Lecture 14: Course Allocation

Indiana University,

Emerson Melo

- In this lecture, we focus on the HR (Hospitals/Residents) problem in the context of *course allocation* as this application will be familiar to most of you.
- In course allocation problems, the students and the course organizers have preferences over each other, and each student needs to be assigned to one course.
- The one-to-many problem is a generalization of the Stable Marriage Matching problem.

- It is known as the hospitals/residents problem (HR), because the assignment of residents or students to hospitals has become an influential application.
- The one-to-many version of the DA algorithm, where students propose assignments to hospitals, is also referred to as *student-optimal stable mechanism* (SOSM).
- Gale and Shapley (1962) showed that a DA algorithm yields a unique stable matching that weakly Pareto-dominates any other stable matching from the viewpoint of the students when the preferences of both sides are strict in a student-proposing version.

- A matching is stable if every individual is matched with an acceptable partner and if there is no student-hospital pair each member of which would prefer to match with the other rather than their assigned partner.
- The algorithm has a number of additional properties if priorities are strict: it is strategy-proof for students (the proposing side) and it is not Pareto-dominated by any other Pareto efficient mechanism that is strategy-proof.

- A variety of non-stable matching methods, such as first-come first-served policies and course bidding, are in use in practice.
- In course bidding systems (Budish and Cantillon, 2012; Krishna and Unver, 2008) students get equal budgets of an artificial currency, and then they bid on different courses.
- This currency does not have a value outside the matching market.
- The highest bidders for each course get a place.
- Course bidding has a number of strategic problems.

- For example, on the one hand bidders might be tempted to submit all their budget on their top preferences.
- Let's assume there is a very popular course and everybody bids their budget on this course.
- Only a few students can win, and the others might get courses which are very low on their preference list.
- On the other hand, students might preemptively bid on courses with lower preference because they believe they have a higher probability to win.
- So, various types of manipulation are possible.

- A one-to-many matching problem (or course allocation problem) consists of a finite set of students (or agents)

$$S = \{s_1, s_2, \ldots, s_n\}$$

and a finite set of courses (or objects)

$$C = \{c_1, c_2, \ldots, c_m\}$$

with maximum capacities

$$q = (q_{c_1}, q_{c_2}, \ldots, q_{c_n}).$$

- To ensure that a feasible matching exists we assume $q_c \geq 0$ for all $c \in C$ and $n \leq \sum_{c \in C} q_c$.

- Each student has a preference relation $\succeq_s$ over the courses C (called a student preference) and each course (organizer) has a preference relation $\succeq_c$ over the students S (called a course priority).
- The preferences of the students and courses $(\succeq_s, \succeq_c)$ do not have to be strict in general; they could also contain indifferences.
- The vectors for these relations are denoted $\succeq s = (\succeq_s)_{s \in S}$ and $\succeq_C = (\succeq_c)_{c \in C}$.

- Let $\mathscr{P}$ denote the set of all possible preference relations over C and $\mathscr{P}^{|S|}$ the set of all preference vectors for all students.
- Similarly, $\mathscr{P}^{|C|}$ is the set of all preference vectors for all course organizers.

Definition

A matching is a mapping μ of students S and courses C that satisfies:

(i) $\mu(s) \in C$ for all $s \in S$

(ii) $\mu(c) \subseteq S$ for all $c \in C$, and

(iii) for any $s \in S$ and $c \in C, \mu(s) = c$ if and only if $s \in \mu(c)$

- The Gale-Shapley student-optimal stable mechanism (SOSM) is a modified version of the Gale-Shapley deferred acceptance algorithm of Gale and Shapley (1962), which allows for one-to-many assignments.
- This "student-proposing" deferred acceptance algorithm works as follows

- A matching is feasible if $|\mu(c)| \leq q_c$ for all $c \in C$, which means that no course is overcrowded.
- Let M denote the set of all feasible matchings.
- One desirable property of matchings is Pareto efficiency, which is such that no student can be made better off without making any other student worse off.
- In school choice (and similarly in course allocation) problems, it is often assumed that schools have priorities but that they are not strategic.
- Also, Pareto efficiency is typically analyzed from the students' point of view.

Definition (Pareto efficiency of matchings (for students))

A matching μ is Pareto efficient with respect to the students if there is no other feasible matching μ' such that $\mu'(s) \succeq_s \mu(s)$ for all students $s \in S$ and $\mu'(s) \succ_s \mu(s)$ for some $s \in S$, where $\succ_s$ describes a strict preference.

- Stability means that there should be no unmatched pair of a student and a course (s, c) where student s prefers course c to her current assignment and she has a higher priority than some other student who is assigned to course c.
- Stability can be seen as a property of a solution that has no justified envy.
- In other words, stability means that there are no incentives for some pair of participants to undermine an assignment by joint action.

Definition (Stability)

A matching μ is stable if $\mu(s') \succ_s \mu(s)$ implies $s' \succ_{\mu(s')} s$ for all s, s'.

- The stability of a matching is closely connected to competitive equilibria in assignment markets (as discussed in previous lectures.
- Stable matchings, in which no pair of players would prefer to switch partners, are a subset of Pareto-optimal matchings considering both sides.
- Note that the notion of Pareto efficiency introduced above concerns only the students.
- A stable matching is Pareto optimal for both sides taken together but not necessarily for the students only.

- Matching is related to assignment markets, and we can also formulate the stable matching problem as a linear program.
- For the following linear program of the course assignment (CA) problem, we assume that each course has a capacity of only one and students need to be assigned to at most one course:

$$
\begin{aligned}
\max \sum_{(s,c)\in A} x_{s,c} & \qquad (CA) \\
\text{s.t.} \quad \sum_{s\in S} x_{s,c} \leq 1 & \qquad \forall c \in C \\
\sum_{c\in C} x_{s,c} \leq q & \qquad \forall s \in S \\
x_{s,c} = 0 & \qquad \forall (s,c) \in (S\times C)\backslash A \\
\sum_{j\in C: j\succ_s c} x_{s,j} + \sum_{i\in S: i\neq_c s} x_{i,c} + x_{s,c} \geq 1 & \qquad \forall (s,c) \in A \\
x_{s,c} \geq 0 & \qquad \forall (s,c) \in (S\times C)
\end{aligned}
$$

- The variables $x_{s,c}$ describe the assignment μ of a student s to a course c if they assume the value 1 and are 0 otherwise; A denotes the set of acceptable pairs.
- A matching μ is called individually rational if no student or course organizer prefers being unmatched (the third constraint in CA).
- A matching μ is stable if it is individually rational and if there is no pair (s,c) in A such that both $s \succ_c \mu(c)$ and $c \succ_s \mu(s)$ hold (the fourth constraint in CA).
- It can be shown that the solution x to the CA problem is a stable matching if and only if it is an integer solution of CA.

- We describe a fractional solution to CA as a fractional matching or random matching.
- A random matching $\mathbf{x}$ can be decomposed into a convex combination of deterministic matchings.
- Next we discuss mechanisms to compute matchings and their properties.
- A mechanism is an algorithm which computes a matching for given preferences of students and courses

- More formally, a two-sided matching mechanism χ is a function $\chi : \mathscr{P}^{|S|} \times \mathscr{P}^{|C|} \to M$ that returns a feasible matching of students to courses for every preference profile of the students.
- We assume that course organizers have public priorities which are not manipulable.
- For a submitted preference profile $\succeq = (\succeq s, \succeq c), \chi(\succeq)$ is the associated matching.
- For a student s the assigned course is $\chi_s(\succeq s) \in C$.
- For a course c the set of assigned students is $\chi_c\,(\succeq_S) \subseteq S$

- A mechanism is Pareto efficient if it always selects a *Pareto-efficient* matching.
- Also, a mechanism is stable if it always selects a stable matching.
- Another important property of a mechanism is *strategy-proofness*: there is no incentive for any student not to submit her truthful preferences, no matter which preferences the other students report.

Definition (Strategy-proofness)

A mechanism χ is strategy-proof if for any $\succ_S \in \mathscr{P}^{|S|}$ with $s \in S$ and $\succ'_s \in \mathscr{P}$ we have $\chi_s(\succ_S) \succeq_s \chi_s(\succ'_s, \succ_{S\setminus\{s\}})$

- Matching mechanisms cannot always be Pareto efficient for students and also stable, as the following example shows.
- **Example:** Consider the course allocation problem with three students $S = \{s_1, s_2, s_3\}$ and three courses $C = \{c_1, c_2, c_3\}$, each course having one place. The course priorities $(\succ_c)$ and the student preferences $(\succ_s)$ are given as follows:

Figure: (a) Course priorities ($\succ_c$) and (b) student preferences ($\succ_s$)

(a)

$\succ_{c_1}$	$\succ_{c_2}$	$\succ_{c_3}$
$\underline{s_1}$	$\underline{s_2}$	s_2
s_3	s_1	s_1
s_2	s_3	$\underline{s_3}$

(b)

$\succ_{s_1}$	$\succ_{s_2}$	$\succ_{s_3}$
c_2	c_1	c_1
$\underline{c_1}$	$\underline{c_2}$	c_2
c_3	c_3	$\underline{c_3}$

- The matching

$$\mu_1 = \begin{pmatrix} c_1 & c_2 & c_3 \\ s_1 & s_2 & s_3 \end{pmatrix}$$

is the only stable matching; it is underlined in table a.

- However, the matching Pareto dominated by

$$\mu_2 = \begin{pmatrix} c_1 & c_2 & c_3 \\ s_2 & s_1 & s_3 \end{pmatrix}$$

- Stability forces students s_1 and s_2 to share the courses in an inefficient way. If these students, s_1 and s_2, were assigned the courses c_2 and c_1 respectively, then student s_3 would prefer course c_1 to her assignment c_3 and she has a higher priority for course c_1 than student s_2.
- As a result, stability may conflict with Pareto efficiency.

Serial Dictatorship

- We now use the concept of *serial dictatorship* (SD).
- Serial dictatorship can be used for one-sided or two-sided matching problems.
- For random serial dictatorship (RSD, aka random priority), the students are ordered with a lottery.
- As a randomized mechanism RSD is universally truthful (intuition?)

- The SD algorithm proceeds as follows
  - Consider a particular ordering of the students S, i.e., a particular permutation ϕ of $\{1, 2, \ldots, n\}$.
  - We assume strict student preferences.
  - For x from 1 to $n = |S|$ do: Assign student $s_{\phi(x)}$ to her top choice among the remaining slots.
- The following example illustrates the serial dictatorship mechanism.

Example

- Consider the course allocation problem with four students $S \equiv \{s_1, s_2, s_3, s_4\}$ and four courses $C \equiv \{c_1, c_2, c_3, c_4\}$, each course having one place.
- The course priorities $(\succ_c)$ and the student preferences $(\succ_s)$ are given in table below; the ellipses indicate that the ranking is arbitrary from the point where they begin.

Figure: (a) Course priorities ($\succ_c$) and (b) student preferences ($\succ_s$)

(a)

$\succ_{c_1}$	$\succ_{c_2}$	$\succ_{c_3}$	$\succ_{c_4}$
s_4	s_2	s_3	$\underline{s_1}$
s_1	$\underline{s_3}$	$\underline{s_4}$	$\vdots$
$\underline{s_2}$	$\vdots$	$\vdots$	$\vdots$

(b)

$\succ_{s_1}$	$\succ_{s_2}$	$\succ_{s_3}$	$\succ_{s_4}$
c_1	$\underline{c_1}$	$\underline{c_2}$	$\underline{c_3}$
$\underline{c_4}$	c_2	c_3	c_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- Consider the permutation

$$\phi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}$$

  which could be the result of a lottery.
- This would lead to the following ordering of the students: (s_2, s_1, s_3, s_4).
- Then the mechanism would work as described in the table.

- The matching

$$\mu_1 = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 \\ s_2 & s_3 & s_4 & s_1 \end{pmatrix}$$

would be the result.

- The table entries corresponding to this result are underlined in the table.

- The SD mechanism is Pareto efficient for students with strict preferences, because no student can be made better off without hurting another student.
- One disadvantage is that course-specific priorities are not considered.
- Furthermore, the SD mechanism is not stable, as the following example shows.

Figure: Example of the serial dictatorship mechanism. The boxes describe an assignment

Step	Student to be assigned	c_1	c_2	c_3	c_4
1	s_2	$\boxed{s_2}$			
2	s_1	$\vdots$			$\boxed{s_1}$
3	s_3	$\vdots$	$\boxed{s_3}$		$\vdots$
4	s_4	$\vdots$	$\vdots$	$\boxed{s_4}$	$\vdots$

- Student s_1 and course c_1 are unmatched, in that student s_1 would prefer course c_1 to her current assignment (c_4) and she has higher priority to c_1 than student s_2, who is actually assigned to course c_1.
- Assuming $s = s_1$ and $s' = s_2$ we obtain

$$c_1 = \mu(s_2) = \mu(s') \succ_{s_1} \mu(s) = \mu(s_1) = c_4$$

and

$$s_2 = s' \prec_{c_1} s = s_1$$

which contradicts the definition of stability.

The Gale-Shapley Student-Optimal Stable Mechanism

- Each student proposes her first-choice course. For each course c with a capacity q_c, those q_c proposers who have the highest priority for c are tentatively assigned to c; the remaining proposers are rejected.
- In general, at step $k, k \geq 2:$ Each student who was rejected in the previous step, $k-1$ proposes her next-choice course. For each course c, from the new proposers and those who were tentatively assigned at a previous step, the q_c with the highest prisrity are tentatiyely assigned to c : the rest are rejected.
- **The algorithm terminates when there are no further rejections of students or proposals.**

Example

- Consider the problem

Figure: (a) Course priorities $(\succ_c)$ and (b) student preferences $(\succ_s)$

(a)

$\succ_{c_1}$	$\succ_{c_2}$	$\succ_{c_3}$	$\succ_{c_4}$
$\underline{s_4}$	$\underline{s_2}$	$\underline{s_3}$	$\underline{s_1}$
s_1	s_3	s_4	$\vdots$
s_2	$\vdots$	$\vdots$	$\vdots$

(b)

$\succ_{s_1}$	$\succ_{s_2}$	$\succ_{s_3}$	$\succ_{s_4}$
c_1	c_1	c_2	c_3
$\underline{c_4}$	$\underline{c_2}$	$\underline{c_3}$	$\underline{c_1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- The resulting matching is

$$\mu_2 = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 \\ s_4 & s_2 & s_3 & s_1 \end{pmatrix}$$

and the corresponding entries are underlined in table.

Properties of the SOSM

- The student-optimal stable mechanism is widespread because it satisfies a number of desirable properties such as strategy-proofness for the students (since in SOSM students do the proposing) and stability.
- In addition, it weakly Pareto-dominates any other stable matching mechanism for students.

Theorem (Gale and Shapley, 1962)

There always exists a stable matching for the two-sided matching problem.

- The above mechanism is used in several public school systems in the USA, and its stability is often used as an explanation for its popularity.

Theorem (Gale and Shapley, 1962)

The SOSM Pareto-dominates any other stable matching mechanism.

- *Proof*. Let us call a course "possible" for a particular student if there is a stable matching that assigns her to this course.
- The proof is by induction. Assume that up to a given point in the algorithm no student has yet been turned away from a course that is possible for her.
- At this point suppose that course organizer c_j, having received proposals from $q = q_{c_j}$ of his preferred students s'_l with $0 < l \leq q_{c_j}$, rejects student s_i.
- We must show that c_j is impossible for s_i.

- We know that each s'_l prefers course c_j to all the others, except for those that have previously rejected her and hence are impossible for her.
- Consider a hypothetical assignment that sends s_i to c_j and everyone else to courses that are possible for them.
- At least one of the s'_l will have to go to a less desirable place than c_j.
- But this assignment is unstable, because s'_l and c_j could both benefit by being assigned to each other. Therefore, c_j is impossible for s_i.
- As a consequence, a stable matching mechanism only rejects students from courses which they could not possibly be assigned to in any stable matching. The resulting matching is therefore Pareto optimal.

- Although no stable matching mechanism can be strategy-proof for all agents, the SOSM is strategy-proof for the proposing side (the students in our examples).

Theorem (Roth, 1982)

No stable matching procedure for the general matching problem exists for which the truthful revelation of preferences is a dominant strategy for all participants.

Proof

- We aim to show that there is a matching problem for which no stable matching procedure has truthful revelation as a dominant strategy.
- Suppose there are students $S = \{s_1, s_2, s_3\}$ and courses $C = \{c_1, c_2, c_3\}$.
- Let μ be an arbitrary stable matching procedure which, for any reported preference profile $\succeq = (\succeq_S, \succeq_C)$ selects some outcome $\chi(\succeq)$ from the set of stable outcomes with respect to the reported preference profile $\succeq$.
- Suppose that the preferences of the participants are given in the next table

Figure: (a) Course priorities ($\succ_c$) and (b) student preferences ($\succ_s$)

(a)

$\succ_{c_1}$	$\succ_{c_2}$	$\succ_{c_3}$
s_1	s_3	s_1
s_3	s_1	s_2
s_2	s_2	s_3

(b)

$\succ_{s_1}$	$\succ_{s_2}$	$\succ_{s_3}$
c_2	c_1	c_1
c_1	c_2	c_2
c_3	c_3	c_3

- There are two stable outcomes, μ_1 and μ_2

$$\mu_1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ c_2 & c_3 & c_1 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ c_1 & c_3 & c_2 \end{pmatrix}$$

- Note that the students prefer μ_1 while the course organizers prefer μ_2. since $\chi(\succeq)$ is a stable matching procedure, one of the two matchings is selected.

- Suppose that c_1 reports $s_1 \succ s_2 \succ s_3$ instead of $s_1 \succ s_3 \succ s_2$.
- The new preference profile is referred to as $\succeq'$.
- Now, the outcome μ_2 is the unique stable outcome with respect to the preference profile $\succeq'$ and any stable matching mechanism can only select $\succeq'$.
- In a similar way, $\succeq''$ is the preference profile which differs from $\succeq$ only in that $\succeq''$ replaces the report by s_1 with $c_2 \succ c_3 \succ c_1$.
- Then the outcome μ_1 is the unique stable outcome with respect to $\succeq''$.

- As a consequence, if in the original problem μ_1 is selected by $\chi(\succeq)$ then c_1 has an incentive to state the preference relation $\succeq'$ instead of the true preference $\succeq$ in order to change the outcome from μ_1 to μ_2.
- If, however, $\chi(\succeq)$ selects μ_2 then s_1 has an incentive to report $\succeq$" to change the outcome from μ_2 to μ_1. $\square$

- The proof also shows that the set of agents favored by a stable matching procedure has no incentive to misreport preferences.
- In the SOSM, no student can be matched with a course that is better than the course assigned.

Theorem (Roth, 1982)

The SOSM is strategy-proof for the students.

- In the literature on school choice, schools are typically assumed to have publicly known priorities but they are assumed not to be strategic, while the relevant students are possibly strategic.
- While these are positive results, SOSM is not without problems.
- There are potential tradeoffs between stability and Pareto efficiency for the students, and tie-breaking is an issue.

Emerson Melo (emelo@iu.edu)