Available online at www.sciencedirect.com
ScienceDirect
Journal of Economic Theory 145 (2010) 1739–1756
JOURNAL OF Economic Theory
www.elsevier.com/locate/jet

The roommates problem revisited

Thayer Morrill *,[1]

Department of Economics, North Carolina State University, Raleigh, NC 27695, United States

Received 31 October 2008; final version received 11 January 2010; accepted 18 January 2010

Available online 18 February 2010

Abstract

One of the oldest matching problems is Gale and Shapley's (1962) [8] "roommates problem": is there a stable way to assign $2N$ students into N roommate pairs? Unlike the classic marriage problem or college admissions problem, there need not exist a stable solution to the roommates problem. However, stability ignores the key physical constraint that roommates require a room and is therefore too restrictive. This motivates a new matching problem: matching agents subject to an initial assignment. A particularly important example is kidney exchange where after an assignment has been made, subsequent tests may determine that a patient and donor are incompatible. This paper introduces an efficient algorithm for finding a Pareto improvement starting from any *status quo* roommates assignment.
© 2010 Elsevier Inc. All rights reserved.

JEL classification: C78; D02

Keywords: Matching; Roommates problem; Market design; Kidney exchange; Edmonds' blossom algorithm

1. Introduction

Two-sided matching theory has been well studied by economists, mathematicians, and computer scientists who have created an elegant and applicable theory. Unfortunately, one-sided

* Fax: +1 919 515 5613.
E-mail address: thayermorrill@gmail.com.

[1] This is the first chapter of my Ph.D. thesis at the University of Maryland-College Park. I am grateful to my adviser Lawrence Ausubel for his guidance and support throughout the project. I would also like to thank Peter Cramton, Melinda Morrill, Daniel Vincent, Daniel Aromi, Emel Filiz Ozbay, Jonah Gelbach, and Erkut Ozbay. Two anonymous referees provided a number of thoughtful comments that greatly benefited this paper.

0022-0531/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jet.2010.02.003

matching theory has been comparatively neglected.[2] This neglect is possibly due to the very paper that introduced it. In their classic 1962 article *College Admissions and the Stability of Marriage*, Gale and Shapley introduce both the marriage problem and the roommates problem. While Gale and Shapley prove a stable match always exists in a two-sided market, they introduce the roommates problem to demonstrate that a stable pairing need not exist in a one-sided market. Since a stable match need not exist, economists have been stymied in their attempts to find and analyze solutions to this important assignment problem.[3] Unfortunately, this has led many economists to turn their attention elsewhere, and as a result, the economics literature on this classic problem is sparse.

This paper introduces an alternative solution concept to stability for a wide class of matching models including the roommates problem. Gale and Shapley define a set of marriages as unstable if either there exist a man and woman who are not married but prefer each other to their current spouse or there exists someone who would prefer to be single than married to her current partner. Stability in the roommates problem is borrowed from the marriage model. A pairing is unstable if two students prefer to live with each other rather than their current assignment.[4] This paper is concerned with the following question: *ex-post*, what types of coalitions will be able to block a given assignment? A key consideration is whether bilateral approval is required to dissolve the original partnership. For example, roommates face an additional constraint that married couples do not; roommates must have a room in which to live. A student may prefer another to her assigned roommate; however, she needs a room in which to live and presumably does not have the right to evict her current roommate. Therefore, after an assignment has already been made, the traditional notion of stability is too restrictive.

I will present Gale and Shapley's original example to highlight this point.

Example *(A stable assignment need not exist).* (See Gale and Shapley [8].) Suppose there are four students: α, β, γ and δ. α's top choice is β, β's top choice is γ, γ's top choice is α, and all three rank δ last. Gale and Shapley define an assignment to be unstable if two students are not currently roommates but prefer each other to their current assignment. Under this definition, there does not exist a stable assignment since whoever is assigned to δ prefers the other two students to δ and is the top choice of one of these students. In the words of Gale and Shapley:

> "... whoever has to room with δ will want to move out, and one of the other two will be willing to take him in."

[2] Roth and Sotomayor [15] is an excellent introduction to the two-sided matching literature. Gusfield and Irving [9] is also a nice introduction. Interestingly, although the economics literature on the roommates problem is very small, there is a comparatively large computer science literature on it. Roth and Sotomayor, two economists, mention the roommates problem only as an example. In contrast, Gusfield and Irving, two computer scientists, devote nearly a quarter of the book to the roommates problem. Finding a traditionally-stable roommate pairing (if one exists) is considered a "hard" algorithmic question. The bulk of their presentation is a polynomial-time algorithm for finding a traditionally-stable pairing when one exists. Tan [17] establishes a necessary and sufficient condition for the existence of a stable pairing. Chung [5] extends Tan's result to a sufficient condition for the existence of a stable pairing when preferences are weak.

[3] Another reasonable explanation for the lack of attention to the roommates problem is that economic applications of the marriage problem are far more common than applications of the roommates problem.

[4] I am interested in the case where each student is required to have a roommate. Consequently, I do not include in my definition of stability the additional requirement that each student prefers her assignment to being unassigned.

While one of the other two may be willing to take him in, it is quite a different matter whether this student is able to take him in. In order to take him in, either his current roommate must voluntarily leave, be evicted, or an additional room must be available. With a scarcity of rooms and with no student willing to change his assignment to δ, the original assignment will not be disturbed after all.

This does not contradict Gale and Shapley's original analysis but rather suggests a new class of assignment problems. Gale and Shapley looked at agents and preferences and asked if there exists a stable assignment. This paper looks at agents, preferences, and a *status-quo* assignment and asks the following question: if each agent has the option to maintain her *status-quo*, what is the solution to this coalitional game?

If the *status-quo* assignment can be dissolved unilaterally, then stability is the natural solution concept. If an agent finds someone she prefers who also prefers her, then both parties will dissolve their current partnership and pair together. However, if a partnership requires bilateral agreement to dissolve, then two agents wanting to change their assignment is not enough to block the original pairing. If bilateral agreement is required, an assignment will only be changed if all reassigned parties agree. Since an agent will only agree if the new assignment makes her better off, any deviation from the original set of assignments must be a Pareto improvement. Therefore, when bilateral agreement is required to dissolve a partnership, a natural solution concept is Pareto optimal, Pareto improvements of the initial assignment.

The theoretical contribution of this paper is to introduce a new class of matching problems: matching agents subject to an initial assignment. These problems are important to market design as they extend several classic matching theory applications to a dynamic environment. Typically, a match is made based on a fixed population and set of preferences. However, in real-world applications invariably both preferences and the population being matched change. This is precisely the scope of this paper: a *status-quo* assignment exists and must be honored, but the primitives of the problem have changed so that the match may no longer be a steady state.

The practical contribution of this paper is to present a fast algorithm for Pareto improving any inefficient roommates assignment. This algorithm will be important for extending any application of the roommates problem to a dynamic environment. Section 3.1 discusses a particular important application of this algorithm to kidney exchange.

The paper is organized as follows. Section 2 formally introduces the problem. Section 3 details the Roommate Swap algorithm and discusses applications. Section 4 examines the strategic implications of mechanisms for selecting a core assignment, and Section 5 concludes. Appendix A provides several technical proofs and a discussion of the computational complexity of the Roommate Swap algorithm.

2. The model and applications

2.1. Matching subject to an initial assignment

This section formally introduces the coalitional game.[5] Each game consists of the triple $(S, \succ, \mu)$. $S = \{s_1, \ldots, s_{2N}\}$ is a set of students. The second element, $\succ = (\succ_1, \ldots, \succ_{2N})$ is the list of preferences of each student where $\succ_i$ denotes the strict preferences of student i over the $2N - 1$ other students. The last element, μ is an initial matching of students. A **matching**

[5] The framework presented here is modeled after Abdulkadiroglu and Sönmez [3].

is an assignment of students such that every student is assigned to exactly one other student and assignments are symmetric. More formally, it is a function from S to S such that for every student s, $\mu(s) \neq s$ and $\mu(\mu(s)) = s$. Let $\mathcal{M}$ denote the set of matchings. For convenience of notation, I fix S and $\succ$ and denote the coalitional game by μ.

Given an initial matching μ, a matching η is **individually rational** if $\eta(s) \succeq_s \mu(s)$ for all $s \in S$. Let $IR(\mu)$ denote the set of individually rational matchings subject to an initial assignment μ. A matching μ is **Pareto efficient** if there does not exist a $\eta \in \mathcal{M}$ such that $\eta(s) \succeq_s \mu(s)$ for all $s \in S$ and $\eta(s) \succ_s \mu(s)$ for some $s \in S$. Let PE denote the set of Pareto-efficient matchings. Given an initial assignment μ, this paper focuses on $IR(\mu) \cap PE$ as a solution concept to the coalitional game.

Note that for any Pareto-efficient assignment μ, $\mu \in IR(\mu) \cap PE$. Therefore there are a number of ways to make an assignment μ, such that $\mu \in IR(\mu) \cap PE$. One mechanism with particular desirable properties is the *Serial Dictatorship*.[6] Assign every student a priority (randomly or otherwise). Assign the student with highest priority her most preferred roommate and remove them both from consideration. From students who remain, assign the student with highest priority her most preferred roommate among those students that are unassigned. Remove these two from consideration and repeat until no students remain. If μ is the assignment that results from the Serial Dictatorship, then $\mu \in IR(\mu) \cap PE$.[7]

In contrast to the ease of making a Pareto-efficient assignment, it is difficult to determine if a given roommates assignment is efficient. Preferences between students need not interact when assigning students, but they interact directly when determining if one assignment Pareto improves another. At this point the reader may object as there is an obvious algorithm to determine if an assignment is Pareto efficient: look at each possible reassignment and determine if it Pareto improves the original. If no assignment Pareto improves the original, then the original is efficient. Unfortunately, this algorithm is of no practical use as the growth of the number of assignments relative to students being assigned is factorial. Specifically, given $2N$ students there exists $\frac{(2N)!}{2^N(N!)} = (2N-1)(2N-3)(2N-5)\cdots(3)(1)$ many ways of assigning them to be roommates.[8] Even for small N, this is prohibitively large. For example, there exists on order of 6 quadrillion (6×10^{15}) many ways to assign 30 students to be roommates. Therefore, a more sophisticated process is required.

3. The Roommate Swap algorithm

This section demonstrates an $O(n^2)$ algorithm for determining if a roommate assignment is efficient. Moreover, when an assignment is inefficient I provide an $O(n^3)$ algorithm, the Roommate Swap, for finding a Pareto improvement.[9]

[6] See Abdulkadiroglu and Sönmez [3].

[7] To see that $\mu \in PE$, note that if a student is involved in a Pareto improvement, then necessarily her roommate must be involved as well. The student with highest priority, s_1, receives her top choice, s_2, so neither she nor her choice can be involved in a Pareto improvement. Let s_3 be the student who chooses second. Since neither s_1 or s_2 are involved in any Pareto improvements, if s_3 is part of a Pareto improvement she must be reassigned to a student among $S \setminus \{s_1, s_2\}$. However, s_3 already receives her top choice among this set. Therefore, s_3 (and consequently the student she chooses) is not part of any Pareto improvement. Similarly, the student who chooses third is not part of any Pareto improvement, and so on. Of course, for all $\mu \in \mathcal{M}$, $\mu \in IR(\mu)$.

[8] A short proof appears in Appendix A.

[9] A discussion on the computational complexity of the algorithm appears in Appendix A.

The extra structure inherent in a two-sided match makes it simple to find a Pareto improvement for an inefficient assignment. I present one algorithm here as it provides intuition for the more complicated one-sided case.[10] For a given man m, define a woman w to be **achievable** for m if w weakly prefers m to her current husband. Have each male point to his most-preferred, achievable woman. Note that every man has a woman to point to as his current assignment is achievable. Have each woman point to her current assignment. There must exist a cycle since there are only a finite number of agents and each agent is pointing to someone. If the cycle is trivial (the man is pointing to the woman he is currently married to), then neither the man nor the woman can be involved in a Pareto improvement and we can remove them from consideration. If the cycle is non-trivial, then reassigning each man in the cycle to the woman he is pointing to is a Pareto improvement.

The arrow from a man to an achievable woman ensures that both prefer the new match. The arrow from a woman ensures her old match will be reassigned. Therefore, a non-trivial cycle that alternates between "male" arrows and "female" arrows will be a Pareto improvement. Moreover, the structure inherent in a two-sided match guarantees that any cycle will alternate between these two types of arrows. This need not be the case for a one-sided match, but the intuition still holds: a cycle that alternates between a preferred, achievable agent and her old assignment is a Pareto improvement.

I will define a graph to aid our search for a Pareto improvement.[11] Given a set of preferences $\succ$ and assignment μ, induce a graph, $G^{\mu}_{\succ}$, as follows:

- Each vertex corresponds to a student. Label the vertices s_1 through s_n. When referring to the graph, I will use the term vertex and student interchangeably.
- A *solid* edge is drawn between roommates. By definition, each vertex is incident to exactly one solid edge.
- Draw a *dashed* edge between any two students that form a blocking pair to μ. That is to say, if s_i prefers s_j to her current roommate and vice versa.

When the preferences and assignment are clear from the context, I will just refer to the graph as G. I will call a path that alternates between dashed and solid edges (or vice versa) an **alternating path**. Similarly, a cycle that alternates between dashed and solid edges is an **alternating cycle**. Given an assignment μ, a set of students X is **closed under roommates** if $s \in X$ implies $\mu(s) \in X$.

Lemma 1. *An assignment μ is Pareto efficient under preferences $\succ$ if and only if $G^{\mu}_{\succ}$ contains no alternating cycle. Moreover, if μ' Pareto improves μ and s is a student such that $\mu(s) \neq \mu'(s)$, then s is contained in an alternating cycle in $G^{\mu}_{\succ}$.*[12]

The intuition for sufficiency is captured in Fig. 1. In an alternating cycle, we can simply "swap" roommates. We eliminate the solid edges, make the dashed edges in the cycle solid,

[10] This algorithm is a slight variation of the Top-Trading Cycles algorithm (Shapley and Scarf [16]) and is very similar to the algorithm presented in Erdil and Ergin [7].

[11] I refer the reader to *Introduction to Graph Theory*, second edition, by Douglas West for a detailed introduction to graph theory.

[12] This lemma was discovered independently by Abraham, D.J. and Manlove, D.F. (2004), "Pareto optimality in the roommates problem," mimeo, University of Glasgow.

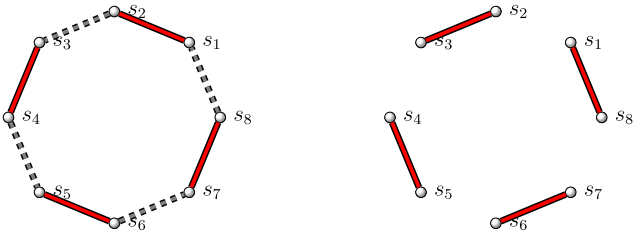


Fig. 1. An alternating cycle with its corresponding Pareto improvement.

and leave everyone outside the cycle unchanged. This is a well-defined reassignment that Pareto improves the initial assignment.

Proof. Suppose $G^{\mu}_{\succ}$ contains an alternating cycle C. An alternating cycle is closed under roommates as each vertex is incident to a solid edge in the cycle. This implies $V(G) \setminus C$ is closed under roommates as well ($V(G)$ denotes the vertex set of G). We will construct a Pareto improvement μ'. For every $v \in V(G) \setminus C$ let $\mu'(v) = \mu(v)$. This is well defined since $V(G) \setminus C$ is closed under roommates. For every $v \in C$, let $\mu'(v)$ be the vertex it shares a dashed edge with in the cycle C. This is well defined as each vertex is incident to exactly one dashed edge in the cycle and sharing a dashed edge is a reciprocal relationship. A dashed edge indicates that both vertices prefer each other to their original assignment. Therefore, μ' Pareto improves μ.

Suppose that μ' is a Pareto improvement of μ. Let G' be the subgraph consisting of all solid edges in $G^{\mu}_{\succ}$ and only the dashed edges between vertices not paired by μ that are paired by μ' (since μ' is a Pareto improvement, there must be a dashed edge between such vertices). Note that any vertex v in G' either has degree[13] 1 (if $\mu(v) = \mu'(v)$) or degree 2 (if $\mu(v) \neq \mu'(v)$). Moreover, for any vertex v, if $d(v) = 2$, then $d(\mu(v)) = d(\mu'(v)) = 2$. Choose any vertex t such that $d(t) = 2$. t is connected via a solid edge to $\mu(t)$. Since $d(t) = 2$, $d(\mu(t)) = 2$ and so $\mu(t)$ must be connected via a dashed edge to $\mu'(\mu(t))$. $\mu'(\mu(t))$ must be connected via a solid edge to $\mu(\mu'(\mu(t)))$ which must be connected to a dashed edge via $\mu'(\mu(\mu'(\mu(t))))$, and so on. Eventually this process must cycle as there are a finite number of vertices. However, a cycle to any vertex $s \neq t$ would mean the degree of s is at least three which is not possible. Therefore, the process must cycle back to our first vertex t. Moreover, it must cycle via a dashed edge as we have already exhausted t's solid edge. By construction, this is an alternating cycle. □

Two vertices are **connected** if there is a path between them. A graph is connected if all vertices are connected. For a graph G, a vertex v is a **cut-vertex** if G is connected but $G \setminus \{v\}$ is not. A **block** is a maximal subgraph containing no cut vertex. Note that the subgraph consisting of two vertices and an edge between them contains no cut-vertex, so any edge is either a block or a subset of a block. I will refer to any block containing only two vertices as a **trivial block**. Since every vertex in our graph has at least one edge, this is the smallest block possible. Fig. 2 shows an example where the blocks have been circled.

[13] The degree of a vertex v, denoted $d(v)$, is the number of edges v is incident to.

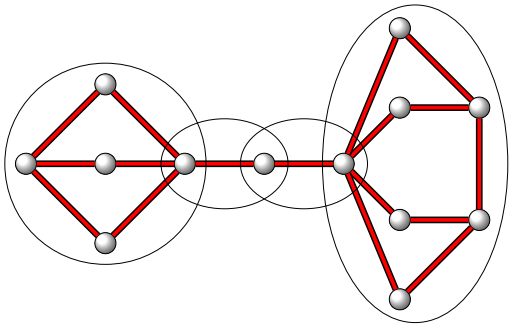


Fig. 2. An example of a graph with four blocks.

Lemma 2. *Let t be any student.*

1. *t and $\mu(t)$ are contained in a unique block, B_t.*
2. *If t is part of an alternating cycle C, then $C \subseteq B_t$.*
3. *If t is involved in a Pareto improvement, then B_t is non-trivial. That is to say if there exists an assignment μ' such that μ' Pareto improves μ and $\mu'(t) \neq \mu(t)$, then $|B_t| > 2$.*

Proof. 1. Since there is an edge between t and $\mu(t)$, they are in at least one block together. Since the intersection of two blocks contains at most one student,[14] t and her roommate must be in exactly one block together. Call this block B_t.

2. A cycle contains no cut-vertex, so it must be a subset of a block. An alternating cycle containing t must contain $\mu(t)$ since t lies on a solid edge in the alternating cycle. Since B_t is the unique block containing t and $\mu(t)$, the cycle must be contained in B_t.

3. If t is involved in a Pareto improvement, then by Lemma 1 t is contained in an alternating cycle. By (2) this alternating cycle is contained in B_t, so B_t must contain more than just t and $\mu(t)$. □

Lemma 2, part (2) says that if a student t is part of a Pareto improvement (and consequently an alternating cycle), then she must be reassigned to a member of B_t. Therefore, no edge between t and a vertex outside of B_t can be part of an alternating cycle. Let G' be the graph obtained by deleting all edges between t and any vertex not in B_t. Then G contains an alternating cycle if and only if G' contains an alternating cycle. This motivates the following procedure.

Pruning a graph.

1. Start with a graph G.
2. Determine the set of blocks $B_1, B_2, \ldots, B_m$.
3. For each student-roommate pair, $\{s, \mu(s)\}$, locate the unique block that both are in. Remove *all* edges from either s or $\mu(s)$ to any student outside this block.

[14] See West p. 156. The intuition is that if two blocks B_1 and B_2 share two vertices, then after cutting a vertex, at least one of the two must remain. Call this vertex v. v is connected to all remaining vertices as it is in a block with each of them. But if every vertex has a path to v, then all vertices are connected. Therefore $B_1 \cup B_2$ has no cut-vertex contradicting the maximality of a block.

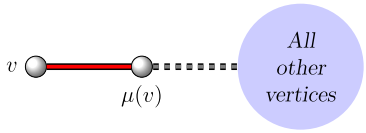


Fig. 3. In a non-trivial block closed under roommates, any vertex v must be incident to at least one dashed edge or else $\mu(v)$ would be a cut-vertex.

By iterating the pruning process we end up with a graph in which all blocks are closed under roommates. This follows because if at any iteration a student s is in a block B with $\mu(s) \notin B$, then Step 3 of the pruning process removes all edges between s and the other vertices in B. After removing the edges, B is no longer connected, and therefore, B is no longer a block. The process only stops when each student s is contained in exactly one block, the unique block containing both s and $\mu(s)$.

The blocks that remain after the pruning process may be trivial, but by Lemma 2, the students in such a block are not involved in any Pareto improvements.

Proposition 1. *Any non-trivial block closed under roommates contains an alternating cycle.*

The algorithm in this proof was inspired by Edmonds' Blossom Algorithm from graph theory[15] and Gale's Top-Trading Cycles Algorithm.[16] Edmonds' Blossom Algorithm has a wide range of applications, but it was originally designed to find a maximum matching for a graph. Given an initial match, the algorithm alternates between edges used in the match and those not used searching for what is called an augmenting path. The key insight of the algorithm is how to get "unstuck" if you ever cycle back to a previously visited vertex. In the current paper's algorithm, we naively alternate between solid and dashed edges until we cycle. We are searching for an even-cycle,[17] so we use Edmonds' trick if we ever get stuck in an odd-cycle.

Proof. Suppose we have a non-trivial block G closed under roommates with $|V(G)| = 2N$. First note that in any non-trivial block closed under roommates, every vertex is incident to exactly one solid edge and at least one dashed edge. Each vertex v has a solid edge to $\mu(v)$ as the block is closed under roommates. There must be additional vertices as the block is non-trivial. If v had no dashed edge, then $\mu(v)$ would be a cut-vertex as removing $\mu(v)$ would disconnect v from these other vertices. (See Fig. 3.)

The proof proceeds by induction on N.

Base step: $N = 2$. Fig. 4 illustrates the argument that follows. The smallest non-trivial block closed under roommates contains four vertices, $\{s_1, \ldots, s_4\}$. Without loss of generality, $\mu(s_1) = s_2$, $\mu(s_3) = s_4$, and there is a dashed edge between s_2 and s_3. If there is a dashed edge between s_1 and s_4 then we are done. If not, then the only remaining possibility is s_1 is adjacent to s_3 and s_4 is adjacent to s_2, but in this case, $\{s_1, s_3, s_4, s_2\}$ is an alternating cycle.

Inductive step. Assume $N > 2$. Our inductive hypothesis is that any graph H that is closed under roommates, has no cut-vertex, and has strictly fewer than $2N$ vertices must contain an alternating cycle.

[15] Edmonds [6]. A discussion of the Blossom algorithm appears in West, p. 142.

[16] Shapley and Scarf [16].

[17] A cycle is even if it contains an even number of vertices.

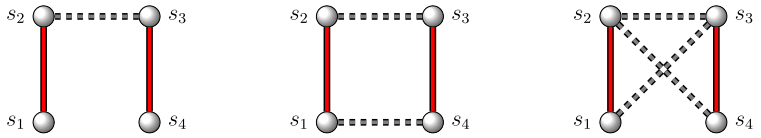


Fig. 4. The base inductive step.

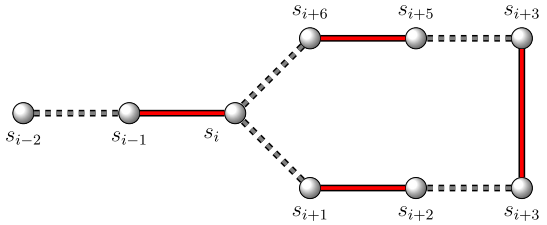


Fig. 5. A "Blossom".

Start with any vertex s_1, take its solid edge to $s_2 = \mu(s_1)$, and continue on a dashed edge to some s_3. Continue alternating between dashed and solid edges until we cycle. We must eventually cycle since there are a finite number of vertices.

If our cycle is even, then we are done. By construction, an even-cycle alternates between dashed and solid edges and is therefore an alternating cycle. Therefore, assume our cycle is odd, $\{s_i, s_{i+1}, s_{i+2}, \ldots, s_{i+2m}\}$. By construction, any odd-cycle looks like Fig. 5, except possibly of different length. Edmonds refers to this as a blossom. The vertices $\{s_1, s_2, \ldots, s_i\}$ are the stem, s_i is the base of the blossom, and s_i must connect to s_{i+1} and s_{i+2m} via dashed edges. By construction, $\{s_i, \ldots, s_{i+2m}\}$ is an alternating path starting with a dashed edge. Therefore s_{i+2j-1} is connected to s_{i+2j} via a solid edge for $1 \leqslant j \leqslant m$. This implies the blossom minus the base, $\{s_{i+1}, s_{i+2}, \ldots, s_{i+2m}\}$, is closed under roommates as each vertex is incident to a solid edge in our blossom.

Intuitively, the way we proceed is to treat the odd-cycle $\{s_i, \ldots, s_{i+2m}\}$ as a single vertex (I call it a super-vertex) and then continue our search for an alternating cycle. After contracting, the new graph contains strictly fewer vertices than the original graph. Therefore, if we can show the contracted graph is closed under roommates and contains no cut-vertex, then the conditions for the inductive hypothesis are satisfied; this implies the contracted graph must contain an alternating cycle. This is the outline for the remainder of the proof. First, I formally define how to contract an odd-cycle. Second, I verify that the contracted graph is closed under roommates and contains no cut-vertex.[18] At this point, we know by the inductive hypothesis that the contracted graph contains an alternating cycle. The final step of the proof is to verify that any alternating cycle in the contracted graph can be expanded to an alternating cycle in the original graph.

Contract the odd-cycle as follows: replace the set of vertices $\{s_i, \ldots, s_{i+2m}\}$ with the single vertex S_i. (See Fig. 6.) Define a new graph G' with $V(G') = V(G) \setminus \{s_i, \ldots, s_{i+2m}\} \cup S_i$. For all $u, v \in V(G) \setminus \{s_i, \ldots, s_{i+2m}\}$, draw a dashed (respectively solid) edge between u and v if and only if there was a dashed (solid) edge between u and v in G. Add a solid edge between s_{i-1} and S_i, and add a dashed edge between S_i and any other vertex v if and only if

[18] More precisely, the contracted graph contains a subgraph that is closed under roommates and contains no cut-vertex.

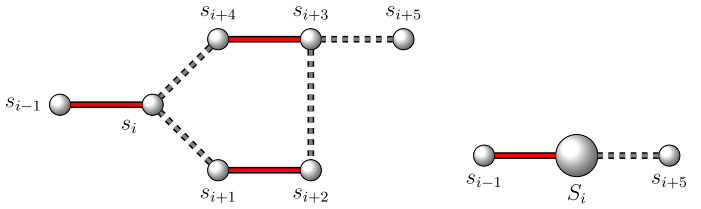


Fig. 6. A blossom before and after contraction.

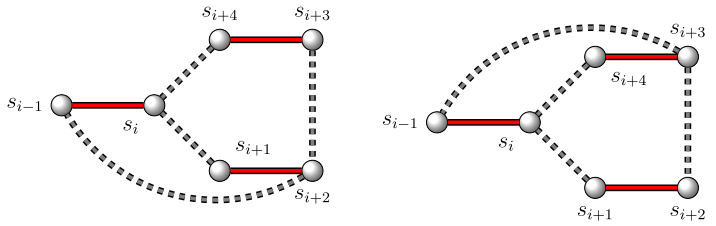


Fig. 7. If there is a dashed edge between s_{i-1} and a vertex in $\{s_i, \ldots, s_{i+2m}\}$, there must be an alternating cycle. By definition, there cannot be a dashed edge between s_{i-1} and s_i.

there was a dashed edge between v and any vertex in $\{s_i, \ldots, s_{i+2m}\}$.[19] s_{i-1} may now have both a solid and a dashed edge with S_i, but if this is the case then we have found our alternating cycle. See Fig. 7 for an example, but more formally, if s_{i-1} has a dashed edge to $s_{i+2j} \in \{s_i, \ldots, s_{i+2m}\}$, then $\{s_{i-1}, s_i, s_{i+1}, s_{i+2}, \ldots, s_{i+2j}\}$ is an alternating cycle. If s_{i-1} is incident to $s_{i+2j+1} \in \{s_i, \ldots, s_{i+2m}\}$, then $\{s_{i+2j+1}, s_{i+2j+2}, \ldots, s_{i+2m}, s_i, s_{i-1}\}$ is an alternating cycle. In either case, we have found an alternating cycle, so we will assume there is no dashed edge between s_{i-1} and any vertex in $\{s_i, \ldots, s_{i+2m}\}$. Therefore, there are no multiple edges in our new graph and it is unambiguous to call s_{i-1} and S_i roommates as they share a solid edge.

It is important to note that the graph $G' \setminus \{S_i\}$ is identical to the graph $G \setminus \{s_i, \ldots, s_{i+2m}\}$. G' may contain a cut-vertex,[20] but we can find a subgraph of G' which contains no cut-vertex and is closed under roommates. A vertex v's component, $\mathcal{C}(v)$, is the set of all vertices v is connected to. Let $\mathcal{C}(s_{i-1})$ be s_{i-1}'s component in the graph $G' \setminus \{S_i\}$.

Claim. *$\mathcal{C}(s_{i-1}) \cup \{S_i\}$ is closed under roommates and contains no cut-vertex.*

Closed under roommates:
s_{i-1} and S_i are roommates and both are in $\mathcal{C}(s_{i-1}) \cup \{S_i\}$. Any other vertex $v \in \mathcal{C}(s_{i-1}) \cup \{S_i\}$ is connected to s_{i-1} by definition of component and connected to $\mu(v)$ by its solid edge. Therefore $\mu(v)$ and s_{i-1} are connected as connected is a transitive relationship. Therefore, $\mu(v) \in \mathcal{C}(s_{i-1}) \cup \{S_i\}$.

No cut-vertex:
S_i cannot be a cut-vertex as $\mathcal{C}(s_{i-1})$ is connected by the definition of component. Look at any $u \in \mathcal{C}(s_{i-1})$. We will show that u is not a cut-vertex by showing that all vertices in

[19] In the graph G, the only solid edge between our odd-cycle and any vertex outside of the odd-cycle is between s_{i-1} and s_i.

[20] In particular, S_i may be a cut-vertex. I would like to thank an anonymous referee for pointing this out.

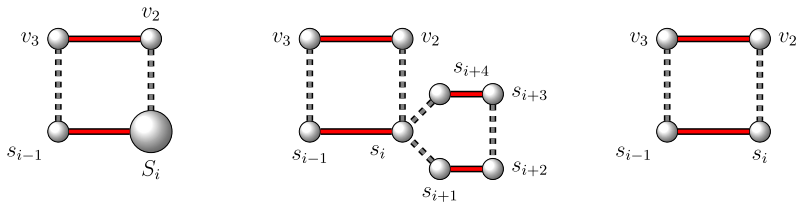


Fig. 8. Expanding a contraction if we enter the odd-cycle through the base.

$\mathcal{C}(s_{i-1}) \cup \{S_i\} \setminus \{u\}$ are connected to S_i and therefore connected to each other. Look at any $v \in \mathcal{C}(s_{i-1}) \setminus \{u\}$. G contains no cut-vertex, so $G \setminus \{u\}$ is connected. Therefore, there is a path in $G \setminus \{u\}$ from v to every vertex in $\{s_i, \ldots, s_{i+2m}\}$. Among all paths in $G \setminus \{u\}$ from v to any vertex in $\{s_i, \ldots, s_{i+2m}\}$, let $\{v, p_1, \ldots, p_n, t\}$ be one of minimal length. This implies that $\{v, p_1, \ldots, p_n\} \cap \{s_i, \ldots, s_{i+2m}\} = \emptyset$, since otherwise there would exist a shorter path to a vertex in $\{s_i, \ldots, s_{i+2m}\}$. Since $\{v, p_1, \ldots, p_n\} \cap \{s_i, \ldots, s_{i+2m}\} = \emptyset$, $\{v, p_1, \ldots, p_n\}$ is a path in $G \setminus \{s_i, \ldots, s_{i+2m}\} = G' \setminus \{S_i\}$. Therefore, $\{v, p_1, \ldots, p_n\} \subseteq \mathcal{C}(s_{i-1})$ as they are all connected to $v \in \mathcal{C}(s_{i-1})$. Moreover, since p_n is adjacent to $t \in \{s_i, \ldots, s_{i+2m}\}$ in G, p_n has a dashed edge with S_i in G'. So indeed, we can remove any u and still find a path that lies entirely in $\mathcal{C}(s_{i-1}) \cup \{S_i\} \setminus \{u\}$ from any v to S_i. Since all vertices in $\mathcal{C}(s_{i-1}) \cup \{S_i\} \setminus \{u\}$ are connected to S_i, all vertices are connected and u is not a cut-vertex. Therefore $\mathcal{C}(s_{i-1}) \cup \{S_i\}$ contains no cut-vertices.

$\mathcal{C}(s_{i-1}) \cup \{S_i\}$ must be non-trivial since s_{i-1} has a dashed edge to some vertex not in $\{s_i, \ldots, s_{i+2m}\}$.[21] Moreover, it contains strictly fewer vertices than G as it is a subset of G' which was the result of contracting three or more vertices into one vertex. Therefore, by the inductive hypothesis $\mathcal{C}(s_{i-1}) \cup \{S_i\}$ contains an alternating cycle.[22]

Since the contracted graph must contain an alternating cycle, the final step in the proof is to demonstrate that any alternating cycle in the contracted graph may be expanded to an alternating cycle in the original graph. Label the alternating cycle $\{v_1, \ldots v_{2n}\}$. Remember $S_i = \{s_i, \ldots, s_{i+2m}\}$ is the blossom we contracted, and s_{i-1} is s_i's roommate. If $S_i \notin \{v_1, \ldots, v_{2n}\}$, then $\{v_1, \ldots, v_{2n}\}$ is an alternating cycle in G. If $S_i \in \{v_1, \ldots, v_{2n}\}$, then $s_{i-1} \in \{v_1, \ldots, v_{2n}\}$ since S_i's solid edge is to s_{i-1}, and $\{v_1, \ldots, v_{2n}\}$ is an alternating cycle. Without loss of generality, assume $s_{i-1} = v_1$ and $S_i = v_2$. Therefore, there is a dashed edge between S_i and v_3. By construction this implies there is a dashed edge between v_3 and a vertex $u \in \{s_i, \ldots, s_{i+2m}\}$. If $u = s_i$, then $\{s_{i-1}, s_i, v_3, \ldots, v_{2n}\}$ is an alternating cycle in G. See Fig. 8 for an example.

If $u = s_j \neq s_i$, then j is either even or odd. If j is even, then $\{s_{i-1}, s_i, s_{i+1}, \ldots, s_j, v_3, \ldots, v_{2n}\}$ is an alternating cycle. If j is odd, then $\{s_{i-1}, s_i, s_{2m}, s_{2m-1}, \ldots, s_j, v_3, \ldots, v_{2n}\}$ is an alternating cycle. See Figs. 9 and 10 for examples. The intuition for this argument can be garnered by looking at Fig. 5, a representative blossom. Since we enter the blossom with a dashed edge, we need an alternating path to s_{i-1} that starts with a solid edge. If we enter via s_i, this is trivial. For any other vertex, it should be clear by inspection that we can always find this by proceeding

[21] Remember, we are assuming s_{i-1} does not have a dashed edge with any vertex in $\{s_i, \ldots, s_{i+2m}\}$ or else we would have already found our alternating cycle.

[22] Proving existence by induction makes the proof cleaner, but it obscures the underlying algorithm for finding the alternating cycle. Now that we have contracted the odd-cycle, we can simply continue alternating between solid and dashed edges until we cycle. If it is an odd-cycle, we contract and continue. If it is an even-cycle, we stop. Every time we contract we have strictly fewer vertices, so eventually the process must conclude with us finding an alternating cycle.

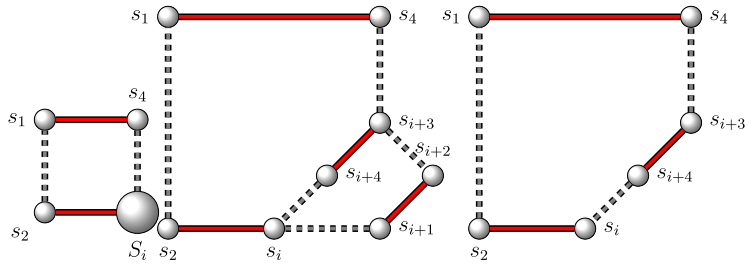


Fig. 9. Expanding a contraction if we enter the odd-cycle on an odd vertex. Continue around the odd-cycle counterclockwise.

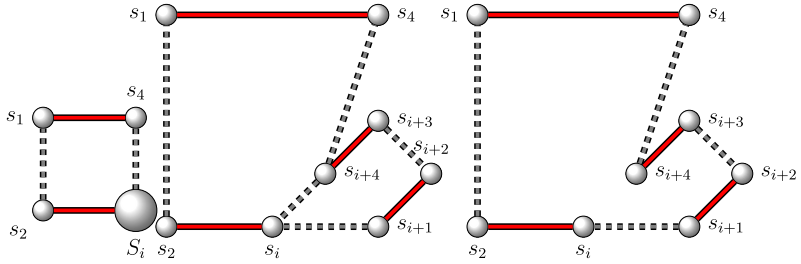


Fig. 10. Expanding a contraction if we enter the odd-cycle on an even vertex. Continue around the cycle clockwise.

either clockwise or counterclockwise around the blossom depending on the parity of the vertex through which we enter. Therefore, we can always expand our contracted blossom to a proper alternating path. □

Given an assignment μ, the proof of Proposition 1 provides an algorithm that determines if μ is Pareto efficient and Pareto improves μ if it is not. Since there can be only a finite number of Pareto improvements, after a finite number of iterations, the algorithm outputs a Pareto-efficient assignment. Label this assignment $RS(\mu)$.

Corollary 2. *For any $\mu \in \mathcal{M}$, $RS(\mu) \in IR(\mu) \cap PE$.*

A natural question to ask is whether or not the Roommate Swap characterizes the entire set $IR(\mu) \cap PE$. When implementing the algorithm, there is never more than one solid edge we may proceed on; however, a student may be incident to multiple dashed edges. On possible decision rule is if there are multiple dashed edges, then choose among the possibilities with equal probability. I call this implementation **the Random Roommates Swap**. Starting with any assignment μ, label the output of the Random Roommates Swap $RRS(\mu)$. The next result affirms that the Random Roommates Swap characterizes the entire set of $IR(\mu) \cap PE$ matchings.

Proposition 3. *Let $\mu \in \mathcal{M}$ be any initial matching. For any $\eta \in IR(\mu) \cap PE$, $RRS(\mu) = \eta$ with a probability that is uniformly bounded away from zero.*

Proof. Let $\eta \in IR(\mu) \cap PE$. If $\eta = \mu$, then μ is Pareto efficient and $RRS(\mu) = \mu$. If $\eta \neq \mu$, then for every student s such that $\eta(s) \neq \mu(s)$, $\{s, \eta(s), \mu(\eta(s)), \eta(\mu(\eta(s))), \ldots, \mu(s)\}$ is an

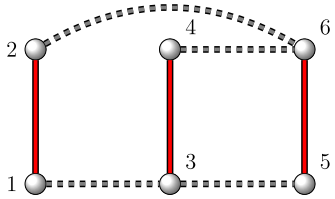


Fig. 11. A non-trivial block closed under roommates, but s_1 and s_2 are not contained in any alternating cycle.

alternating cycle. In any given iteration of the Roommates Swap, the probability we start with s is greater than or equal to $\frac{1}{2N}$. Similarly, the probability the dashed line we chose is to $\eta(s)$ is greater than or equal to $\frac{1}{2N}$. If the length of our alternating cycle is $2m$, then we have m many dashed edges. Therefore, the probability that an iteration selects exactly our alternating cycle is greater than or equal to $\frac{1}{2N}^{m+1}$. Students that have been reassigned under η may form multiple cycles, however these cycles must be disjoint. If the probability we select an alternating cycle of length $2m$ on a given iteration is greater than or equal to $\frac{1}{2N}^{m+1}$, then the probability that we select each cycle in succession is greater than or equal to $\frac{1}{2N}^{N+N}$ □

A given student might like to know if *she* can be part of a Pareto improvement. Unfortunately, Proposition 1 does not generalize to the statement if a student t is contained in a non-trivial block closed under roommates, then t is involved in a Pareto improvement. Fig. 11 is a non-trivial block that is closed under roommates, but s_1 and s_2 are not part of any Pareto improvements.

3.1. Applications

The practical importance of matching theory with an initial assignment is that it enables the market designer to extend a matching theory application to a dynamic environment. For example, in a kidney exchange, preference over a kidney change if tests determine the patient and donor are incompatible. The population being matched changes if a patient passes away. In either instance, the initial assignment must be honored for unaffected patients; however, after the primitives of the matching problem have changed, the initial assignment may no longer be Pareto optimal.

As described in Roth, Sönmez, and Unver, hereafter RSU, [13,14] there are a number of regional kidney-exchange programs. These clearinghouses are designed to make Pareto-efficient matchings that maximize the number of patients receiving a kidney subject to technological and logistical constraints that limit the size of possible exchanges. Although exchanges with more than two agents are routinely performed, two-way exchanges are the most common. RSU [12] introduces a mechanism for two-way exchanges. In a kidney exchange an agent consists of a patient in need of a kidney along with her (incompatible) donor. When exchanges are limited to two agents, it becomes a Roommates Problem. Here I describe two ways in which the Roommate Swap may be used to extend the mechanism proposed in RSU [12].

A patient needing a kidney may be incompatible with a donor if their blood-types are incompatible or if the patient has an antibody against one or more of the donor's proteins (this is called a positive crossmatch).[23] Blood type incompatibility follows a very organized structure: type O

[23] RSU [14] includes a nice description of the factors behind kidney-exchange compatibility.

patients can only receive a kidney from a type O donor, type A patients can only receive a kidney from type O or type A donors, type B patients are compatible with type O and type B, and type AB patients can receive a kidney from all types. However, positive crossmatches are much less predictable as they result from individual donor proteins.

While there are tests to determine if a positive crossmatch exists, these tests often fail. This is especially the case for highly sensitized patients. Unfortunately, positive crossmatches are relatively common. For example, Zenios et al. [19] estimate the probability two strangers will have a positive crossmatch to be 11 percent.[24] Since each assignment requires two donor-patient pairs to be compatible, roughly 20 percent of the pairings in any initial assignment will be incompatible. Therefore, positive crossmatches are a significant obstacle to any kidney-exchange mechanism.

After an assignment is made, compatible matches should not be forced to give up their initial assignment. Therefore, this is exactly the domain of the Roommate Swap. There exists a *status-quo* assignment which must be honored; however, after preferences over kidneys have been updated to reflect positive crossmatches, the initial assignment may not be Pareto efficient. The Roommate Swap determines whether or not the initial kidney-exchange assignment is still Pareto optimal and Pareto improves it if the assignment is not.[25]

A second way in which the Roommate Swap may extend the mechanism in RSU [12] is by enriching each agent's preferences. RSU's design assumes that a patient is indifferent between all compatible kidneys. As they discuss in their paper, there is some controversy over the validity of this assumption. If nothing else, a patient will want to travel the minimum distance possible in order to most easily recover from such a serious surgery. Even if we believe that preferences over kidneys are of secondary importance and that our primary objective should be to maximize the size of any match, the Roommates swap may improve the match quality of the mechanism in RSU [12]. Their mechanism produces an assignment of agents, and this paper's algorithm Pareto improves it if possible.

This paper has focused on a one-sided match subject to an initial assignment. However, matching subject to an initial assignment is relevant to two-sided applications as well. Matching mechanisms are used by the New York City Department of Education[26] and the Boston public school system[27] to assign students to schools. A key difference between these mechanisms is that New York City schools have preferences over students while the Boston public schools do not. For New York City, the 2009–2010 high school match will be made in March and April.[28] In between April and the start of the school year both the preferences of the students and the

[24] The probability of a crossmatch is difficult to measure precisely. RSU [13] used the characteristics of the current waiting list to indicate the scope of the problem. The OPTN/SRTR database catagorizes participants as either low, medium, or high PRA (Percent Reactive Antibody). Low PRA patients have a positive crossmatch with less than 10 percent of the population, medium have a positive crossmatch with 10 to 80 percent, and high percent have a positive crossmatch with more than 80 percent of the population. Roth et al. [13] reports that the distribution of patients on the OPTN/SRTR registry over the years from 1993 to 2002 was 70.19 percent low, 20 percent medium PRA, and 9.81 percent high PRA.

[25] There are at least two downsides to using this paper's algorithm to improve match quality. First, using the algorithm would require shifting a number of patients to a different donor assignment. A legitimate question is whether the improvement in match quality would be worth the time and expense to reassign patients. Second, doctors are now increasingly able to conduct large exchanges, and the Roommates Swap only applies to two-way exchanges. However, even with larger exchanges included, the Roommate Swap may still be used to determine if the set of two-way exchanges is Pareto efficient.

[26] See Abdulkadiroglu, Pathak, and Roth [2].

[27] See Abdulkadiroglu and Sönmez [4], Abdulkadiroglu, Pathak, Roth and Sönmez [1], and Pathak and Sönmez [10].

[28] From http://schools.nyc.gov/ChoicesEnrollment/default.htm accessed January 3, 2010.

student body itself are subject to change. In either case, all students have the right to keep their initial assignment, but the initial assignment may no longer be Pareto optimal. Especially for the New York City school assignment problem where both students and schools have preferences that must be honored, this is now a two-sided match subject to an initial assignment problem.[29]

4. Strategic implications

In this section I examine the strategic implications of any mechanism for selecting a core assignment of the coalitional game μ. Unfortunately, although not surprisingly, I obtain a negative result: there does not exist a mechanism for selecting a core assignment where truthfully reporting one's preferences is always a dominant strategy. This result follows very closely the results for two-sided matching theory presented in Roth and Sotomayor [15].

Following the matching literature, I will use dominant strategy as my solution concept.

Definition 1. A **dominant strategy** is a strategy that is a best response to all possible strategies of the other agents. An assignment mechanism is **strategy proof** if it is a dominant strategy for each agent to reveal her preferences truthfully.

Lemma 3. *There does not exist a strategy-proof mechanism for selecting a Pareto improvement of an inefficient assignment.*

Lemma 3 is proved in Appendix A. This is quite a general result, but it is rather easy to proof. A strategy-proof mechanism must be able to handle any initial assignment and any profile of preferences. Following the path of Roth [11], I demonstrate a case that no mechanism is able to handle.

5. Conclusion

This paper studies a new class of matching problems: matchings subject to an initial assignment. When bilateral approval is required to dissolve a partnership, the core consists of Pareto-optimal matches. A particularly appealing example, and the focus of this paper, is the roommates problem. After an assignment has been made, roommates have common rights to the room. When the assignment cannot be changed without mutual consent, Pareto optimality and not stability defines the core.

The key practical contribution of this paper is to introduce a fast algorithm that determines whether or not a particular roommates assignment is Pareto efficient. If a given assignment is not in the core, the algorithm finds a core assignment. This has an important application to kidney exchange as it enables any assignment mechanism to respond to positive crossmatches.

Appendix A

Lemma 4. *There are* $\frac{(2N)!}{2^N(N!)} = (2N-1)(2N-3)(2N-5)\cdots(3)(1)$ *many ways to assign* $2N$ *students to be roommates.*

[29] The motivating algorithm at the beginning of Section 3 may be used to find a Pareto improvement of an inefficient two-sided match.

Proof. The proof is by induction. When $N = 1$, the result is trivial as there is only one way to assign two students to be roommates. Assume $N > 1$ and by induction there are $(2N-3)(2N-5)\cdots(3)(1)$ many ways to assign $2(N-1)$ many students to be roommates. Select a student s. There are $2N-1$ possible roommates for s, and by assumption, for any roommate we pick, there are $(2N-3)(2N-5)\cdots(3)(1)$ many ways to assign the remaining $2N-2$ many students. Therefore, there is a total of $[2N-1]\times[(2N-3)(2N-5)\cdots(3)(1)]$ many ways of assigning roommates. □

Lemma 3 There does not exist a strategy-proof mechanism for selecting a Pareto improvement of an inefficient assignment.

Proof. Suppose there are four students, a, b, c, and d, and an initial assignment, μ_1 pairing a with b and c with d. Moreover, suppose the student's preferences are as follows.

$$a : c \succ d \succ b$$
$$b : c \succ d \succ a$$
$$c : b \succ a \succ d$$
$$d : b \succ a \succ c$$

With four students, there are three possible assignments. Note that an assignment is completely determined by who a (or any other student) is assigned to. Let μ_2 denote the assignment where a is paired with c and μ_3 denote the assignment where a is paired with d. In our original assignment μ_1, each person is paired with their least preferred roommate, so μ_1 is Pareto dominated by both of the other assignments. Suppose for contradiction that their exists a strategy-proof mechanism M for selecting an efficient, Pareto-improving assignment. Note that if a submits the preferences $c \succ b \succ d$ and all other students submit true preferences, then μ_2 is the only assignment that Pareto improves μ_1 (relative to the submitted preferences). In such a case, M must select μ_2. Similarly, if b submits the preferences $c \succ a \succ d$ and all other students submit true preferences, then M must select μ_3 as it is now the only Pareto-improving assignment. When all students submit true preferences, M must select either μ_2 or μ_3. If M selects μ_2, then b can do better by deviating and submitting the preferences $c \succ a \succ d$. If M selects μ_3, then a can do better by submitting preferences $c \succ b \succ d$. Either way, M is not strategy proof which is a contradiction. □

A.1. Computational complexity

The purpose of this section is to demonstrate that the Roommate Swap is a polynomial time algorithm. I demonstrate that it is at worst an $O(N^3)$ algorithm where N is the number of students.

Each iteration of the algorithm involves the following steps, performed in sequence:

1. *Induce the graph.* This is at worst $O(N^2)$ as a graph is defined by its edges and there are at most $\frac{N(N-1)}{2}$ many edges.
2. *Iteratively prune the graph until all blocks are closed under roommates.* West [18], p. 157, details an $O(N)$ algorithm for determining blocks. We need to iterate at most N times as each iteration eliminates at least one student from each block or stops looking at a block

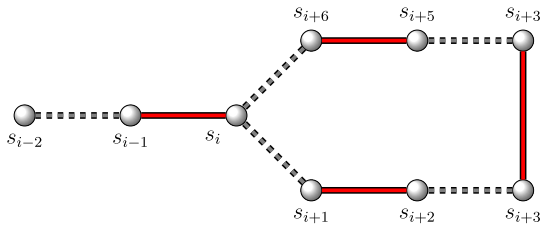


Fig. 12. An odd-cycle.

if it is already closed under roommates. Therefore iteratively pruning the graph is at worst $O(N^2)$.

3. *Find an alternating cycle.* This process is $O(N)$. At each step we either travel to a previously unvisited vertex, which we can do at most N times, or contract a minimum of two vertices, which we can do at most $\frac{N}{2}$ times. So the algorithm must conclude in at most $N + \frac{N}{2}$ steps. As it takes at most N steps to expand a cycle containing super-vertices to a proper cycle, finding an alternating cycle concludes in $O(N)$ time.

Therefore each iteration is $O(N^2)$.

Observation 1. *In each iteration of the Roommate Swap, at least one student can be reassigned her top achievable student.*

Proof. The search process ends with an alternating cycle that may or may not contain super-vertices. Choose the dashed edges from standard vertices to be the vertex's most preferred student among those who prefer her to their current assignment. Therefore, if the alternating cycle contains no super-vertices, then half the students receive their top achievable match. A gray edge from a super-vertex is not necessarily the student's most preferred achievable student. However, if the alternating cycle contains a super-vertex and we need to expand our contractions, then there must be a *last* odd-cycle that needs to be expanded.

Fig. 12 shows a last cycle with six vertices, but the analysis is the same for fewer or greater vertices. Our alternating path must go through s_i and either s_{i+1} or s_{i+6}. None of these edges involve super-vertices (this is our last expansion) so by construction, s_{i+1} is s_i's top achievable student and s_i is s_{i+6}'s top achievable choice. Either way, at least one student receives her top achievable choice. □

The significance of this is that once a student has been assigned her top achievable choice, neither she nor her roommate can ever be involved in another Pareto improvement. Therefore we can eliminate them both from consideration. Since we eliminate at least two students after every iteration, there can be at most $\frac{N}{2}$ iterations.

The algorithm performs $O(N)$ many iterations of an $O(N^2)$ process. Therefore it is, at worst, $O(N^3)$.

References

[1] A. Abdulkadiroglu, P.A. Pathak, A.E. Roth, The New York City high school match, Amer. Econ. Rev. Papers Proc. 95 (2005) 368–371.

[2] A. Abdulkadiroglu, P.A. Pathak, A.E. Roth, Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match, Amer. Econ. Rev. 99 (2009) 1954–1978.
[3] A. Abdulkadiroglu, T. Sönmez, Random serial dictatorship and the core from random endowments in house allocation problems, Econometrica 66 (1998) 689–701.
[4] A. Abdulkadiroglu, T. Sönmez, School choice: a mechanism design approach, Amer. Econ. Rev. 93 (2003) 729–747.
[5] K.-S. Chung, On the existence of stable roommate matchings, Games Econ. Behav. 33 (2000) 206–230.
[6] J. Edmonds, Paths, trees, and ßowers, Canad. J. Math. 17 (1965) 449–467.
[7] A. Erdil, H. I Ergin, What's the matter with tie-breaking? (Improving efficiency in school choice), Amer. Econ. Rev. 21 (2008) 669–689.
[8] D. Gale, L.S. Shapley, College admissions and the stability of marriage, Amer. Math. Monthly 69 (1962) 9–15.
[9] D. Gusfield, R. Irving, The Stable Marriage Problem: Structure and Algorithms, MIT Press, Cambridge, 1989.
[10] P.A. Pathak, T. Sönmez, Leveling the playing field: Sincere and strategic players in the Boston mechanism, Amer. Econ. Rev. 98 (2008) 1636–1652.
[11] A.E. Roth, The economics of matching: stability and incentives, Math. Oper. Res. 7 (1982) 617–628.
[12] A.E. Roth, T. Sönmez, M.U. Ünver, Pairwise kidney exchange, J. Econ. Theory 125 (2005) 151–188.
[13] A.E. Roth, T. Sönmez, M.U. Ünver, A kidney exchange clearinghouse in New England, Amer. Econ. Rev. Papers Proc. 95 (2005) 376–380.
[14] A.E. Roth, T. Sönmez, M.U. Ünver, Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences, Amer. Econ. Rev. 97 (2007) 828–851.
[15] A.E. Roth, M. Sotomayor, Two-Sided Matching: A Study in Game-Theoretic Modelling and Analysis, Cambridge University Press, Cambridge, 1990.
[16] L.S. Shapley, H. Scarf, On cores and indivisibility, J. Math. Econ. 1 (1974) 23–37, 1: 23–28.
[17] J.J.M. Tan, A necessary and sufficient condition for the existence of a complete stable matching, J. Algorithms 12 (1991) 154–178.
[18] D.B. West, Introduction to Graph Theory, Prentice Hall, Englewood Cliffs, 2001.
[19] S.A. Zenios, E.S. Woodle, L.F. Ross, Primum non nocere: avoiding harm to vulnerable waitlist candidates in an indirect kidney exchange, Transplantation 72 (2001) 648–654.