorithm:

1. Each hospital "proposes" to their favorite doctor on their list.

2. Each doctor who receives at least one proposal "gets engaged" to the hospital they prefer among those who propose; "rejects" the rest. Doctors with no proposals do nothing.

3. If no hospital is rejected, stop. Finalize the engagements into the stable matching. Otherwise, rejected hospitals cross the name of the doctor who rejected them off their list and then propose to the favorite among those remaining.

4. Return to Step 2.

List all of your observations about the algorithm here (think: loop invariants):

- Every doctor d remains engaged from the point at which they receive their first proposal; and the sequence of partners to which they are engaged gets better and better (in terms of their preference list).

- The sequence of doctors to whom h proposes gets worse and worse (in terms of his preference list). However, when matched, they are engaged to whomever they most prefer out of those that will accept them.

- If h is free at some point in the execution of the algorithm, then there is a doctor to whom they have not yet proposed.

Claim 1. The Gale-Shapley algorithm returns a stable matching.

Proof. A stable matching is a perfect matching with no blocking pairs.

To see that GS returns a perfect matching, observe that we terminate when there are no free hospitals. By our third observation, if there is a free hospital, then there is a free doctor it has not proposed to, and a free doctor always tentatively accepts, just as a free doctor always proposes, hence they would match.

To see that GS returns a matching with no blocking pairs, suppose that (h, d) is a blocking pair—they are not matched to each other in the returned matching but prefer one another to their given match. There are two cases.

Case 1: In the execution of GS, h proposed to d. Then for (h, d) to not be matched, it must be that d rejected h for some preferred proposal/engagement, and by the second observation, one d is engaged, d stays engaged to hospitals of increasing preference, thus d must be engaged to some $h' \succ_d h$, and (h, d) cannot be a blocking pair.

Case 2: In the execution of GS, h never proposed to d. Because h proposed in decreasing order on his preference list until they are matched, then h didn't propose because they got and stayed engaged to some other doctor $d' \succ_h d$ that h preferred. Thus (h, d) cannot be a blocking pair.

In both cases, we have a contradiction, hence GS returns a stable matching. □

Claim 2. The Gale-Shapley algorithm terminates in time $O(n^2)$.

Proof. Each hospital proposes to each doctor at most once, and thus proposes to at most n doctors. There are n hospitals, so there are at most $O(n^2)$ proposals. □

Claim 3. The hospital-proposing algorithm returns the hospital-optimal stable matching. (Each hospital is matched to their best eligible partner.)

Proof. We call a partner d *eligible* for h if there exists *some* stable matching M' in which (d, h) are partners. Suppose, seeking a contradiction, that Gale-Shapley with hospitals proposing returns a matching which is not hospital-optimal, that is, some hospital is not paired with its best eligible partner.

Consider the first time that some hospital h is rejected by an eligible partner d. In order to reject, d must have another offer from some $h' \succ_d h$ which they prefer. Consider the stable matching M that pairs (h, d), and let d' be the partner of h' in that matching.

Because h was the first hospital rejected by an eligible partner when d rejected it, then h' has not been rejected by an eligible partner at this point. That is, $d \succ_{h'} d'$, as h' has proposed to d at this point, and there is some matching in which (h', d') are matched, so d' is an eligible partner and has yet to reject h' (and therefore be proposed to). But in that case, M is not a stable matching, as (h', d) would be a blocking pair, so this is a contradiction. □

Claim 4. The hospital-proposing algorithm returns the doctor-pessimal stable matching. (Each doctor is matched to their worst eligible partner.)

Proof. Suppose in Gale-Shapley that there is a pair (h, d), but h is not the worst eligible partner for d. The, there exists a stable matching M with pair (h', d) where $h' \prec_d h$. Let d' be the partner of h' in M. By Claim 3, the GS matching is hospital-optimal, so $d \succ_h d'$. Hence h and d prefer each other to their partners in M. M is thus not stable, and we have a contradiction. □

Other fun topics about Stable Matching

- There are many different stable matchings! Hospitals get worse preferences and doctors get better. . . what's a *fair* stable matching?
- How *many* stable matchings are there? Can we bound them? Lots of research here! [7]
- The different stable matchings give a set of doctors and hospitals and their preferences form an incredible lattice structure. [6]
- Other algorithms for obtaining different stable matchings. [3, 1]
- What if preferences aren't fully known, how many interviews do we need to conduct? What if preferences are random? [10, 2, 4]
- This algorithm is "strategyproof" or "truthful"—one cannot reorder their preferences to get a better match. But is there anything else they could do? Truncate their preferences? (Yes! [5]) Form a coalition? (Yes! [9])
- Even though it's strategyproof and pretty easy to explain as such, there's evidence of people irrationally listing missorted preferences in practice (in really obvious ways—"I prefer BU without a scholarship to BU with a scholarship"). This sparked a new direction of research on "obviously strategyproof" algorithms. [8]

References

[1] Atila Abdulkadiroglu and Tommy Andersson. School choice. Technical report, National Bureau of Economic Research, 2022.

[2] Hedyeh Beyhaghi and Éva Tardos. Randomness and fairness in two-sided matching with limited interviews. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[3] Piotr Dworczak. Deferred acceptance with compensation chains. *Operations Research*, 69(2):456–468, 2021.

[4] Hugo Gimbert, Claire Mathieu, and Simon Mauras. Two-sided matching markets with strongly correlated preferences. In *International Symposium on Fundamentals of Computation Theory*, pages 3–17. Springer, 2021.

[5] Yannai A Gonczarowski. Manipulation of stable matchings using minimal blacklists. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 449–449, 2014.

[6] Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms.* MIT press, 1989.

[7] Anna R Karlin, Shayan Oveis Gharan, and Robbie Weber. A simply exponential upper bound on the maximum number of stable matchings. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 920–925, 2018.

[8] Shengwu Li. Obviously strategy-proof mechanisms. *American Economic Review*, 107(11):3257–87, 2017.

[9] Marek Pycia. Stability and preference alignment in matching and coalition formation. *Econometrica*, 80(1):323–362, 2012.

[10] Baharak Rastegari, Anne Condon, Nicole Immorlica, Robert Irving, and Kevin Leyton-Brown. Reasoning about optimal stable matchings under partial information. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 431–448, 2014.