Harvard CS 1360, Spring 2026
Instructor: Ariel Procaccia
Scribe: Kelsey Chen, Jakob de Raaij

Stable Matchings

Lecture 15

In this lecture we discuss two-sided matchings, where two sets of agents need to be matched to each other. In contrast to the previous lectures, now both 'sides' of the matching have preferences. Because of that, the goal no longer is to find a maximum-cardinality matching; in fact, we assume that any pair of agents from the two sides can be matched and only consider perfect matchings (where no one is left unmatched). Instead, we will try to find *stable* matchings, where no pair of agents would be better off matching by themselves instead of participating in the matching mechanism. Real-world examples include matching students to schools, matching TFs to classes, or matching doctors to hospitals.

1 Stable Matchings

Definition 1 (Stable Matching)**.** An instance of *two-sided matching* consists of

- n students $S = \{s_1, ..., s_n\}$ and n courses $T = \{t_1, ..., t_n\}$,
- where each student $s \in S$ has a ranking σ_s over courses T, and each course $t \in T$ has a ranking σ_t over courses S.

A *matching* $\pi : S \cup T \to S \cup T$ maps each student to a course and vice versa. That is, for all $s \in S$ and $t \in T$, $\pi(s) = t \Leftrightarrow \pi(t) = s$. A *blocking pair* for π is $(s, t) \in S \times T$ such that $s \succ_{\sigma_t} \pi(t)$ and $t \succ_{\sigma_s} \pi(s)$. That is, a student-course pair (s, t) who both prefer each other over their current match in π. A matching is *stable* if there exists no blocking pair.

In other words, a stable matching is a one-to-one assignment of students to courses so that no student and course that are not matched would prefer each other over their respective current matches.

Example 1 (Stable Matching)**.** Assume there are $n = 3$ students and courses with the following rankings

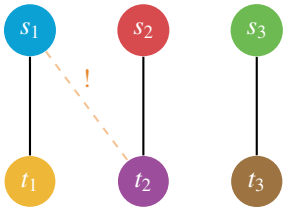
S preferences

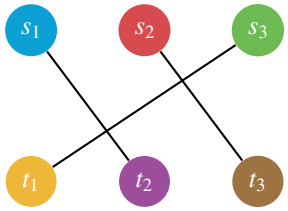
s_1	s_2	s_3
t_2	t_1	t_1
t_1	t_3	t_2
t_3	t_2	t_3

T preferences

t_1	t_2	t_3
s_1	s_3	s_1
s_3	s_1	s_3
s_2	s_2	s_2

Consider the two matchings below. The matching on the left is not stable, since the pair (s_1, t_2) is blocking: Student s_1 prefers course t_2 over their current match, t_1, and course t_2 prefers student s_1 over their current match, s_2. The matching is not stable, since student s_1 and course t_2 would be better off if they matched with each other than under this matching. In contrast, the matching on the right is stable: Students s_1 and s_3 are matched to their favorite choice, so only s_2 could be part of a blocking pair. There is only one course, t_1, that s_2 prefers over their current match t_3; this course t_1 prefers their current match s_3 over s_2. Thus, there are no blocking pairs.





While we were able to find a stable matching in this example, it is not at all obvious from the definition that such stable matchings always exist. We will now discuss an algorithm that is guaranteed to find a stable matching, thereby constructively proving that a stable matching always exists.

Algorithm 1 Student-Proposing Deferred-Acceptance (DA) Algorithm

```
Start with all students and courses being unmatched.
All students propose to their most-preferred course.
Each course that receives at least one proposal, tentatively accepts their most-preferred student.
while there still exists an unmatched student do
    Each unmatched student proposes to their most preferred course that they haven't proposed to so far.
    Each course that received at least one new proposal proceeds as follows:
    if the course already tentatively accepted a student and prefers them over all new proposals then
        the course keeps this current, tentatively accepted student.
    else the course tentatively accepts its favorite student from the new proposals.
    end if
end while
Return the matching where each student is matched to the course that tentatively accepted it.
```

Intuitively, in the deferred-acceptance algorithm, students propose to courses in decreasing order of preference, and each course at any point tentatively accepts its favorite student that proposed to it so far, until a more preferred student proposes. This version is known as the student-proposing DA algorithm; one can analogously define the course-proposing DA algorithm by switching the role of courses and students.

Example 2 (Deferred-acceptance algorithm)**.** We revisit the preferences from Example 1 and consider what the student-proposing deferred-acceptance algorithm will do. In round 1, each student proposes to their most preferred course (on the left). Course t_1 receives two proposals and tentatively accepts their preferred student of the two, s_3. Course t_2 tentatively accepts their only proposal, by student s_1. The tentative acceptances are shown on the right.

In the next round, only student s_2 is unmatched. They propose to their next-favorite course, t_3. Course t_3 tentatively accepts the proposal, since it is its only proposal.

Since no more students are unmatched, the algorithm terminates and the tentative acceptances become the returned matching.

Theorem 1. *The student-proposing deferred-acceptance algorithm terminates with a stable matching.*

Proof. First, note that the algorithm always terminates. In any round that is not the last, at least one proposal is rejected, and no student repeats a proposal.

Let π be the matching returned by the algorithm and assume towards a contradiction that (s, t') is a blocking pair. Let $\pi(s) = t$ and $\pi(t') = s'$ be the matches of s and t' in π. Since s prefers t' to t (by the definition of a blocking pair), s proposed to t' before it proposed to t. Since t' is paired with s', t' prefers s' to s (since the tentatively accepted students only improve in each course's ranking throughout the algorithm). However, t' preferring its match s' in π to s is a contradiction to (s, t') being a blocking pair for π. □

2 The Lattice Property of Stable Matchings

So far, things are looking great! We have found an efficient algorithm that gives us stable matchings. However, there is a clear asymmetry in the algorithm between the role of the students and the role of the courses, so we may wonder which side this benefits.

To analyze this, we will show that the set of stable matchings satisfies a remarkable property: There exists a stable matching that is the most-preferred for every student, simultaneously, and least-preferred by every course, simultaneously, and another matching in which the exact opposite is true.

Definition 2. For two matchings π and π',

- we write $\pi \geq_S \pi'$ if $\pi(s) \geq_{\sigma_s} \pi'(s)$ for all $s \in S$, i.e., each student prefers their match in π over their match in π'. We write $\pi =_S \pi'$ if and only if $\pi = \pi'$. We call $\geq_S$ the *student-respecting preference ordering*.
- the *join* $\pi^j = \pi \vee \pi'$ of π and π' is a stable matching π^j such that $\pi^j \geq_S \pi, \pi^j \geq_S \pi'$, and for every stable matching $\pi^\star$ satisfying these two inequalities, $\pi^\star \geq_S \pi^j$. Out of all stable matchings that are weakly preferred by all students over π and π', the stable matching π^j is the least-preferred by the students (the "lowest-possible upper bound").
- the *meet* $\pi^m = \pi \wedge \pi'$ of π and π' is a stable matching π^m such that $\pi^m \leq_S \pi, \pi^m \leq_S \pi'$, and for every stable mathcing $\pi^\star$ satisfying these two inequalities, $\pi^\star \leq_S \pi^m$. Out of all stable matchings that the students prefer (weakly) less than π and π', the stable matching π^m is the most-preferred by the students (the "greatest-possible lower bound").

Note that it is not oblivious whether the join and meet exist for all matching π and π'.

Theorem 2. *The meet and join exist for any pair of stable matchings.*

This is known as the *lattice property* of stable matchings, with respect to $\geq_S$. Given any two, distinct stable matchings, there is a unique matching (their join) that is minimally preferred to both. Similarly, we can find another matching (their meet) that is minimally less preferred than both. In particular, this implies that there is a unique stable matching that is most preferred by *every* student, since we can combine any two distinct stable matchings to obtain another stable matching that is preferred to both.

Example 3 (Meet and Join)**.** Consider the following preferences for $n = 4$ students and courses,

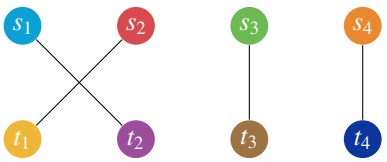
S preferences

s_1	s_2	s_3	s_4
t_1	t_2	t_3	t_4
t_2	t_1	t_4	t_3
t_3	t_4	t_1	t_2
t_4	t_3	t_2	t_1

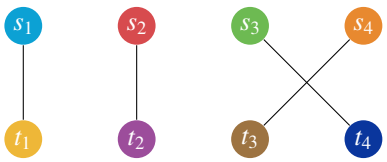
T preferences

t_1	t_2	t_3	t_4
s_4	s_3	s_2	s_1
s_3	s_4	s_1	s_2
s_2	s_1	s_4	s_3
s_1	s_2	s_3	s_4

and the following two matchings π and π'.

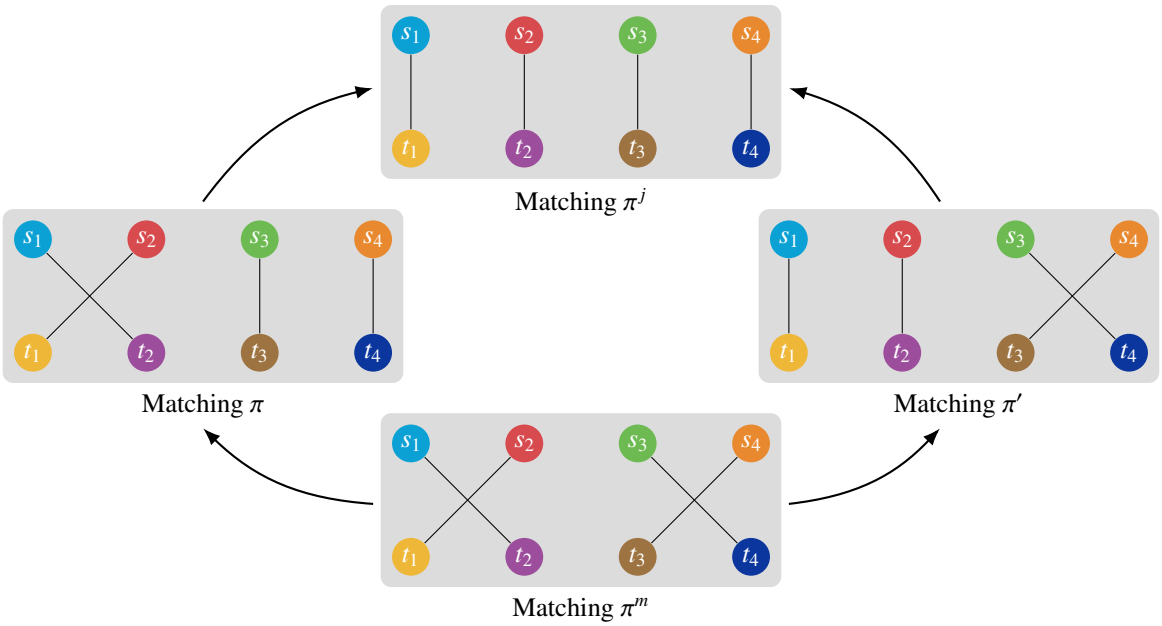


Matching π



Matching π'

Their join and meet are shown in the lattice below. One can check that indeed $\pi^j \geq_S \pi, \pi'$ and $\pi, \pi' \geq_S \pi^m$, and that π^j and π^m are the least-preferred and most-preferred stable matching to do so.



Let's now prove that the join and meet always exist for stable matchings π and π'.

Proof of Theorem 2. We define a *pointing operator* λ for π and π': For $s \in S$, $\lambda(s)$ is the course out of $\pi(s)$ and $\pi'(s)$ that is more preferred by s; for $t \in T$, $\lambda(t)$ is the student out of $\pi(t)$ and $\pi'(t)$ that is less preferred by t. We'll now prove that λ is the join, by showing that it is a matching, then that it is stable, and finally, that it is the join.

First, we prove that λ is a matching, i.e., that each student and course is matched to exactly one course and student, respectively. Consider any $s \in S$ with $\lambda(s) = t$. W.l.o.g., assume that (s,t) are matched in π. If (s,t) are also matched in π', we know $\lambda(t) = s$. Else, let (s,t') and (s',t) be matched in π'. If $\lambda(t) = s$, then t prefers s over s' and s prefers t over t', by the definition of λ, so (s,t) is a blocking pair in π' —a contradiction. It follows that $\lambda(t) = s$, so λ maps each s to exactly one t, and vice versa.

Now we show λ is stable, i.e., that there is no blocking pair. Suppose towards a contradiction that (s,t) is a blocking pair in λ. Course t is matched in λ with either $\pi(t)$ or $\pi'(t)$, so w.l.o.g. assume $\lambda(t) = \pi(t)$. It follows that $s \succ_{\sigma_t} \pi(t)$ since (s,t) is a blocking pair. Furthermore, since s is paired in λ with its more preferred of $\pi(s)$ and $\pi'(s)$, we know that $t \succ_{\sigma_s} \pi(s)$ and $t \succ_{\sigma_s} \pi'(s)$ (since (s,t) is a blocking pair). However, $t \succ_{\sigma_s} \pi(s)$ and $s \succ_{\sigma_t} \pi(t)$ imply that (s,t) is a blocking pair in π —a contradiction. Thus, λ is stable.

Lastly, note that λ is the join since every matching $\pi^\star$ satisfying $\pi^\star \geq_S \pi$ and $\pi^\star \geq_S \pi'$ must at least take the student-wise more preferred of $\pi(s)$ and $\pi'(s)$ for any s, so at least be as preferred, for each student s, as λ.

To prove that the meet operator always exists, we define a pointing operator that returns the less preferred course out of $\pi(s)$ and $\pi'(s)$ for each student $s \in S$ and the more preferred student out of $\pi(t)$ and $\pi'(t)$ for each course $t \in T$; the proof that this indeed is the meet is analogous. □

From the equivalences between the pointing operators and the join and meet, respectively, we also learn that moving up in the lattice (the join) is improving the matching for all students while worsening the matching for all courses. At the same time, moving down in the lattice (the meet) is worsening the matching for all students and improving the matching for all courses. In particular, the join with respect to the students' preferences $\geq_S$, is equivalent to the meet with respect to the courses' preferences $\geq_T$ (defined analogously), and vice versa.

It follows that there exists a *student-optimal (course-pessimal) stable matching* $\bar{\pi}$ such that $\bar{\pi} \geq_S \pi$ for every stable matching π, and a *student-pessimal (course-optimal) stable matching* $\underline{\pi}$ such that $\underline{\pi} \leq_S \pi$ for every stable matching π.

Theorem 3. *The student-proposing deferred-acceptance algorithm terminates with a student-optimal stable matching. The course-proposing deferred-acceptance algorithm terminates with a course-optimal stable matching.*

Proof. Students propose in order of decreasing preference, so if s is matched with t, s was before rejected by all t' such that $t' \succ_{\sigma_s} t$. Thus, it suffices to show that no student is ever rejected by an *achievable course*, i.e., a course that the student could be matched with in some stable matching.

We prove by induction on the number of rounds of the algorithm that at the beginning of every round ℓ of the algorithm, the courses t that rejected a student s so far are not achievable for s. The base case for the first round, $\ell = 1$, is trivial: When the algorithm starts, no student has been rejected yet.

Now, in round ℓ, suppose some course t rejects some student s. Thus, t is matched after this round with another student s' with $s' \succ_{\sigma_t} s$. We want to show that no stable matching can pair (s, t). Since student s' also proposes in the order of their preferences, they already proposed to every t' that they prefer to t in a previous round, so by the inductive assumption all those t' are not achievable for s'. Thus, s' prefers t to every achievable course $t' \neq t$. Assume towards a contradiction that there is a stable matching π with $\pi(s) = t$. It follows that $\pi(s') = t'$ for an achievable t' with $t \succ_{\sigma_{s'}} t'$. However, then (s', t) is a blocking pair in π since $t \succ_{\sigma'_s} t'$ and $s' \succ_{\sigma_t} s$ — a contradiction. We thus know that no student s in round ℓ was rejected by an achievable course. The theorem for the student-proposing DA algorithm follows.

The proof for the course-proposing DA algorithm is analogous, since the problem and algorithm is fully symmetric in students and courses (in particular, if we exchange all occurrences of the two words, the proof follows verbatim). □

To illustrate the course-proposing DA outputting a course-optimal stable matching, different than the student-optimal stable matching, we consider an example.

Example 4 (Course-proposing deferred-acceptance algorithm)**.** We once again consider the preferences from Example 1,

S preferences

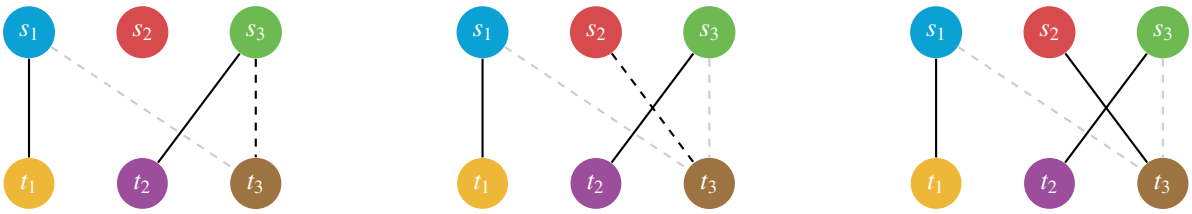
s_1	s_2	s_3
t_2	t_1	t_1
t_1	t_3	t_2
t_3	t_2	t_3

T preferences

t_1	t_2	t_3
s_1	s_3	s_1
s_3	s_1	s_3
s_2	s_2	s_2

and consider what the course-proposing deferred-acceptance algorithm will do. In round 1, each course proposes to their most preferred student (on the left). Student s_1 receives two proposals and tentatively accepts their preferred course of the two, t_1. Student s_3 tentatively accepts their only proposal, by course t_2. The tentative acceptances are shown on the right.

In the next round, only course t_3 is unmatched. They propose to their next-favorite student, s_3 (on the left). Student s_3 already tentatively accepted course t_2, which they prefer, so they reject the proposal. Ultimately, course t_3 proposes to s_2 (in the middle), who is unmatched and thus accepts the proposal (on the right).



Since no more students are unmatched, the algorithm terminates and the tentative acceptances become the returned matching. One can check that this stable matching is more preferred by courses t_1 and t_2 (and equally preferred for t_3) than the stable matching obtained from the student-proposing DA algorithm in Example 2, while students s_1 and s_3 prefer the latter over the earlier (and student s_2 is indifferent).

3 Strategic Stable Matchings

In practice, it is not unlikely that students or courses that are aware of the algorithm used to find a matching may try to misreport their preferences to manipulate the outcome. Thus, similarly to previous topics, we consider strategic players.

Theorem 4. *Truthful reporting is a dominant strategy for students in the student-proposing deferred-acceptance algorithm.*

While we won't prove this formally, consider the following intuition: Each student is matched to their most-preferred achievable course in the matching returned by the student-proposing DA algorithm; no student can change which courses are achievable for them by misreporting.

However, courses may misreport their preferences to achieve a better outcome under the student-proposing DA algorithm.

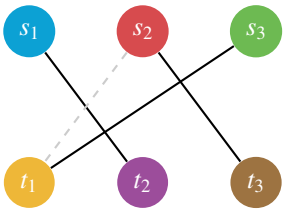
Example 5 (Strategic manipulation in DA algorithm)**.** Again, consider the preferences from Example 1 and the matching from the student-proposing DA algorithm, as seen in Example 2,

S preferences

s_1	s_2	s_3
t_2	t_1	t_1
t_1	t_3	t_2
t_3	t_2	t_3

T preferences

t_1	t_2	t_3
s_1	s_3	s_1
s_3	s_1	s_3
s_2	s_2	s_2

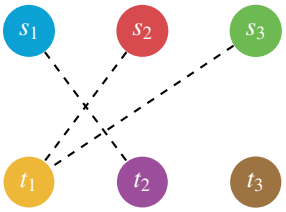


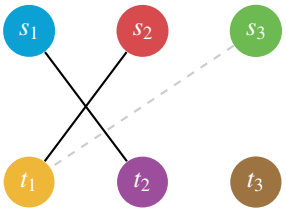
Now, let's assume that course t_1 misreports their preferences, to get

T preferences

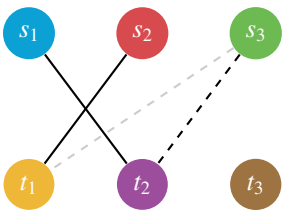
t_1	t_2	t_3
s_1	s_3	s_1
~~s_3~~ s_2	s_1	s_3
~~s_2~~ s_3	s_2	s_2

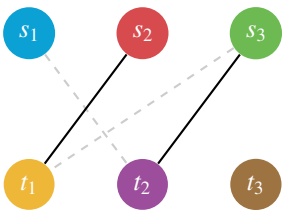
Let's consider what the student-proposing deferred-acceptance algorithm will do when course t_1 misreports their preferences. In round 1, each student still proposes to their most preferred course (on the left). Course t_1 receives two proposals and — since they misreported their preferences — now tentatively accepts s_2. Course t_2 tentatively accepts their only proposal, by student s_1. The tentative acceptances are shown on the right.





In the next round, only student s_3 is unmatched. They propose to their next-favorite course, t_2. Course t_2 prefers s_3 over their currently tentatively accepted student s_1, so they retract the tentative acceptance to s_1 and instead accept s_3.





Now, only student s_1 is unmatched. They propose to their next-favorite course, t_1. Course t_1 prefers s_1 over their currently tentatively accepted student s_2, so they retract the tentative acceptance to s_2 and instead accept s_1. This is where the strategic manipulation of course t_1 pays off: Since they accepted s_2 earlier over their more-preferred s_3, s_3 got to propose to t_2, thereby "freeing" s_1 from t_2. s_1 is now worse off, proposing to only their second favorite choice t_1.

The remaining unmatched student s_2 now propose to their next-favorite course, t_3. Course t_3 is still unmatched, so they tentatively accept s_2.

Since no more students are unmatched, the algorithm terminates and the tentative acceptances become the returned matching. The resulting matching now is strictly more preferred by course t_1, who is matched with s_1 instead of s_3, so t_1 benefited from misreporting their preferences (indeed, we obtained the course-optimal stable matching).

Unfortunately, strategic manipulation is not fully avoidable in stable matching algorithms.

Theorem 5. *No bipartite matching mechanism with two-sided preferences is strategyproof for all players and always returns a stable matching.*

You will prove this — with guidance — in assignment 4.

4 Stable Matchings in Practice

The National Resident Matching Program (NRMP) is a real-world application of the deferred-acceptance algorithm, used to match medical school graduates (the *residents*) to hospital residency programs in the United States. Every year, thousands of residents submit preferences over programs and hospitals submit rankings over residents. The system then computes a stable matching between residents and programs. The NRMP uses a modified version of the deferred-acceptance algorithm that allows for a fixed number of residents being matched to the same hospital, ties in the rankings, and couples (two residents) requesting to be placed in the same hospital. They use the resident-proposing (many times also called *applicant-proposing*) version of the deferred-acceptance algorithm, which ensures the matching is optimal for the residents.

Another important application of matching theory is in school choice, where districts use centralized systems to assign students to high schools. This is a two-sided matching problem: While students rank schools, schools also express priorities over students, often to account for factors like under-represented minorities, sibling attendance, or proximity.

Several large school systems have implemented versions of the student-proposing deferred-acceptance algorithm. New York City adopted such a system in 2003–04, followed by Boston Public Schools in 2005–06. Prior to this, Boston used an immediate acceptance mechanism, where students' first-choice applications were processed first, then second choices, and so on, with schools not being able to 'tentatively' accept a student and then 'replace' them with another student in a later round (of the algorithm). This approach was not strategy-proof — students could benefit from misrepresenting their preferences. When Boston Public Schools switched to the student-proposing deferred-acceptance algorithm, the task force emphasized a key reason:

> *"A strategy-proof algorithm 'levels the playing field' by reducing the disadvantage faced by families who don't strategize — or who don't strategize well."*

This was an explicit appeal to fairness, with strategyproofness framed as essential for ensuring equal access, especially for families with less familiarity or comfort navigating the system. For students and families, it removes the burden of gaming the process; for policy makers, it simplifies guidance and helps gather more accurate data on school preferences.