THE ROOMMATES PROBLEM DISCUSSED

NATHAN SCHULZ

Abstract. The stable roommates problem as originally posed by Gale and Shapley [1] in 1962 involves a single set of even cardinality $2n$, each member of which ranks every other member in order of preference. A stable matching is then a partition of this single set into n pairs such that no two unmatched members both prefer each other to their partners under the matching. However, a simple counterexample quickly proves that a stable matching need not exist in the stable roommates problem. In 1984, Irving published an algorithm that determines in polynomial time if a stable matching is possible on a given set, and if so, finds such a matching. However, others have made efforts to redefine the concept of a "stable matching," or even reframe the problem altogether to give it new real-world significance. The present paper describes both Irving's algorithm, and look at other reappraisals of this problem.

Contents

1. Background

In their 1962 paper "College Admissions and the Stability of Marriage," David Gale and Lloyd Shapley propose the stable marriage problem [1]. The problem concerns n men and n women, each of whom is to marry one partner. Each man and woman ranks each woman and man, respectively, from 1 to n in order of preference. A *matching* is defined as a set of n disjoint pairs containing one woman and one man each. For convenience sake, we'll refer to the woman's partner in the matching as her "husband" and the man's partner in the matching his "wife." A matching is *stable* when no woman x prefers a man y to her husband in the present matching where y also prefers x to his present wife. In the paper, Gale and Shapley furnished an algorithm that always provides a stable matching in polynomial time.

In the same paper, Gale and Shapley propose the related stable problem (henceforth, simply referred to as the "roommates problem"). This problem concerns $2n$ participants who each rank the other $2n-1$ members in order of preference. A *matching* in this context is just a set of n disjoint pairs of participants. The two participants in a pair will henceforth be referred to as "partners" in the matching.

Date: August 17, 2008.

In this problem, a matching is *stable* when no participants x and y exist who prefer each other to their present partners. However:

Proposition 1.1. *There need not exist a stable matching in the roommates problem.*

This is demonstrated with an illustrative counterexample.

Proof. Imagine there are four participants, α, β, γ, and δ, where the preference lists are given as:

$$\begin{aligned} \alpha :& \beta, \gamma, \delta \\ \beta :& \gamma, \alpha, \delta \\ \gamma :& \alpha, \beta, \delta \\ \delta :& \text{(Unimportant)} \end{aligned}$$

Then whoever is paired in a room with δ prefers someone in the other room who also prefers them. Therefore, a stable matching cannot always be constructed in the problem. □

Since there need not exist a stable matching in the roommates problem, there cannot be an algorithm that always constructs a stable matching. This fact has led to variety of reappraisals of this problem. This paper looks at a variety of approaches to the problem, including:

(1) Irving's algorithm, which determines in polynomial time if a stable matching is possible on a given set of participants, and if so, finds such a matching,
(2) Morrill's approach, which works by redefining the concept of a stable matching, and

2. Irving's Method Pt. I

Irving's algorithm breaks down into two distinct phases. The first phase consisting of a sequence of "proposals" not unlike the algorithms employed in stable marriage problems [2].

Algorithm 2.1 (Phase 1)**.**
(1) If x receives a proposal from y, then:
    (a) x rejects it at once if x already holds a better proposal;
    (b) x holds it for consideration otherwise, simultaneously rejecting any poorer proposal x currently hold.
(2) An individual x proposes to the others in the order in which they appear in x's preference list, stopping when a promise of consideration is received; any subsequent rejection causes x to continue immediately with the sequence of proposals.

When this first phase terminates, there are two possibilities:

(1) every person holds a proposal
(2) one person is rejected by everyone

Lemma 2.2. *If y rejects x in the proposal sequence described above, then x and y cannot be partners in a stable matching.*

Proof. Suppose that of all the rejections involving two participants who are partners in a stable matching, the rejection of x by y is chronologically first. Denote M a stable matching in which x and y are partners. For y to have rejected x, y must have already held or later received a better proposal, denoted by z. But if y prefers z to x, then z must prefer his own partner in M, denoted by w, to y. Therefore, z must have been rejected by w before z proposed to y. But this rejection must have come before y rejected x, which is a contradiction since the rejection of x by y is assumed to be the first. □

Corollary 2.3. *If, at any stage of the proposal process, x proposes to y, then in a stable matching:*

(1) x cannot have a better partner than y
(2) y cannot have a worse partner than x

Proof. If x has proposed to y, x has been rejected by everyone higher on their preference list than y, and so by Lemma 2.2 may not have a better partner than y, proving (1). If y and z are partners in a stable matching, and y prefers x to z, then since x prefers y to x's own partner, stability is violated. So (2) is demonstrated by contradiction. □

An obvious consequence of Lemma 2.2 is that if the first phase of the algorithm terminates with one person x having been rejected by everyone else, then no stable matching exists, as x could have no possible partner.

Corollary 2.4. *If every person holds a proposal, then we reduce their preference lists as follows: If the first phase of the algorithm terminates with every person holding a proposal, then the preference list of possible partners for y, who holds a proposal from x, can be "reduced" by deleting from it*

(1) all those to whom y prefers x;
(2) all those who hold a proposal from a person whom they prefer to y. In the resulting lists:
(3) y is first on x's list and x is last on y's
(4) in general, b appears on a's list iff a appears on b's.

Proof. (1) and (2) follow from Corollary 2.3. (3) follows from (1) and (2), and (4) is self-evident. □

Lemma 2.5. *If in the reduced preference lists, every list contains one person, then the lists specify a stable matching.*

Proof. This is a consequence of Corollary 2.4 part (4). Suppose x prefers y to the sole participant on x's list. Then x was rejected by y, meaning y had obtained a better proposal. The final proposal held by y is from the remaining participant on y's reduced preference list, meaning y prefers this participant to x. Therefore, the matching specified by the reduced preference lists is never unstable. □

3. Irving's Method Pt. II

In not every case where a stable matching exists are the reduced preferences listed created by the first phase sufficient to denote a matching. We must further reduce the preference lists, making use of an iterative process that will result in either one person running out of people to propose to (in which case no stable matching exists), or every preference list shrinking to a single person, in which case they specify a stable matching.

A set of preference list is said to be *completely reduced* if:

(1) they have been subjected to phase 1 reduction as described in Corollary 2.4,
(2) they have been subjected to zero or more phase 2 reductions, as described below.

The idea behind the reductions that characterize the second phase of Irving's algorithm is to recognize a cycle $a_1, \dots, a_r$ of distinct participants such where:

(1) for $i = 1, \dots, r-1$, the second participant in a_i's current reduced preference list is the first participant in a_{i+1}'s; henceforth denoted by b_{i+1}.
(2) the second participant in a_r's current reduced list is the first in a_1's; henceforth denoted by b_1.

We define $a_1, \dots, a_r$ an *all-or-nothing* cycle relative to the current reduced lists. To find an all-or-nothing cycle, let p_1 be an arbitrary participant whose current reduced list contains more than one participant. Generate the sequences by taking q_i = second person in p_i's list, and p_{i+1} = last person in q_i's list. We then let $a_i = p_{s+i+1} (i = 1, 2, \dots)$, where p_s is the first element in the p sequence that is repeated, and define $p_1, p_2, \dots, p_{s-1}$ as the cycle's "tail."

Algorithm 3.1 (Phase 2)**.** The second phase reduction involves a set of reduced lists and a particular all-or-nothing cycle. This phase involves forcing each b_1 to reject the proposal held from a_1, forcing each a_1 to propose to b_{i+1}, the second person in the reduced list. As a result, all successors of a_1 in b_{i+1}'s reduced lists can be deleted (as in Corollary 2.4, and b_{i+1} can be eliminated from their lists. If a_i achieves no better partner than b_{i+1}, then for stability's sake, b_{i+1} can do no worse than a_i. It follows that parts (3) and (4) of Corollary 2.4 apply also to these further reduced lists.

The significance of a completely reduced set of preference lists is that if the original problem instance allows for a stable matching, then there is a stable matching in which every participant is partnered by someone on their reduced list. We say that such a matching is *contained* in the reduced lists.

Lemma 3.2. *Let $a_1, \dots, a_r$ be an all-or-nothing cycle, and denote b_1 the the first person in a_1's reduced list. Then:*

(1) in a stable matching contained in these reduced lists, either a_i and b_i are partners for all values of i or for no value of i,
(2) if there is such a stable matching in which a_i and b_i are partners, then there is another in which they are not.

Proof. Considering subscripts (modulo r), suppose that for a fixed i, a_i and b_i are partners in a stable matching contained in the reduced lists. Since b_i is second on a_{i-1}'s it follows that a_{i-1} is at least present on b_i's reduced list. Additionally, since a_i is last on b_i's reduced list, it follows that b_i prefers a_{i-1} to a_i. Then for stability, a_{i-1} must be partnered by someone he prefers to b_i, and the only qualifying participant in a_{i-1}'s list is b_{i-1}. Repeating this argument shows that a_i and b_i must be partners for all values of i. We define $A = \{a_1, \dots, a_r\}$, $B = \{b_1, \dots, b_r\}$. If A union B is non-empty, say $a_j = b_k$, then it is impossible for all a_i to have their first remaining preference, since b_k has has last preference when a_k has first. So, the fact that A union B is non-empty implies that no a_i and b_i can be partners, so we may as well assume A union B is non-empty. Suppose M is a stable matching in the completely reduced lists, in which a_i and b_i are partners for all i. Denote M' the matching in which each a_i is partnered by b_{i+1}, and any person not in A intersect B has the same partner as in M. We claim M' is stable. Each participant of B finds a better partner in M' from their point of view than the one in M. The only individuals who do worse are the members of A, so any instability in M' must involve some a_i. If a_i prefers a participant y over b_{i+1}, then there are three cases:

(1) a_i and y were partners in M, in which case, y prefers the new partner, a_{i-1} to a_i.
(2) a_i prefers y to b_i, in which case y is not in a_is reduced list.
(3) a_i prefers b_i to y, in which case y lies between b_i and b_{i+1} in a_i's original preference list, but is not in the reduced list. This means y must have obtained a proposal from someone better than a_i, so x must prefer the partner in M' to a_i.

□

Corollary 3.3. *If the original problem instance allows for a stable matching, then there is a stable matching contained in a completely reduced set of preference lists.*

Corollary 3.4. *If one or more among a reduced set of preference lists is empty, then the original problem instance admits no stable matching.*

The last lemma delineates the circumstances under which a stable matching can be created following Irving's method.

Lemma 3.5. *If in a completely reduced set of preference lists, every list contains just one person, then the lists specify a stable matching.*

Proof. This is as a consequence of Corollary 2.4 (4). Suppose that y prefers x to the participant left on y's list. Then, like in the case (2) in the second part of Lemma 3.2, we demonstrate that x must prefer the participant remaining on x's list. Therefore, no instability is admitted. □

4. One Economist's Idea

The paper "The Roommates Problem Revisited," authored by economist Morrill in 2007 provides a new look at the roommates problem. In the paper he quickly acknowledges the real-life problem posed by the fact that a stable matching need not always exist in the roommates problem. Morrill addresses some practical concerns overlooked by the original definition of stability, stating: "the traditional notion of stability ignores the key physical constraint that roommates require a room, and it is therefore too restrictive," [3]. According to Morrill, the best way to approach this problem on a practical level employs the concept of *Pareto efficiency.*

Definition 4.1. In the context of the roommates problem, a *Pareto improvement* on a matching is a re-pairing of the inhabitants of two or more rooms that doesn't come at the expense of any participants involved. A matching is said to be *Pareto efficient* if no further Pareto improvements can be made.

Proposition 4.2. *For any set of participants there exists a Pareto efficient matching.*

Proof. Assign each participant a priority. Then, in order of priority, ask each participant who they would like to be paired with. Then pair those participants. Continue down the prioritized list until every participant is paired. It then is easy to show that this matching is Pareto efficient. □

The author then describes an algorithm whereby successive changes can be applied to any status quo matching to result in a Pareto efficient matching in $O(n^3)$ time [3]. Though Proof 4 gives an easy method for obtaining a Pareto efficient matching, Morrill's method has its advantages. From a practical perspective, we notice differences in the proved method of re-matching everyone using a prioritized list versus the incremental changes of Morrill's algorithm. Morrill's approach upsets a given status quo less than the total re-matching of the method proven above.

References

[1] D. Gale and L.S. Shapley: "College Admissions and the Stability of Marriage," American Mathematical Monthly 69, 9-14, 1962.

[2] Robert W. Irving. An Efficient Algorithm for the "Stable Roommates" Problem. Academic Press Inc. 1985.

[3] Thayer Morrill. The Roommates Problem Revisited. University of Maryland. 2007.