MATH4994 — Capstone Projects in Mathematics and Economics

Topic Two: Matching schemes

2.1 Marriage problems

- Deferred acceptable algorithm
- Stable solution

2.2 College admission and school choice problems

- Gale-Shapley student optimal stable mechanism
- Top trading cycles mechanism
- Boston school choice mechanism
- Chinese parallel mechanism

2.3 House allocation with existing tenants

2.4 Kidney exchange

2.5 Roommate problems and Irving algorithm

2.1 Marriage problems

2012 Nobel Awards in Economics go to Shapley and Roth on their works on matching schemes

- Each man has *strict* preferences over the women. Each woman has *strict* preferences over the men.

A *matching* is a bijection (one-to-one correspondence) M between m and w. There may be some men or women left unattached when the numbers of men and women are not the same.

A man (woman) prefers the matching scheme M to another matching scheme M' if he (she) prefers the partner he (she) is matched to in M to the one he (she) is matched to in M'.

Desirable property: stability of a matching

A man-woman pair (not one of the pairs in the matching scheme M) *blocks* M if they prefer each other to their spouses under M. A matching M is *stable* if there is no man-woman pair blocking M.

Example

1	2	4	1	3	1	2	1	4	3
2	3	1	4	2	2	4	3	1	2
3	2	3	1	4	3	1	4	3	2
4	4	1	3	2	4	2	1	4	3

Men's Preferences Women's Preferences

The matching $\{(\mathbf{1},\mathbf{4}),\ (2,3),\ (\mathbf{3},\mathbf{2}),\ (4,1)\}$ is stable. Stability may be verified by considering each man in turn as a potential member of a blocking pair.

Man 1 could form a blocking pair only with woman 2, but she prefers her partner, man 3 to man 1 under this matching. Each of men 2 and 3 is matched with his favorite woman, so neither can be in a blocking pair. Finally, man 4 could form a blocking pair only with woman 4, but she would rather stick with her partner, man 1.

This matching favors men but not women, a property of the man-oriented stable matching. In the reverse sense, each woman has the worst partner that she can have in any stable matching.

- A second example of a stable matching, indeed the only other stable matching in this case, is $\{(\mathbf{1}, \mathbf{4}), (2, 1), (\mathbf{3}, \mathbf{2}), (4, 3)\}$, as may be verified in a similar way.

- The matching $\{(1, 1), (2, 3), (3, 2), (4, 4)\}$, for example, is unstable because of the blocking pair $(1, 4)$; man 1 prefers woman 4 to his partner, woman 1, and woman 4 prefers man 1 to her partner, man 4.

- Some other unstable matchings may have many more blocking pairs: for example, the matching $\{(1, 1), (2, 2), (3, 4), (4, 3)\}$ has six.

Stable pair and fixed pair

A man m and a woman w constitute a *stable pair* if and only if m and w are partners in some stable matching. In these circumstances, m is a *stable partner* of w, and vice versa. If some man m and woman w are partners in *all* stable matchings, then (m, w) is called a *fixed pair*. In the above example, since there are only two stable matching solutions, $(1, 4)$ and $(3, 2)$ are seen to be fixed pairs.

Deferred acceptance algorithm

Gale, D. and Shapley, L., "College admissions and stability of marriage", *American Mathematical Monthly*, vol. 69, p.9-15 (1962).

Man-oriented version

1. Each man proposes to his favorite woman.

2. Each woman "engages" her favorite man among her proposers and rejects the others. The woman holds the most preferred man at this stage without commitment and the acceptance of a more preferred man can be made in later rounds of proposals.

3. Each rejected man proposes to his next favorite woman.

4. Repeat steps 2 and 3 until all women have been proposed to. That is, there are all rejected men have exhausted their own individual lists.

For a given man m, we writer $P_M(m)$ as the partner (woman) assigned to m under the matching scheme M.

Example

Boys' Preferences			
Adam	**Bob**	**Charlie**	**Don**
Mary	Jane	Mary	Mary
Jane	Mary	Kate	Kate
Kate	Kate	Jane	Jane

Girls' Preferences		
Mary	**Jane**	**Kate**
Adam	Adam	Don
Bob	Charlie	Charlie
Charlie	Don	Bob
Don	Bob	Adam

Each boy approaches the first girl on their lists.

In the first round, Mary gets three offers, and holds on to Adam, while rejecting the other two. Jane receives one offer, from Bob, so she asks him to wait. Kate receives no offers, so at the end of the day she has nobody. A girl does not commit to anyone, but may improve her choice from later male proposers.

	Round 1	Round 2	Round 3	Round 4	Round 5
Mary	**Adam** (Charlie & Don rejected)	**Adam** (no new proposal)	**Adam** (no new proposal)	**Adam** (Bob rejected)	**Adam** (no new proposal)
Kate	No proposal	**Don** (Charlie rejected)	**Don** (no new proposal)	**Don** (no new proposal)	**Don** (Bob rejected)
Jane	**Bob**	**Bob** (no new proposal)	**Charlie** (Bob rejected)	**Charlie** (no new proposal)	**Charlie** (no new proposal)

In the second round, the two boys who are not being held onto approach the second girls on their lists. In this case, Kate receives both proposals, holds onto Don and sends Charlie away. Since nobody proposed to Mary or Jane, they hold onto their boys from the first round.

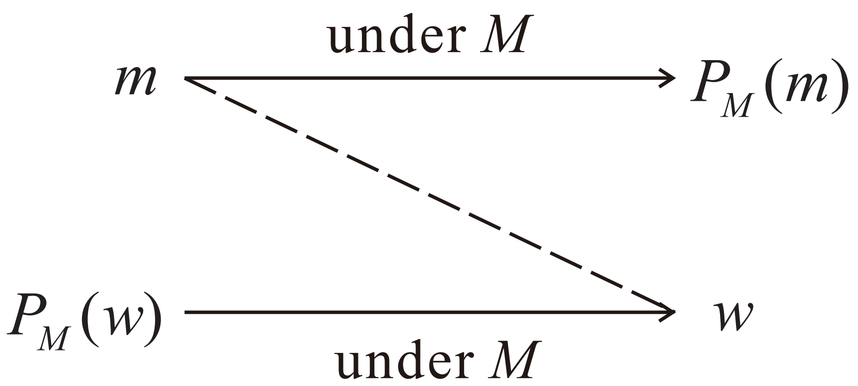
In the third round, Charlie (the only boy not currently held by a girl), asks Jane. Since Jane ranks Charlie ahead of Bob (who she's held since the first round), she releases Bob, and holds onto Charlie.

In the fourth round, Bob asks Mary, but is rejected by her since she ranks Adam higher. In the fifth round Bob asks his last choice, Kate, but she rejects him as well, since she ranks Don higher.

All three girls have boys on hold, and the one unattached boy has been rejected by all three girls. The process is done. Adam and Mary end up together, as do Don and Kate, and Charlie and Jane. Bob ends up alone. Poor Bob, he cannot hold on Jane even Jane is less popular among the other 3 boys.

Matching under man-oriented scheme M is stable (non-existence of a blocking pair)

For any m, we show that one cannot find w such that (m, w) blocks M. As a result, the scheme is stable. Assume contrary, there exists m such that $w \underset{m}{\succ} P_M(m)$, where m prefers w to the woman (partner) $P_M(m)$ at this matching scheme M. By the procedure of the man-oriented scheme, m must have proposed to w earlier under M before being assigned to $P_M(m)$, but m must be rejected by w before matching with $P_M(m)$. This gives $P_M(w) \underset{w}{\succ} m$, so (m, w) cannot form a blocking pair. A contradiction is encountered.



Both $(m, P_M(m))$ and $(P_M(w), w)$ are under M. Though w is more preferred by m, however, w prefers $P_M(w)$ more than m.

In Shapley's example, (Don, Kate) and (Adam, Mary) are paired under M, but (Don, Mary) cannot form a blocking pair. Note that Don has been rejected by Mary since Mary prefers Adam over Don. In the man-oriented matching solution, Don is paired with his second choice Kate.

Set of achievable partners

There may be multiple stable matching solutions. We say that m and w are achievable partner to each other if they are paired in a stable matching. In analyzing optimality of the choice of partner under stable matching, we only need to consider the set of achievable partners of individual players.

In the first example, the set of achievable partners of man 2 is $\{$woman 1, woman 3$\}$. The set of achievable partners of man 3 is singleton $\{$woman 2$\}$ since $(3, 2)$ is a fixed pair.

M-optimality of man-oriented scheme M_0

Every man gets his best achievable partner under the man-oriented scheme M_0.

Remark It suffices to show that no man is ever rejected by an achievable woman. By the man-oriented procedure, since each man proposes to women in sequential preference order starting from the best choice to the less preferred choice, so every man gets his best achievable partner.

Proof
To construct the proof, we use an induction argument and assume that up to certain step in the procedure, no man has yet been rejected by a woman in his set of achievable partners. In the next step of the procedure, suppose a man m is rejected by a woman w in his set of achievable partners, we try to establish a contradiction. In other words, rejection by an achievable woman would not occur. The proof is done.

Let A_m denote the set of achievable partners of man m and $w \in A_m$.

1. At the first occurrence of rejection by a woman in the achievable set, suppose w rejects m; and currently, w engages another man m'. This mean $m \underset{w}{\prec} m'$.

2. Since $w \in A_m$, it is guaranteed that there exists a stable matching M^* such that m is paired with w while another woman w' is paired with m' in this M^*-stable matching solution (since m has been paired with w).

The final step is to show that (m', w) blocks M^*, where $m' \underset{w}{\succ} m$ and $w \underset{m'}{\succ} w'$, so that M^* is not stable. That is, m' and w prefer each other more than the assigned pairings: (m, w) and (m', w'). This leads to a contradiction.

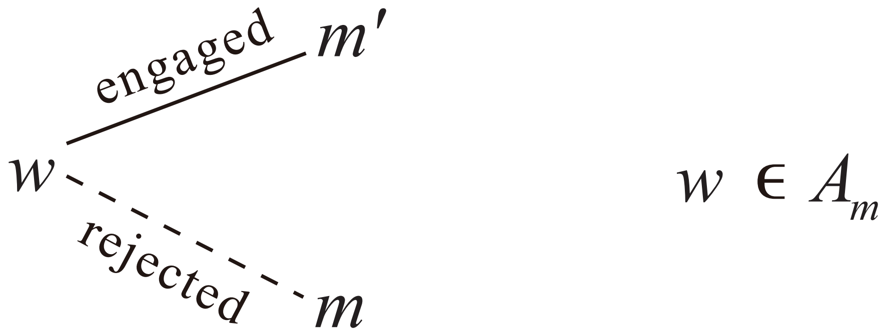
(a) w prefers m' to m as stated in Point (1);

(b) m' prefers w to w' (hard).

Recall that m' engages with w currently. According to the procedure of the man-oriented scheme, up to the current step, m' *prefers* w *to any woman except for those who have previously rejected* m'. Based on the induction assumption, no man has been rejected by a woman in his set of achievable partners. In other words, those women who have rejected m' are not in his set of achievable partners.

We argue that m' prefers w to any other woman in $A_{m'}$. This is because none of the women that have rejected m' are in $A_{m'}$ and $w' \in A_{m'}$. That is, w' cannot be one of those who have previously rejected m', so $w' \underset{m'}{\prec} w$.

Summary: Story upon the first occurrence of rejection of a man by an achievable woman



Suppose there exists another stable scheme M^* such that

$$m \leftrightarrow w \quad \text{and} \quad m' \leftrightarrow w', \text{ where } w' \in A_{m'}.$$

We would like to show that (m', w) blocks M^*, so a contradiction is encountered. Based on the observations:

- m' prefers w to any woman except for those who have rejected m',
- None of the women that have rejected m' are in $A_{m'}$ (deduced from the assumption in the induction argument);

We deduce that m' prefers w to w'. This is because w' could not have rejected m' earlier since $w' \in A_{m'}$.

Uniqueness of M-optimal stable solution under the man-oriented scheme

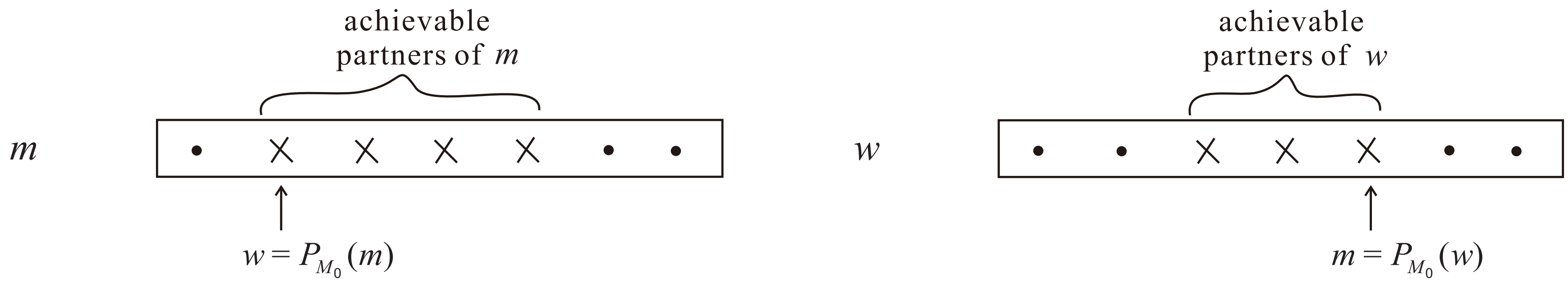
There can only be one M-optimal stable matching (independent of the order in which the men proposing). Suppose the man-oriented scheme gives non-unique solutions, where m is assigned to w_1 and w_2 in two different stable solutions. Either one of w_1 and w_2 is more preferred by m based on strict preferences. By virtue of M-optimality and strict preferences of choices, the pair where m is matched with less preferred woman cannot appear in the matching solution under the man-oriented scheme. Therefore, we observe uniqueness of M-optimal stable solution under the man-oriented scheme.

Termination after finite number of steps

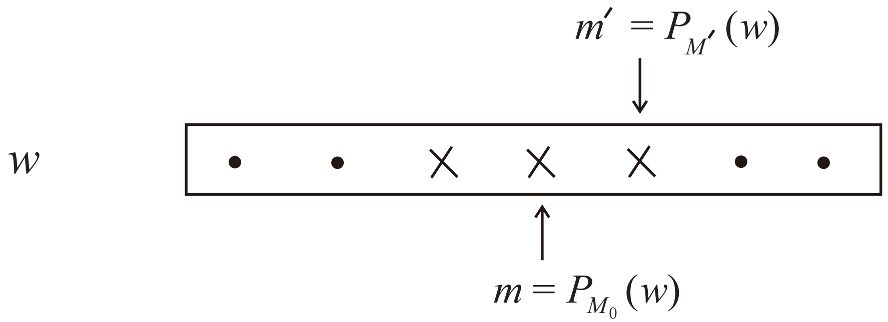
Since no boy proposes to any girl after she has rejected him, this algorithm will reach a stable solution in a finite number of steps. In the above example, it takes five rounds.

Theorem In the man-optimal stable matching, each woman has the worst partner that she can have in any stable matching.

Graphical interpretation of the theorem



Assume contrary, suppose $P_{M_0}(w)$ is not the worst partner within the set of achievable partners of w, it is guaranteed to find an achievable partner m' (behind m) under a stable matching M' such that $m' = P_{M'}(w)$ and m is paired with another woman w'.



We argue that (m, w) blocks M', so M' cannot be a stable matching:

(i) m is a better partner of w than $m' = P_{M'}(w)$;

(ii) Recall that w is the best achievable partner of m under M_0. Therefore, w is a better partner of m than $w' = P_{M'}(m)$ since both w and $w' \in A_m$.

Degenerate case of single stable matching

It can happen that the man-oriented and woman-oriented versions of the deferred acceptance algorithm yield the same stable matching.

Under this scenario, for all men and women, the set of achievable partners becomes a singleton. This is because the best achievable partner and worst achievable partner coincide in each achievable set of partners. We have only one stable solution.

When there is only one stable matching, there is no issue of distinguishing "best" or "worst" partners among various stable solutions.

Reference

Roth, A.E. and Sotomayor, M., *Two-sided matching: A study in game-theoretic modeling and analysis*, Cambridge University Press, United Kingdom (1991).

2.2 College admission and school choice problems

College admission is an extension of the marriage problem if we assume that males can have several wives, as each degree program can admit multiple students.

Absence of justified envy among students

There is no unmatched student-school pair (i, s), where student i prefers school s to her assignment and she has higher priority than some other student who is assigned a seat at school s (equivalently, school s accepts a student of lower priority).

The existence of such pair (i, s) indicates envyness of i against the inferior student admitted and this would form a blocking pair. It is desirable for the college admission matching solution to observe envyfreeness.

Example

Mary is admitted into MATH but finds that Tom is admitted into MAEC (a program more preferred by Mary) while Tom has a lower priority as ranked by MAEC. Definitely, Mary envies Tom. At the same time, MAEC admits a weaker student (Tom) rather than Mary. (Mary, MAEC) (though not a matched pair under the matching scheme) forms a blocking pair with regard to the matched pairs (Tom, MAEC) and (Mary, MATH). Absence of justified envyness prevails if no such blocking pair occurs.

Pareto efficiency

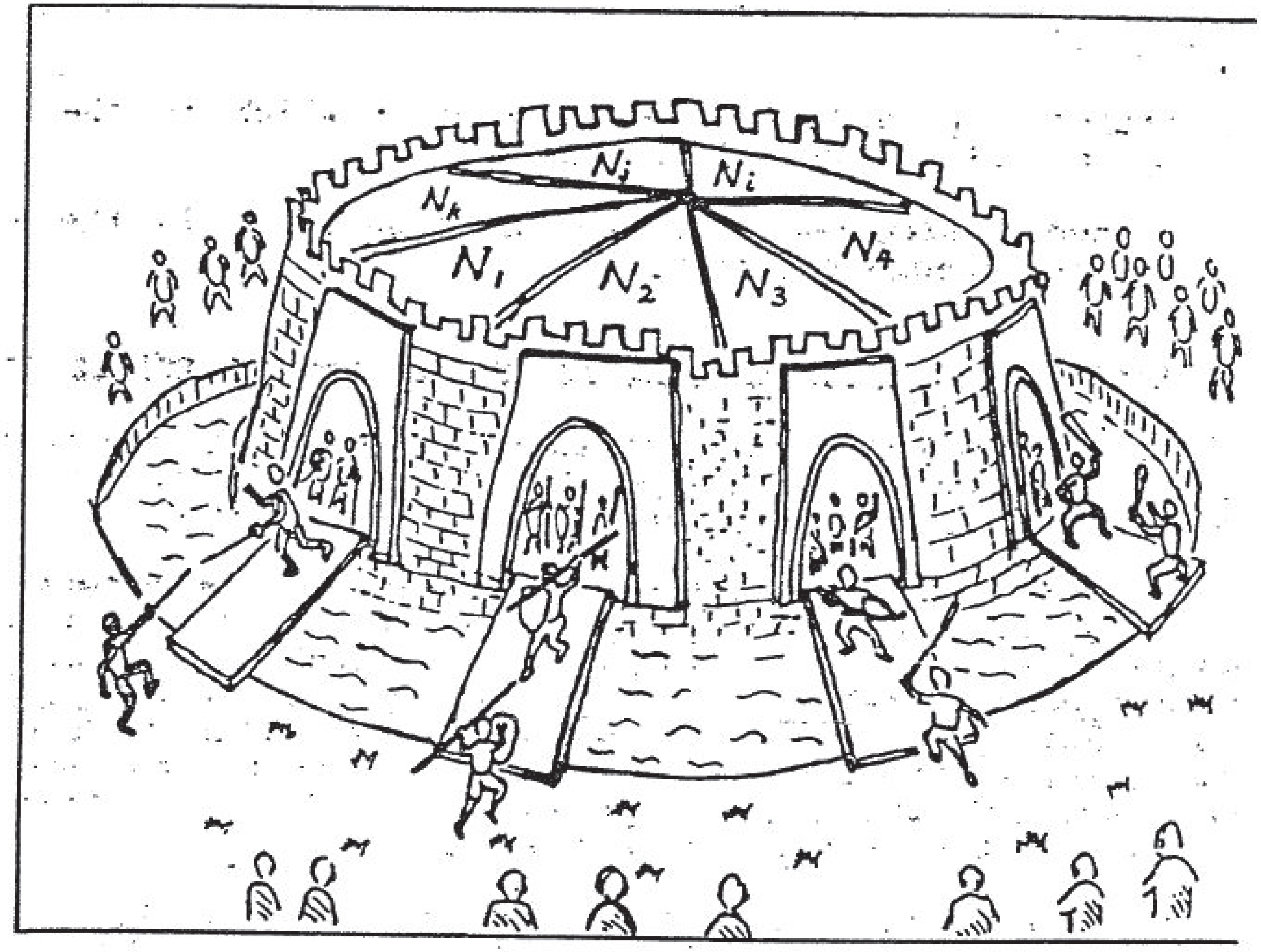
A matching μ is Pareto efficient if it is not possible to improve the allocation of a student without making another student worst off. A Pareto improvement is another allocation that all students are at least as good and one student is strictly better. Pareto efficiency means nonexistence of Pareto improvement.

Equivalently, there is no other matching μ' such that

(i) all students weakly prefer μ' to μ (all students are either indifferent between μ' and μ or prefer μ' to μ);

(ii) there is at least one student who strictly prefer μ' to μ, meaning at least one student who is not assigned to the same school under μ' and μ and prefers the school she is assigned to under μ'.

Student optimal approach: Candidates fight to attach to a program of higher preference



A solider tries to break into the castle through one of the gangways (in order of his preference). When a compartment in full, then the intruder will fight with the weakest person admitted in that compartment.

- The intruder is victorious by kicking out the weakest occupant.
- The intruder is defeated by the weakest occupant, he will try the next preferred gangway (until having exhausted all his choices).

Each student can be admitted at most into one program. He tries to attach to one of his dreamed programs in his order of preferences. However, he may be knocked out by a stronger competitor which is ranked higher than him by the program. Alternatively, he can knock out the weakest (in terms of ranking by the program) who has been tentatively attached to the program when his ranking in that program is higher than that of this weakest student. Equivalently, the attachment to a program by a student is tentative until all students have exhausted their choices and he is not knocked out by a stronger student in all later rounds.

Example – Student optimal stable mechanism

The inputs are (i) students' preferences, (ii) schools' priorities, (iii) school capacities. The priorities of the schools and the preferences of the students are as follows:

$$\begin{aligned} s_1 &: i_1 - \textcircled{\mathbf{i_2}} - i_3 - i_4 - \textcircled{\mathbf{i_5}} - i_6 - i_7 - i_8 \\ s_2 &: i_3 - i_5 - i_4 - i_8 - i_7 - i_2 - i_1 - i_6 \\ s_3 &: i_5 - i_3 - i_1 - i_7 - i_2 - i_8 - i_6 - i_4 \\ s_4 &: i_6 - i_8 - i_7 - i_4 - i_2 - i_3 - i_5 - i_1 \end{aligned}$$

i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
s_2	$\textcircled{\mathbf{s_1}}$	s_3	s_3	$\textcircled{\mathbf{s_1}}$	s_4	$\textcircled{\mathbf{s_1}}$	$\textcircled{\mathbf{s_1}}$
s_1	s_2	s_2	s_4	s_3	s_1	s_2	s_2
s_3	s_3	s_1	s_1	s_4	s_2	s_3	s_4
s_4	s_4	s_4	s_2	s_2	s_3	s_4	s_3

Number of students that can be admitted:

$$n(s_1) = 2,\ n(s_2) = 2, n(s_3) = 3, n(s_4) = 3.$$

Step 1: Students i_2, i_5, i_7, i_8 propose to school s_1, student i_1 proposes to school s_2, students i_3, i_4 propose to school s_3 and student i_6 proposes to school s_4.

Since $n(s_1) = 2$, school s_1 tentatively assigns its seats to students i_2, i_5 and rejects students i_7, i_8. Since school s_1 is the only school with excess proposals, all other students are tentatively assigned seats at schools that they propose.

Step 2: Having been rejected at Step 1, each of students i_7, i_8 propose to school s_2 which is their next choice. School s_2 considers student i_1 whom it has been holding together with its new proposers i_7, i_8. Since $n(s_2) = 2$, school s_2 tentatively assigns its seats to students i_7 and i_8, and rejects student i_1.

Step 3: Having been rejected at Step 2, student i_1 proposes to school s_1 which is her next choice. School s_1 considers students i_2, i_5 whom it has been holding together with its new proposer i_1. School s_1 tentatively assigns its seats to students i_1, i_2 and rejects student i_5.

Step 4: Having been rejected at Step 3, student i_5 proposes to school s_3 which is her next choice. School s_3 considers students i_3, i_4 whom it has been holding together with its new proposer i_5. Since school s_3 has 3 seats, it tentatively assigns its seats to these students.

Since no student proposal is rejected at Step 4, the algorithm terminates. Each student is assigned her final assignment:

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 \\ s_1 & s_1 & s_3 & s_3 & s_3 & s_4 & s_2 & s_2 \end{pmatrix}.$$

A college A is said to be achievable by a student α if there is a stable assignment that sends α to A. His achievable set C_α is the set of all achievable colleges.

The definition of stability in college admission matching requires "nonwasteful" and "individual rationality", besides envyfreeness. A matching is nonwasteful if a student prefers a program to her matching program, then that program must have filled its quota. A matching is individually rational if no student prefer the no college option to his assignment. Nonwasteful and individual rationality follow from Pareto efficiency.

- Assume failure of nonwastefulness, then an empty seat is available that fails to admit the student who prefers the program more. Pareto improvement exists by allocating the student to that program.

- Suppose some student prefers no college to his assignment, we then take away his assigned college and the vacant seat can improve the assignment of another student. Therefore, Pareto improvement exists.

Student-optimality (Gale and Shapley, 1962)

The student-optimal scheme is optimal for students. That is, every admitted student gets his best achievable college.

Similar to the earlier proof for the marriage problem, it suffices to show that no student is ever rejected by an achievable college. Equivalently, the procedure only rejects students from colleges which they could not be admitted to in any stable assignment (colleges that are outside the achievable set). By the student optimal procedure, each student approaches to colleges in sequential preference order. Therefore, every student gets into his best achievable college.

Key steps in the proof

By induction, suppose there has been no student rejected by an achievable college so far and A has received a full quota of better qualified students $\beta_1, \beta_2, \ldots, \beta_q$ and rejects α. We would like to show that A is not achievable for α. The rejection of any student by an achievable college would never occur.

It suffices to show that $A \notin C_\alpha$, where C_α is the set of achievable colleges of α. Assume contrary, suppose $A \in C_\alpha$ so that α is admitted by A under some stable matching M^* and some other student β_i goes to a less desirable college (call it B) than A.

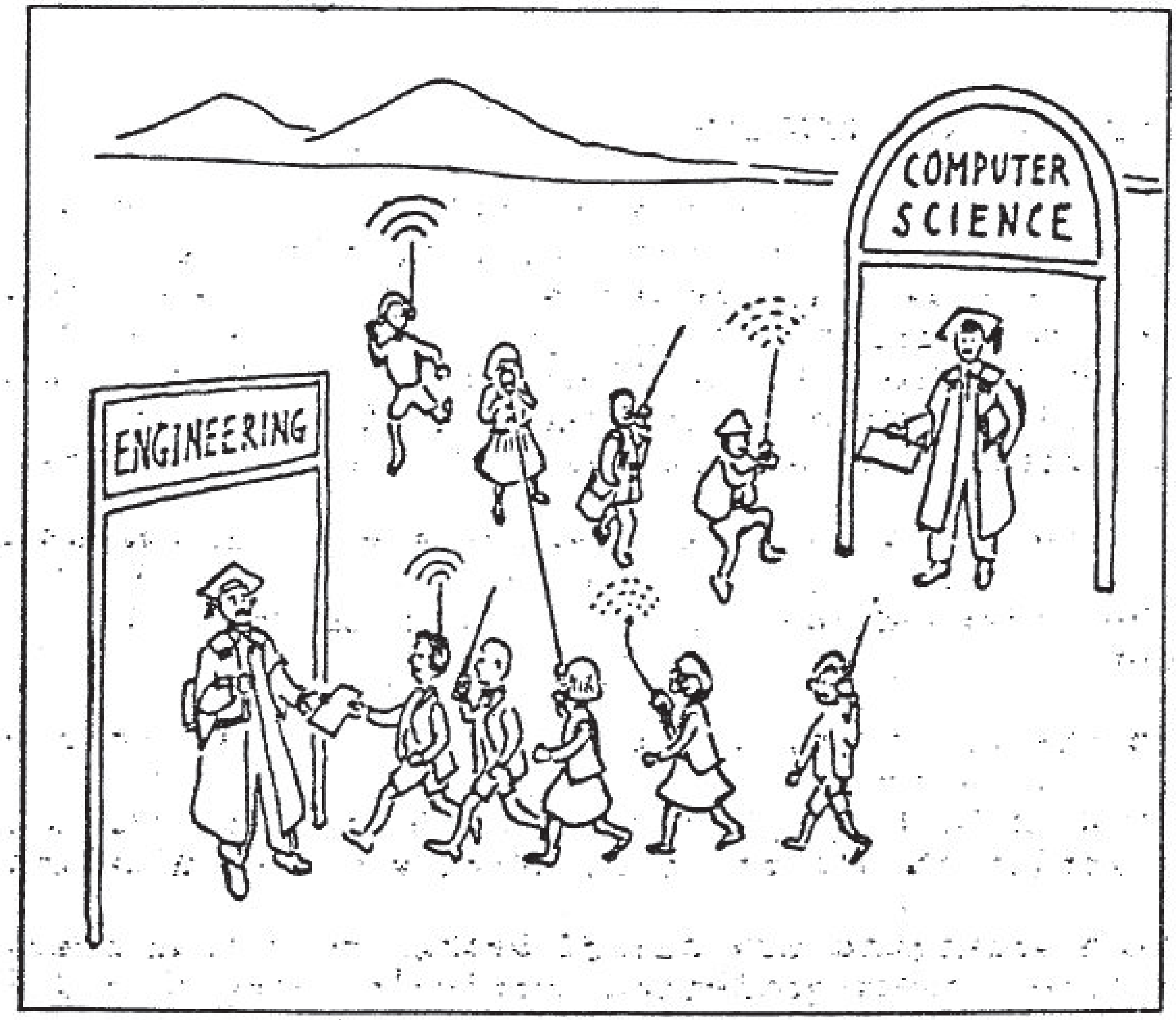
Now, (α, A) and (β_i, B) are paired under M^*. We argue that (β_i, A) blocks M^*. This is seen to be true since

(i) A accepts a student (α in this case) of lower priority since β_i has higher priority than α at college A (at the current step, A keeps β_i instead of α);

(ii) β_i prefers A to any college except those colleges who have previously rejected β_i.

Since a student approaches the colleges in sequential preference order and only be rejected by non-achievable colleges, the student is admitted into the most preferred college among the set of achievable colleges. In other words, every student admitted into a college gets his best achievable college.

Program optimal approach: Each program gives out all K offers to the top K candidates



- Each admissions officer gives out all the K offers to the top K candidates.

- If there are more than one offer is given to the candidate, then the candidate will return the less preferred offers to the admissions officers, who will then give them to the next eligible candidates in the queues. No candidate holds more than one offer. When this is achieved, then those who have an offer in hand are the ones who are selected for admission.

The program tries to keep the best students, and may lose some of them if these stronger students give up the offers and go to the individually more desirable program. In other words, a program cannot get a more eligible student willing to accept its offer to replace the weakest one already accepted.

In this college optimal deferred acceptance scheme, the candidates are passive. They wait for better offers from more desirable colleges to arrive.

Serial dictatorship scheme

Generate a priority ordering (rank list) of students based on either test scores or random drawing. All colleges observe the same priority ordering of students.

Round 1: For the first student in the ranking, assign his first choice. This student and his assignment are removed from the system.

. .

. .

. .

Round k: Only consider the k^{th} student in the ranking. Assign her first choice among the remaining college slots. Again, this student and assignment are removed from the system.

The algorithm terminates when all students have been matched or when no college slot remains.

Strategy-proofness – Individual student cannot gain by gaming around

The best strategy for students is to report their most preferred colleges in sequence. Regardless of the behavior of others, the student simply needs to truthfully reveal preference, which is the dominant strategy. It is impossible to get into more preferred school once they are taken earlier by stronger students.

Elimination of justified envy

By assigning students one after another according to ranking, the central authority will not go to the next student without considering all choices of the higher ranked students. If one student s prefers another college c to her assigned $\mu(s)$, her test score must be lower than the student $\mu^{-1}(c)$ assigned to college c.

Pareto efficiency

The student with the highest test score cannot be better off since he gets his most preferred college. Any change in his assignment would make him worse off, as there is a strict preference.

The student with the second highest score cannot be better off since she gets her most preferred college among college slots. Any change in her assignment would make her worst off. The same argument can be applied to students that are ranked lower.

In summary, a student can be better off only by getting a more preferred school from a stronger student. That student must become worst off.

In reality, the admissions algorithm is complicated by two-step admissions process, with college acceptance first and faculty allocation second. It may happen that a student chooses according to

$$\begin{aligned}\text{engineering at Peking University} &\succ \text{engineering at Tianjin University}\\ &\succ \text{business at Peking University}\end{aligned}$$

Turkey's centralized college optimal multi-category serial dictatorship scheme

In Turkey, the centralized student placement office that assigns students to colleges, in fact to the particular faculties (for example, engineering, medical, dental, business, etc.) of colleges, with no student is assigned to more than one college-faculty.

The standardized examination consists of several component tests, including mathematics, science, verbal aptitude, etc.. Faculties use different combinations of tests to arrive at rankings of the students since different faculties prize different skills. The preference of a student over colleges includes the no-college option, which may be placed ahead of some colleges (maybe none).

Every faculty-type ranks the applying students identically, strictly according to the relevant scores in the category skills. The term college means a college-faculty (e.g. HKUST-Science) so that a college is associated with a particular well defined skill category.

- Let q^t be the quota of colleges in category t. With the ranking induced by the test scores in this category t, assign (only) the colleges in category t to (at most q^t) students using the serial dictatorship applied to the ranking. That is, student with the highest score in category t is assigned his top choice among those colleges in category t, the student with the next highest score is assigned her top choice among the remaining slots in this category, and so on.

- Do the same procedure for all categories.

- Assign the no-college option c_0 to all students who are not assigned to a college. This leads to a tentative assignment since a student may be assigned slots in two or more colleges.

This scheme is college oriented where students are passive, similar to women in man-oriented deferred acceptance scheme.

In the next step, we consider the following two scenarios for each student.

(i) If a student is assigned no slot, then his preference list remains the same.

(ii) If a student is assigned one or more slots, then move the no-college option c_0 directly after the rest of the assigned slots in the previous slot. Therefore, the student is guaranteed to receive the best among the assigned slots or even later in case some better choice becomes available when it is released by other student that has multiple assigned slots.

The rankings of students by the colleges are not changed. The algorithm terminates when no student is assigned more than one slot.

Numerical example

$S = \{s_1, s_2, s_3, s_4, s_5\}$, $C = \{c_1, c_2, c_3\}$, $q = \{q_{c_1}, q_{c_2}, q_{c_3}\} = (2, 1, 1)$, $T = \{t_1, t_2\}$, $t(c_1) = t_1$, $t(c_2) = t(c_3) = t_2$.

$s_1 : \quad c_2 \succ c_1 \succ c_0 \succ c_3, \quad f_1 = (90, 90);$
$s_2 : \quad c_1 \succ c_2 \succ c_3 \succ c_0, \quad f_2 = (80, 60);$
$s_3 : \quad c_1 \succ c_3 \succ c_2 \succ c_0, \quad f_3 = (70, 70);$
$s_4 : \quad c_1 \succ c_2 \succ c_0 \succ c_3, \quad f_4 = (60, 80);$
$s_5 : \quad c_2 \succ c_3 \succ c_1 \succ c_0, \quad f_5 = (50, 50).$

s_1 dislikes c_3 more than c_0 and same for s_4, so c_3 is the least popular college among the students. Note that s_5 is the weakest student.

The academic scores induce the following rankings in t_1 and t_2:

$$t_1 \; : \; s_1 \succ s_2 \succ s_3 \succ s_4 \succ s_5$$
$$t_2 \; : \; s_1 \succ s_4 \succ s_3 \succ s_2 \succ s_5$$

Step 1

$$t_1: \quad \begin{array}{cc} s_1 & s_2 \\ c_1 & c_1 \end{array} \qquad t_2: \quad \begin{array}{ccc} s_1 & s_4 & s_3 \\ c_2 & - & c_3 \end{array}$$

$$\left(\begin{array}{ccccc} s_1 & s_2 & s_3 & s_4 & s_5 \\ c_1, c_2 & c_1 & c_3 & c_0 & c_0 \end{array} \right)$$

Poor s_4, who loses to s_1 in t_2 and both s_2 and s_3 in t_1.

Modified preference lists for students that are assigned at least one slot

$$\begin{array}{ccc} s_1 & : & c_2 \succ c_0 \succ c_1 \succ c_3 \\ s_2 & : & c_1 \succ c_0 \succ c_2 \succ c_3 \\ s_3 & : & c_1 \succ c_3 \succ c_0 \succ c_2 \end{array}$$

The preference lists for s_4 and s_5 (with no assigned slot) remain the same.

Step 2

$$t_1: \quad \begin{matrix} s_1 & s_2 & s_3 \\ - & c_1 & c_1 \end{matrix} \qquad t_2: \quad \begin{matrix} s_1 & s_4 & s_3 \\ c_2 & - & c_3 \end{matrix}$$

$$\begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ c_2 & c_1 & c_1, c_3 & c_0 & c_0 \end{pmatrix}$$

$$\begin{aligned} s_1 &: \quad c_2 \succ c_0 \succ c_1 \succ c_3 \\ s_2 &: \quad c_1 \succ c_0 \succ c_2 \succ c_3 \\ s_3 &: \quad c_1 \succ c_0 \succ c_3 \succ c_2 \\ s_4 &: \quad c_1 \succ c_2 \succ c_0 \succ c_3 \\ s_5 &: \quad c_2 \succ c_3 \succ c_0 \succ c_1 \end{aligned}$$

Step 3

$$t_1: \quad \begin{matrix} s_1 & s_2 & s_3 \\ - & c_1 & c_1 \end{matrix} \qquad t_2: \quad \begin{matrix} s_1 & s_4 & s_3 & s_2 & s_5 \\ c_2 & - & - & - & c_3 \end{matrix}$$

Final matching solution: $\begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ c_2 & c_1 & c_1 & c_0 & c_3 \end{pmatrix}$. Note that s_4 with scores $(60, 80)$ is not assigned due to his dislike of c_3 (ranked below c_0) while s_5 with scores $(50, 50)$ is assigned to the least popular college c_3.

Failure to respect improvement

$S = \{s_1, s_2\}$, $C = \{c_1, c_2\}$, $q = (q_{c_1}, q_{c_2}) = (1, 1)$,
$T = (t_1, t_2)$, $t(c_1) = t_1$, $t(c_2) = t_2$.

$$\begin{array}{lll} s_1 : & c_1 \succ c_2 \succ c_0, & f_1 = (80, 90); \\ s_2 : & c_2 \succ c_1 \succ c_0, & f_2 = (90, 80). \end{array}$$

The outcome under the multi-category serial dictatorship mechanism is

$$\begin{pmatrix} s_1 & s_2 \\ c_2 & c_1 \end{pmatrix}.$$

Both students are assigned to their **second choice** since c_1 prefers s_2 and c_2 prefers s_1. Suppose we modify $\widehat{f}_1 = (70, 70)$. The new outcome becomes $\begin{pmatrix} s_1 & s_2 \\ c_1 & c_2 \end{pmatrix}$. Paradoxically, student s_1 is rewarded by getting his **top choice** as a result of worse performance. This college optimal scheme respect the preferences of students less.

Example – 18 candidates and 4 programs

PREFERENCES				
CAND NO.	CURRICULUM (K1)	(K2)	(K3)	(K4)
101	1	2	3	4
102	2	1	3	4
103	1	3	2	4
104	4	3	2	1
105	3	1	2	4
106	1	2	3	4
107	1	-	3	2
108	2	-	-	1
109	-	2	1	-
110	-	-	1	2
111	-	3	2	1
112	-	2	1	3
113	1	2	3	-
114	2	1	4	3
115	3	2	1	4
116	2	3	-	1
117	4	1	2	3
118	2	1	-	-

Letter 'C' indicates the College Optimal Stable Solution. Letter 'S' indicates the Student Optimal Stable Solution. Letter 'M' indicates a Third Stable Solution.

RANK	CURRICULUM (K1) $N_1 = 4$			CURRICULUM (K2) $N_2 = 4$			CURRICULUM (K3) $N_3 = 3$			CURRICULUM (K4) $N_4 = 3$		
	CAND NO.	PRE-FER-ENCE	STA-TUS	CAND NO.	PRE-FER-ENCE	STA-TUS	CAND NO.	PRE-FER-ENCE	STA-TUS	CAND NO.	PRE-FER-ENCE	STA-TUS
1	101	(1)	CSM	102	(1)	CSM	101	(3)		101	(4)	
2	102	(2)		101	(2)		104	(2)		102	(4)	
3	103	(1)	CSM	104	(3)		103	(2)		103	(4)	
4	104	(4)		103	(3)		105	(2)		104	(1)	CSM
5	105	(3)		105	(1)	CSM	102	(3)		108	(1)	CSM

For the stronger candidates, the assignments to the programs are identical under any stable assignment schemes. The corresponding achievable set is a singleton, where the best college and worst college coincide.

6	117 (4) C	106 (2) CSM	107 (3)	107 (2) C
7	116 (2) CSM	109 (2)	109 (1) CSM	105 (4)
8	107 (1) SM	111 (3)	110 (1) CSM	106 (4)
9	106 (1)	116 (3)	106 (3)	110 (2)
10	108 (2)	112 (2) CSM	111 (2) C M	112 (3)
11	114 (2)	113 (2)	117 (2) S	117 (3) M
12	115 (3)	114 (1)	112 (1)	111 (1) S
13	113 (1)	118 (1)	113 (3)	115 (4)
14	118 (2)	115 (2)	114 (4)	116 (1)
15		117 (1)	115 (1)	114 (3)

A small number of weaker candidates (107, 111, 117 at the border line of acceptance) are assigned to different programs by these stable solutions.

Three candidates, 107, 111 and 117, are assigned to different programs under C, S and M.

(i) 107 is assigned to his second choice K_4 under C; however, he is assigned to his first choice K_1 under S and M. 107 gets into better college under the student optimal stable solution.

(ii) 111 is assigned to his first choice K_4 under S; however, he is assigned to his second choice K_3 under C and M.

(iii) 117 is assigned to his fourth choice K_1 under C, third choice K_4 under M and second choice K_3 under S. The achievable set of programs of 117 is $\{K_3, K_4, K_1\}$, while K_2 is non-achievable since K_2 places 117 as its lowest priority. 117 gets the worst choice under the college optimal stable solution.

A student fails to get admitted into a college if his achievable set is the empty set. In this numerical example, the same set of students would be admitted into colleges, irrespective to the use of any one of the three stable matching schemes.

Absence of justified envy may conflict with Pareto efficiency

$$\begin{aligned} s_1 &: \; i_1 - i_3 - i_2 \qquad & i_1 &: \; s_2 \; s_1 \; s_3 \\ s_2 &: \; i_2 - i_1 - i_3 \qquad & i_2 &: \; s_1 \; s_2 \; s_3 \\ s_3 &: \; i_2 - i_1 - i_3 \qquad & i_3 &: \; s_1 \; s_2 \; s_3 \end{aligned}$$

There is only one stable matching:

$$\begin{pmatrix} i_1 & i_2 & i_3 \\ s_1 & s_2 & s_3 \end{pmatrix}.$$

Note that s_1 and s_2 get their respective best student already while s_3 is least welcomed by all students. Therefore, all schools would not complain.

i_3 may envy i_1 since i_1 gets into s_1. However, this is not a justified envy since s_1 prefers i_1 to i_3. The same argument applies to the potential envy of i_2 by i_3. Again, such envyness is not justified.

Free of justified envy may force students to share schools that are not Pareto efficient. The above stable matching is Pareto dominated by

$$\begin{pmatrix} i_1 & i_2 & i_3 \\ s_2 & s_1 & s_3 \end{pmatrix}$$

since i_1 and i_2 can both receive better assignment (Pareto improvement). Hence, Pareto efficiency is not observed. Since 2 out of 3 students have been assigned to their respective best choice, non-existence of Pareto improvement is guaranteed.

However, under the Pareto efficient matching, (i_3, s_1) forms a blocking pair with regard to (i_3, s_3) and (i_2, s_1) since i_3 prefers s_1 to s_3 and s_1 prefers i_3 to i_2. The Pareto efficient matching is not free of justified envyness.

Free of justified envy does not imply Pareto efficiency and vice versa.

Top trading cycles mechanism

The top trading cycles mechanism is Pareto efficient but does not completely eliminate justified envy.

Step 1: Assign a *counter* for each school which keeps track of how many seats are still available at the school. Initially set the counters equal to the capacities of the schools.

- Each student points to her favorite school under her announced preferences.

- Each school points to the student who has the highest priority for the school.

Formation of cycles

A *cycle* is an ordered list of distinct schools and distinct students $(s_1, i_1, s_2, \ldots, s_k, i_k)$ where s_1 points to i_1, i_1 points to $s_2, \ldots, s_k$ points to i_k, i_k points to s_1.

Since each vertex has one out-going edge, by starting with any vertex and following edges formed, we must find a cycle. We will not get "stuck" since there is always one out-going edge and there are a finite number of vertices, we must *repeatedly* visit some vertex to end up in a cycle.

Moreover, each school can be part of at most one cycle. Similarly, each student can be part of at most one cycle.

Every student in a cycle is assigned a seat at the school she points to and is removed. The counter of each school in a cycle is reduced by one and if it reduces to zero, the school is also removed. Counters of all other schools stay put.

Step k: Each remaining student points to her favorite school among the remaining schools and each remaining school points to the student with highest priority among the remaining students. There is at least one cycle.

- Every student in a cycle is assigned a seat at the school that she points to and is removed.

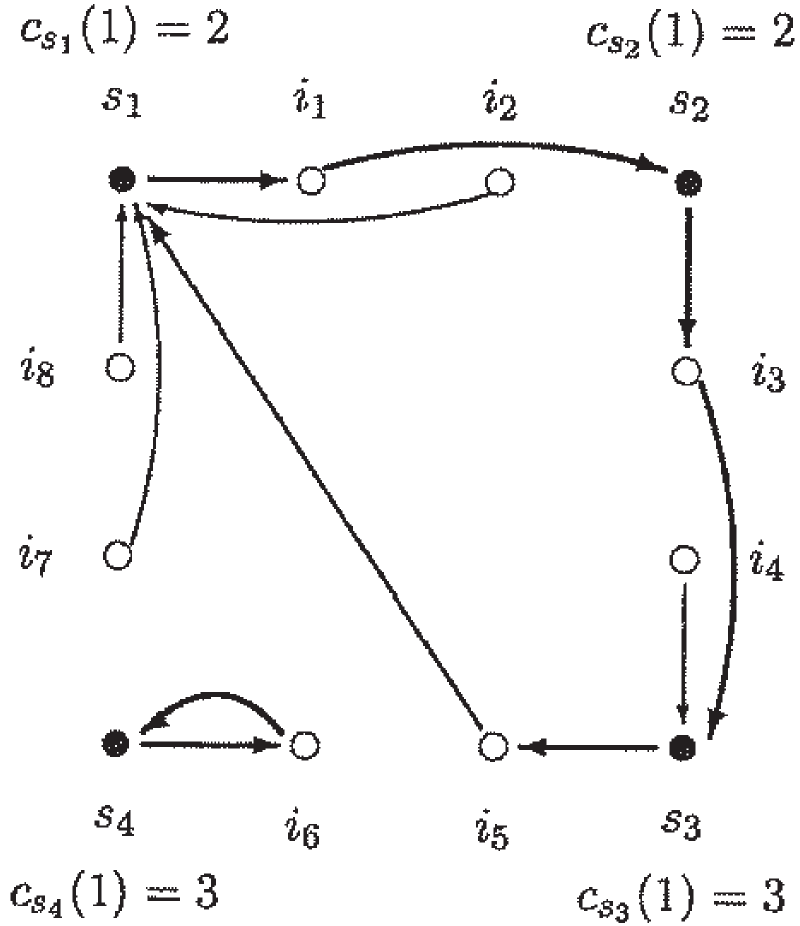
- The counter of each school in a cycle is reduced by one and if it reduces to zero the school is also removed. Counters of all other schools stay put.

The algorithm terminates when all students are assigned a seat. The number of steps cannot be more than the number of students.

Example

Use the same schools' priorities and students' preferences as that on p.23. Let c_{s_1}, c_{s_2}, c_{s_3} and c_{s_4} indicate the counters of the schools.

Step 1:



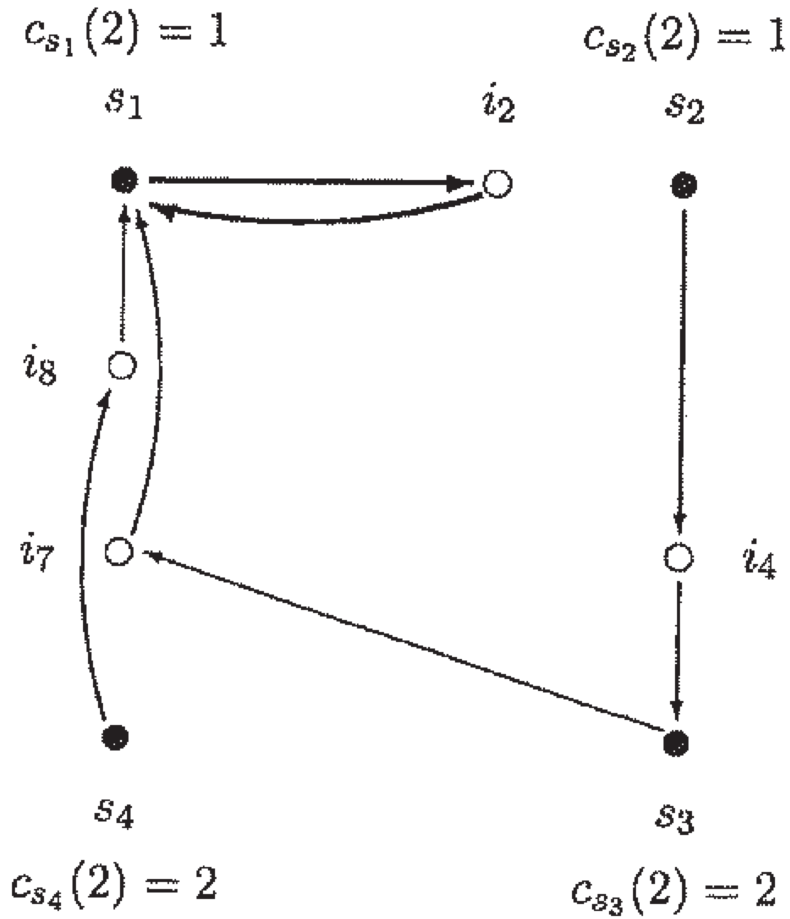
By starting at i_2, i_5, i_7 or i_8, we end up at s_1. When the procedure of pointing to the most desirable school continues, it is guaranteed that eventually one student points to some school that has been visited earlier in the pointing process.

There are two cycles formed in Step 1: $(s_1, i_1, s_2, i_3, s_3, i_5)$ and (s_4, i_6).

- Students i_1, i_3, i_5, i_6 are assigned one slot at schools s_2, s_3, s_1, s_4 respectively and removed.

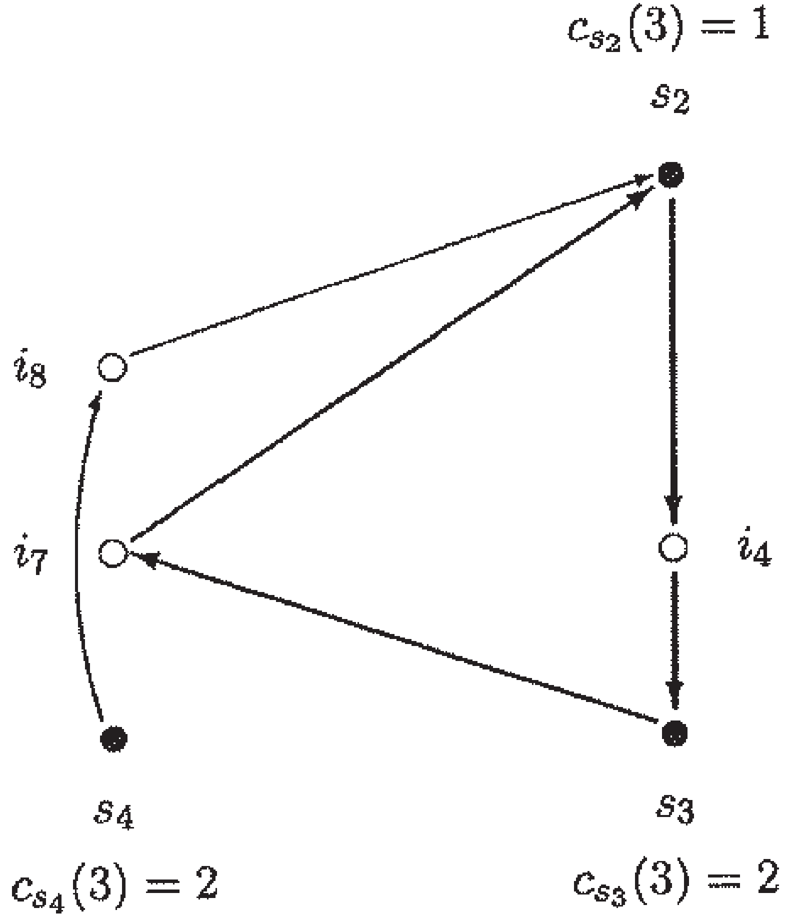
- In step 1, every school participates in a cycle. All counters are reduced by one for the next step.

Step 2:



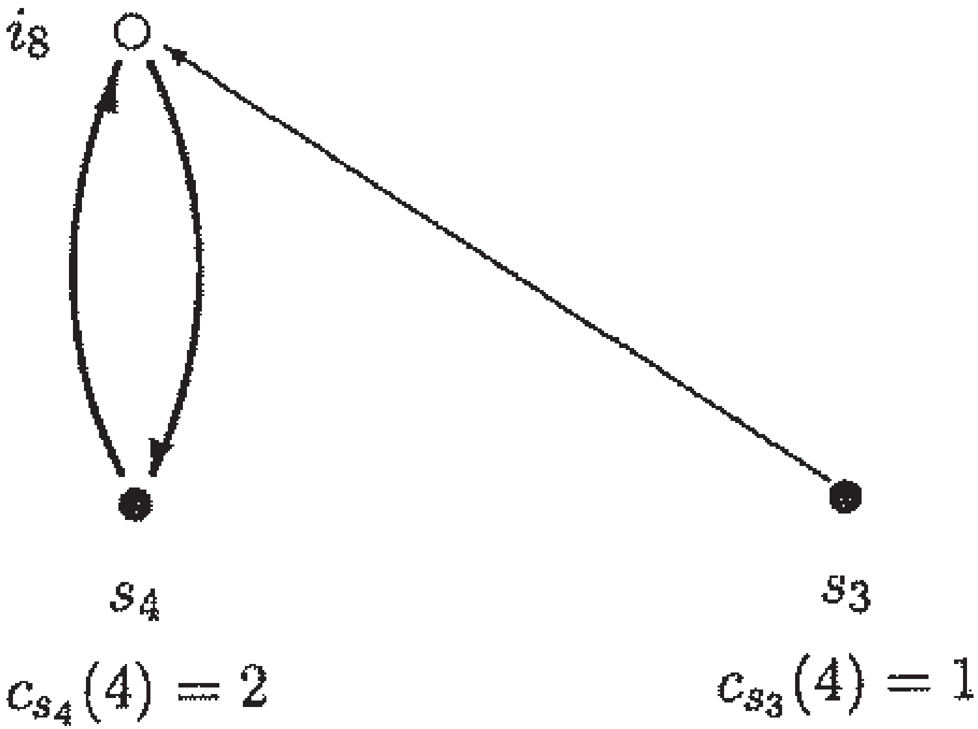
There is only one cycle formed in Step 2: (s_1, i_2). As a result, student i_2 is assigned one slot at school s_1 and removed. The counter of school s_1 is reduced by one to zero and it is removed. All other counters stay put.

Step 3:



There is only one cycle formed in Step 3: (s_3, i_7, s_2, i_4). Students i_7, i_4 are assigned one slot at schools s_2, s_3, respectively, and removed. The counters of schools s_2 and s_3 are reduced by one. Since there are no slots left at school s_2, it is removed. Counters of schools s_3 and s_4 stay put.

Step 4:



There is only one cycle in Step 4: (s_4, i_8). Therefore student i_8 is assigned one slot at school s_4 and removed. There are no remaining students so the algorithm terminates. The final matching is

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 \\ s_2 & s_1 & s_3 & s_3 & s_1 & s_4 & s_2 & s_4 \end{pmatrix}.$$

The matching outcomes ensure Pareto efficiency.

Top trading cycles mechanism is Pareto efficient

It suffices to show that it is always impossible to make a student better off without hurting the other student. As a result, Pareto improvement cannot be achieved, so Pareto efficiency is established.

- Any student who leaves at Step 1 is assigned her top choice and cannot be made better off.

- Any student who leaves at Step 2 is assigned her top choice among those seats remaining at Step 2. However, She cannot get the better school that has left in Step 1. In other words, she cannot be made better off without hurting someone who left at Step 1 since all students in Step 1 got their best choice.

- Proceeding in a similar way, no student can be made better off without hurting someone who left at an earlier step. Therefore the *top trading cycles mechanism* is Pareto efficient.

Top trading cycles mechanism is strategy-proof

Consider a student i with true preferences P_i. Fix an announced preference profile $Q_{-i} = (Q_j)_{j \in I \setminus \{i\}}$ for every student except i.

Let T be the step at which student i leaves under a game preference Q_i, $(s, i_1, s_1, \ldots, s_k, i)$ be the cycle she joins, and thus school s be her assignment.

Let T^* be the step at which she leaves under her true preferences P_i. We want to show that her assignment under P_i is at least as good as school s, so gaming strategy does not help.

Case 1: $T^* < T$

Since student i fails to participate in a cycle prior to Step T^*, we deduce that the same schools remain in the algorithm at Step T^* whether student i announces Q_i or P_i.

The assignments made earlier than T^* are NOT affected by the adoption of either P_i or Q_i by student i. There is no impact on schools available to others since no schools (or quotas) have been removed due to a cycle formed by student i prior to T^*.

Student i is assigned a school at his best choice remaining at T^* under P_i. This must be at least as good as school s.

Case 2: $T^* \geq T$

The key observation is that the same students and schools remain at Step T whether i announces Q_i or P_i. At Step T, the pointing pattern $s \to i_1$, $i_1 \to s_1$, ..., $s_k \to i$ is observed. The key is that this pointing *pattern persists* as long as student i remains.

Under Q_i, student i points to school s within a closed cycle, so i receives s. We argue that this is not necessarily the best school of i among remaining schools at step T.

Indeed, when student i announces P_i truthfully, he can either receive a better assignment than school s or at least s in this round or a later round. This is because school s remains after T since the pointing pattern $s \to i_1$, $i_1 \to s_1$, ..., $s_k \to i$ persists. Since student i truthfully points to his best remaining choice of school at each step, either i gets a better choice than s or s takes in i when i goes down to s as his best choice.

School choice mechanisms

Primary school graduating students choose their secondary schools under a school choice mechanism. The priorities are based on home location, siblings, religious attachment, academic scores, etc. It is different from the College admission problem, where colleges have preferences over students. Here, the priorities are enforced by laws and schools have no say on how the priority order is determined.

Schools are viewed as objects to be consumed. Students are the only "economic agents", not schools. School choice matching is considered as one-sided matching.

The priority of student i is violated (or disrespected) at a matching if there exists another student j who is assigned to school s such that (i) student i prefers school s to his current assignment, (ii) student i has higher priority than student j for school s. Violation of priorities may induce parents to seek legal action for such envyness.

Boston student assignment mechanism

1. Each student submits a preference ranking of the schools.
2. For each school a priority ordering is determined according to the following hierarchy:
   - First priority: sibling and walk zone.
   - Second priority: sibling.
   - Third priority: walk zone.
   - Fourth priority: students other than above.

Students in the same priority group are ordered based on a previously announced lottery.

A secondary school has to take in students who have chosen the school as their first choice before admitting other students.

Political correctness: Government may boost that a high portion of students can get into their first choice.

Round 1: Only the first choices of the students are considered. For each school, consider the students who have listed it as their first choice and assign seats of the school to these students one at a time following their priority order until either there are no seats left or there is no student left who has listed it as her first choice.

Round 2: Only the second choices of these students are considered. For each school with still available seats, consider the students who have listed it as their second choice and assign the remaining seats to these students one at a time following their priority order until either there are no seats left or there is no student left who has listed it as her second choice. The same procedure is repeated with the k^{th} choice, $k = 3, 4, \ldots$.

Once a student misses her first choice, she may be admitted into a school which may be quite low in her preference list. Students (and parents) have to game around the mechanism. If the dreamed first choice is too competitive, the student may cheat by putting a less preferred choice as the first choice in order to secure admission into a reasonably good but not the best school.

JUPAS

Students can choose up to 20 programs. Programs can rank students based on their test scores and interview performance. However, programs do not have the full information of the preference lists of students on their ranking orders on programs, except in the form of 5 bands. Band A contains the top 3 choices, Band B contains the next 3 choices, etc. A program is informed whether a student has selected the program to be among the top 3 choices in Band A, but not the exact order of first, second or third choice.

The categorization into bands help programs to conduct interviews for students who show strong interest in their programs. As the common practice, most elite programs only admit students who choose the specific program as their Band A choices.

Similar to the school choice problems, JUPAS applicants have to game around to set their Band A choices to programs whose expected minimum admission scores are close but lower than their examination scores.

School tiers system in China

- Colleges are categorized into different tiers.
  Key colleges (National 985 and 211 universities) belong to the first tier and admit students first.
  Ordinal colleges belong to the second tier.
  Vocational training colleges are included in the third tier.

- College admissions in China proceed sequentially in peers. Only when assignments in the first tiers are finalized, admissions in the second tier start; and so on.

Chinese parallel mechanism

We consider the following Shanghai version of the Chinese parallel mechanisms in its simplest version, with two parallel choices.

- An application to the first ranked school is sent for each student at the first step.

- Throughout the allocation process, a school can hold no more applications than its capacity. If a school receives more applications than its capacity, it retains the students with the highest priority up to its capacity and rejects the remaining students.

- Whenever a student is rejected from his first choice school, her application is sent to her second choice school. Whenever a student is rejected from her second choice school, she can no longer make an application in this round.

- Throughout each round, whenever a school receives new applications, these applications are considered together with the retained applications for that school. Among the retained and new applications, the ones with the highest priority up to the capacity are retained.

- If a student is rejected by his first two choices in the initial round, then he participates in a new round of applications together with other students who have also been rejected from their first two choices, and so on. At the end of each round the assigned students and the slots assigned to them are removed from the system.

The assignment process ends when no more applications can be rejected.

We call it to be application-rejection mechanism with e schools, which coincides with

1. Boston mechanism when $e = 1$,
2. Chinese parallel mechanism when $2 \leq e < \infty$ (Shanghai version when $e = 2$),
3. deferred acceptance mechanism when $e = \infty$.

Within the family of application-rejection mechanism ϕ^e:

1. $e = 1$, Boston mechanism is Pareto efficient.
2. $e = \infty$, deferred acceptance mechanism is both strategy-proof and stable.

For any e, ϕ^e is more manipulable than $\phi^{e'}$ whenever $e' > e$.

Therefore, among application-rejection mechanisms, Boston mechanism is the most manipulable and the deferred acceptance mechanism is the least manipulable member.

A “successful” strategy for a student is one that ensures that he is assigned to his “target school” at the end of the initial round. In this sense, missing the first choice in the Boston mechanism could be more costly to a student than in a Chinese parallel mechanism such as the Shanghai, which offers a “second chance” to the student before he loses his priority advantage. On the other hand, at the other extreme of this class lies the deferred acceptance mechanism, which completely eliminates any possible loss of priority advantage for a student.

It is in the best interest of each student to put his within-round choices in their true order. More precisely, for a student facing ϕ^e, any strategy that does not list the first e choices (that are considered in the initial round) in their true order is dominated by the otherwise identical strategy that lists them in their true order.

Timing of preference submission by students

- Before the examination (2 provinces).
- After the examination but before knowing the examination s-cores (3 provinces).
- After knowing the examination scores (26 provinces).

Complaint from a parent in China on the old college admission scheme (similar to the Boston mechanism): "My child achieved a score of 632 in the college entrance examination last year. Unfortunately, he was not accepted by his first choice. After his first choice rejected him, his second and third choices were already full. My child had no choice but to repeat his senior year."

The application-rejection algorithm is praised by one student in Shanghai: "I could give Peking University a try ... Even though I failed to be admitted by Peking University, Fudan University accepted me at the end, thanks to the parallel choice algorithm."

Example

Consider a problem with three students i_1, i_2, and i_3 and three schools s_1, s_2, and s_3, each with one seat. Student preferences are:

$$\begin{aligned} i_1 &: s_2 - s_1 - s_3 \\ i_2 &: s_1 - s_2 - s_3 \\ i_3 &: s_1 - s_2 - s_3, \end{aligned}$$

and priorities are:

$$\begin{aligned} s_1 &: i_1 - i_3 - i_2 \\ s_2 &: i_2 - i_1 - i_3 \\ s_3 &: i_3 - i_1 - i_2. \end{aligned}$$

Under the *student-proposing deferred acceptance mechanism*, the matching produced is

$$\mu_{\mathsf{DA}} = \begin{pmatrix} i_1 & i_2 & i_3 \\ s_1 & s_2 & s_3 \end{pmatrix}.$$

In this matching, none of the students obtain their top choice. The matching is not Pareto efficient. However, since there are no blocking pairs, it is stable.

Under the *top trading cycles mechanism*, the matching produced is

$$\mu_{\mathsf{TTC}} = \begin{pmatrix} i_1 & i_2 & i_3 \\ s_2 & s_1 & s_3 \end{pmatrix}.$$

This matching is Pareto efficient since both student i_1 and i_2 have obtained their top choice. However, student i_3 and school s_1 form a blocking pair, so the matching is not stable.

Under the *Boston mechanism*, the matching produced is

$$\mu_{\mathsf{BOS}} = \begin{pmatrix} i_1 & i_2 & i_3 \\ s_2 & s_3 & s_1 \end{pmatrix}.$$

Poor i_2, he gets the worst choice s_3 since he loses to i_3 in competing for the first choice s_1 and his second choice s_2 has been taken by i_1 in the first-choice round.

This matching is Pareto efficient since i_1 and i_3 have obtained their first choice. However, student i_2 and school s_2 form a blocking pair, so the matching is not stable. Moreover, had students i_2 reported that s_2 was her top choice, that is, $s_2 - s_1 - s_3$, she would have received an assignment there, which demonstrates that the mechanism is not strategy-proof:

$$\mu_{\text{BOS}}^{\text{game}} = \begin{pmatrix} i_1 & i_2 & i_3 \\ s_3 & s_2 & s_1 \end{pmatrix}.$$

The outcome of the Boston mechanism is Pareto efficient, provided that students truthfully reveal their preferences. Proof of Pareto efficiency can follow similar argument as that of the Top Trading Cycle algorithm. However, truthful revelation is rarely in the best interest of students, and efficiency loss is expected.

Game around the system

With gaming on both sides (initiated by programs, followed by s-tudents), envy-freeness would not be guaranteed. A program may admit students with lower scores while missing other applicants with higher scores. Mediocre programs and students may benefit from the gaming strategies on both sides.

Social welfare enhanced or undermined? How to quantify social welfare?

To what extent can programs admit reasonably good quality of students with high interest in the programs?

Information collection on the minimum admission scores is important for students to help make better informed decision. How much an uninformed student loses her chance of getting into a program of reasonable level of desirability?

2.3 House allocation with existing tenants

A house allocation problem with existing tenants consists of

1. A finite set of existing tenants, I_E;
2. A finite set of new applicants, I_N;
3. A finite set of occupied houses, $H_0 = \{h_i\}_{i \in I_E}$;
4. A finite set of vacant houses, H_V;
5. A list of preference relations, $P = (P_i)_{i \in I_E \cup I_N}$.

We assume no consumption externalities, where agents care only the houses but not their neighbors. Also, there is an ordering f of the agents, where this order may indicates seniority or priority of the agents. Otherwise, an ordering can be generated by lottery.

In some earlier mechanisms, those who want to move are asked to give up their houses before they are assigned another one.

If there is no guarantee of getting a better house, the tenants may simply keep their current houses, which may result in loss of potential gains from trade. We would like to ensure that the new assigned houses would not be worst off than the original assignment for existing tenants. This is called *individual rationality*.

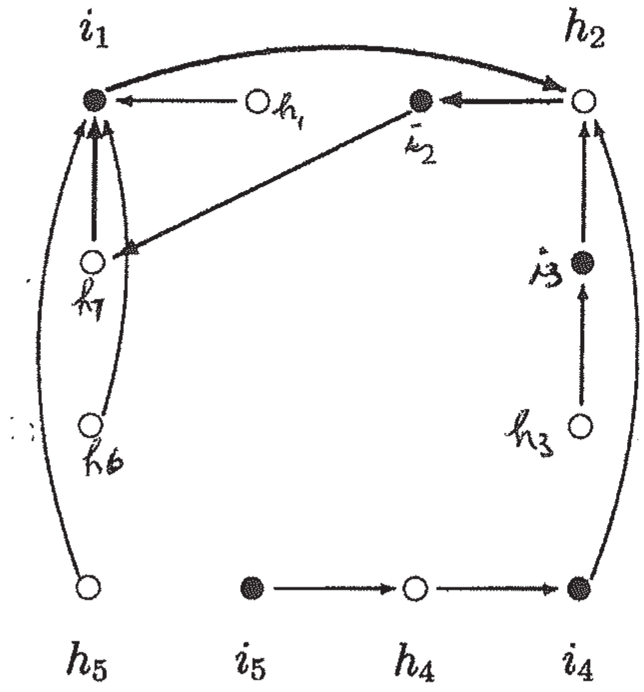
Pareto efficient: There is no other allocation that makes all agents weakly better off and at least one agent strictly better off.

Strategy proof: Truth telling is a weakly dominant strategy for every agent.

Let $I_E = \{i_1, i_2, i_3, i_4\}$, $I_N = \{i_5\}$, $H_0 = \{h_1, h_2, h_3, h_4\}$, and $H_V = \{h_5, h_6, h_7\}$. The existing tenant i_k occupies the house h_k for $k = 1, \ldots, 4$.

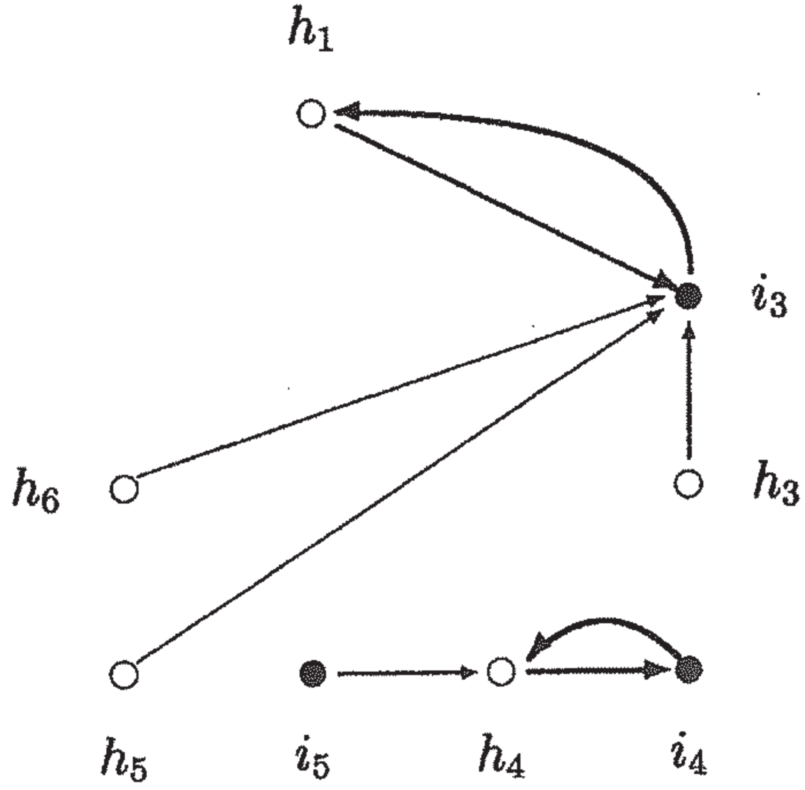
Let the ordering f order the agents as $i_1 - i_2 - i_3 - i_4 - i_5$ and let the preferences be as follows:

P_{i_1}	P_{i_2}	P_{i_3}	P_{i_4}	P_{i_5}
h_2	h_7	h_2	h_2	h_4
h_6	h_1	h_1	h_4	h_3
h_5	h_6	h_4	h_3	h_7
h_1	h_5	h_7	h_6	h_1
h_4	h_4	h_3	h_1	h_2
h_3	h_3	h_6	h_7	h_5
h_7	h_2	h_5	h_5	h_6

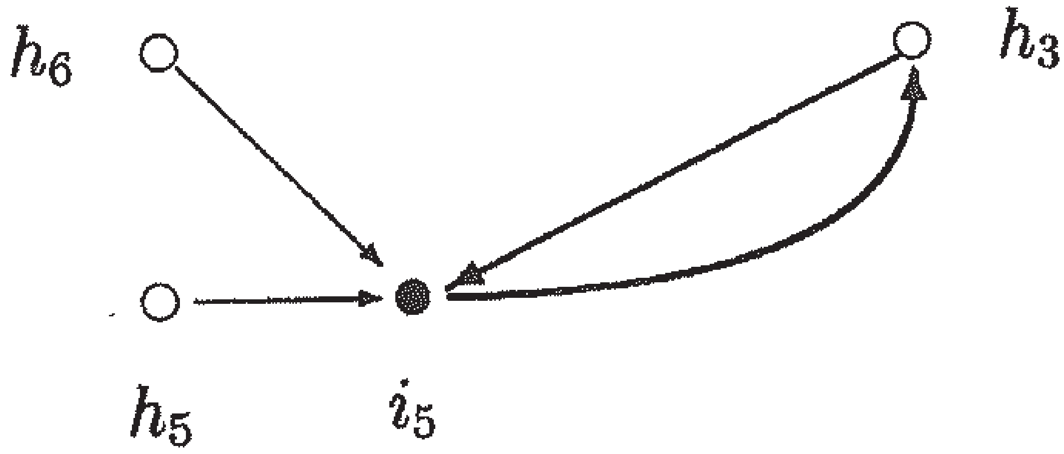


- All vacant houses h_5, h_6 and h_7 all point to i_1 (the agent with highest priority). In this way, i_1 would not lose the top priority to get into one of the vacant houses.
- All occupied houses point to their existing tenants, like h_2 points to i_2. The existing tenant would not lose the occupied house unless he finds better house.
- All agents point to their top choice; like i_1 points to h_2 and i_3 points to h_2.

A closed cycle is formed: i_1 gets h_2 and i_2 gets h_7.



- Since i_1 and i_2 move out in the first round, h_1 becomes available.
- Now, i_3 is the highest ranking agent, so h_1, h_5 and h_6 all point to i_3.
- There are two cycles (i_3, h_1) and (i_4, h_4). Therefore i_3 is assigned h_1 and i_4 is assigned his own house h_4.



- The available houses h_3, h_5, and h_6 all point to the only remaining agent i_5. The only cycle is (i_5, h_3). Therefore i_5 is assigned h_3.

Final outcome

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 \\ h_2 & h_7 & h_1 & h_4 & h_3 \end{pmatrix}$$

The new assignment for an existing tenant would not be worst off since an existing tenant always has his original assigned house pointing to him until he leaves. The new assignment of i cannot be worse than his original assignment.

MIT · NH4 mechanism

The following mechanism is used at the residence NH4 of MIT. It works as follows:

(i) An ordering f of agents is chosen from a given distribution of agents.

(ii) The first agent is *tentatively* assigned his or her top choice among all houses, the next agent is *tentatively* assigned his top choice among the remaining houses, and so on, until a *squatting conflict* occurs.

(iii) A *squatting conflict* occurs if it is the turn of an existing tenant but every remaining house is worse than his or her current house. That means someone else, the *conflicting agent*, is tentatively assigned the existing tenant's current house.

When this happens

(a) the existing tenant is assigned his or her current house and removed from the process, and

(b) all tentative assignments starting with the conflicting agent and up to the existing tenant are erased.

At this point the squatting conflict is resolved and the process starts over again with the conflicting agent. Every squatting conflict that occurs afterwards is resolved in a similar way.

(iv) The process is over when there are no houses or agents left. At this point all tentative assignments are finalized.

Let $I_E = \{i_1, i_2, i_3, i_4\}$, $I_N = \{i_5\}$, $H_0 = \{h_1, h_2, h_3, h_4\}$, and $H_V = \{h_5\}$. Here the existing tenant i_k occupies the house h_k for $k = 1, 2, 3, 4$. Let the ordering f order the agents as $i_1 - i_2 - i_3 - i_4 - i_5$ and let the preferences be as follows:

P_{i_1}	P_{i_2}	P_{i_3}	P_{i_4}	P_{i_5}
h_3	h_4	h_5	h_3	h_4
h_4	h_5	h_3	h_5	h_5
h_5	h_2	h_4	h_4	h_3
h_1	h_3	h_2	h_2	h_1
h_2	h_1	h_1	h_1	h_2
h_0	h_0	h_0	h_0	h_0

Round one

$$\begin{array}{cccc} i_1 & i_2 & i_3 & i_4 \\ \hline h_3 & h_4 & h_5 & h_3 \\ & & & h_5 \\ & & & \boldsymbol{h_4} \end{array} \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{occupied}$$

i_4 is assigned to h_4 and leaves. The house held by i_2 (conflicting agent) is reassigned.

Round two

$$\begin{array}{ccc} i_1 & i_2 & i_3 \\ \hline h_3 & h_5 & h_5 \\ & & \boldsymbol{h_3} \end{array} \text{ occupied}$$

i_3 is assigned to h_3 and leaves. The house held by i_1 (conflicting agent) is reassigned.

Round three

$$\begin{array}{ccc} i_1 & i_2 & i_5 \\ \hline h_5 & h_2 & h_1 \end{array}$$

i_2 is assigned h_2 since h_4 and h_5 are gone. The new comer is assigned the fourth choice h_1. Poor i_3, though he is at the top of the order, he only gets the third choice since the top two choices are assigned to the existing occupants due to resolution of squatting conflicts.

Therefore the final matching is

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 \\ h_5 & h_2 & h_3 & h_4 & h_1 \end{pmatrix}$$

which is Pareto dominated by both

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 \\ h_3 & h_2 & h_5 & h_4 & h_1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} i_1 & i_2 & i_3 & i_4 & i_5 \\ h_4 & h_2 & h_5 & h_3 & h_1 \end{pmatrix}.$$

2.4 Kidney exchange

Most transplanted kidneys are from cadavers, but there are also many transplants from live donors. Live donor kidneys are preferable to cadaver kidneys due to a higher survival rate from surgery. A successful transplant requires the donor and recipient to be compatible in blood and tissue types. There has always been a considerable shortage of kidneys, compared with demand.

Direct exchange

This involves two patient-donor pairs in which a transplant from the donor to the intended patient is infeasible, but successful transplants are possible using the kidney from the other patient-donor pair.

Indirect exchange

This involves an exchange between one incompatible patient-donor pair and the cadaver queue. The patient in the (donor, patient) pair receives high priority on the cadaver queue, in return for the donation of his donor's kidney to someone on the queue.

This improves the welfare of the patient in the pair, compared with having a long wait for a suitable cadaver kidney. It also benefits the recipient of the live kidney, and others on the queue who benefit from the increase in kidney supply due to an additional living donor. In general, patients gain more if they are given wider choices of kidneys under the exchange mechanism.

Reference

Roth, A.E., Sönmez, T. and Ünver, M.U. (2004). "Kidney exchange", *Quarterly Journal of Economics*, vol. 119(2), p.457-488.

U. S. KIDNEY TRANSPLANTS

Year	Cadaveric donors	Cadaveric transplants	Live donors	All wait-list patients	New wait-list additions
1992	4,276	7,202	2,535	22,063	15,224
1993	4,609	7,509	2,851	24,765	16,090
1994	4,797	7,638	3,009	27,258	16,538
1995	5,003	7,690	3,377	30,590	17,903
1996	5,038	7,726	3,649	34,000	18,328
1997	5,083	7,769	3,912	37,438	19,067
1998	5,339	8,017	4,361	40,931	20,191
1999	5,386	8,023	4,552	43,867	20,986
2000	5,490	8,089	5,324	47,596	22,269
2001	5,528	8,202	5,924	51,144	22,349
2002	5,630	8,534	6,233	54,844	23,494

Cadaver queue: The waiting list for cadaver kidneys is highly structured, with scores being assigned to each candidate based on factors, like blood type, tissue, age, size. These factors affect whether a transplant is likely to succeed. This allows for strict preferences within any kidney exchange mechanism.

Parallels between the kidney exchange system and college dormitories allocation

- The tenant of the requested dormitory room is moved to the top of the queue ahead of the person who requested that room. This feature ensures every existing tenant a room that is no worse than his own. If a donor gives a kidney to the cadaver queue, his intended recipient jumps to the top of the queue.

- The number of rooms in the room allocation is fixed, whereas the number of kidneys is not. We do not know how long one has to wait in the cadaver queue until a compatible kidney becomes available.

- There are unoccupied rooms and new students who do not have occupied rooms. The counterpart of new students are patients who have no living donors. The counterpart of vacant rooms are cadaveric kidneys that are not targeted for specific patients. While occupied rooms and vacant rooms can be simultaneously allocated, this is not possible in the context of kidney exchange due to uncertainty in the arrivals of kidneys.

Top trading cycles and chains (TTCC) mechanism

There are n donor-recipient pairs, (k_i, t_i), $i = 1, 2, \ldots, n$. The donor k_i is interpreted as kidney k_i and intended recipient t_i as patient t_i.

Given a patient t_i, let $K_i \subset K$ denote the set of living donor kidneys that are compatible with patient t_i. Let w denote the option of entering "the waiting list with priority", reflecting the donation of his donor's kidney k_i. Let P_i denote his strict preferences over $K_i \cup \{k_i, w\}$.

- If patient t_i ranks kidney k_i at the top of his preferences, that means he and his donor do not wish to participate in an exchange.

- If patient t_i ranks k_i on top of w, that means he and his donor do not consider exchanging kidney k_i with priority in the cadaveric kidney waiting list.

Cycles and w-Chains

In each round each patient t_i points either toward a kidney in $K_i \cup \{k_i\}$ or toward w, and each kidney k_i points to its paired recipient t_i. At least, t_i can point back to k_i as the last resort. However, he always hopes to receive better kidney.

A *cycle* is an ordered list of kidneys and patients $\{k'_1, t'_1, k'_2, t'_2, \ldots, k'_m, t'_m\}$ such that kidney k'_1 points to patient t'_1, patient t'_1 points to kidney $k'_2, \ldots$, kidney k'_m points to patient t'_m, and patient t'_m points to kidney k'_1. Note that each kidney or patient can be part of at most one cycle and thus *no two cycles intersect*.

A w-*chain* is an ordered list of kidneys and patients $\{k'_1, t'_1, k'_2, t'_2, \ldots, k'_m, t'_m\}$ such that kidney k'_1 points to patient t'_1, patient t'_1 points to kidney $k'_2, \ldots$, kidney k'_m points to patient t'_m, and patient t'_m points to w. We refer to the pair (k'_m, t'_m) whose patient receives a cadaver kidney in a w-chain as the *head* and the pair (k'_1, t'_1) whose donor donates to someone on the cadaver queue as the *tail* of the w-chain.

We need to have a well-defined chain selection rule. Selection of longer w-chains will benefit more patients, so this choice of a chain selection rule has efficiency implications.

For a given kidney exchange problem, the TTCC mechanism determines the exchanges as follows.

1. Initially, all kidneys are *available*, and all agents are active. At each stage of the procedure each remaining *active* patient t_i points to his most preferred remaining unassigned kidney or to the wait-list option w, whichever is more preferred. Each remaining kidney k_i points to its paired recipient t_i.

2. *There is either a cycle, or a w-chain, or both.*

   (a) Proceed to Step 3 if there are no cycles. Otherwise, locate each cycle, and carry out the corresponding exchange. Remove all patients in a cycle together with their assignments.

   (b) Each remaining patient points to his top choice among remaining kidneys, and each kidney points to its paired recipient. Locate all cycles, carry out the corresponding exchanges, and remove them. Repeat until no cycle exists.

3. If there are no pairs left, we are done. Otherwise, each remaining pair is the tail of a w-chain. Select *only one* of the chains with the *chain selection rule*. The assignment is *final* for the patients in the selected w-chain. The chain selection rule also determines whether the selected w-chain is removed and the associated exchanges are all immediately assigned (including the kidney at the tail, which is designated to go to a patient on the cadaver queue), or if the selected w-chain is kept in the procedure although each patient in it is passive henceforth.

4. After a w-chain is selected, new cycles may form. Repeat Steps 2 and 3 with the remaining active patients and unassigned kidneys until no patient is left.

At the end of the procedure, each patient with a living donor is assigned a kidney (or a high priority place on the waiting list).

Consider a kidney exchange problem with 12 pairs $(k_1, t_1), \ldots, (k_{12}, t_{12})$ with preferences as follows:

t_1:	k_9	k_{10}	k_1				
t_2:	k_{11}	k_3	k_5	k_6	k_2		
t_3:	k_2	k_4	k_5	k_6	k_7	k_8	w
t_4:	k_5	k_9	k_1	k_8	k_{10}	k_3	w
t_5:	k_3	k_7	k_{11}	k_4	k_5		
t_6:	k_3	k_5	k_8	k_6			
t_7:	k_6	k_1	k_3	k_9	k_{10}	k_1	w
t_8:	k_6	k_4	k_{11}	k_2	k_3	k_8	
t_9:	k_3	k_{11}	w				
t_{10}:	k_{11}	k_1	k_4	k_5	k_6	k_7	w
t_{11}:	k_3	k_6	k_5	k_{11}			
t_{12}:	k_{11}	k_3	k_9	k_8	k_{10}	k_{12}	.

Note that patient t_i either assigns k_i as the lowest priority, or otherwise w. The later case corresponds to the case where k_i is not compatible, so t_i is put at the top of the cadaver queue.

The final matching is

$$\begin{pmatrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} & t_{12} \\ k_9 & k_{11} & k_2 & k_8 & k_7 & k_5 & k_6 & k_4 & w & k_1 & k_3 & k_{10} \end{pmatrix}.$$

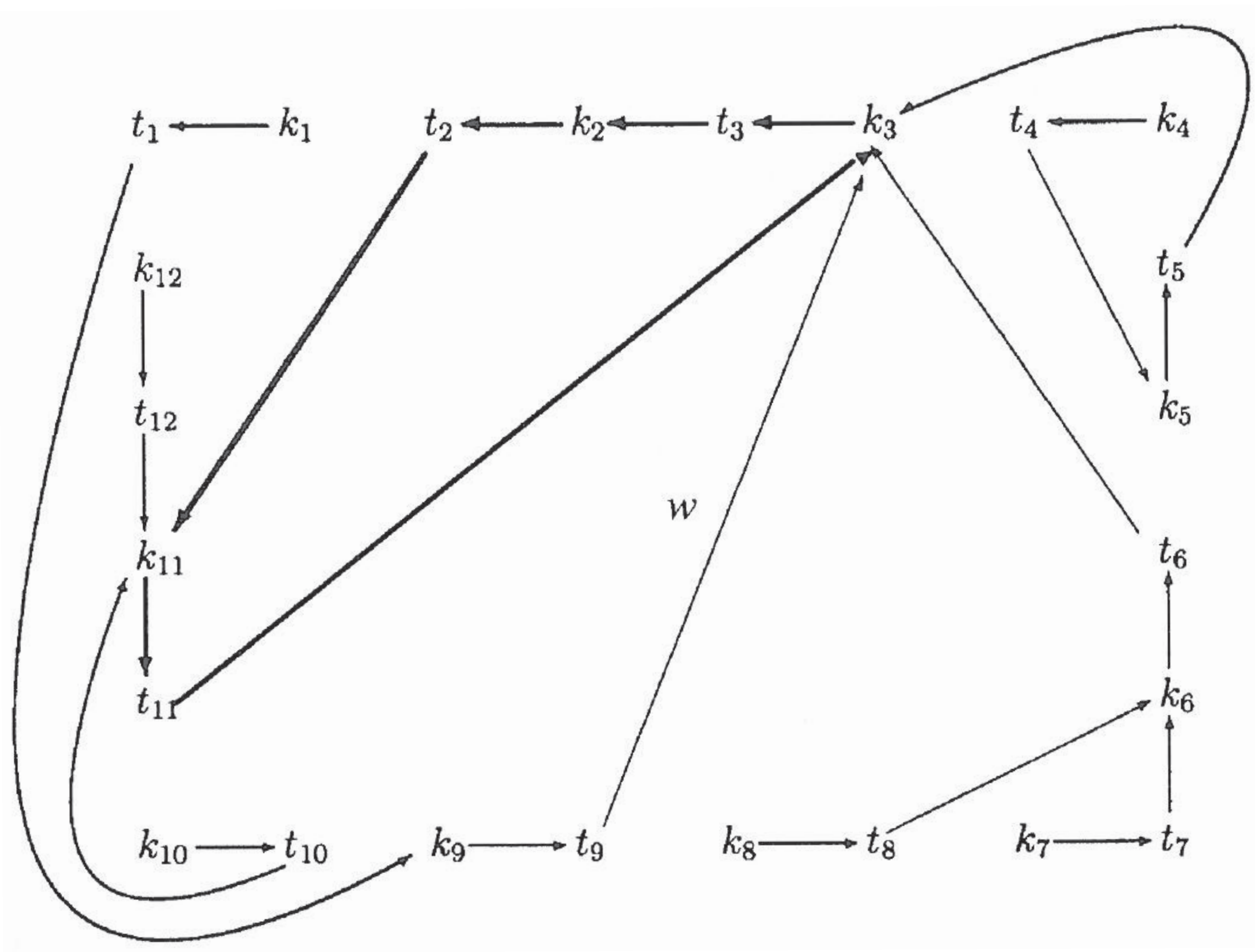


FIGURE I
Example 1, Round 1

There is a single cycle $C_1 = (k_{11}, t_{11}, k_3, t_3, k_2, t_2)$. Remove the cycle by assigning k_{11} to t_2, k_3 to t_{11}, and k_2 to t_3.

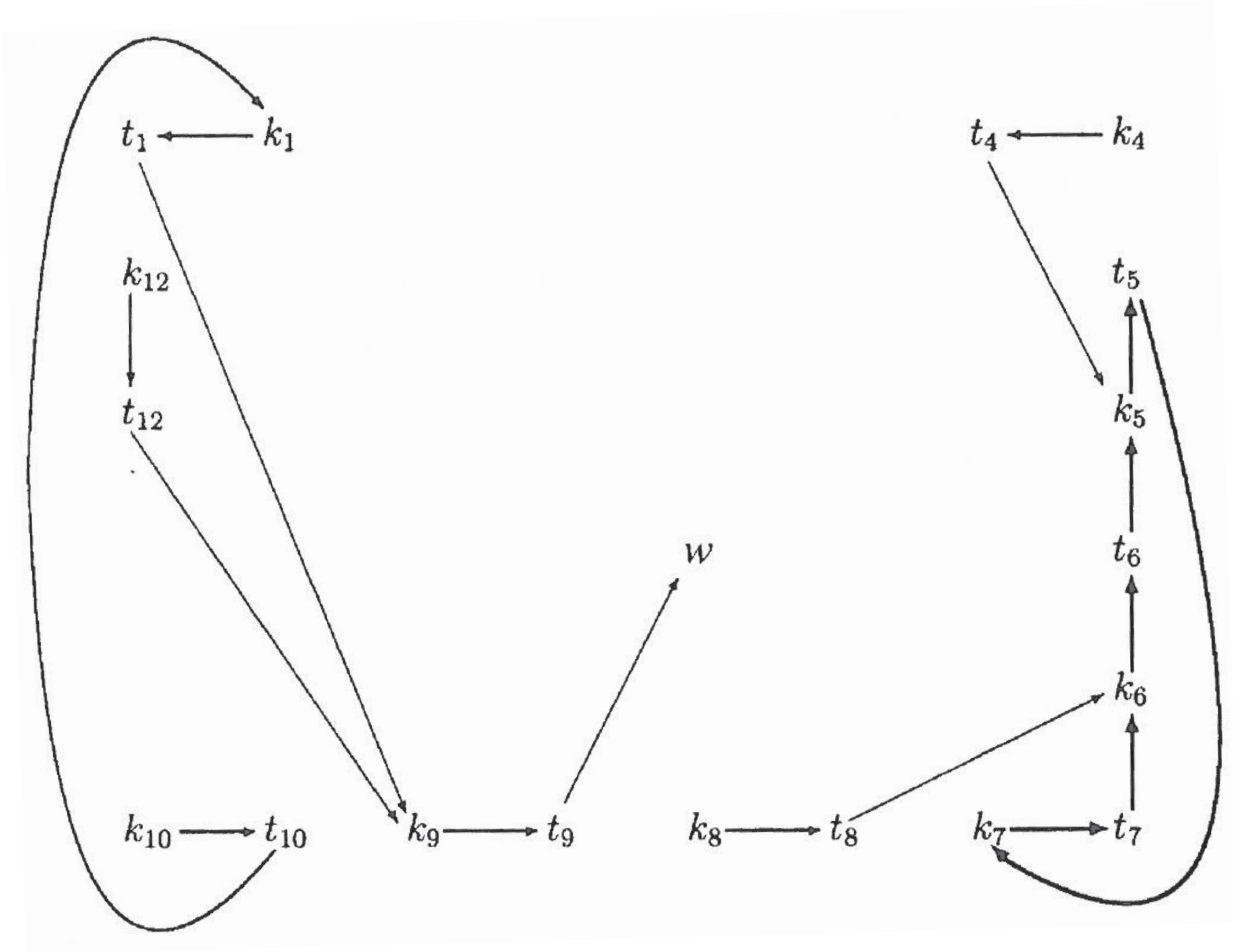


FIGURE II
Example 1, Round 2

Upon removing cycle C_1, a new cycle $C_2 = (k_7, t_7, k_6, t_6, k_5, t_5)$ form-s. Remove it by assigning k_7 to t_5, k_6 to t_7, and k_5 to t_6.

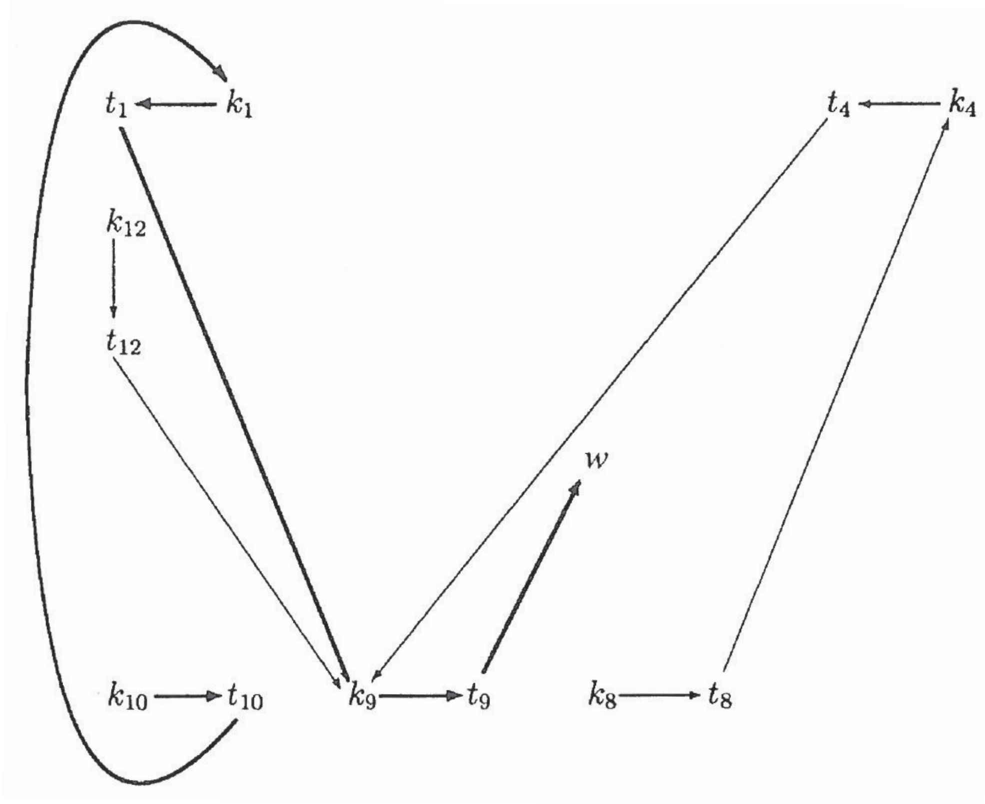


FIGURE III
Example 1, Round 3

No new cycle forms, and hence each kidney-patient pair starts a w-chain. The longest w-chains are $W_1 = (k_8, t_8, k_4, t_4, k_9, t_9)$ and $W_2 = (k_{10}, t_{10}, k_1, t_1, k_9, t_9)$. Since t_1, the highest priority patient, is in W_2 but not in W_1, choose and fix W_2. Assign w to t_9, k_9 to t_1, and k_1 to t_{10} but do not remove them. Kidney k_{10}, the kidney at the tail of W_2, remains available for the next round.

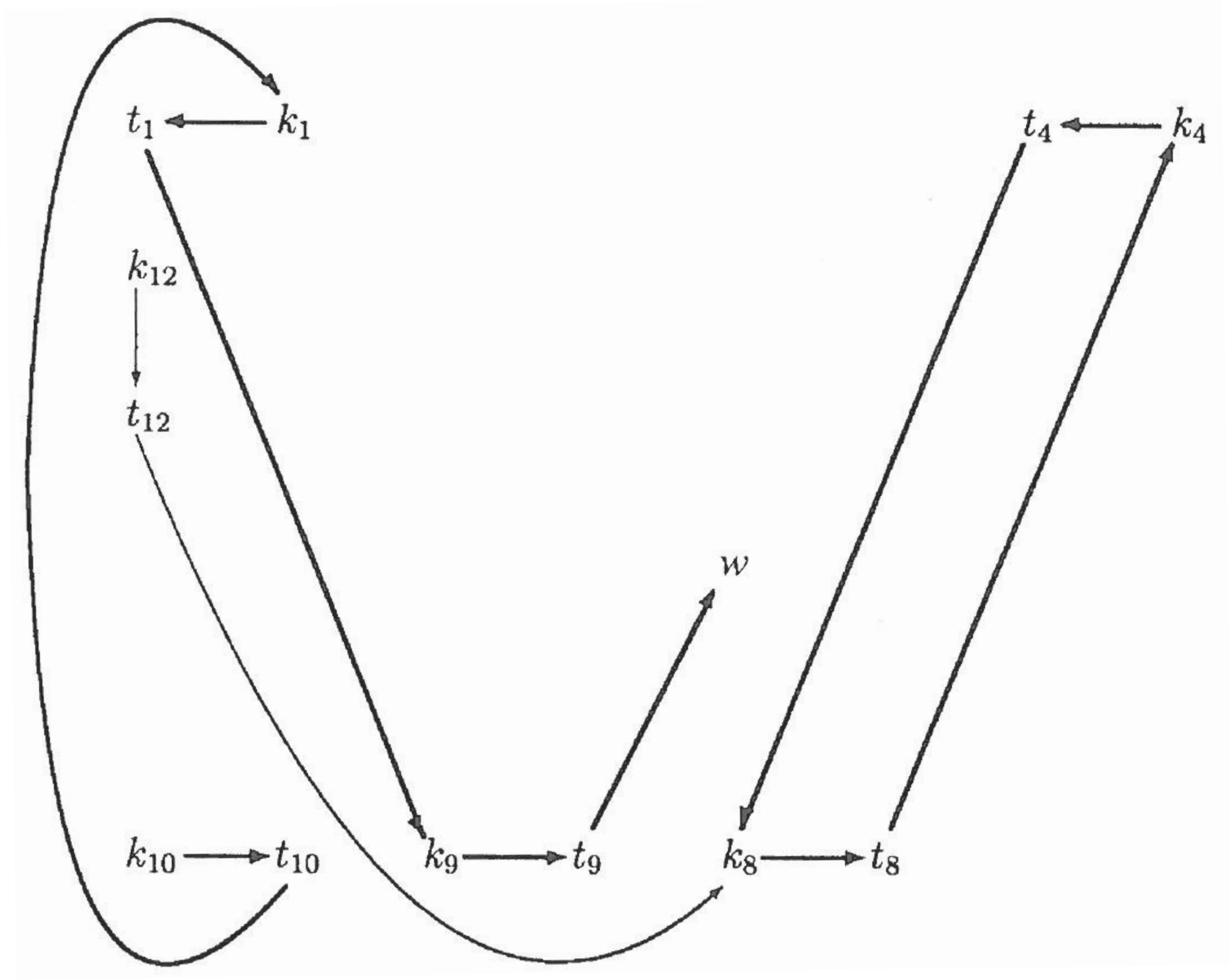


FIGURE IV
Example 1, Round 4

Upon fixing the w-chain W_2, a new cycle $C_3 = (k_4, t_4, k_8, t_8)$ forms. Remove it by assigning k_4 to t_8, and k_8 to t_4.

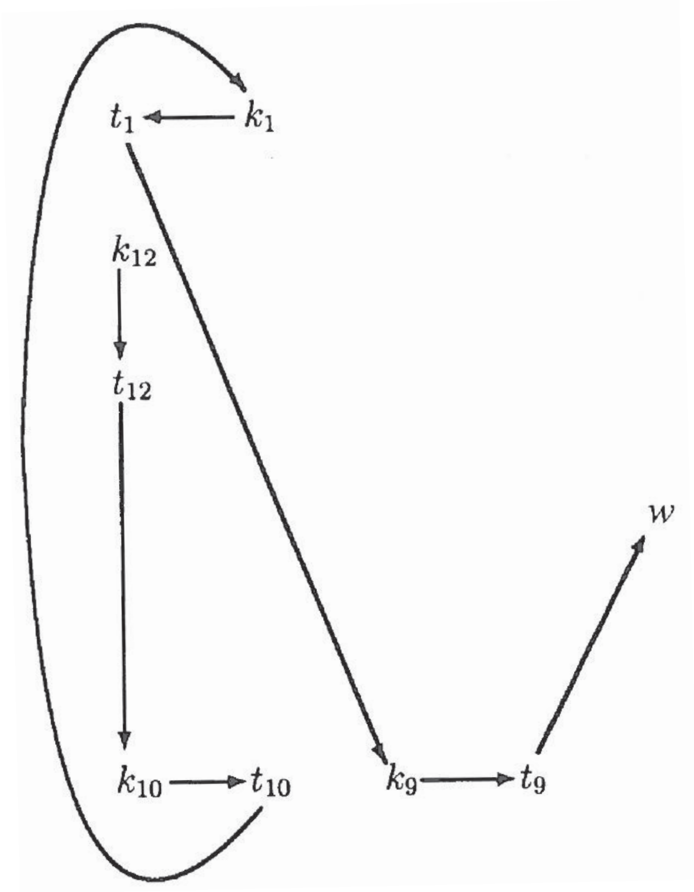


FIGURE V
Example 1, Round 5

No new cycles form, and the pair (k_{12}, t_{12}) "joins" W_2 from its tail to form the longest w-chain $W_3 = (k_{12}, t_{12}, k_{10}, t_{10}, k_1, t_1, k_9, t_9)$. Fix W_3, and assign k_{10} to t_{12}. Since no patient is left, w-chain W_3 is removed, and kidney k_{12} at its tail is offered to the highest priority patient at the cadaveric waiting list.

2.5 Roommate problems and Irving algorithm

A roommate problem involves a set of $2n$ persons, where n is positive integer. Each person has strict preferences over the others.

We define a *matching* to be a partition μ of the people into n pairs. The roommate problem is to find a matching which is stable in the sense that there are no two persons which are not roommates and they prefer each other to their assigned roommates.

The roommate problem and marriage problem are different since the roommates problem allows matching between any two persons in the same group while marriage problem allows matching between one man and one woman in two separate groups.

Example: 6-person roommate problem

The preference lists of the 6 persons are given as follows:

	Preferences				
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
1	4	6	2	5	3
2	6	3	5	1	4
3	4	5	1	6	2
4	2	6	5	1	3
5	4	2	3	6	1
6	5	1	4	2	3

Main goals: Develop an algorithm in identifying ''stable matching'' (divide the 6 persons into 3 pairs) in this roommate problem

Irving algorithm

The algorithm is a series of elimination steps to rule out some impossible pairs that cannot appear in the stable matching.

First phrase
This stage is very similar to the Gale-Shapley algorithm in the stable marriage problem: Each person (say x) takes turn to make proposal to another person (say y) based on his/her own preference (from the first choice to the last choice).

1. When x proposes to y, we have two possible scenarios:
   - If y does not hold any other proposal, then y will keep x's proposal.
   - If y holds a proposal from another person other than x, then y will keep the better proposal and reject the poorer proposal (based on y's preference list). Similar to the marriage problem, each person can hold at most one proposal only.
2. If x's proposal is rejected by y, then x makes another proposal to his next preference and repeats the process until his proposal is accepted by someone or finds no one to accept the proposal.

This is similar to the man-oriented marriage matching where each player (serving as man) treats other players as women (dual role as man and woman). On one hand, his proposal is accepted (best match); on the other hand, he accepts someone's proposal (worst match).

The iteration stops when either

- every person holds a proposal or
- one person is rejected by everyone.

Remark If a stable matching does not exist, then an unlucky guy is not assigned a partner. Given the practical concern that you cannot throw away a student applying for a dormitory room, we need to find a matching solution using another criterion.

Example

Players	*Preference List*
A	B C M
B	C A M
C	A B M
M	arbitrary

As a remark, non-existence of stable solution of the roommates problem is revealed in the first phase already when M is rejected by all others.

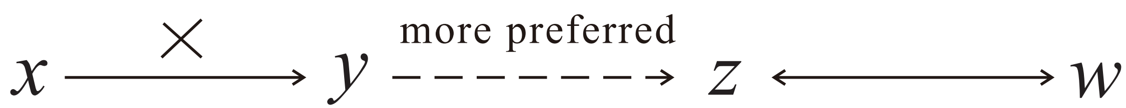
	keeps proposal from	*proposal accepted by*
A	C	B
B	A	C
C	B	A

Proposition

Suppose A rejects B in the proposal sequence, then A and B can not be paired in a stable matching.

Proof

Assume contrary, there are pairs in a stable matching that involve earlier rejection of one partner by another partner in an assigned pair. Suppose that, of all the rejections that involve two students who are partners in some stable matching, the rejection of x by y is the first chronologically. This means y prefers another person z to x and engages with z; that is, z proposes to y earlier than x. In this stable matching M, suppose that x is paired with y and z is paired with w. For stability of M, z must prefer w to y; otherwise (y, z) forms a blocking pair, a contradiction to stability of M. We deduce that z chooses to propose to y after rejected by w. Now, there exists another pair (z, w) where rejection of z by w occurs earlier. A contradiction to the assumption that (x, y) is the first pair is encountered.



By applying the above rule to the numerical example, the results are as follows. Notation: $5 \rightarrow 4, \times 1$, means 5 proposes to 4 and 4 rejects 1 and keeps 5.

$1 \rightarrow 4$ (first choice, kept by 4 temporarily)

$2 \rightarrow 6$ (first choice, kept by 6 temporarily)

$3 \rightarrow 4, \times 3$ (4 prefers 1 to 3, keeps old proposer 1 and rejects new proposer 3) $(3, 4)$ is ruled out

$3 \rightarrow 5$ (goes to the second choice)

$4 \rightarrow 2$ (first choice, kept by 2 temporarily)

$5 \rightarrow 4, \times 1$ (4 prefers 5 to 1, rejects 1) $(1, 4)$ is ruled out

$1 \rightarrow 6, \times 2$ (1 proposes to the next best choice 6, 6 kicks 2 out) $(2, 6)$ is ruled out

$2 \rightarrow 3$ (second choice, kept by 3 temporarily)

$6 \rightarrow 5, \times 6$ (6 proposes to 5 but fails) $(5, 6)$ is ruled out

$6 \rightarrow 1$ (goes to the second choice)

Note that "6" keeps "1" while "1" keeps "6". We expect that $(1, 6)$ forms a pair.

After step 10, each person holds a proposal and the iteration stops.

	Preferences					Proposal
	1st	2nd	3rd	4th	5th	kept
1	4	6	2	5	3	"6"
2	6	3	5	1	4	"4"
3	4	5	1	6	2	"2"
4	2	6	5	1	3	"5"
5	4	2	3	6	1	"3"
6	5	1	4	2	3	"1"

"4" has been proposed by "1", "3" and "5". According to the preferences of "4", "4" keeps "5". Given that "1" and "3" have been rejected by "4", by virtue of the earlier proposition, $(1, 4)$ and $(3, 4)$ cannot form a pair in a stable matching solution.

"6" keeps the proposal from "1" and kicks off the earlier proposer "2". "5" rejects "6" and keeps the proposer "3" at the end of the first phase (worst match) but "5" may want to get better partner.

The purpose of the first phase is to rule out some pairs that never appear in the stable matching solution. Suppose that x is rejected by y, one should expect that (x, y) cannot form a pair since y can find a better partner.

Accordingly, the pairs $(1,4),(2,6),(3,4),(5,6)$ cannot be paired in any stable matching. Hence, one can delete the corresponding entries in the lists and the preference lists reduce to

	Preferences					Proposal kept
	1st	2nd	3rd	4th	5th	
1	~~4~~	6	2	5	3	"6"
2	~~6~~	3	5	1	4	"4"
3	~~4~~	5	1	6	2	"2"
4	2	6	5	~~1~~	~~3~~	"5"
5	4	2	3	~~6~~	1	"3"
6	~~5~~	1	4	~~2~~	3	"1"

Best possible match and worst possible match

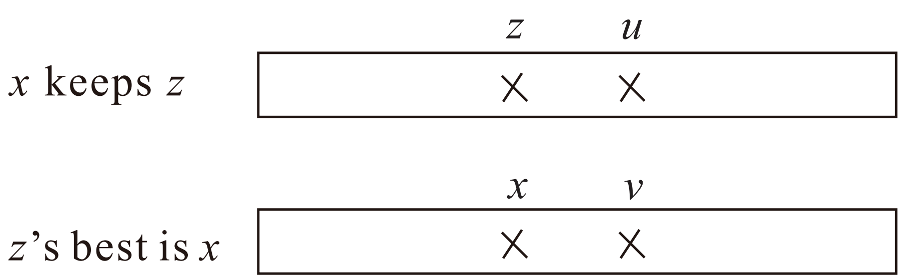
- Suppose that x's proposal is accepted by y after the first phrase, after perhaps several earlier rejections by others, then x cannot have a better partner than y. So y will be the "upper bound" of x's potential partners.

  This is obvious since x's proposal has been rejected by any person w better than y, where x and w cannot be partners. Since the proposal goes sequentially from the top choice to less good choice, the first keeper of x's proposal is the best match for x.

- Suppose x keeps a proposal from another person z after the first phrase. Since x is the "upper bound" of z's potential partners, x is secured to have at least z. Perhaps, x may reject z in the second phase when x can find a better partner. Therefore, x cannot have a partner worse than z. In other word, z will be the "lower bound" of x's potential partners.

Proof of the lower bound property

We prove by contradiction. Suppose that x gets a partner worse (say u) than z in the final roommate allocation. Since z is not paired with x, we let v be z's final roommate. Since z's proposal goes to x in the first phase, it follows that v must be worse than x since x is the upper bound of z's partners. Note that (x, z) can form a blocking pair since x prefers z to u and z prefers x to v, and the matching is unstable.



(x, z) forms a blocking pair

As an example, we consider preference list of person 2 in the example. Since he proposes to 3 and it is kept by "3", while keeps a proposal from 4, so "3" is the best partner that 2 can get and "4" is the worst partner that 2 can get. So partner of person 2 can be 3, 5, 1 or 4.

$$2| \quad 6 \quad \underbrace{3}_{best} \quad 5 \quad 1 \quad \underbrace{4}_{worst}.$$

The above observation allows us to rule out additional invalid pairs. Given the preference list of x and suppose that x gets z's proposal and his proposal is keep by y. We can

– delete all people that are worse than z. At the same time, one can delete "x" from the preference list of person who is deleted.

– delete all people that are better than y. This have been done in the first step already.

For the given example, one can apply this procedure and further reduce the preference lists as follows:

	Preferences					Proposal kept
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	
1		6	~~2~~	~~5~~	~~3~~	"6"
2		3	5	~~1~~	4	"4"
3		5	~~1~~	~~6~~	2	"2"
4	2	~~6~~	5			"5"
5	4	2	3		~~1~~	"3"
6		1	~~4~~		~~3~~	"1"

- For "1", since he keeps "6" as the proposer, so "6" is already the worst partner. We can cancel all players lower in "1"'s preference list.

- Once we know that $(1, 2)$ is out, we can delete "1" from "2"'s preference list.

	Preferences					Proposal kept
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	
1		$\underline{6}$				“6”
2		3	5		$\underline{4}$	“4”
3		5			$\underline{2}$	“2”
4	2		$\underline{5}$			“5”
5	4	2	$\underline{3}$			“3”
6		$\underline{1}$				“1”

The first phrase allows us to reduce the preference lists by eliminating invalid pairs so that one can identify the stable matching in the second phase.

- Only the pair $(1, 6)$ is fixed, while partners for “2”, “3”, “4” and “5” have not yet fixed.

The reduced preference lists obtained from the first phrase algorithm have the following properties:

(i) If x's proposal is kept by y, then x is last on y's list (x is lower bound of y's potential partner). y may improve his choice in later rounds. However, x consider y as the best choice (except for those who have rejected x). y is the upper bound of x's potential partners.

(ii) a appears in b's list if and only if b appears on a's list.

When there is only one person in each list in the reduced preference lists, a stable matching is achieved by assigning the person to the sole person in his list. This is seen in the pair $(1, 6)$.

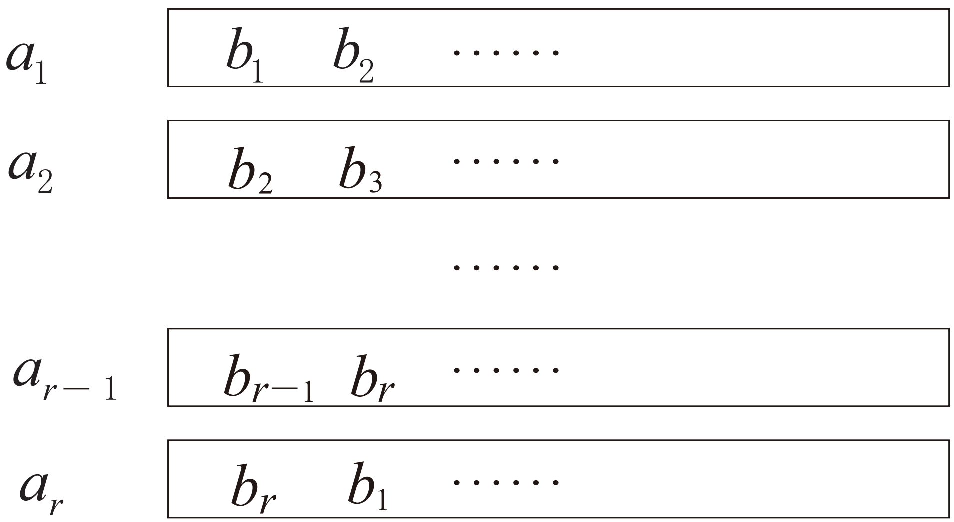
Based on the proposals kept by them, we observe that each person $(2, 3, 4, 5)$ keeps the proposal from the worst person. In the context of stable matching, each person tries to find a partner as good as possible. The purpose of the second phrase of *rotational elimination* is to find whether it is possible for them to identify better partners.

All-or-nothing cycle

Given a reduced preference lists, we let $\{a_1, a_2, \ldots, a_r\}$ be a subset of people drawing from the set of people whose lists contain more than 1 person. We let b_i be the first preference of a_i.

The sequence $\{a_1, a_2, \ldots, a_r\}$ is an all-or-nothing cycle if and only if

- for any $i = 1, 2, \ldots, r-1$, the second person in a_i's current preference list is the first person in a_{i+1}'s;
- the second person in a_r's current reduced preference list is the first in a_1's.



How to find all-or-nothing sequence? It amounts to finding a cycle where each person can secure the second best from his list.

One can adopt the following algorithm to identify such a cycle:

Step 1: Choose an arbitrary person a_1.

Step 2: Find b_2, which is the second person on a_1's list.

Step 3: Once b_2 is identified, find its worst match. This is a_2. Accordingly, b_2 is the first person in a_2's list.

Step 4: (General Case) Given a_i, we find b_{i+1}, which is the second person in a_i's list. We determine a_{i+1} such that b_{i+1} is the first person on a_{i+1}'s list.

Repeat Step 4 until the iterates a_j' start to repeat. The all-or-nothing cycle can be obtained by taking out the "repeated component".

2	3	5	<u>4</u>	
3	5	<u>2</u>		
4	2	<u>5</u>		
5	4	2	<u>3</u>	

We consider the reduced lists in the numerical example. We take $a'_1 = 2$. Then we have

i	$\boldsymbol{a_i}$	b_i	b_{i+1}
1	**2**	3	5
2	**3**	5	2
3	**4**	2	5
4	**3**	5	2
5	**4**	2	5
6	**3**	5	2
7	**4**	2	5

The sequence $\{a_1, a_2\} = \{3, 4\}$ is the desired all-or-nothing sequence.

How can all-or-nothing algorithm improve the outcome?

Given an all-or-nothing cycle $\{a_1, a_2, \ldots, a_r\}$ and the preference list of each of $a_i s$ (shown below),

	Preferences				
	1^{st}	2^{nd}	3^{rd}		
$\boldsymbol{a_1}$	b_1	b_2	$\cdots$		
$\boldsymbol{a_2}$	b_2	b_3			
$\boldsymbol{a_3}$	b_3	b_4			
$\vdots$					
$\boldsymbol{a_r}$	b_r	b_1			

	Preferences				
	1^{st}	2^{nd}	3^{rd}	Last	
$\boldsymbol{b_1}$			$\cdots$	a_1	
$\boldsymbol{b_2}$				a_2	
$\boldsymbol{b_3}$				a_3	
$\vdots$					
$\boldsymbol{b_r}$				a_r	

Recall that while b_1 is the best match for a_1, however, a_1 is the worst match for b_1. Since a_1 cannot compete better than a_r, so a_1 is rejected by b_1. Next, a_1 proposes to b_2 (second best person in a_1's list).

One can improve his partner as follows:

- We start with person b_1. b_1 can improve his partner by rejecting a_1's proposal. Recall that a_1 proposes to b_1 who is the first preference in $a_1's$ list while a_1 is the worst choice in b_1's list. To close the loop, we observe that $\mathbf{b_1}$ **accepts** $\mathbf{a}_r$.
- Since a_1 is rejected by b_1, a_1 proposes to b_2 (second best).
- b_2 will accept a_1's proposal and reject a_2's proposal since a_1 is better than a_2. Recall that a_2 is the worst choice for b_2.
- Since a_2 is now rejected by b_2, a_2 proposes to b_3.
- By continuing the process, a_r is rejected by b_r and a_r proposes to b_1. Lastly, b_1 accepts a_r and b_1 has rejected a_1 earlier.

We need to ensure that the sequence is *cyclic* so that b_1 (initial one) is able to get a new partner after rejecting a_1.

	1st	2nd	3rd	4th	5th
$a_1 = 3$		5			2
$a_2 = 4$	2		5		

	Preferences				
		New proposal received		Last proposal rejected	
$b_1 = 5$		4	$\cdots$	3	
$b_2 = 2$		3		4	

In the new round, 3 proposes to 2 (next best) and 4 proposes to 5 (next best);
Both 2 and 5 are happy since both get their better choice.

From the example, we see that $\{a_1, a_2\} = \{3, 4\}$ is the all-or-nothing cycle.

- Since "5" (b_1) is the first preference of "3" (a_1) in the reduced list, we force "5" to reject "3"'s proposal. (So 3 and 5 cannot form a pair by Lemma 1)

- Then "3" proposes to "2". Then "2" accepts "3"'s proposal and rejects "4"'s proposal (2 and 4 cannot form a pair by Lemma 1).

- Then "4" proposes to "5" and "5" accepts "4"'s proposal and reject "3".

Here, we observe that both "5" (b_1) and "2" (b_2) can get a better proposal (partner). That is,

- "5" receives "4"'s proposal (instead of "3")

- "2" receives "3"'s proposal (instead of "4")

Since each of “2” and “5” have received a better proposal, so each of them rejects the old proposal that is held by him (i.e. “5” rejects “3” and “2” rejects “4”). We can further delete the corresponding entries in the reduced preference lists.

	Preferences					Proposal held
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	
1		$\underline{6}$				“6”
2		$\underline{3}$	5		~~4~~	“3”
3		~~5~~			$\underline{2}$	“2”
4	~~2~~		$\underline{5}$			“5”
5	$\underline{4}$	2	~~3~~			“4”
6		$\underline{1}$				“1”

Any entries that are behind the proposal being held should be deleted since the proposal held is already the worst partner. We then delete "5" from the list of "2" and "2" from the list of "5", the final list becomes

	Preferences					Proposal held
	1st	2nd	3rd	4th	5th	
1		6				"6"
2		3				"3"
3					2	"2"
4			5			"5"
5	4					"4"
6		1				"1"

Since there is only one entry in each list, thus we conjecture that the pairs $(1,6)$, $(2,3)$ and $(4,5)$ constitutes the stable matching.

Wisdom of the Irving algorithm

The Irving algorithm breaks the deadlock between {proposer is engager's worst choice} and {engager is proposer's best choice}.

- After the first phase, each person knows his best potential partner (who accepts his proposal) and his worst potential partner (whose proposal has been engaged).

- In the second phase, the proposers go down to their *second* best partner and seek for cyclic relations between proposers and engagers. With the new proposers, engagers improve their choices. On the other hand, the proposers go down their ranked lists of potential partners by one. The iteration ends when every person has only one partner in his list.

- Stable matching is guaranteed since the ranked list choices go down from the best choice one by one. Blocking pair would not exist. Hopefully, this is intuitively acceptable.

Multiple stable roommate matching solutions

Stable solution may not exist. It can be proven rigorously that if A is any stable roommate assignment, then there is an execution of Irving's algorithm that produces A. For the following 8-person roommate problem, a former MAEC student tried with different choices of a_1, (starting person in the second phase) and managed to obtain 3 stable matching solutions:

1	2 5 4 6 7 8 3
2	3 6 1 7 8 5 4
3	4 7 2 8 5 6 1
4	1 8 3 5 6 7 2
5	6 1 8 2 3 4 7
6	7 2 5 3 4 1 8
7	8 3 6 4 1 2 5
8	5 4 7 1 2 3 6

The 3 stable matching solutions are

$\{(1,2),\ (3,4),\ (5,8),\ (6,7)\}$,

$\{(1,4),\ (2,3),\ (5,6),\ (7,8)\}$,

$\{(1,5),\ (2,6),\ (3,7),\ (4,8)\}$.