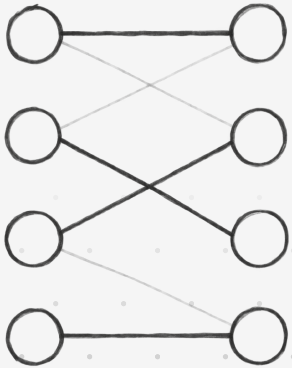
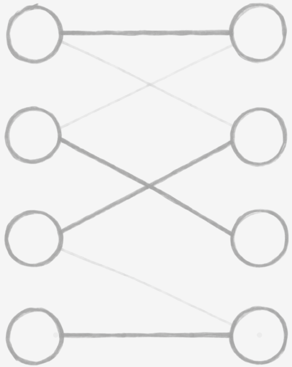


Algorithms ▶ Algorithms and Incentives

Lecture 5

- Allocating Rooms
- Two-Sided Matching
- Properties and Applications



Algorithms ▶ Algorithms and Incentives

Lecture 5

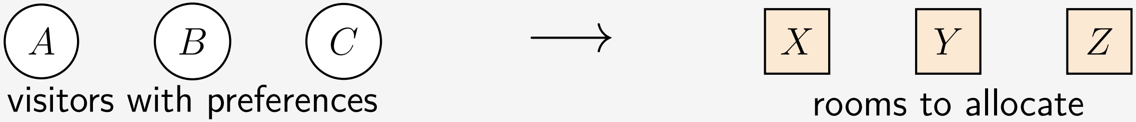
- Allocating Rooms
- Two-Sided Matching
- Properties and Applications

Imagine a summer school with n visitors and n single rooms, all at the same price. Some rooms are quieter; others are closer to classes.

Your job is to design an algorithm to allocate rooms to visitors, in a “reasonable” way

Problem

Each visitor strictly ranks all n rooms. Assign one room to each visitor, using each room once. Everyone prefers any room to no room.



Remark

What properties would you want from the assignment procedure?

Property 1: Pareto Optimality

Definition: Pareto improvement

An alternative assignment that makes someone strictly better off and makes nobody worse off.

Definition: Pareto optimality

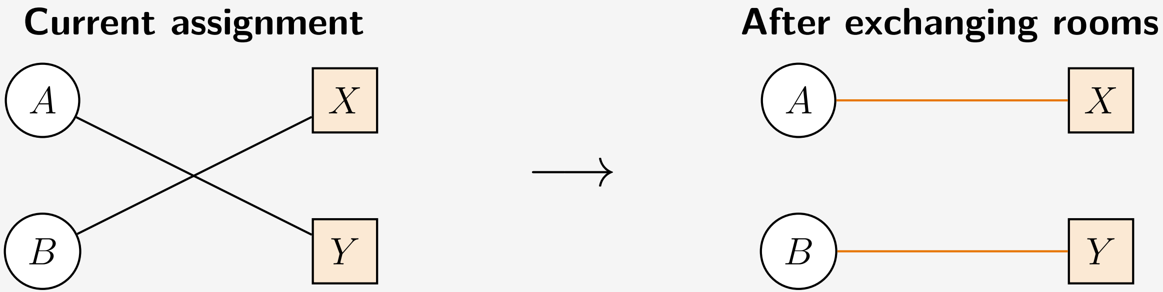
An assignment is Pareto optimal if no Pareto improvement exists.

Remark

A useful minimum requirement: if someone can benefit without hurting anyone else, improve the assignment. This alone does not guarantee fairness.

An Improvement Everyone Accepts

A prefers X to Y; B prefers Y to X. All other visitors keep their rooms.



Both A and B improve, and nobody else is affected. The original assignment was not Pareto optimal.

Property 2: Strategyproofness

Each visitor has a **true ranking**, but submits a **reported ranking**. The algorithm sees only the report, which may be a lie.

Incentives are reasons the rules give participants to choose one action over another. Could lying get you a better room?

$$\text{True: } X \succ Y \succ Z \qquad \longrightarrow \qquad \text{Report: } Y \succ X \succ Z\,?$$

Definition: Strategyproofness (truthfulness)

Fix everyone else's reports. Reporting your true ranking gives an outcome at least as good for you as any alternative report, judged by your true ranking.

Remark

Can we get both Pareto optimality (with truthful reports) and strategyproofness?

Algorithm: Serial Dictatorship

Algorithm: Serial dictatorship

1. Fix an order of the visitors, **independently of their reports**.
2. In that order, give each visitor its highest-ranked room among those still available.

The order could be chosen by a lottery. Our claims hold for every fixed order.

A Serial Dictatorship in Action

Visitor	Rooms, best to worst
A	$X \succ Y \succ Z$
B	$Y \succ X \succ Z$
C	$Y \succ Z \succ X$

For order A, B, C:

A chooses X from $\{X, Y, Z\}$ → B chooses Y from $\{Y, Z\}$ → C receives Z the remaining room

Remark

C would prefer Y, but giving it to C would hurt B. A different order can produce a different assignment.

Serial Dictatorship Is Pareto Optimal

Recall: Pareto optimal means no alternative makes someone better off without making anyone worse off.

Theorem:

With truthful rankings, serial dictatorship produces a Pareto optimal assignment.

Proof

Suppose another assignment is a Pareto improvement. Let i be the **first visitor in the order** whose room changes.

Earlier visitors keep their rooms. Therefore the new room for i was available when i chose in serial dictatorship.

But i received its favorite available room. Since preferences are strict, any different available room makes i worse off: a contradiction.

Serial Dictatorship Is Strategyproof

Recall: Strategyproof means truthful reporting is at least as good as any lie, for every fixed set of others' reports, judged by your true preferences.

Theorem:

Serial dictatorship is strategyproof when its order is independent of the reports.

Proof

Fix visitor i and everyone else's reports. Changing i's report changes neither its position in the order nor any earlier visitor's choice.

Thus the set of available rooms when i chooses is independent of its report. Truthful reporting gives i its favorite room in that set. No alternative report can give it a better one.

Definition: Mechanism

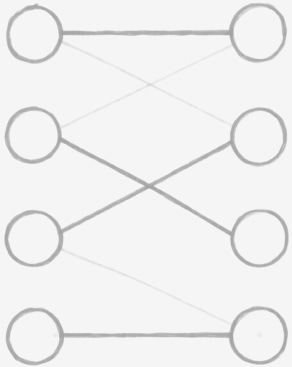
Definition: Mechanism

A procedure that chooses an outcome as a function of participants' reported preferences.

Reported rankings → Serial dictatorship → Room assignment

- **Pareto optimality** is a property of the outcome.
- **Strategyproofness** is a property of the procedure.
- The rules create **incentives** by linking reports to outcomes.

So far, only the visitors have preferences. What if the other side has preferences too?



Algorithms ▸ Algorithms and Incentives

Lecture 5

- Allocating Rooms
- Two-Sided Matching
- Properties and Applications

Definition: Two-Sided Matching

Problem

There are two groups of n participants each. Every participant strictly ranks all members of the other group. Pair everyone with exactly one participant on the other side.

Definition: Two-sided market

Two distinct groups of participants, each with their preferences.

The Model: Students and Hospitals

Medical students rank residency programs; hospitals rank students. The National Resident Matching Program coordinates these assignments.

- Write S for the **student side**, H for the **hospital side**.
- Our model has $|S| = |H| = n$ and one position per hospital. Real residency programs can have several positions.

Definition: Perfect matching

A set $M \subseteq S \times H$ in which every student and hospital occurs once. Write $M(s)$ for student s's hospital and $M(h)$ for hospital h's student.

Is Pareto optimality enough? Here we count **both sides** when asking whether anyone becomes worse off.

Example: Is Pareto Optimality Enough?

Student	Ranking	Hospital	Ranking
A	$Y \succ X$	X	$A \succ B$
B	$Y \succ X$	Y	$A \succ B$

The only other perfect matching hurts B (and X), so this one is Pareto optimal.

Definitions: Blocking Pairs and Stability

For hospitals h, h', write $h \succ_s h'$ when student s prefers hospital h to hospital h'

Definition: Blocking pair

A pair $(s, h) \notin M$ blocks M if

$$h \succ_s M(s) \qquad \text{and} \qquad s \succ_h M(h).$$

Each prefers the other to its current partner.

Definition: Stable matching

A perfect matching with no blocking pair.

Remark

Does a stable matching always exist? How could we find one?

Algorithm: Deferred Acceptance

Algorithm: Student-proposing deferred acceptance (DA)

Initially everyone is unmatched.

1. While an unmatched student has an untried hospital:
   - The student proposes to its favorite untried hospital.
   - The hospital holds its favorite of the new proposer and its current student (if any), rejecting the other.
2. Make all held pairs final.

Acceptances are **tentative**. A hospital can replace its held student; a rejected student continues down its list.

Any eligible student can propose next. The algorithm can run in $O(n^2)$ time.

Example: Preferences for Deferred Acceptance

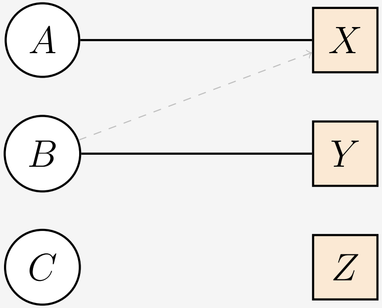
Students		proposals $\longrightarrow$	Hospitals	
A	$X \succ Y \succ Z$		X	$C \succ A \succ B$
B	$X \succ Y \succ Z$		Y	$C \succ B \succ A$
C	$Y \succ X \succ Z$		Z	$A \succ B \succ C$

Remark

Let A propose first, then B, then C. What happens when C proposes to Y?

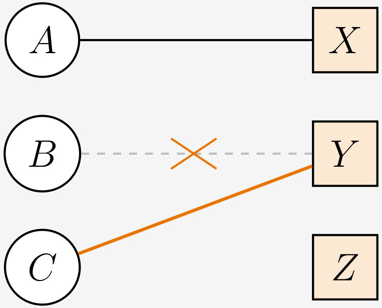
Example: A Run of Deferred Acceptance

1. A, then B propose



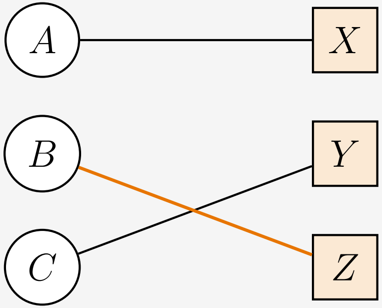
X holds A, rejects B.
B then goes to Y.

2. C proposes to Y



Y prefers C to B.
B becomes unmatched.

3. B tries Z



Z holds B.
All students are matched.

Solid lines: held pairs. Dashed lines: rejected proposals.

DA Produces a Perfect Matching

Recall: A perfect matching pairs every student with exactly one hospital and every hospital with exactly one student.

Lemma:

Deferred acceptance always terminates with a perfect matching.

Proof

Termination: no student proposes to the same hospital twice, so the process ends. Held pairs always form a matching.

Completeness: once a hospital holds a student, it stays occupied: it releases a student only to take another.

If a student were unmatched at termination, it would have proposed to all hospitals. Every hospital would then hold a distinct student. That accounts for all n students, a contradiction.

It remains to show that no pair blocks the final matching.

DA Produces a Stable Matching

Recall: A matching is stable if no student–hospital pair would both prefer each other to their assigned partners.

Theorem:

Deferred acceptance always returns a stable perfect matching.

Proof

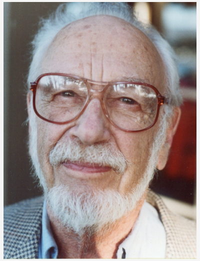
Let M be the final matching. Suppose student s prefers hospital h to $M(s)$.

Student s must have proposed to h before proposing to $M(s)$. Hospital h rejected s, immediately or later, in favor of a student it preferred.

A hospital's held student only improves. Hence h prefers its final student $M(h)$ to s. So (s, h) cannot be a blocking pair.

The algorithm also proves that a stable matching always exists in this model.

From Stable Matchings to Market Design



David Gale

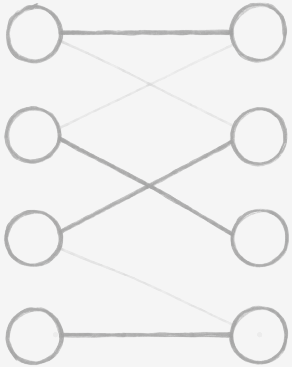
Lloyd Shapley

Alvin E. Roth

- **Gale and Shapley (1962):** deferred acceptance and stable matchings.
- **Roth:** evidence on stability and the design of practical matching markets.

Shapley and Roth shared the 2012 Nobel Memorial Prize in Economics for the theory of stable allocations and the practice of market design.

Gale died in 2008; he was not a recipient of the prize.



Algorithms ▶ Algorithms and Incentives

Lecture 5

- Allocating Rooms
- Two-Sided Matching
- Properties and Applications

Stability Implies Pareto Optimality

Theorem:

With strict preferences, every stable matching is Pareto optimal when we count **all students and hospitals**.

The converse is false: our two-student example was Pareto optimal but unstable.

Whose welfare are we measuring? A stable matching can still admit an improvement for students alone that makes some hospitals worse off.

Even student-proposing DA need not be Pareto optimal for students among *all* matchings.

Proposer Optimality

Theorem:

For fixed strict preference lists, student-proposing DA simultaneously gives:

- each student its **best** partner in any stable matching;
- each hospital its **worst** partner in any stable matching.

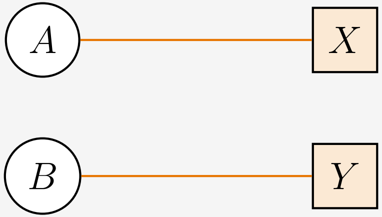
The comparison is only over **stable matchings for those same lists**.

Reversing the proposing side reverses these guarantees. Who proposes is a consequential choice in the rules.

Two Stable Outcomes

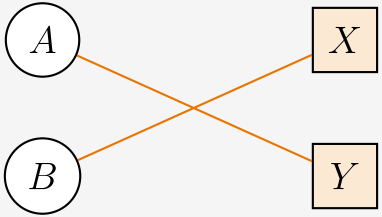
Student	Ranking	Hospital	Ranking
A	$X \succ Y$	X	$B \succ A$
B	$Y \succ X$	Y	$A \succ B$

Students propose



students get first choices

Hospitals propose



hospitals get first choices

Both are stable. Each side prefers a different one.

Strategyproofness of DA

Students and hospitals submit rankings; DA uses these reports, not their private true preferences.

Theorem:

In student-proposing DA:

- For each student, no lie improves its assignment under its true preferences, whatever everyone else reports.
- A hospital can sometimes obtain a better student by misreporting.

Remark

The two sides face different incentives: students cannot gain by lying; hospitals sometimes can.

From Residencies to College Admissions

Residency matching	**College admissions**
Students rank hospitals	Applicants rank colleges
Hospitals rank students	Colleges rank applicants
Hospitals have positions	Colleges have places

Assume each college has a fixed ranking of applicants and always wants its highest-ranked applicants up to its capacity.

- Replace a college with q places by q separate slots, all with the college's ranking.
- Applicants replace each college in their ranking by its slots, consecutively in the order $1, \ldots, q$.

Run one-to-one DA and merge each college's slots back together. Equivalently, each college holds its favorite q applicants so far.

A Changed Model: At Most B Applications

Problem

Keep n students and n colleges, one place per college. Each student may apply to at most B colleges. Run student-proposing DA, allowing a student to propose only to colleges it applied to.

- Full underlying preferences still exist.
- Students must choose **which schools to list**, as well as their order.
- An unmatched student stops once its submitted list is exhausted.

Remark

Which guarantees survive: a complete assignment, stability, Pareto optimality, strategyproofness?

Revisiting the Stability Proof

With unmatched participants, stability still means no blocking pair. Everyone prefers any partner to remaining unmatched.

The original proof used this implication:

If s prefers h to its final assignment, then s must have proposed to h.

Remark

Where did we use complete preference lists? Does this implication still hold when the submitted list has length at most B?

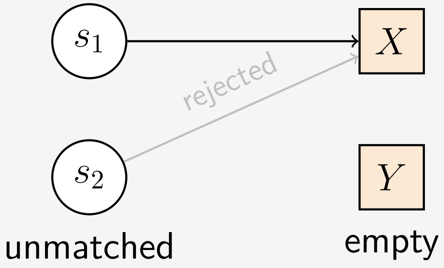
A preferred college may be **missing from the list**. Then s never proposed, so we cannot conclude that h rejected s for someone better.

The proof still works for colleges on the submitted list. It no longer rules out blocking pairs involving an omitted college.

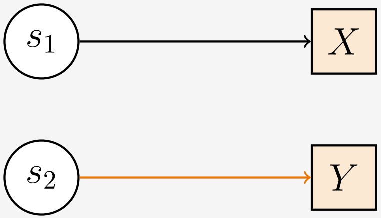
A Safety School Can Matter

Take $B = 1$, with one place per college. Both students prefer X to Y; both colleges prefer s_1 to s_2. Any match is better than no match.

Both list their favorite



s_2 lists Y instead



Student s_2 never proposed to Y, even though both would prefer to match. The application budget gives s_2 an incentive to list the safety school Y instead of its favorite X.

Which Guarantees Survive?

Fix the applications and assume truthful rankings on the submitted lists.

Property	**With at most B applications**
Complete assignment	Not guaranteed, even with enough total places.
Stability	Holds among applied-to pairs; an unlisted pair may block.
Pareto optimality	Holds for both sides over assignments using only applied-to pairs; may fail in the full market.
Strategyproofness	Truthful ordering of a fixed application set is safe; choosing that set is strategic.

The previous example leaves s_2 unmatched and Y empty. Allowing the omitted pair improves both without hurting anyone.

Lessons: Small Changes to the Model

A small change in the definition of a model can change its properties. Revisit the proof to find exactly which assumption did the work.

- With complete lists, a preferred hospital must have received a proposal.
- With a budget, it may never have been considered.
- Stability in the full market and truthful top-B reporting can both fail.

Remark

When a problem definition changes, which steps of your old proof survive?