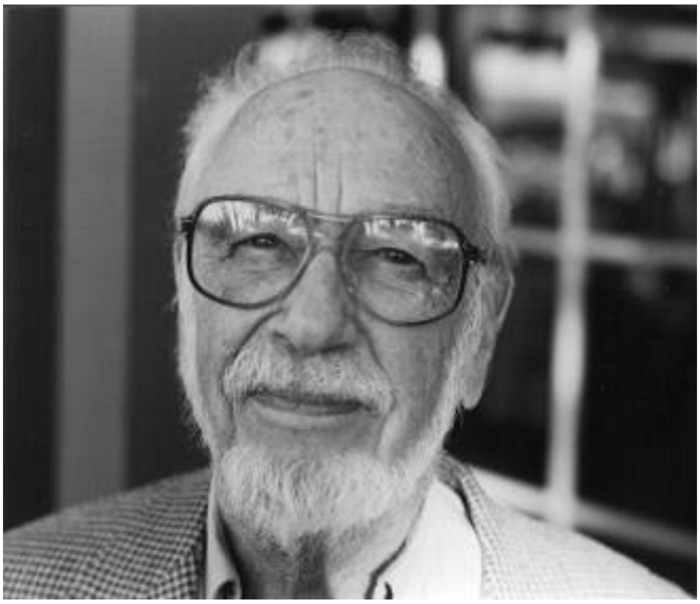
MATCH-UP 2008

Matching Under Preferences
– Algorithms and Complexity –

Satellite workshop of ICALP 2008
July 6, 2008, Reykjavík, Iceland

Sponsored by

School of Informatics, Kyoto University
School of Business, Reykjavík University
Department of Computing Science, University of Glasgow



David Gale, 1921 – 2008

MATCH-UP: Matching Under Preferences

Dedicated to the memory of David Gale, December 13, 1921 – March 7, 2008

Preface

This volume contains the papers presented at the ICALP Workshop "MATCH-UP: Matching Under Preferences", held at the University of Reykjavík on July 6, 2008.

Matching problems with preferences occur in widespread applications such as the assignment of school-leavers to universities, junior doctors to hospitals, students to campus housing, children to schools, kidney transplant patients to donors and so on. The common thread is that individuals have preference lists over the possible outcomes and the task is to find a matching of the participants that is in some sense optimal with respect to these preferences.

The remit of this workshop was to explore matching problems with preferences, with an emphasis on the algorithms and complexity perspective, but a key objective was also to bring together the computer science and economics communities who have tended to follow different paths when studying these problems previously. The timing of the workshop reflects the growing interest in such problems among researchers in these communities that has led to a wealth of publications in the past few years.

The opening talk at the workshop was to have been given by one of the key pioneers in the field, Professor David Gale, of the University of California, Berkeley. Tragically, David Gale died suddenly on 7 March 2008, while preparations for the workshop were under way. We are honoured to dedicate the workshop to his memory.

In addition to David Gale, Professor Kurt Mehlhorn, Max Planck Institute für Informatik, Professor Al Roth, Harvard University, and Professor Marilda Sotomayor, Universidade de São Paolo, all kindly agreed to give an invited talk at the workshop. Marilda Sotomayor agreed to open the workshop by paying tribute to the life and work of David Gale, in particular describing his contribution to the theory of matching problems.

Our call for papers generated much interest, and we were pleased with both the quality and quantity of submitted papers. These originated in roughly equal measure from the computing science and economics communities. The tight time constraints imposed by a one-day workshop made the selection process difficult, and forced us to reject a number of good papers that, in other circumstances, we would have been happy to accept. The final choice of 15 contributed papers appearing in these proceedings represents, we feel, an excellent snapshot of the current state of the art regarding research in the area of matching problems with preferences.

We would like to conclude by thanking the invited speakers and the authors of all submitted papers for helping to make this workshop a success.

Magnús M. Halldórsson
Rob Irving
Kazuo Iwama
David Manlove

ICALP Workshop: Match-Up, July 6, 2008
Workshop Program

Session 1: 08:30 – 10:30 (Chair: Rob Irving)

08:30 Opening remarks
08:40 **Invited talk**: My encounters with David Gale
M. Sotomayor 1
09:30 Unravelling in two-sided matching markets and similarity of preferences
H. Hałaburda 5
09:50 Successful manipulation in stable marriage model with complete preference lists
H. Kobayashi and T. Matsui 17
10:10 On the house allocation markets with duplicate houses
K. Cechlárová and T. Fleiner 23
10:30 *Coffee break* (30 minutes)

Session 2: 11:00 – 12:50 (Chair: Kazuo Iwama)

11:00 **Invited talk**: Assigning papers to reviewers
K. Mehlhorn 3
11:50 Better and simpler approximation algorithms for the stable marriage problem
Z. Király 36
12:10 Optimal popular matchings
T. Kavitha and M. Nasre 46
12:30 The hospitals/residents problem with quota lower bounds
K. Hamada, K. Iwama and S. Miyazaki 55
12:50 *Lunch* (70 minutes)

Session 3: 14:00 – 15:50 (Chair: David Manlove)

14:00 **Invited talk**: Kidney exchange:
Design and evolution of a computer-assisted matching mechanism
A. Roth 4
14:50 Circular stable matching and 3-way kidney transplant
C-C. Huang 67
15:10 Three-sided stable matchings with cyclic preferences
P. Biró and E. McDermid 79
15:30 Approximating matches made in heaven
N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian and A. Rudra 91
15:50 *Coffee break* (30 minutes)

Session 4: 16:20 – 17:20 (Chair: Péter Biró)

16:20 Smith and Rawls share a room: stability and medians
B. Klaus and F. Klijn 102
16:40 The stability of the roommate problem revisited
E. Inarra, C. Larrea and E. Molis 114
17:00 An algorithm for a super stable roommates problem
T. Fleiner, R.W. Irving and D.F. Manlove 126
17:20 *Short break*

Session 5: 17:30 – 18:30 (Chair: Tamás Fleiner)

17:30 Faster algorithms for stable allocation problems
B.C. Dean and S. Munshi 133
17:50 Finding all stable pairs for the (many-to-many) stable matching
P. Eirinakis, D. Magos, I. Mourtos and P. Miliotis 145
18:10 A note on existence and uniqueness of vNM stable sets in marriage games
J. Wako 157
18:30 Closing remarks

My Encounters with David Gale

Marilda Sotomayor, Universidade de São Paolo

Abstract

In this talk I will tell what I know about David Gale and what I learned from him as a friend and a mathematician.

Extracts from a letter from Marilda Sotomayor to Bernhard von Stengel on David Gale's work, dated March 12 2008:

Over the last fifty years David Gale played a leading role in developing some of the themes of fundamental importance to economic theory. An example is matching theory which he introduced to me in 1983. We wrote several papers together. From this time on I used to send to him my manuscripts, before submitting them. He always used to read them and to make comments and suggestions. The Gale's Feast I organized at Stony Brook was a way to thank him for everything I received from him. I also edited a special issue in honor to him for the International Journal of Game Theory, which was published these days. I think David did not see the publication, but I gave to him during the dinner of the Gale's Feast, as a symbolic gift, a compilation of the copies of all 17 papers. You can find a paper of mine there too: The Stability Of The Equilibrium Outcomes In The Admission Games Induced By Stable Matching Rules. This special issue is on David's work but most of them are on matching. Probably we will also have a book published by Springer.

There is some thing that I am sure you can tell in your appreciation on Gale's work. Once, in 1975, when David finished a talk about the stable marriage problem, some physician, who had attended the talk, approached him and told him that the Gale-Shapley algorithm was very similar to the one that was being used by the National Resident Matching Program in Illinois. Then, he wrote a letter to the NRMP asking them about that. They answered David and from the description of the algorithm used by the NRMP he could see that it was mathematically equivalent to the Gale-Shapley algorithm, but in the reverse: instead of producing the optimal stable matching for the students it produced the optimal stable matching for the hospitals. This fact was spread orally. When I arrived in Berkeley, February of 1983, there were papers on the walls of the Department of Mathematics congratulating David for having been elected for the National Academy of Sciences. In these papers it was written that David Gale had discovered an algorithm which was being used to make the allocation of the interns and hospitals in the United States. At this time, there was a colloquium in the Department of Mathematics and David gave a talk about the stable marriage problem. I attended such a talk. Then he talked about the mathematical equivalence of the two algorithms and made clear that the Gale-Shapley algorithm was independently discovered 11 years after the discovery of the NRMP.

These facts were reported in the second paper David wrote about the stable marriage problem, in 1983, co-authored with me: Some remarks on the stable matching problem, Discrete Applied Mathematics. This paper was only published in 1985 and was very important for the developing of the theory of the discrete matchings. In my opinion this was, after the first one by Gale and Shapley, the most important paper that was written on this subject. It presents a concise theory whose results and arguments used in the proofs have been used by the authors until nowadays. The existent theory only considered special cases of the marriage model (the same number of men and women and/or complete preference lists of acceptable partners). Our paper presents the general marriage model where the lists of acceptable partners do not need to be complete and we may have any number of agents in each side of the model. Then the paper generalizes all existent results, presents new proofs with different arguments, and also new results. Almost all results have two proofs: one by making use of the algorithm and another one without the algorithm, by only using the theory that is constructed in the paper. A new and very short proof of the non-manipulability theorem by Dubins and Freedman allows to teach this result in only one class. The original proof had 20 pages. This theorem opened space for investigation on the strategic aspects of the Gale-Shapley algorithm. This was done in the following paper by us, also written in 1983 and published in 1985: Ms. Machiavelli and the stable matching problem, American Math. Monthly.

His first work on continuous matching models was with Gabrielle Demange: The strategy structure of two-sided matching markets, Econometrica, 1985. This is a very precious paper because it allowed to us to use the similarities and differences between several results for the marriage model and for the continuous model to understand better the structure of the matching models.

Another important work on the continuous matching models was written with me and Demange: Multi-item auctions, Journal of Political Economy. It is about two dynamic auction mechanisms to produce the minimum competitive equilibrium price for the case where buyers have quasi-linear utilities and only wish one object and each seller owns one object.

An important thing to be written about David Gale is that all his works reflect his extraordinary creativity, his ability to combine precision and rigour with an elegant style of exposition and to provide simpler alternatives to complicated proofs. An example is the paper of Shapley and Scarf (1974), where he presented a short and simple proof of the non-emptiness of the core of the Housing market, as an alternative to the more complicated proof of the authors. His proof is done by means of the well-known "Top trading cycles algorithm" which has been applied to allocation problems of students to schools and of kidneys to patients. Another example is the suggestion to John Nash to demonstrate the existence of Nash equilibria using the Kakutani Fixed Point Theorem to simplify his proof.

Assigning Papers to Reviewers

Kurt Mehlhorn
Max Planck Institut für Informatik

Abstract

CS conferences typically have a program committee (PC) that selects the papers for the conference from among the submitted papers. The work of the PC is supported by a conference support system, e.g., EasyChair. It is the task of the program chair to assign the papers to the members of the PC. In order to achieve an effective assignment, the PC members classify the papers according to interest. The EasyChair conference system knows four levels: strongly interested, weakly interested, not interested, conflict of interest. Given the classification of the papers by the PC members, the chair seeks a good assignment. What are the right objectives? Here are some: balancing the load of the PC members, making the task of the PC members worthwhile by assigning their high interest papers, guaranteeing each paper a sufficient number of reviews, guaranteeing each paper a sufficient total level of interest, and so on. I will discuss several versions of the problem; I will pose more questions than I give answers.

Kidney Exchange: Design and Evolution of a Computer-assisted Matching Mechanism

Al Roth, Harvard University

Abstract

I will give an overview of the recent development of regional kidney exchanges in the United States, which help patients with incompatible (or poorly matched) donors arrange kidney transplants from other patients' donors. Advances in matching theory interacted with changes in surgical practice as these kidney exchanges evolved. Legislation in 2007 removed barriers to a national kidney exchange, which raises new market design and computational issues, some of which are still open questions.

Unravelling in Two-Sided Matching Markets and Similarity of Preferences

Hanna Hałaburda*

Abstract

This paper investigates the causes and welfare consequences of unravelling in two-sided matching markets. It shows that similarity of preferences is an important factor driving unravelling. In particular, it shows that under the ex-post stable mechanism (the mechanism that the literature focuses on), unravelling is more likely to occur when participants have more similar preferences. It also shows that any Pareto-optimal mechanism must prevent unravelling, and that the ex-post stable mechanism is Pareto-optimal if and only if it prevents unravelling.

1 Introduction

The hiring process calls for collecting information in order to choose the best individual from among the candidates. In certain markets, however, firms hire workers long before all the pertinent information is available. For instance, in the market for hospital interns before 1945, appointments have been made even as early as two years before students' graduation and the actual start of the job (Roth, 1984, 2003). This phenomenon of contracting long before the job begins and before relevant information is available, is called *unravelling*. Those early matches often turn out to be inefficient when the job starts.

Unravelling has been recognized as a serious problem in numerous markets.[1] Measures designed to preclude this phenomenon have not always been successful. Unravelling prevails in certain markets because some employers see a better chance to hire their most-preferred candidates when they contract early than when they wait. Meanwhile, other markets for entry-level professionals appear never to have experienced unravelling, including markets for new professors in finance, economics and biology. Studying what factors lead to unravelling in some markets but not in others can help design better measures to prevent unravelling.

Much of the existing research focuses on stability as the key to understanding unravelling. A matching is ex-post stable if every agent prefers his match to being unmatched, and if there is no blocking pair, that is, a worker and a firm that both strictly prefer each other to their assigned partners. Roth (1991) and Kagel and Roth (2000) argue that ex-post stable matching implemented upon arrival of pertinent information should preclude early contracting. This argument is known as the "stability hypothesis." An ex-post stable matching can be produced in a market through a clearinghouse.[2] However, some clearinghouses with an ex-post stable algorithm have failed to stop unravelling.[3] Roth and Xing (1994) also offer theoretical examples of unravelling even when ex-post stable matching is expected upon the arrival of pertinent information. There is no consensus, however, on whether these examples are single anomalies, or if instead there is some systematic reason for the stability hypothesis to fail.[4]

*Harvard Business School, hhalaburda@hbs.edu.

[1]For an extensive list, see Roth and Xing (1994).

[2]In a clearinghouse, firms and workers submit their preferences, and a matching among all participants is produced by an algorithm.

[3]Examples include the U.S. gastroenterology market, whose clearinghouse was abandoned in 1996 (Niederle and Roth, 2003), and the Canadian market for new lawyers (Roth and Xing, 1994).

[4]The stability hypothesis is not the only explanation of unravelling in the literature. In Damiano, Li and Suen (2005), early contracting is the result of costly search. Li and Rosen (1998), Li and Suen (2000) and Suen (2000) point to workers' risk aversion as the main cause of the phenomenon. Although risk aversion plays an

This paper shows that the similarity of preferences is an important factor contributing to unravelling. The more similar are firms' preferences, the more unravelling will occur in the market, even with an ex-post stable clearinghouse in operation. This paper also shows that unravelling leads to a loss in welfare, and a mechanism must preclude unravelling to be Pareto-optimal. In some markets it means that an ex-post stable mechanism is Pareto-dominated by an ex-post unstable mechanism.

This paper examines a two-sided matching market populated by firms and workers. The agents on each side are heterogenous and they have preferences over agents on the other side of the market. Their aim is to match with the best possible agent on the other side. Workers' preferences over firms are identical: all workers agree on which firm is the best firm, the next-to-best or the worst firm. Firms, however, may have different preferences over the workers. The similarity of firms' preferences over workers is a comparative statics parameter. There are two periods. Firms and workers can contract in either period, but firms only learn their preferences in the second period. The firms and workers who contract in the first period exit the market. The agents who remain in the second period participate in a mechanism that produces a matching between them. In this model, contracting during the first period, before firms have learned their preferences, constitutes unravelling.

The first part of the paper investigates unravelling when the mechanism in the second period produces the ex-post stable matching. In the environment considered here there always exists a unique ex-post stable matching.

It is shown that the nature of equilibria depends crucially on the level of similarity: unravelling occurs only in markets where firms' preferences are sufficiently similar. With very similar preferences, many firms are likely to prefer the same workers. Amid such competition, worse firms may have a better chance to hire their top candidates if they contract before rankings are known.

The second part of the paper studies the problem of mechanism design in markets where unravelling is possible. Before the game starts, a mechanism is chosen for the second period. The mechanism is announced at the outset of the game, so that firms and workers are aware of it during the first period. The goal is to provide a Pareto-optimal outcome from the ex-ante perspective of the beginning of period 1.

It turns out that any Pareto-optimal mechanism must preclude unravelling. The first part of the paper shows that the ex-post stable mechanism may unravel. When this is the case, this mechanism cannot be Pareto-optimal. There exists another — ex-post unstable — mechanism for such a market that does not unravel and Pareto-improves upon the ex-post stable mechanism. In every market there exists a mechanism producing a Pareto-optimal outcome. In some markets, however, all Pareto-optimal mechanisms are ex-post unstable.

Section 2 of this paper presents the model. Section 3 investigates unravelling under an ex-post stable mechanism. Section 4 analyzes the problem of mechanism design in markets where unravelling is possible. Section 5 offers some concluding observations.

2 The Model

To investigate unravelling, I construct a two-stage game between two types of agents: firms and workers. Firms and workers can contract during the first stage. If they do, they leave the market. In the second stage, the remaining agents are matched by a mechanism. The game, described in this section, is represented in Figure 1.

The market is populated by F firms, $f \in \{1, \ldots, F\}$, and W workers, $w \in \{1, \ldots, W\}$. Let $\mathcal{F} \subseteq \{1, \ldots, F\}$ denote an arbitrary subset of firms. Similarly, let $\mathcal{W} \subseteq \{1, \ldots, W\}$ denote an arbitrary subset of workers. There are more workers than firms, $W > F$. Each firm has exactly one position to fill, and each worker can take at most one job.

Workers have identical preferences over firms: all workers consider firm F the most desirable, firm $F-1$ — the second-best, and so on. The

important role and may be an additional cause of early contracting, it is not a necessary condition for the phenomenon. The model in this paper assumes risk-neutrality in order to distinguish incentives to unravel driven by similarity of preferences from those attributable to risk aversion.

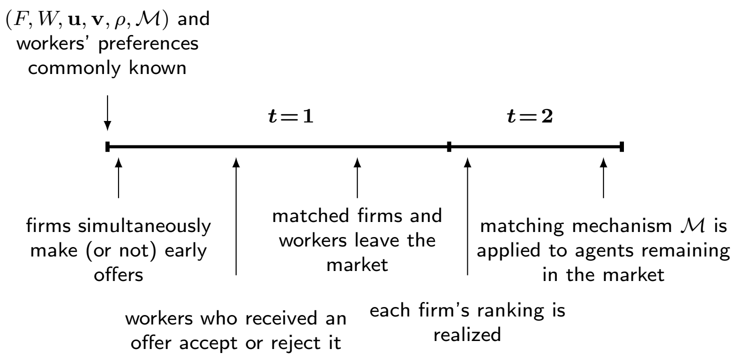


Figure 1: Timeline of the game

utility for a worker from being matched to firm f is u_f, and the utility from being unmatched is 0. Workers prefer being hired by the worst firm to not being hired at all, i.e., $0 < u_1 < u_2 < \ldots < u_F$. Let $\mathbf{u} \equiv [u_1, u_2, \ldots, u_F]$.

Firms may have different preferences over workers. Firm f's preferences are described by its ranking, denoted by $\mathcal{R}^f = (r_1^f, r_2^f, \ldots, r_W^f)$ — an ordered list of length W, where r_1^f represents the lowest-ranked worker, and r_W^f represents the highest-ranked worker in firm f's ranking. Let $\mathbf{R} = [\mathcal{R}^1, \ldots, \mathcal{R}^F]$ be the vector of all firms' rankings. For a subset of firms $\mathcal{F}$, let $\mathbf{R}^{\mathcal{F}}$ be the corresponding vector of the rankings of the firms in $\mathcal{F}$.

The value to firm f of being matched to worker r_k^f is v_k.[5] It is better to hire the worst worker than to keep a vacancy, i.e., $0 < v_1 < v_2 < \ldots < v_W$. Let $\mathbf{v} \equiv [v_1, v_2, \ldots, v_W]$. The matching value vectors, $\mathbf{u}$ and $\mathbf{v}$, are publicly known. There are no transfers between firms and workers.

Definition 1. *A* **matching** *between $\mathcal{F}$ and $\mathcal{W}$ is a function $\mu^{\mathcal{F},\mathcal{W}} : \mathcal{F} \to \mathcal{W} \cup \{\emptyset\}$ that uniquely assigns workers to firms. That is, for any two firms f and f' in $\mathcal{F}$ such that $f \neq f'$*

$$\begin{array}{ll} \textit{either} & \mu^{\mathcal{F},\mathcal{W}}(f) \neq \mu^{\mathcal{F},\mathcal{W}}(f') \\ \textit{or} & \mu^{\mathcal{F},\mathcal{W}}(f) = \mu^{\mathcal{F},\mathcal{W}}(f') = \emptyset \end{array}$$

Expression $\mu^{\mathcal{F},\mathcal{W}}(f) = \emptyset$ means that in matching $\mu^{\mathcal{F},\mathcal{W}}$, firm f is not matched with any worker. When $\mu^{\mathcal{F},\mathcal{W}}(f) = w \in \mathcal{W}$, then firm f is matched with worker w.

Much of the literature emphasizes the importance of ex-post stability in matching. The notion of ex-post stability[6] was introduced by Gale and Shapley (1962). A matching is called *ex-post unstable* if it results in a firm and a worker who would prefer to be matched to each other than to remain in their current matches. A matching is called *ex-post stable* if it is not ex-post unstable.

Which matching is ex-post stable depends on firms' preferences, $\mathbf{R}^{\mathcal{F}}$. A well established result in the literature[7] states that in a market where workers' preferences are identical, for any given firms' preference profile there exists a unique ex-post stable matching between $\mathcal{F}$ and $\mathcal{W}$.[8] This matching can be characterized in the following way: The best firm — the firm most desired by workers — in $\mathcal{F}$ is matched with its highest-ranked worker in $\mathcal{W}$. Then, the next-best firm is matched with its highest-ranked worker from among the remaining workers, and so on. Every firm in $\mathcal{F}$ is matched to its highest-ranked worker remaining in the pool after all the better firms in $\mathcal{F}$ have been matched.

A matching is defined between a subset of firms and a subset of workers. A special case is *matching outcome*, which refers to a matching between all firms, $\{1, \ldots, F\}$, and all workers, $\{1, \ldots, W\}$, realized at the end of the two-stage game. The *ex-post stable outcome* — denoted by o_S — is the ex-post stable matching between all workers, $\{1, \ldots, W\}$, and all firms, $\{1, \ldots, F\}$, in the market: $\mathsf{o}_S \equiv \mu_S^{\{1,\ldots,W\},\{1,\ldots,F\}}$. I drop $\mathbf{R}$ from the notation, keeping in mind that ex-post stable matching depends on rankings.

When a matching is produced by a matching mechanism, $\mathcal{M}$, also called a clearinghouse, it is based on the rankings reported by firms. A matching mechanism is *incentive compatible* if no firm benefits from misreporting its preferences. A mechanism is called *ex-post stable* — and denoted

[5]The assumption that every firm has the same value of being matched with k-th worker on its list is needed for clarity of exposition. The general results remain true for differing matching values.

[6]Gale and Shapley (1962) call this property "stability." Here it is called "ex-post stability" to emphasize the fact that a matching satisfying this property may nevertheless unravel, and thus in a sense may be "ex-ante" unstable though it is "ex-post" stable.

[7]E.g., Gusfield and Irving (1989) or Roth and Sotomayor (1990).

[8]With arbitrary workers' preferences, ex-post stable matching does not need to be unique.

$\mathcal{M}_S$ — if it applies ex-post stable matching to the reported rankings. In this model the ex-post stable mechanism is incentive compatible. Therefore, the ex-post stable mechanism operating over $\mathcal{F}$ and $\mathcal{W}$ will produce ex-post stable matching between $\mathcal{F}$ and $\mathcal{W}$.

There are two periods in the model: $t = 1, 2$. Workers' preferences are commonly known in both periods. Firms learn their own preferences, in the form of rankings, only at the beginning of period 2. Each firm's ranking is its private information.

With W workers there are $W!$ possible rankings. Denote as $\mathfrak{R}$ the set of all possible rankings over workers. The rankings for all F firms, $(\mathcal{R}^1, \ldots, \mathcal{R}^F)$, are drawn from a joint distribution G over $\mathfrak{R}^F$. The model focuses on distributions where the marginal distributions of individual rankings are always uniform, allowing for different levels of similarity between the rankings. Two special cases of such distributions are identical preferences and independent preferences.

Let G_1 be the joint distribution where all firms' rankings are identical and the marginal distribution of any individual ranking is uniform on $\mathfrak{R}$. That is, every ranking in $\mathfrak{R}$ is drawn with equal probability of $\frac{1}{W!}$ and all firms will have the same ranking.

Let G_0 be the joint distribution such that any firm's ranking is drawn from the uniform distribution independently of other firms' rankings.

Between the identical and the independent rankings, there is a continuum of cases of intermediate similarity, G_ρ.

Definition 2. *For* $\rho \in [0, 1]$,

$$G_\rho = \rho G_1 + (1 - \rho) G_0$$

The parameter ρ is a measure of preference similarity[9] and will be a comparative statics parameter in the analysis below. Preferences are said to be *more similar* under $G_{\rho'}$ than under G_ρ when $\rho' > \rho$. Since ρ completely characterizes G_ρ, the two are used interchangeably.

[9] Similarity of preferences, as measured by ρ is similar to the concept of correlation. However, correlation for rankings is not well defined. Since preferences are expressed as rankings, *rankings* and *preferences* are used interchangeably.

Figure 1 illustrates how the game unfolds. Market characteristics $(F, W, \mathbf{u}, \mathbf{v}, \rho, \mathcal{M})$ and workers' preferences are commonly known at any time. At the beginning of period 1, firms simultaneously decide whether or not to make an early offer, and if so, to which worker. Each firm can make at most one offer. After the early offers are released, each worker observes the offers he has received, if any. He does not see offers made to other workers. Every worker presented with an offer accepts or rejects it. He may accept at most one offer. If an offer is accepted, the matched firm and worker leave the market. Firms whose offers were rejected or who did not make an offer in period 1, remain in the market for period 2. In period 2, firms' rankings are realized and a matching mechanism $\mathcal{M}$ operates on the agents remaining in the market. Section 3 assumes the ex-post stable mechanism in period 2. Section 4 considers other mechanisms. There is no discounting between the periods and making offers is costless.

This paper considers only incentive compatible mechanisms, where firms truthfully report their rankings in period 2. In period 1, every firm decides whether or not to make an offer and if so, to which worker. The analysis focuses on sequential equilibria in pure strategies. The strategy of any firm f is $\sigma_f \in \{1, \ldots, W\} \cup \{\emptyset\}$. Let $\Omega_w \subset \{1, \ldots, F\}$ be the set of firms that have made an early offer to worker w. A strategy of worker w, $\sigma_w(\Omega_w) \in \Omega_w \cup \{\emptyset\}$, is the offer that he accepts. Strategy $\sigma_w(\Omega_w) = \emptyset$ means that the worker rejects all offers.

Every firm's payoff depends on many variables: market characteristics $(F, W, \mathbf{u}, \mathbf{v}, \rho, \mathcal{M})$, firms' realized rankings $\mathbf{R}$, and the strategies played by all agents in the market. For clarity, most of this notation is suppressed and only the variables essential to the current analysis are retained.

A definition of sequential equilibrium applied to this model is a profile of strategies and a system of beliefs such that

(1) strategies are sequentially rational given the beliefs, i.e.

 (f) every firm $f \in \{1, \ldots, F\}$ chooses σ_f^* that maximizes its expected payoff, i.e.

$$E\pi_f(\sigma_f^*) \geq E\pi_f(\sigma_f) \qquad \forall \sigma_f \in \{1, \ldots, W\} \cup \{\emptyset\}$$

(w) each worker $w \in \{1, \ldots, W\}$ chooses his strategy, conditionally on the set of received offers, $\sigma_w^*(\Omega_w)$, such as to maximize his expected utility, i.e.

$$EU_w\left(\sigma_w^* \,|\, \Omega_w\right) \geq EU_w\left(\sigma_w \,|\, \Omega_w\right) \qquad \forall\, \sigma_w \in \{\Omega_w\} \cup \{\emptyset\}$$

(2) beliefs are consistent with the strategies played.

Offers made and accepted in period 1 constitute unravelling.

3 Unravelling under Ex-Post Stable Mechanism

Given that the literature focuses on ex-post stable mechanisms, this section investigates unravelling under the ex-post stable matching mechanism.

The ex-post stable mechanism, $\mathcal{M}_S$, is not only incentive compatible, but in all equilibria it also produces ex-post stable matching among the agents remaining in period 2. Unless unravelling occurs in period 1, it produces the ex-post stable outcome, o_S.

The ex-post stable outcome has two properties that are of particular interest here. One is that lower-ranked firms receive lower expected payoffs in the ex-post stable matching, and the other is that firms' expected payoffs decrease as preferences become more similar.

Because of the first property, worse firms are more likely to prefer early contracting under $\mathcal{M}_S$ than better firms. To unravel, firms need to be good enough to be accepted in period 1 and bad enough to want to contract early. And because of the second property, more firms prefer to contract early as preference similarity increases.

Let $E\pi_f(\mathsf{o}_S|\rho)$ denote firm f's expected payoff in the ex-post stable outcome in a given market. Then the following lemma summarizes the properties of o_S.

Lemma 1 (properties of o_S).

(1) *In any market $(F, W, \mathbf{u}, \mathbf{v}, \rho, \mathcal{M}_S)$, for any $f > 1$, $E\pi_{f-1}(\mathsf{o}_S \,|\, \rho) < E\pi_f(\mathsf{o}_S \,|\, \rho)$.*

(2) *Holding other market parameters constant, for any $f < F$,*

$$\rho < \rho' \quad \Longrightarrow \quad E\pi_f(\mathsf{o}_S \,|\, \rho) > E\pi_f(\mathsf{o}_S \,|\, \rho')$$

Proof. See the Appendix, page 11.

3.1 Equilibria without Unravelling

An equilibrium has no unravelling when either no firm makes an early offer, or all early offers are rejected. This subsection explores conditions under which such an equilibrium exists, i.e., conditions under which there is no profitable deviation from o_S.

Consider a worker who receives an offer from firm f in period 1, when in equilibrium all firms are expected to wait for period 2. If the worker accepts the offer, he receives utility u_f. If he rejects the offer, all firms and all workers participate in the period 2 matching mechanism. Because the workers are a priori identical, a worker's expected utility from rejecting f's offer is $\frac{1}{W}\sum_{i=1}^{F} u_i$. He accepts the offer when

$$u_f > \frac{1}{W}\sum_{i=1}^{F} u_i$$

Let $L^0_{(W,\mathbf{u})}$ denote the lowest ranked firm whose offer will be accepted in period 1. All firms worse than L^0 will be rejected in period 1. Firm L^0 and all firms better than L^0 will be accepted. Call $\left\{L^0_{(W,\mathbf{u})}, \ldots, F\right\}$ the *acceptance set.*

The incentives for firms to contract in period 1 depend on the joint distribution of rankings, G_ρ. Recall that $E\pi_f(\mathsf{o}_S|\rho)$ denotes firm f's expected payoff from the ex-post stable outcome under G_ρ.

Since all workers are ex ante the same, an offer made to any worker in period 1 — if it is accepted — yields

$$\pi^0 \equiv \frac{1}{W}\sum_{k=1}^{W} v_k$$

Firm f prefers early contracting to ex-post stable outcome when $\pi^0 > E\pi_f(\mathsf{o}_S|\rho)$.

Firm F never has incentives to make an offer in period 1, since in the ex-post stable outcome it always hires its most-preferred worker. Other firms may have something to gain from an early offer, depending on ρ and $\mathbf{v}$.

Example 1. *Consider firm $F-1$. This firm gets its most-preferred worker unless that worker is firm F's most-preferred worker as well. Its ex-*

pected payoff from the ex-post stable matching is

$$E\pi_{F-1}(\mathsf{o}_S\,|\,\rho) =$$
$$= (1-\rho)\left(1-\frac{1}{W}\right)v_W + \left(\rho + (1-\rho)\frac{1}{W}\right)v_{W-1}$$

In a market with 2 *firms and* 3 *workers where* $\mathbf{v} = [1,2,6]$, $E\pi_1(\mathsf{o}_S\,|\,\rho) = \frac{14}{3}(1-\rho) + 2\rho$ *and* $\pi^0 = 3$. *Thus, firm* 1 *would prefer early contracting to the ex-post stable outcome when* $\rho > \frac{5}{8}$. ■

The lower ranked the firm, the lower its expected payoff in the ex-post stable outcome (Lemma 1(1)). Thus, if firm f prefers early contracting to the ex-post stable outcome, then all firms worse than f do too. The set of all firms that prefer early contracting under G_ρ and $\mathbf{v}$ — called the *offer set* — is an interval $\left\{1, \ldots, H^0_{(\rho,\mathbf{v})}\right\}$, where $H^0_{(\rho,\mathbf{v})}$ is the highest-ranked firm that prefers early contracting.

A profitable deviation from o_S is possible only when there exists a firm that belongs to both the acceptance set and the offer set. This happens when the two sets have nonempty intersection, i.e. $L^0_{(W,\mathbf{u})} \leq H^0_{(\rho,\mathbf{v})}$.

The H^0, and thus the offer set, depend on the similarity of preferences, ρ. The remainder of this section shows that under independent preferences, G_0, the offer set is empty: no firm wants to contract in period 1. Under identical preferences, G_1, by contrast, there may be firms willing to contract early, depending on $\mathbf{v}$. For intermediate cases, $H^0_{(\rho,\mathbf{v})}$ increases with ρ.

For independent preferences, G_0, no firm prefers early contracting to the ex-post stable outcome. Thus the offer set is empty. Therefore, in any market with independent preferences, there is an equilibrium without unravelling.

Lemma 2. *For any* F, $\mathbf{v}$ *and* $W > F$, *if the preferences are independent,* G_0, *then* $H^0_{(G_0,\mathbf{v})} < 1$. *I.e.,*

$$\forall\, F, \mathbf{v}, W \quad s.t.\ W > F \qquad \pi^0 < E\pi_f(\mathsf{o}_S|G_0) \quad \forall\, f$$

Proof. See the Appendix, page 11.

Under identical preferences, $E\pi_f(\mathsf{o}_S|G_1) = v_{W-F+f}$. Thus, condition $\pi^0 > E\pi_f(\mathsf{o}_S|\rho)$ reduces to:

$$\frac{1}{W}\sum_{k=1}^{W} v_k > v_{W-F+f}$$

This inequality may be satisfied for some firms and some values of $\mathbf{v}$.

Example 2 shows a market characterized by identical preferences of firms, where there exists a profitable deviation.

Example 2. *Consider a market with 3 firms and 4 workers and with matching values vectors* $\mathbf{v} = [1,\ 2,\ 3,\ 4]$ *and* $\mathbf{u} = [4,\ 5,\ 6]$, *and with identical firms' preferences,* G_1.
The ex-post stable outcome is

$$\mathsf{o}_S(f_3) = r^3_4 \quad \Longrightarrow \quad \pi_3(\mathsf{o}_S) = 4$$
$$\mathsf{o}_S(f_2) = r^2_3 \quad \Longrightarrow \quad \pi_2(\mathsf{o}_S) = 3$$
$$\mathsf{o}_S(f_1) = r^1_2 \quad \Longrightarrow \quad \pi_1(\mathsf{o}_S) = 2$$

An early offer yields expected payoff of 2.5. *Thus, firm 2 has no incentive to make an early offer, but firm 1 prefers to contract in period 1. That is,* $H^0_{(G_1,\mathbf{v})} = 1$.

A worker's expected utility from period 2 matching is $\frac{1}{W}\sum_{f=1}^{F} u_f = \frac{15}{4} < 4 = u_1$. *This means that firm 1's offer in period 1 will be accepted by any worker. Thus,* $L^0_{(4,\mathbf{u})} = 1$ *and the acceptance set is* $\{1,2,3\}$. *Since the acceptance and the offer sets overlap at* $H^0_{(G_1,\mathbf{v})} = L^0_{(4,\mathbf{u})} = 1$, *there exists a profitable deviation from* o_S *in this market.* ■

However, a profitable deviation from o_S may not exist even when firms' preferences are identical. For example, if the matching utilities in Example 2 were $\mathbf{u}' = [2,3,4]$, then firm 1 would be rejected by any worker in period 1. So there would be no profitable deviation.

Thus, under identical preferences a profitable deviation from o_S may but need not exist. That is, there are markets characterized by G_1, in which there exists an equilibrium without unravelling, but there also are markets with G_1 in which any equilibrium must exhibit unravelling.

Now, consider intermediate similarity of firms' preferences. By Lemma 1(2), the expected value of o_S for firms decreases as similarity of preferences increases. As a consequence, holding other parameters of the market constant, more firms prefer early contracting as similarity of preferences increases. Workers' incentives to accept an offer in period 1 do not depend on similarity of preferences.

For any market parameters $(F, W, \mathbf{v}, \mathbf{u})$, there exists a threshold ρ^{**} such that a profitable deviation from o_S exists for any similarity higher than the threshold but not for similarity lower than the threshold.

Lemma 3. *For any market parameters* $(F, W, \mathbf{v}, \mathbf{u})$, *there exists* $\rho^{**} \in (0, 1]$ *s.t.*

> *for all* $\rho \leq \rho^{**}$, *there exists an equilibrium without unravelling, and*
>
> *for all* $\rho > \rho^{**}$, *there is no equilibrium without unravelling.*

Proof. See the Appendix, page 11.

For G_0 there are no market parameters $(F, W, \mathbf{v}, \mathbf{u})$ for which a profitable deviation from o_S exists. This follows from the fact that under G_0, the offer set is always empty. But as similarity of preferences, ρ, increases, firms' expected payoffs from ex-post stable outcome decrease and unravelling becomes more profitable.

3.2 Equilibria with Unravelling

Firms and workers that contract early exit the market before period 2. The remaining agents participate in the ex-post stable matching mechanism. In equilibrium, worker w who receives offers Ω_w in period 1 either accepts the best offer in Ω_w or rejects all of them, depending on which of the two options maximizes his expected utility.

For any given equilibrium, define *equilibrium unravelling set* as the set of firms that contract early in this equilibrium, and denote this set by $\mathcal{U}$. The remaining firms, $\{1, \ldots, F\} \setminus \mathcal{U}$, participate in $\mathcal{M}_S$ in period 2 with workers still remaining in the market. If the equilibrium unravelling set is empty, such an equilibrium does not involve unravelling.

It is a property of any equilibrium that the unravelling set is an interval, that is, $\mathcal{U}$ has no "holes." For the given equilibrium unravelling set $\mathcal{U}^*$, let firm H^* be the highest-ranked firm in $\mathcal{U}^*$, and firm L^* — the lowest-ranked in $\mathcal{U}^*$. The fact that $\mathcal{U}^*$ is an interval means that all firms worse than H^* but better than L^* belong to $\mathcal{U}^*$ as well. This result is formally stated in Lemma 4(1) below.

Thus, any nonempty $\mathcal{U}^*$ can be characterized by the best firm (H^*) and the worst firm (L^*) that contract early in such equilibrium: $\mathcal{U}^* \equiv \{L^*, \ldots, H^*\}$, for $L^* \leq H^*$.

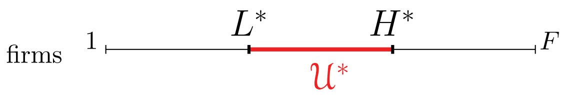


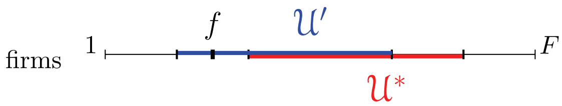
Figure 2: The structure of an equilibrium

In every market there is at least one equilibrium. This result is formally stated in Lemma 4(2) below. Moreover, in a typical market there is more than one equilibrium.

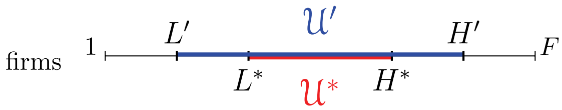
Example 3. *Consider a market with 5 firms and 6 workers where* $\mathbf{u} = [2, 5, 6, 9, 10]$, $\mathbf{v} = [2, 3, 4, 5, 8, 17]$ *and firms' preferences are identical,* G_1. *In this market there are two possible unravelling sets in pure strategy equilibria:* $\mathcal{U}^* = \{3\}$ *and* $\mathcal{U}' = \{2, 3, 4\}$. ■

In Example 3 both equilibrium unravelling sets were nonempty. But this does not need to be the case. There are markets with multiple equilibria where some equilibria exhibit unravelling and others do not.

However, equilibrium unravelling sets cannot be arbitrary. For any two equilibrium unravelling sets in a given market, one needs to be fully included in the other. In particular, two equilibrium unravelling sets for the same market cannot "overlap." Lemma 4(3) states this result formally.



(a) An impossible configuration of multiple equilibrium unravelling sets



(b) A possible configuration of multiple equilibrium unravelling sets

Figure 3: Multiple equilibria with unravelling

The following lemma summarizes properties of equilibria in an arbitrary market with the ex-post stable matching mechanism, $(F, W, \mathbf{u}, \mathbf{v}, \rho, \mathcal{M}_S)$.

Lemma 4. *Given a market* $(F, W, \mathbf{u}, \mathbf{v}, \rho, \mathcal{M}_S)$*:*

(1) **(convexity of unravelling set)** *In any equilibrium, the equilibrium unravelling set,* $\mathfrak{U}$*, is an interval.*

(2) **(existence of equilibrium)** *There exists an equilibrium in pure strategies.*

(3) **(multiple equilibria)** *If there are two equilibrium unravelling sets,* $\mathfrak{U}^*$ *and* $\mathfrak{U}'$ *where* $\mathfrak{U}^* \neq \mathfrak{U}'$*, then either* $\mathfrak{U}^* \subset \mathfrak{U}'$ *or* $\mathfrak{U}' \subset \mathfrak{U}^*$*. Moreover, if both unravelling sets are nonempty,* $\mathfrak{U}^* = \{L^*, \dots, H^*\}$ *and* $\mathfrak{U}' = \{L', \dots, H'\}$ *then*

$$L' < L^* \iff H^* < H'$$

Proof. Available upon request.

The last property of multiple equilibria leads to conclusions about how increasing similarity of preferences drives changes in equilibrium outcomes.

Comparative statics on ρ

Equilibrium unravelling — as measured by the size of $\mathfrak{U}$ — weakly increases with the similarity of preferences.

In any market characterized by independent preferences, all equilibria result in no-unravelling. Lemma 3 implies that as ρ increases, equilibria with $\mathfrak{U} = \emptyset$ exist for a smaller range of market parameters $(F, W, \mathbf{u}, \mathbf{v})$.

By the property of multiple equilibrium unravelling sets (Lemma 4(3)), every equilibrium unravelling set in a given market (if there is more than one) has a different number of firms contracting early. Thus, for any market, all equilibria can be ordered by the size of $\mathfrak{U}$. The *maximum* equilibrium ($\mathfrak{U}^{MAX}$) and the *minimum* equilibrium ($\mathfrak{U}^{MIN}$) can be distinguished. The former is the class of equilibria with maximum unravelling, i.e., the largest $\mathfrak{U}$, and the latter is the class of equilibria with minimum unravelling, i.e., the smallest $\mathfrak{U}$. It may happen in a market that $\mathfrak{U}^{MAX} \equiv \mathfrak{U}^{MIN}$, that is, that all equilibria in this market result in the same unravelling set. For instance, in any market with G_0, $\mathfrak{U}^{MAX} \equiv \mathfrak{U}^{MIN} = \emptyset$.

As similarity of preferences increases, both minimum and maximum equilibrium unravelling sets increase. The maximum equilibrium unravelling set increases from empty to non-empty at some ρ^*. Thus, for $\rho > \rho^*$ an equilibrium with unravelling appears in the market. Moreover, when similarity of preferences increases, the minimum equilibrium unravelling set may also increase from empty to non-empty — at some $\rho^{**} \geq \rho^*$. When this occurs, "no unravelling" is no longer an equilibrium in markets with preference similarity $\rho > \rho^{**}$. This relationship between equilibrium unravelling sets in a market and the level of preference similarity is illustrated by Figure 4.

Let $\mathfrak{U}(\rho)$ be an equilibrium unravelling set in a market with similarity of preferences ρ.

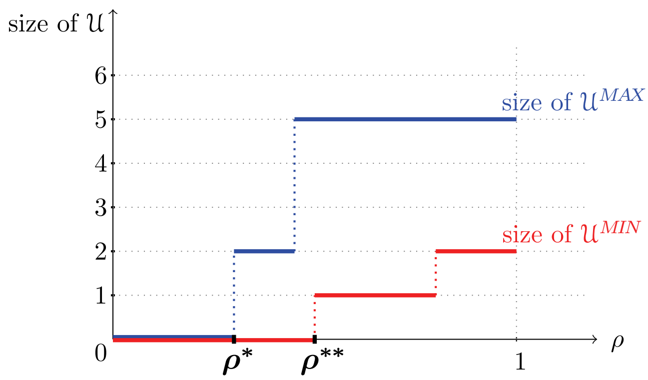


Figure 4: The relationship of $\mathfrak{U}^{MIN}$, $\mathfrak{U}^{MAX}$ and ρ in a typical market

Proposition 1. *Under* $\mathcal{M}_S$*, for any market parameters* $F, W, \mathbf{u}, \mathbf{v}$*, there exist* ρ^* *and* ρ^{**} *such that* $0 < \rho^* \leq \rho^{**} \leq 1$ *and*

$$\begin{aligned} \rho \in [0, \rho^*] &\implies \mathfrak{U}^{MAX}(\rho) = \emptyset \\ \rho \in (\rho^*, \rho^{**}] &\implies \mathfrak{U}^{MIN}(\rho) = \emptyset \quad \& \quad \mathfrak{U}^{MAX}(\rho) \neq \emptyset \\ \rho \in (\rho^{**}, 1] &\implies \mathfrak{U}^{MIN}(\rho) \neq \emptyset \end{aligned}$$

Proof. See the Appendix, page 11.

In some markets characterized by high similarity of preferences, all equilibria under the ex-post stable mechanism involve early contracting. In those markets only an ex-post *unstable* matching mechanism can prevent unravelling. Section 4 focuses on the welfare consequences of unravelling and on characterizing Pareto-optimal mechanisms for markets where unravelling is possible.

4 Mechanism Design

This section turns to the problem of mechanism design in markets where unravelling may occur. It shows that unravelling is Pareto-inefficient, for a broad class of mechanisms called *anonymous* mechanisms. For example, the ex-post stable mechanism is anonymous, as are all mechanisms ever used in real markets.

This section shows that the ex-post stable matching mechanism is Pareto-optimal if and only if it does not induce unravelling. Moreover, in every market there always exists a mechanism that produces a Pareto-optimal outcome. In the markets where the ex-post stable clearinghouse unravels, there exists an ex-post *unstable* mechanism that will stop unravelling and improve the welfare of the market participants.

An outcome, o, is a function from the profile of rankings to randomization over matchings between all firms and all workers.

The previous section considered a special case of an outcome function — the ex-post stable outcome, o_S. This section also examines other outcomes and mechanisms.

Firm f's payoff from an outcome depends on the realized rankings, $\mathbf{R}$, and is denoted by $\pi_f(\mathsf{o}|\mathbf{R})$. The ex-ante expected payoff of an outcome is the expectation over all possible ranking realizations. Let $E\pi_f(\mathsf{o})$ be firm f's expected payoff from outcome o, then

$$E\pi_f(\mathsf{o}) = \sum_{\mathbf{R}\in\mathfrak{R}^F} \pi_f(\mathsf{o}|\mathbf{R}) \cdot Prob(\mathbf{R}|\rho)$$

Similarly, worker w's expected utility of outcome o is

$$EU_w(\mathsf{o}) = \sum_{\mathbf{R}\in\mathfrak{R}^F} U_w(\mathsf{o}|\mathbf{R}) \cdot Prob(\mathbf{R}|\rho)$$

An outcome o' *strictly Pareto-dominates* (ex-ante) outcome o'' when

$$\Big(\forall f \;\; E\pi_f(\mathsf{o}') \geq E\pi_f(\mathsf{o}'') \quad and \quad \forall w \;\; EU_w(\mathsf{o}') \geq EU_w(\mathsf{o}'')\Big)$$

and

$$\Big(\exists f \;\; E\pi_f(\mathsf{o}') > E\pi_f(\mathsf{o}'') \quad or \quad \exists w \;\; EU_w(\mathsf{o}') > EU_w(\mathsf{o}'')\Big)$$

A matching outcome o is *Pareto-optimal* in a given market when there does not exist an outcome in that market that strictly Pareto-dominates o.

The social planner designs a mechanism to achieve the best outcome in the Pareto sense (ex-ante). For an incentive compatible mechanism in period 2, an equilibrium under $\mathcal{M}$ is described by the first-period strategies of agents. Let $\boldsymbol{\sigma}$ denote a vector of period 1 strategies for all agents. A mechanism may possibly implement many equilibria. For example, it was demonstrated that the game with the ex-post stable mechanism usually has multiple equilibria. Let $\boldsymbol{\Sigma}^{\mathcal{M}}$ be the set of all possible equilibria under mechanism $\mathcal{M}$. A *mechanism-equilibrium* pair $(\mathcal{M},\boldsymbol{\sigma})$, where $\boldsymbol{\sigma} \in \boldsymbol{\Sigma}^{\mathcal{M}}$, determines a unique outcome $\mathsf{o}_{(\mathcal{M},\boldsymbol{\sigma})}$.

A mechanism-equilibrium pair $(\mathcal{M},\boldsymbol{\sigma})$ is *unconstrained Pareto-optimal* when it produces a Pareto-optimal outcome. However, a social planner is constrained to inducing outcomes by means of a mechanism. A mechanism-equilibrium pair $(\mathcal{M},\boldsymbol{\sigma})$ is *constrained Pareto-optimal* when there is no other mechanism-equilibrium pair $(\mathcal{M}',\boldsymbol{\sigma}')$ such that its outcome $\mathsf{o}_{(\mathcal{M}',\boldsymbol{\sigma}')}$ strictly Pareto-dominates $\mathsf{o}_{(\mathcal{M},\boldsymbol{\sigma})}$.

Define a mechanism to be *anonymous* if it assigns workers to firms based only on firms' rankings, not on workers' identities. For example, the ex-post stable mechanism, $\mathcal{M}_S$, is anonymous. Define a vector of strategies, $\boldsymbol{\sigma}$, as anonymous if any firm that contracts with a worker in period 1 selects a worker at random, ignoring his identity.[10] A mechanism-equilibrium pair $(\mathcal{M},\boldsymbol{\sigma})$ is anonymous when $\mathcal{M}$ and $\boldsymbol{\sigma}$ are anonymous.

Notice that under an anonymous mechanism-equilibrium pair, every worker has the same ex-ante expected utility at the beginning of the first period.

It is said that a mechanism-equilibrium pair $(\mathcal{M},\boldsymbol{\sigma})$ *exhibits unravelling* when there is a positive probability that an early offer is both made and accepted under the vector of strategies $\boldsymbol{\sigma}$.

The following proposition presents the main result of this section: when an anonymous $(\mathcal{M},\boldsymbol{\sigma})$ exhibits unravelling, it cannot be constrained

[10] It is assumed, however, that no two firms that want to contract in period 1 make offer to the same worker.

Pareto-optimal.

Proposition 2. *For any anonymous mechanism-equilibrium pair $(\mathcal{M}, \boldsymbol{\sigma})$ that exhibits unravelling, there exists an anonymous mechanism-equilibrium pair $(\mathcal{M}', \boldsymbol{\sigma}')$ such that it does not exhibit unravelling and that outcome $\mathsf{o}_{(\mathcal{M}', \boldsymbol{\sigma}')}$ strictly Pareto-dominates outcome $\mathsf{o}_{(\mathcal{M}, \boldsymbol{\sigma})}$.*

Proof. Consider an anonymous $(\mathcal{M}, \boldsymbol{\sigma})$ such that $\mathcal{M}$ produces in equilibrium $\boldsymbol{\sigma}$ a non-empty unravelling set $\mathcal{U}^{\mathcal{M}} \neq \emptyset$. Now consider the following mechanism $\mathcal{M}'$:

(1) To all firms in $\mathcal{U}^{\mathcal{M}}$, $\mathcal{M}'$ tentatively assigns a random worker from the set of all workers. This mimics the unravelling outcome for those firms. Notice that with probability $\frac{1}{W}$, a firm is assigned to its least-preferred worker.

(2) All other firms are matched according to $\mathcal{M}$. These firms get the same expected payoff as under $(\mathcal{M}, \boldsymbol{\sigma})$. For these firms it is the final match.

(3) (the "least-preferred workers correction") For all firms in $\mathcal{U}^{\mathcal{M}}$ that were matched to their least-preferred workers, $\mathcal{M}'$ replaces these workers with workers still remaining in the pool. This is feasible because, after all firms are matched, there is at least one worker still in the pool. For firm f tentatively matched with its least-preferred worker, any of the remaining workers is preferable to the tentative match. This way, all firms tentatively matched with their least-preferred workers can improve their payoff. When there are no more firms in $\mathcal{U}^{\mathcal{M}}$ that are matched to their least-preferred worker, the algorithm stops and the matching is finalized.

Notice that $\mathcal{M}'$ is an incentive compatible mechanism, as no firm can gain by misreporting its preferences.

There is an equilibrium without unravelling under $\mathcal{M}'$. This is the case because all firms in $\mathcal{U}^{\mathcal{M}}$ prefer to wait for $\mathcal{M}'$ rather than to unravel given that other firms wait for period 2. Since firms outside $\mathcal{U}^{\mathcal{M}}$ did not unravel when some other firms were contracting early, they do not unravel when all other firms wait for period 2 under $\mathcal{M}'$. Therefore, no unravelling occurs. Denote the equilibrium without unravelling by $\boldsymbol{\sigma}'$.

Notice that since $(\mathcal{M}, \boldsymbol{\sigma})$ is anonymous, $(\mathcal{M}', \boldsymbol{\sigma}')$ is anonymous as well. And since every firm that is matched to a worker under $(\mathcal{M}, \boldsymbol{\sigma})$ is also matched under $(\mathcal{M}', \boldsymbol{\sigma}')$, the expected payoff to every worker does not change. Every firm in $\mathcal{U}^{\mathcal{M}}$ has a strictly higher expected payoff in $\mathsf{o}_{(\mathcal{M}', \boldsymbol{\sigma}')}$ than in $\mathsf{o}_{(\mathcal{M}, \boldsymbol{\sigma})}$. All the other firms have exactly the same expected payoff in both outcomes. Therefore, $\mathsf{o}_{(\mathcal{M}', \boldsymbol{\sigma}')}$ Pareto-dominates $\mathsf{o}_{(\mathcal{M}, \boldsymbol{\sigma})}$. □

Proposition 2 establishes that no-unravelling is a necessary condition for constrained Pareto-optimality of an anonymous $(\mathcal{M}, \boldsymbol{\sigma})$. In particular, when the ex-post stable mechanism — which is anonymous — unravels, it cannot be constrained Pareto-optimal. For the ex-post stable mechanism, any $(\mathcal{M}_S, \boldsymbol{\sigma})$ that does not exhibit unravelling is unconstrained Pareto-optimal.

Proposition 3 guarantees that in any market there exists an unconstrained Pareto-optimal mechanism-equilibrium pair, i.e., one that produces a Pareto-optimal outcome.

Proposition 3. *For any market, there exists a mechanism $\mathcal{M}$ and an equilibrium $\boldsymbol{\sigma} \in \boldsymbol{\Sigma}^{\mathcal{M}}$ such that $(\mathcal{M}, \boldsymbol{\sigma})$ is unconstrained Pareto-optimal.*

Proof. Consider a mechanism $\mathcal{M}$ that first randomly assigns all participating firms a number between 1 and F. Then the mechanism works in the same way as the ex-post stable mechanism but the order in which firms are matched with workers is based on the randomly assigned numbers, not on their position in the market.

Notice that this mechanism is anonymous. It is also incentive compatible, as is the ex-post stable mechanism. Moreover, there exists an equilibrium without unravelling. If all agents participate in the mechanism, then all firms have higher expected payoffs from the mechanism than from unravelling. Thus, no firm wants to unravel when no other firm unravels. Denote the no-unravelling equilibrium as $\boldsymbol{\sigma}$.

Now, notice that $(\mathcal{M}, \boldsymbol{\sigma})$ produces a Pareto-optimal outcome. The sum of workers' expected utilities and the sum of firms' expected payoffs are the same under $(\mathcal{M}, \boldsymbol{\sigma})$ as they are under o_S. Since o_S is Pareto-optimal, so must $\mathsf{o}_{(\mathcal{M}, \boldsymbol{\sigma})}$ be: in both outcomes it is impossible to increase the expected payoff for one agent without decreasing it for some other agent on the same side of the market. □

5 Conclusions

This study investigates the causes and welfare consequences of unravelling in two-sided matching markets. It considers a two-period model in which firms receive pertinent information about workers and specify preferences over them at the beginning of the second period. It is assumed that firms and workers can make and accept offers during the first period if they wish to, and that a clearinghouse mechanism is used in the second period to assign workers to the remaining firms. Unravelling is said to occur when offers are both made and accepted in the first period. Notice that firms that choose to contract early do so in the absence of information on which workers are most-preferred.

Section 3 explores the issue of unravelling when the ex-post stable mechanism operates in the second period. Ex-post stable matching is the clearinghouse mechanism that most of the existing literature focuses on. Section 3 shows that unravelling becomes more likely as firms' preferences over workers grow more similar. This is the case because when firms' preferences are very similar, lower-ranked firms can be matched with their most-preferred worker only by contracting with them early. Despite insufficient information in the first period, it may be worthwhile for such firms to bear the risk and contract early.

Section 4 investigates the issue of Pareto-optimality of matching mechanisms. The main result demonstrates that a necessary condition for an anonymous mechanism to be Pareto-optimal is that it does not induce unravelling. Any anonymous mechanism that induces unravelling is Pareto-inefficient. In particular, the ex-post stable matching mechanism is Pareto-optimal if and only if it does not unravel.

Another result of Section 4 demonstrates that in every market there exists a mechanism that produces a Pareto-optimal outcome. In markets where the ex-post stable clearinghouse unravels, it is an ex-post unstable mechanism that achieves Pareto-optimality.

These findings are particularly noteworthy given the importance that the literature assigns to stability. In some circumstances, an ex-post unstable mechanism that precludes unravelling is actually preferable from a policy standpoint.

Appendix

Proof of Lemma 1 (page 5)

First, note that the probability that firm f is matched with its kth worker in the ex-post stable outcome under independent preferences is:

$$P(W,f,k) \equiv \frac{(F-f)!}{(F-W-f+k)!}\frac{(k-1)!}{W!}(W-F+f)$$

This formula is derived by applying combinatorics.[11]

(1) *Proof.* The probability that firm $f-1$ gets its worker $k > W-F+f$ is

$$(1-\rho)\cdot P(W,f-1,k) =$$
$$= (1-\rho)\cdot P(W,f,k)\cdot\frac{F-f+1}{F-W-f+1+k}\frac{W-F+f-1}{W-F+f}$$

Since F, f and W are fixed, the ratio in the formula decreases with increasing k. The inequality in expected payoffs of firms f and $f-1$ follows from first order stochastic dominance. □

(2) *Proof.* Follows directly from $E\pi_f(\mathsf{o}_S|\rho) = \rho\cdot E\pi_f(\mathsf{o}_S|G_1)+(1-\rho)E\pi_f(\mathsf{o}_S|G_0)$, and

$$E\pi_f(\mathsf{o}_S|G_0) = \sum_{k=W-F+f}^{W} v_k\cdot P(W,f,k) >$$
$$> v_{W-F+f}\sum_{k=W-F+f}^{W}\cdot P(W,f,k) = v_{W-F+f} = E\pi_f(\mathsf{o}_S|G_1)$$

This completes the proof. □

Proof of Lemma 2 (page 6)

Proof. Consider the worst firm, firm 1.

$$Prob(\mathsf{o}_S(f=1) = r^1_k|G_0,W) \equiv P(W,1,k)$$
$$\text{for} \quad k = (W-F+1),\dots,W$$

and 0 for $k < W-F+1$.

By induction, it can be shown that $P(W,1,k) > P(W,1,k')$ for $k > k'$. Therefore, distribution $P(W,1,k)$ first order stochastically dominates distribution $P_0(W,1,k) = \frac{1}{W}$ for any k, which is the distribution for early matches. Thus, $E\pi_1(\mathsf{o}_S|G_0) > \pi^0$ in any market with G_0.

By Lemma 1(1) for any firm better than firm 1, the payoff from the ex-post stable outcome is higher. Therefore, all firms prefer to wait for o_S rather than to unravel. □

Proof of Lemma 3 (page 7)

Proof. Follows from Lemma 1(2), Lemma 2 and monotonicity of H^0 in ρ. □

Proof of Proposition 1 (page 8)

Proof. First, notice that in any market with G_0, $(F,W,\mathbf{u},\mathbf{v},G_0)$, the only equilibrium outcome is $\mathcal{U}^* = \emptyset$. This follows from Lemma 2.

For $\rho > 0$, the proof follows from the fact that $\mathcal{U}^{MIN} \subseteq \mathcal{U}^{MAX}$ and monotonicity of $H^0(\rho,\mathbf{v})$ and

[11] Derivation of this formula is available on the author's website.

$E\pi_f(\mu_S^{\mathcal{F},\mathcal{W}}|\rho)$ in ρ. The part for ρ^{**} follows from Lemma 3. For ρ^*, notice that for any market parameters $(F, W, \mathbf{u}, \mathbf{v})$ under identical preferences, G_1, it must be that either $\mathcal{U}^{MAX} = \emptyset$, or $\mathcal{U}^{MAX} \neq \emptyset$. In the former case, $\rho^* = 1$ satisfies the Proposition.

In the latter case, let $\mathcal{U}^{MAX} = \{L^{MAX}, H^{MAX}\}$. From an equilibrium condition it must be that

$$E\pi_{H^{MAX}}(\mathcal{M}_S \,|\, \mathcal{U}^{MAX} \setminus \{H^{MAX}\}, G_1) < \pi^0$$

By monotonicity of $E\pi_f$ in ρ, for $\rho < 1$

$$E\pi_{H^{MAX}}(\mathcal{M}_S \,|\, \mathcal{U}^{MAX} \setminus \{H^{MAX}\}, \rho) > \\ > E\pi_{H^{MAX}}(\mathcal{M}_S \,|\, \mathcal{U}^{MAX} \setminus \{H^{MAX}\}, G_1)$$

And we also know that

$$E\pi_{H^{MAX}}(\mathcal{M}_S \,|\, \mathcal{U}^{MAX} \setminus \{H^{MAX}\}, G_0) > \pi^0$$

Thus, there must exist a threshold value ρ' such that

$$E\pi_{H^{MAX}}(\mathcal{M}_S \,|\, \mathcal{U}^{MAX} \setminus \{H^{MAX}\}, \rho) \begin{cases} < \pi^0 & \text{if } \rho > \rho' \\ \geq \pi^0 & \text{if } \rho \leq \rho' \end{cases}$$

i.e. for similarity of preferences lower than ρ', H^{MAX} does not belong to $\mathcal{U}^{MAX}$.

Similarly, there exists a threshold ρ'' such that firm $H^{MAX}-1$ does not belong to $\mathcal{U}^{MAX}$ for $\rho \leq \rho''$. And so on. Thus, there must be a threshold value ρ^* such that there is no firm that belongs to $\mathcal{U}^{MAX}$ under $\rho \leq \rho^*$, but $\mathcal{U}^{MAX}$ is nonempty for similarity of preferences higher than ρ^*.

Values of ρ^* and ρ^{**} may be the same or different. But by definitions of $\mathcal{U}^{MIN}$ and $\mathcal{U}^{MAX}$ it is not possible that $\rho^* > \rho^{**}$.

Thus, Proposition 1 holds. □

References

[1] K. Atkins, Committee OKs new school assignment plan, Boston Herald, July 21, 2005.

[2] E. Damiano, H. Li and W. Suen (2005), Unravelling of Dynamic Sorting, Review of Economic Studies, Vol. 72, 1057-1076.

[3] D. Gale and L.S. Shapley (1962), College Admissions and the Stability of Marriage, American Mathematical Monthly, Vol. 69, No. 1, 9-15.

[4] D. Gusfield and R.W. Irving (1989), The Stable Marriage Problem. Structure and Algorithms, MIT Press.

[5] J.H. Kagel and A.E. Roth (2000), The Dynamics of Reorganization in Matching Markets: A Laboratory Experiment Motivated by a Natural Experiment, Quarterly Journal of Economics, Vol. 115, No. 1, 201-235.

[6] H. Li and S. Rosen (1998), Unravelling in Matching Markets, American Economic Review, Vol. 88, No. 3, 371-387.

[7] H. Li and W. Suen (2000), Risk Sharing, Sorting, and Early Contracting, Journal of Political Economy, Vol. 108, 1058-1091.

[8] M. Niederle and A.E. Roth (2003), Unravelling Reduces Mobility in a Labor Market: Gastroenterology with and without a Centralized Match, Journal of Political Economy, Vol. 111, No. 6, 1342.

[9] A.E. Roth (1984), The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory, The Journal of Political Economy, Vol. 92, No. 6, 991-1016.

[10] A.E. Roth (1991), A Natural Experiment in the Organization of Entry-Level Labor Markets: Regional Markets for New Physicians and Surgeons in the United Kingdom, American Economic Review, Vol. 81, No. 3, 415-440.

[11] A.E. Roth (2003), *Matching and Allocation in Medicine and Health Care*, transcript of a talk at the National Academy of Engineering, June 16, 2003.

[12] A.E. Roth and M. Sotomayor (1990), Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis, Cambridge University Press 1990.

[13] A.E. Roth and X. Xing (1994), Jumping the Gun: Imperfections and Institutions Related to the Timing in Market Transactions, American Economic Review, Vol. 84, No. 4, 992-1044.

[14] W. Suen (2000), A Competitive Theory of Equilibrium and Disquilibrium Unravelling in Two-Sided Matching, RAND Journal of Economics, Vol. 31, No. 1, 101-120.

Successful Manipulation in Stable Marriage Model with Complete Preference Lists

Hirotatsu Kobayashi and Tomomi Matsui

Faculty of Science and Engineering, Chuo University,
Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan.
matsui@ise.chuo-u.ac.jp

Abstract. This paper deals with a strategic issue in the stable marriage model with complete preference lists (i.e., a preference list of an agent is a permutation of all the members of the opposite sex). Given complete preference lists of n men over n women, and a marriage μ, we consider the problem for finding preference lists of n women over n men such that the men-proposing deferred acceptance algorithm (Gale-Shapley algorithm) adopted to the lists produces μ. We show a simple necessary and sufficient condition for the existence of a set of preference lists of women over men. Our condition directly gives an $\mathrm{O}(n^2)$ time algorithm for finding a set of preference lists, if it exists.

1 Introduction

In 1962, Gale and Shapley [1] proposed a simple model, called *stable marriage model*, of two-sided matching market. Given two sets of agents, men and women, both of size n, Gale and Shapley discussed a model in which each agent had preferences over agents of the opposite sex. A marriage is a one-to-one mapping between the two sexes such that a man m is mapped to a woman w, if and only if, w is mapped to m. A marriage is called stable, if no man and woman who are not mapped to each other would both prefer to be. Gale and Shapley proposed an algorithm, called deferred acceptance algorithm (Gale-Shapley algorithm), which *always* finds a stable marriage.

The deferred acceptance algorithm is employed in a number of labor market clearinghouses and college admission systems. A notable variation of the algorithm, called men-proposing deferred acceptance algorithm, works by having men make proposals to women and produces men-optimal marriage, which every man likes at least as well as any other stable marriage. Since most applications of the deferred acceptance algorithm involve the participation of independent agents, it is natural to ask whether agents can benefit by being dishonest about their preference lists. It is well-known that stating true preferences is a dominant strategy for the men in men-proposing deferred acceptance algorithm. In settings that allow incomplete preference lists, women on the other hand have incentives to submit false preferences. By contrary, little is known in the case of stable marriage model with complete preference lists.

This paper deals with a strategic issue in the stable marriage model with complete preference lists (i.e., a preference list of an agent is a permutation of all the members of the opposite sex). Given complete preference lists of n men over n women, and a marriage μ, we consider the problem for finding complete preference lists of n women over n men such that the men-proposing deferred acceptance algorithm adopted to the lists produces μ. We show a simple necessary and sufficient condition for the existence of a set of complete preference lists of women over men. Our condition directly gives an $\mathrm{O}(n^2)$ time algorithm for finding a set of preference lists, if it exists.

In the next section, we establish some terminology and definitions and give some background. Section 3 gives our main results.

2 Notations and Definitions

We denote two sets of agents by M and W, called men and women, both of size n. Each agent in $M \cup W$ has a *preference list* which is a totally ordered list of all the members of the opposite sex. Here we note that this paper considers the case with 'complete' preference lists, i.e., a preference list of an agent is a permutation of members of the opposite sex. A *marriage* is a mapping $\mu : (M \cup W) \rightarrow (M \cup W)$ satisfying that (1) $\forall m \in M, \mu(m) \in W$, (2) $\forall w \in W, \mu(w) \in M$, and (3) $w = \mu(m)$ if and only if $m = \mu(w)$. For any agent $i \in M \cup W$, $\mu(i)$ is called the *mate* of i in marriage μ. A pair $(m, w) \in M \times W$ is called a *blocking pair* for a marriage μ, if m prefers w to $\mu(m)$ and w prefers m to $\mu(w)$. A marriage with no blocking pair is called a *stable marriage*.

Gale and Shapley [1] showed that a stable marriage always exists, and a simple algorithm called the *deferred acceptance algorithm* can find a stable marriage. Here we briefly describe a variant of the their algorithm in which men propose to women (these roles can naturally be reversed). In the following algorithm, we introduced an iteration number which will be used in a later section.

Men-Proposing Deferred Acceptance Algorithm

Step 0: Set the iteration number $r := 1$ and unmarried men $U := M$.
Initially, every woman has no current mate.

Step 1: If $U = \emptyset$, then output the current mate of every woman and stop.

Step 2: Choose a man $m \in U$. Let $w \in W$ be m's most preferred woman who hasn't yet rejected m.

Step 3: (Create a proposal from man m to woman w.)
If woman w has no mate, then update $U := U \setminus \{m\}$ and set m be the current mate of w.
Else if w prefers m to her current mate m', then w rejects m', update $U := U \setminus \{m\} \cup \{m'\}$ and set m be the current mate of w.
Else, w rejects m.

Step 4: Update $r := r + 1$ and go to Step 1.

It is known that the order of proposals (choice of $m \in U$ in Step 2) does not affect the output of the algorithm [1]. Conway showed that the set of stable marriages can be partially ordered as a lattice with the pair of extremal elements, called men-optimal and women-optimal marriages (see [9] for example). In fact, men-proposing deferred acceptance algorithm produces men-optimal marriage.

The issues of strategic manipulation in the stable marriage are discussed in many papers (see books [4, 6] and the references therein, for example). Roth [5] showed that when the men-proposing algorithm is used, none of the men benefits by submitting a false preference list, regardless of how the other agents report their preference. Dubins and Freedman [10] proved that no coalition of men could collectively manipulate in such a way as to strictly improve all of their mates in comparison to men-optimal marriage. In settings that allow incomplete preference lists, women on the other hand have incentives to cheat in men-proposing algorithm. Gale and Sotomayor [2] showed that a woman has an incentive to falsify their preferences as long as she has at least two distinct stable mates. In fact, the women can force the women-optimal marriage μ by rejecting all the men except mates in μ (see [3]).

A feature of this paper is that the agents are required to submit complete preference list. Comparing to the above results, little is known in the case of stable marriage model with complete preference lists. Gusfield and Irving ([4], page 65) point to the absence of any general results in this setting. Tadenuma and Toda [8] considered an implementation question. Teo, Sethuraman, and Tan [7] deals with the situation that there exists a specified woman w who is the only deceitful agent, and that she knows the reported preferences of all the other agents. They proposed a polynomial time algorithm for constructing woman w's optimal cheating strategy. They also discussed the Singapore school-admissions problem, where stable marriage model with complete preference lists is a suitable representation of the problem.

3 Main Results

In this paper, we consider the following problem.

Problem $\mathrm{P}(\mathcal{L}^{\mathrm{m}}, \mu)$:

Input: Set of preference lists $\mathcal{L}^{\mathrm{m}}$ of men M over women W and a marriage μ.
Question: If there exists a set of preference lists $\mathcal{L}^{\mathrm{w}}$ of women W over men M such that the men-proposing deferred acceptance algorithm adopted to the lists in $\mathcal{L}^{\mathrm{m}}$ and $\mathcal{L}^{\mathrm{w}}$ produces the marriage μ, then output $\mathcal{L}^{\mathrm{w}}$. If not, say 'none exists.'

We give a simple necessary and sufficient condition for the existence of women's lists. Let $G(\mathcal{L}^{\mathrm{m}}, \mu)$ be a directed bipartite graph with a pair of vertex sets M, W and a set of directed edges A defined by

$$\begin{aligned} A = &\{(w, \mu(w)) \in W \times M \mid w \in W\} \\ &\cup \left\{ (m, w) \in M \times W \,\middle|\, \begin{array}{l} w = \mu(m) \text{ or} \\ m \text{ prefers } w \text{ to } \mu(m) \end{array} \right\}. \end{aligned}$$

Here we note that for any pair of mates $\{i, \mu(i)\}$ in μ, there are parallel directed edges with opposite directions between corresponding vertices in $G(\mathcal{L}^{\mathrm{m}}, \mu)$.

A directed graph is said to be *strongly connected* if, for all pair of vertices i and j, there exists a directed path from i to j. A *strong component* of a directed graph is a strongly connected subgraph which is maximal. In this paper, we denote a strong component by a set of vertices in the corresponding strongly connected subgraph. If a strong component V' has no incoming edge, i.e., every edge connecting vertices $i \in V'$ and $j \notin V'$ is incident from i to j, we say that V' is a *minimal strong component*.

Theorem 1. *Let $\mathcal{L}^{m}$ be a given set of preference lists of men M over women W, and μ a given marriage. There exists a set of preference lists $\mathcal{L}^{w}$ of women over men, such that the men-proposing deferred acceptance algorithm adopted to the lists in $\mathcal{L}^{m}$ and $\mathcal{L}^{w}$ produces the marriage μ, if and only if, every minimal strong component of $G(\mathcal{L}^{m}, \mu)$ consists of exactly one pair of vertices.*

Before proving the above theorem, we describe some properties of strong components in $G(\mathcal{L}^{\mathrm{m}}, \mu)$. Since every pair of vertices $\{i, \mu(i)\}$ induces a strongly connected subgraph of $G(\mathcal{L}^{\mathrm{m}}, \mu)$, vertices i and $\mu(i)$ are contained in a common strong component of $G(\mathcal{L}^{\mathrm{m}}, \mu)$ for any agent $i \in M \cup W$. Thus, for any strong component V' in $G(\mathcal{L}^{\mathrm{m}}, \mu)$, there exists a subset of men $M' \subseteq M$ satisfying that $V' = \cup_{m \in M'}\{m, \mu(m)\}$. Clearly, the equality $|V' \cap M| = |V' \cap W|$ holds. Every vertex $w \in W$ has a unique outgoing edge $(w, \mu(w))$ and every vertex $m \in M$ has a unique incoming edge $(\mu(m), m)$. These properties yield that a pair of vertices $m \in M$ and $\mu(m) \in W$ forms a minimal strong component, if and only if, vertex $\mu(m)$ has a unique incoming edge $(m, \mu(m))$. Lastly, we note that the strong component decomposition of $G(\mathcal{L}^{\mathrm{m}}, \mu)$ is essentially equivalent to the Dulmage-Mendelsohn decomposition of corresponding underlying undirected graph [11].

Proof. First, consider the case that there exists a minimal strong component V' of $G(\mathcal{L}^{\mathrm{m}}, \mu)$ including more than two vertices. From the assumption, there exists a subset of men $M' \subseteq M$ satisfying that $|M'| \geq 2$ and $V' = \cup_{m \in M'}\{m, \mu(m)\}$.

Let $\mathcal{L}^{\mathrm{w}}$ be an arbitrary set of preference lists of W over M. We apply the men-proposing deferred acceptance algorithm to lists in $\mathcal{L}^{\mathrm{m}}$ and $\mathcal{L}^{\mathrm{w}}$ and assume on the contrary that the marriage μ is obtained. For any $m \in M$, $r(m)$ denotes the iteration number when man $m \in M$ proposed to $\mu(m)$. Let m^* be a man in M' who proposed to his mate in μ lastly, i.e., m^* satisfies $r(m^*) = \max_{m \in M'} r(m)$. We denote $r(m^*)$ by r^* for simplicity.

At the beginning of r^*th iteration, man m^* is unmarried and every man $m \in M' \setminus \{m^*\}$ is the current mate of women $\mu(m)$. Now we show that women $\mu(m^*)$ also has a current mate at the beginning of r^*th iteration. We denote $\mu(m^*)$ by w^*.

Since $\{m^*, w^*\}$ is not a strong component, vertex w^* has an incoming edge (m', w^*) different from the edge (m^*, w^*). The minimality of V' yields that the vertex m' is contained in $M' \setminus \{m^*\}$. From the definition of the graph, m' prefers w^* to its mate $\mu(m')$ and thus w^* has rejected m' in an iteration earlier than r^*. In the deferred acceptance algorithm, if a women rejects a man, she holds a current mate in the rest of iterations. Thus women w^* has a current mate at the beginning of r^*th iteration.

From the above, all the women in V' have current mates, denoted by M'', at the beginning of r^*th iteration. Since m^* is unmarried at the beginning of r^*th iteration, $M'' \subseteq M' \setminus \{m^*\}$. Thus, we have that

$$\begin{aligned} |V' \cap W| = |M''| &\leq |M' \setminus \{m^*\}| < |M'| \\ &= |V' \cap M| = |V' \cap W|. \end{aligned}$$

Contradiction.

Next, we show the inverse implication. Let us consider the case that every minimal strong component of $G(\mathcal{L}^{\mathrm{m}}, \mu)$ consists of exactly one pair of vertices. First, we modify the directed graph as follows. We introduce an artificial vertex s and add directed edge (s, w) for each (woman) vertex $w \in W$ satisfying that w is contained in a minimal strong component. For any vertex $i \in M \cup W$, there exists a directed path from s to i. Thus there exists a directed outgoing spanning tree, denoted by T, with root vertex s in the modified graph. (We fix a directed outgoing spanning tree T in the rest of this proof.) For each vertex $i \in M \cup W$, we denote the parent vertex of i in T by $\mathrm{prt}(i)$. Here we note that for any man (vertex) $m \in M$, his parent vertex $\mathrm{prt}(m)$ is equivalent to his mate $\mu(m)$. The parent vertex of a woman (vertex) is either the artificial vertex s or a man (vertex).

Now we construct preference lists $\mathcal{L}^{\mathrm{w}}$ of women as follows. For any woman w contained in a minimal strong component, we employ a preference list (a total order of men M) such that the most preferred man is $\mu(w)$. If a woman w is not contained in any minimal strong component, we adopt a preference list (a total order of men M) of w such that woman w's first choice is $\mu(w)$ and her second choice is $\mathrm{prt}(w)$ in T.

We apply the men-proposing deferred acceptance algorithm to lists in $\mathcal{L}^{\mathrm{m}}$ and $\mathcal{L}^{\mathrm{w}}$. Since each woman w most prefers man $\mu(w)$ in the list $\mathcal{L}^{\mathrm{w}}$, w never rejects man $\mu(w)$ in the algorithm. Thus, if man m proposed to woman w in the algorithm, then $w = \mu(m)$ or m prefers w to $\mu(m)$, and consequently, the graph $G(\mathcal{L}^{\mathrm{m}}, \mu)$ includes the directed edge (m, w).

Let μ' be a marriage obtained by the men-proposing deferred acceptance algorithm adopted to lists in $\mathcal{L}^{\mathrm{m}}$ and $\mathcal{L}^{\mathrm{w}}$. In the rest of this proof, we show that $\mu = \mu'$ by induction on heights of vertices defined below. For any vertex $i \in M \cup W$, $h(i)$ denotes the *height* of i in T, i.e., $h(i)$ is equal to the length of a unique path from s to i in T. We define that $h(s) = 0$.

(1) Let $i \in M \cup W$ be an agent with $h(i) = 1$. Clearly from the definition of the modified graph, i is a woman contained in a minimal strong component and has exactly two incoming edges $\{(s, i), (\mu(i), i)\}$ in the modified graph. Since man $\mu'(i)$ proposed to i in the algorithm, there is a directed edge $(\mu'(i), i)$ in $G(\mathcal{L}^{\mathrm{m}}, \mu)$. From the above, we have that $\mu(i) = \mu'(i)$.

(2) Assume that for any vertex $j \in M \cup W$, $h(j) = h'$ yields $\mu(j) = \mu'(j)$. Let $i \in M \cup W$ be a vertex whose height is $h' + 1$, i.e., $h(i) = h' + 1$.

(2-1) If $h' + 1$ is an even number, i corresponds to a man, denoted by $m \in M$. Since the vertex m has a unique incoming edge $(\mu(m), m)$, $\mu(m)$ is the parent vertex of m, whose height is h'. The assumption of induction yields that $\mu'(\mu(m)) = \mu(\mu(m)) = m$. From the definition of marriage, $\mu'(\mu(m)) = m$ implies that $\mu(m) = \mu'(m)$.

(2-2) If $h' + 1$ is an odd number, i corresponds to a woman $w \in W$. We denote w's parent vertex $\text{prt}(w)$ by $m' \in M$, for simplicity. From the assumption of induction, $\mu'(m') = \mu(m')$ and thus man m' proposed to woman $\mu(m')$ in the algorithm. Since the graph includes the directed edge (m', w), man m' prefers w to $\mu(m')$. Consequently, man m' proposed to woman w and w rejected m' in the algorithm. In the preference list $\mathcal{L}^{\text{w}}$, man m' is w's second choice. Since w rejected m', w's mate obtained in the algorithm is her first choice $\mu(w)$. Thus, we obtained a desired result that w's mate obtained in the algorithm, denoted by $\mu'(w)$, is equivalent to $\mu(w)$. □

The above theorem directly implies that we can solve the problem $\text{P}(\mathcal{L}^{\text{m}}, \mu)$ by constructing the strong component decomposition of the directed graph $G(\mathcal{L}^{\text{m}}, \mu)$, which requires $\text{O}(n^2)$ time [12].

References

1. D. Gale and L. S. Shapley, "College Admissions and the Stability of Marriage," *The American Mathematical Monthly*, vol. 69, pp. 9–15, 1962.
2. D. Gale and M. Sotomayor, "Some Remarks on the Stable Matching Problem," *Discrete Applied Mathematics*, vol. 11, pp. 223–232, 1985.
3. D. Gale and M. Sotomayor, "Ms Machiavelli and the Stable Matching Problem," *American Mathematical Monthly*, vol. 92, pp. 261–268, 1985.
4. D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, Cambridge, MA, 1989.
5. A. E. Roth, "The Economics of Matching: Stability and Incentives," *Mathematics of Operations Research*, vol. 7, pp. 617–628, 1982.
6. A. E. Roth and M. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, Cambridge, Cambridge University Press, 1990.
7. C.-P. Teo, J. Sethuraman, W.-P. Tan, "Gale-Shapley Stable Marriage Problem Revisited: Strategic Issues and Applications," *Management Science*, vol. 47, pp. 1252–1267, 2001.
8. K. Tadenuma and M. Toda, (1998). "Implementable Stable Solutions to Pure Matching Problems," *Mathematical Social Sciences*, vol. 35, pp. 121–132, 1998.
9. D. E. Knuth, *Mariages Stables*, Montreal, Les Presses de l'Universite de Montreal, 1976. (An English translation by Martin Goldstein, *Stable Marriage and its Relation to Other Combinatorial Problems*, CRM Proceedings and Lecture Notes, 10, American Mathematical Society, 1996.)
10. L. E. Dubins and D. A. Freedman, "Machiavelli and the Gale-Shapley Algorithm," *American Mathematical Monthly*, vol. 88, pp. 485–494, 1981.
11. A. L. Dulmage and N. S. Mendelsohn, "Coverings of Bipartite Graphs," *Canadian Journal of Mathematics*, vol. 10, pp. 517–534, 1958.
12. R. E. Tarjan, "Depth First Search and Linear Graph Algorithms," *SIAM Journal on Computing*, vol. 1, pp. 146–160, 1972.

On housing markets with duplicate houses*

Katarína Cechlárová[1] and Tamás Fleiner[2]
[1]Institute of Mathematics, Faculty of Science, P.J. Šafárik University,
Jesenná 5, 041 54 Košice, Slovakia
email: katarina.cechlarova@upjs.sk
Department of Computer Science and Information Theory,
Budapest University of Technology and Economics,
H-1117, Magyar tudósok körútja 2, Budapest, Hungary
email: fleiner@cs.bme.hu

June 10, 2008

Dedicated to the memory of David Gale.

Abstract. Housing market is a special type of exchange economy where each agent is endowed with one unit of an indivisible good (house) and wants to end up again with one unit, possibly the best one according to his preferences. If the endowments of all agents are pairwise different, an equilibrium as well as a core allocation always exist. However, for markets in which some agents' houses are equivalent, the existence problem for the economic equilibrium is NP-complete. In this paper we show that the hardness result is not valid if the preferences of all agents are strict, but it remains true in markets with trichotomous preferences. Further, we extend some known results about housing markets to the case with duplicate houses using graph-theoretical methods.
Keywords: Housing market, Core, Pareto optimality, Economic equilibrium, Algorithm, NP-completeness
AMS classification: 91A12, 91A06, 68Q25

1 Introduction

The study of markets with indivisible goods started by the seminal paper of Shapley and Scarf [14] where a *housing market* was defined. In a housing market there is a finite set of agents, each one owns one unit of a unique indivisible good (house) and wants to exchange it for another, more preferred one; the preference relation of an agent is a linearly ordered list (possibly with ties) of a subset of goods. In such a market, the set of economic equilibria and the core is always nonempty, which was proved constructively by the Top Trading Cycles (TTC for short) algorithm due to Gale (see [14]).

Roth and Postlewaite [13] made a careful distinction between housing markets where preferences of agents contain ties and those where ties are not allowed. Their findings can be summarized as follows: The core is always nonempty, it always contains the set

*This work was supported by the VEGA grants 1/3001/06, 1/3128/06 (Cechlárová) and by OTKA K69027 grant and the MTA-ELTE Egerváry Research Group (Fleiner).

of equilibrium allocations, but may be strictly larger than the latter. The strong core may be empty, but if the agents' preferences do not contain ties, then strong core is nonempty and is equal to the unique equilibrium allocation. Later Quint and Wako [12] provided a polynomial algorithm for deciding the nonemptyness of the strong core and Wako [15] showed that each strong core allocation is an equilibrium allocation.

All the above results use the assumption that each agent's house is unique. If the houses of several agents are equivalent, the situation may change. Fekete, Skutella and Woeginger [5] proved that it is NP-complete to decide whether a *housing market with duplicate houses* admits an economic equilibrium. Duplicate houses mean that in the preference lists of agents some 'compulsory' ties appear (naturally, equivalent houses must be tied) and there is less freedom in assigning prices to houses (equivalent houses must have equal price). However, in the market constructed in the NP-completeness reduction in [5], some additional ties were used. If nonequivalent houses cannot be tied, would the hardness result still be valid? This question is motivated by other allocation markets where the dividing line between efficiently solvable and hard cases is the presence or absence of ties. Let us mention the *stable roommates problem*, where if ties are not present, a polynomial algorithm to decide the existence of a stable matching exists; in case with ties the existence problem is NP-complete (see [7]). A similar situation occurs also for a modification of the classical housing market, where agents have preferences also over the lengths of trading cycles [4]. In such markets, the notion of economic equilibrium is not applicable, but the TTC algorithm in these settings always finds a (strong) core allocation, if ties are not allowed. However, in the presence of ties it is NP-complete to decide whether the core as well as the strong core are nonempty [3]. Another example is the *stable marriage problem*. There, a stable matching always exists and in the case without ties all stable matchings have the same cardinality. On the other hand, in the case with ties, stable matchings may have different cardinality and the problem of finding a maximum cardinality stable matching is NP-hard [9].

The aim of the present paper is to study some algorithmic problems for housing markets. In Section 3 we derive some properties that are common in all housing markets, in particular we review algorithms for computing core and (strongly) Pareto optimal allocations, show how to test the nonemptyness of the strong core and prove that each strong core allocation is an equilibrium allocation. Our approach, unlike the previously published results, does not assume the uniqueness of agents' endowments and provides a unifying method using the language of graph theory.

In Sections 4 and 5 we deal with economic equilibrium in housing markets. We show that in the case of strict preferences the hardness result for markets with duplicate houses [5] is not valid and we propose a simple polynomial time algorithm for deciding the existence of an economic equilibrium. The other end of the spectrum is the case with trichotomous preferences (i.e. all agents consider all acceptable houses equivalent, strictly preferred to their own house). We show that here the existence problem for the economic equilibrium remains NP-complete.

2 Description of the model

Let A be a *set of n agents*, H a *set of m house types*. The *endowment function* $\omega : A \to H$ assigns to each agent the type of house he originally owns. (Notice that in the classical model of Shapley and Scarf [14], $m = n$ and ω is a bijection.) Further, each agent $a \in A$ wishes to have in his possession just one house and it is supposed that his preferences are given as a linear ordering $P(a)$ on a set $H(a) \subseteq H$, the set of *acceptable* house types. We assume that $\omega(a) \in H(a)$ and this is the least preferred house in $H(a)$ for each $a \in A$. The notation $i \succeq_a j$ means that agent a prefers house type i to house type j. If $i \succeq_a j$ and simultaneously $j \succeq_a i$, we say that house types i and j are *tied* in a's preference list and write $i \sim_a j$; if $i \succeq_a j$ and not $j \succeq_a i$, we write $i \succ_a j$ and say that agent a *strictly* prefers house type i to house type j. The n-tuple of preferences $(P(a), a \in A)$ will be denoted by $\mathcal{P}$ and called the *preference profile.*

The *housing market* is a quadruple $\mathcal{M} = (A, H, \omega, \mathcal{P})$.

If $S \subseteq A$, let $\omega(S) = \{\omega(a); a \in S\} \subseteq H$ be the set of house types owned by agents in S. Conversely, for $T \subseteq H$ we set $A_S(T) = \{a \in S; \omega(a) \in T\}$ to be the set of agents in S who own a house whose type is in T. If $A' \subseteq A$, we say that $\mathcal{M}' = \mathcal{M} \backslash A' = (A \backslash A', \omega(A \backslash A'), \omega', \mathcal{P}')$ is a *submarket* of $\mathcal{M}$ if ω' and $\mathcal{P}'$ are restrictions of ω and $\mathcal{P}$ to $A \backslash A'$ and $\omega(A \backslash A')$, respectively.

We say that $\mathcal{M}$ is a housing market with *strict preferences* if there are no ties in $\mathcal{P}$. On the other hand, $\mathcal{M}$ is a housing market with *trichotomous preferences* if for each agent $a \in A$, all the house types $h \in H'(a) = H(a) \backslash \{\omega(a)\}$ are tied. In other words, in a housing market with trichotomous preferences each agent $a \in A$ partitions the set H into three classes: $H'(a)$ are the house types that are *better* than his own house type, the second class contains just $\omega(a)$ and the third one those house types that are *worse* than $\omega(a)$. For each agent $a \in A$ we denote by $f_S(a)$ the set of the most preferred ($\succeq_a$-maximal) house types from $\omega(S)$ and $F_S(a) = A_S(f_S(a))$ is the set of owners of these houses.

Notice that when preferences of agents are strict then $|f_S(a)| = 1$ for each $a \in A$ and $S \subseteq A$; for trichotomous preferences we have $f_A(a) = H'(a)$.

We say that a function $x : S \to H$ is an *allocation* on S if there exists a bijection π on S such that $x(a) = \omega(\pi(a))$ for each $a \in S$. In the whole paper, we assume that allocations are *individually rational*, i.e. $x(a) \in H(a)$ for each $a \in A$. Notice that for each allocation x on S, the set S can be partitioned into directed cycles (*trading cycles*) of the form $K = (a_0, a_1, \ldots, a_\ell)$ in such a way that $x(a_i) = \omega(a_{i+1})$ for each $i = 0, 1, \ldots, \ell$ (here and elsewhere, indices for agents on cycles are taken modulo ℓ). Therefore we shall often represent an allocation as a collection of trading cycles. We say that agent a is trading in x if $x(a) \neq \omega(a)$.

A coalition $S \subseteq A$ *blocks* an allocation x on A if there exists an allocation y on S such that

$$y(a) \succ_a x(a) \text{ for each agent } a \in S,$$

that is, agents of S can reallocate their houses in such a way that everybody in S is strictly better off than in x. A coalition $S \subseteq A$ *weakly blocks* an allocation x on A if there exists an allocation y on S fulfilling the condition

$$y(a) \succeq_a x(a) \text{ for each } \; a \in S \text{ and } y(a) \succ_a x(a) \text{ for at least one } \; a \in S,$$

that is, in a reallocation nobody from S is worse off and at least one agent in S gains. An allocation x on A is in the *core* of market $\mathcal{M}$ if it admits no blocking coalition and it is in the *strong core* of $\mathcal{M}$ if no coalition weakly blocks it. An allocation x on A is *Pareto optimal* for market $\mathcal{M}$ if A does not block x, and x is *strongly Pareto optimal* if

A does not weakly block x. A pair (x, p), where x is an allocation on A and $p : H \to \mathbb{R}$ is a price function, is an *economic equilibrium* for market $\mathcal{M}$ if for each agent $a \in A$, $x(a) \in f_S(a)$ for the set S of agents whose house is affordable to a, i.e.

$$S = \{a' \in A; p(\omega(a')) \leq p(\omega(a))\}.$$

We shall say that allocation x is an *equilibrium allocation* if there exists a price function p such that the pair (x, p) is an *economic equilibrium.*

The following simple property of equilibrium allocations will often be used.

Lemma 1 *If (x, p) is an economic equilibrium for market $\mathcal{M}$ then $p(x(a)) = p(\omega(a))$ for each $a \in A$.*

Proof. Let $K = (a_0, a_1, \ldots, a_\ell)$ be any trading cycle of x. According to the definition of equilibrium, $p(\omega(a_i)) \geq p(\omega(a_{i+1}))$ for each $i = 0, 1, \ldots, \ell$, which implies the assertion of the Lemma. ■

In what follows, we shall denote by $Q(\mathcal{M}), C(\mathcal{M}), SC(\mathcal{M}), PO(\mathcal{M})$ and $SPO(\mathcal{M})$ the set of all equilibrium, core, strong core, Pareto optimal and strongly Pareto optimal allocations for market $\mathcal{M}$. The definitions imply that for each housing market

$$SC(\mathcal{M}) \subseteq C(\mathcal{M}) \subseteq PO(\mathcal{M}) \text{ and } SC(\mathcal{M}) \subseteq SPO(\mathcal{M}) \subseteq PO(\mathcal{M}),$$

and all the inclusions in the above chain can be strict. Further, in the case without duplicate houses, Wako [15] proved $SC(\mathcal{M}) \subseteq Q(\mathcal{M})$.

Example 1 Let us consider the housing market $\mathcal{M}$ given by Figure 1.

a	a_1	a_2	a_3	a_4	a_5
$\omega(a)$	h_1	h_2	h_3	h_4	h_5
$P(a)$	$h_2 \succ h_3 \succ h_1$	$h_1 \sim h_4 \succ h_2$	$h_4 \succ h_2 \succ h_3$	$h_3 \sim h_5 \succ h_4$	$h_2 \succ h_1 \succ h_5$

Figure 1: Endowments and preferences of agents

In this example, either $x(a_1) = h_2$ or $x(a_1) \neq h_2$ for allocation x. In the former case the cycle (a_2, a_4, a_5) is weakly blocking as agent a_5 strictly improves and agents a_2, a_4 are not worse off than in x (they have their most preferred house types on this cycle). In the latter case the set $\{a_1, a_2\}$ is weakly blocking, as a_1 improves and a_2 is not worse off. Hence $SC(\mathcal{M}) = \emptyset$.

Further, $C(\mathcal{M})$ contains six allocations

$$x_1 = (a_1, a_2)(a_3, a_4)(a_5), \quad x_2 = (a_2, a_4, a_5)(a_1)(a_3), \quad x_3 = (a_1, a_3, a_4, a_5, a_2)$$
$$x_4 = (a_1, a_3, a_2, a_4, a_5), \quad x_5 = (a_1)(a_2, a_4, a_3)(a_5), \quad x_6 = (a_1, a_2, a_4, a_5)(a_3).$$

Core allocation x_3 is strongly Pareto optimal, as the only agent who could strictly improve is a_1 by getting h_2, but since a_5 is not allowed to become worse off than in x_3, he must also be assigned his most preffered house h_2, and so there is no weakly blocking allocation on A for x_3.

Allocation $x_7 = (a_1, a_3, a_2)(a_4)(a_5)$ is Pareto optimal (notice e.g. that agent a_2 cannot strictly improve) but not in the core, as the set $\{a_3, a_4\}$ is blocking.

Finally, allocations x_1 and x_2 form economic equilibria with price vectors p_1 and p_2, where $p_1(a_1) = p_1(a_2) = p_1(a_3) = p_1(a_4) = 1$, $p_1(a_5) = 0$ and $p_2(a_1) = p_2(a_3) = 0$ and $p_2(a_2) = p_2(a_4) = p_2(a_5) = 1$.

In the following sections we shall often use digraphs, so let us recall the terminology used; for further details we recommend e.g. [11]. A *digraph* is a pair $G = (V, E)$, where V is a set of *vertices*, E is the set of *arcs*, i.e. ordered pairs of vertices. We allow parallel arcs as well as loops, i.e. arcs of the form (i, i) for some $i \in V$. If $V' \subseteq V$, a subdigraph of G *induced* by V' is a digraph $G(V') = (V', E')$ where $E' = \{(i, j) \in E; i, j \in V'\}$.

A *walk* in G is a sequence of vertices $(i_0, i_1, \ldots, i_k)$ such that $(i_j, i_{j+1}) \in E$ for each $j = 0, 1, \ldots, k-1$. We say that vertex j is a *successor* of a vertex i in G, if G contains a walk from i to j. A vertex j is a *direct successor* of a vertex i in G, if $(i, j) \in E$. A vertex $i \in V$ is a *sink* in G, if it has no successors in G. (Let us remark, that some authors, e.g. [14, 12] use the term *top vertex* in the same meaning.)

A walk $(i_0, i_1, \ldots, i_k)$ is said to be *closed*, if $i_0 = i_k$; if moreover all the vertices $i_1, \ldots, i_k$ are pairwise distinct, it is a *cycle*. A collection $\mathcal{K}$ of vertex-disjoint cycles is a *cycle cover* of a digraph G if each vertex of G is contained in some $K \in \mathcal{K}$. A polynomial-time algorithm to decide whether a digraph has a cycle cover is well-known. The folklore approach described e.g. in [2] first constructs a bipartite (undirected) graph Γ by duplicating each vertex of G and making the pair $\{i, j'\}$ an edge if and only if $(i, j) \in E$, where j' is the copy of vertex $j \in V$. It is straightforward to see that G has a cycle cover if and only if Γ has a perfect matching.

A digraph G with vertex set V is *strongly connected*, if for each pair i, j of distinct vertices of V there is a walk from i to j as well as a walk from j to i in G. A *strongly connected component* (SCC for short) of a digraph G is a maximal strongly connected subdigraph of G. We shall call a SCC *trivial*, if it contains just one vertex and no arcs. (So if a vertex with a loop is a SCC, then it is nontrivial.) The *condensation* $G^* = (V^*, E^*)$ of a directed graph G is obtained by merging the vertices of each SCC of G (and perhaps deleting eventual parallel arcs). For $i \in V$ we shall denote by i^* the vertex of G^* corresponding to the SCC of G containing i. As G^* is acyclic, the vertices of G^* are partially ordered, $i^* \gg j^*$ if and only if i^* is a successor of j^* in G^* and this order can easily by extended to a linear ordering, sometimes called a *topological order*. A SCC is *sink*, if its corresponding vertex in the condensation is a sink. An algorithm to construct a condensation of a digraph and its topological order, linear in the number of its arcs, is described e.g. in [11].

A condensation of a digraph can be used also for the following task. Sometimes we shall have to decide whether in a digraph $G = (V, E)$ a cycle cover exists, that contains at least one arc from the prescribed set $F \subseteq E$; such a cycle cover will be said to be *hitting* F. Using the approach described above, we compute a perfect matching in the bipartite graph Γ. If some of the arcs from F corresponds to a matched edge, the answer for the hitting cycle cover problem is yes. Otherwise we can continue in the following way: direct all the matched edges from the first color class to the second one and all the unmatched edges from the second color class to the first one. For the obtained digraph construct its condensation Γ^*. Now its is obvious that each matched edge that connects two different SCCs of Γ^* is matched in each perfect matching (i.e. is used in each cycle cover of G), each unmatched edge that connects two different SCCs is matched in no perfect matching (i.e. is used in no cycle cover of G), and all the edges within some SCC are in some, but not in all perfect matchings. Hence the answer for the hitting cycle cover problem is no if and only if all arcs from F are unmatched and connect different components of Γ^*.

A strongly connected digraph G is said to be Eulerian, if it contains a closed walk containing all arcs of G exactly once. This is equivalent with the property that for each vertex i in G the number of arcs that enter i is equal to the number of arcs that leave i – this can also be decided linearly in the number of arcs of G [11].

3 Common properties of housing markets

If $\mathcal{M}$ is any housing market, for finding a core (and hence also a Pareto optimal) allocation the famous TTC algorithm [14] can be used. TTC starts with the whole set of agents A. An arbitrary agent $a_0 \in A$ is chosen and he proposes to one (arbitrary) agent in $F_A(a_0)$, say agent a_1. Then a_1 proposes to one of the agents in $F_A(a_1)$, say agent a_2, etc. After a finite number of proposals, a cycle K arises, which will be the first trading cycle. Agents corresponding to K are deleted from the market and the whole procedure is repeated for the submarket $\mathcal{M} \backslash K$ until all agents are assigned to some trading cycle.

Theorem 1 *[14] The core of each housing market is nonempty.*

An $O(L)$ implementation of the TTC algorithm, where $L = \sum_{a \in A} |H(a)|$ is the total length of preference lists, is described in [1] for the no-ties, no-duplicate-houses case. Obviously, this implementation can be used in any market, if at the beginning the ties are broken arbitrarily, respecting the house types.

When TTC is applied to a housing market without duplicate houses and ties, its outcome is unique. This need not be the case when either duplicate houses or ties are present. Moreover, in the classical model each equilibrium allocation is an output of some realization of the TTC algorithm irrespective of ties, but it may happen that some core allocations cannot be obtained in this way. In Example 1, this applies e.g. to allocation x_5: as soon as agent a_3 receives the proposal from a_4, he proposes back to a_4. Let us remark here, that a complete description of the structure of the core of a housing market is not known.

The strong core of a housing market may be empty [13] and an $O(n^3)$ algorithm for testing its nonemptyness was proposed in [12]. Here we give a simpler algorithm for this task. Let $G_F(\mathcal{M}) = (A, E)$ be a digraph defined by $(i, j) \in E$ if $j \in F_A(i)$. That is, in $G_F(\mathcal{M})$ each vertex (agent) i sends an arc to each agent who owns a house that i likes the best.

Lemma 2 *Let $\mathcal{M} = (A, H, \omega, \mathcal{P})$ be any housing market and let D be a sink SCC in $G_F(\mathcal{M})$. Then $SC(\mathcal{M}) \neq \emptyset$ if and only if D admits a cycle cover and $SC(\mathcal{M}') \neq \emptyset$ for the submarket $\mathcal{M}' = \mathcal{M} \backslash V(D)$.*

Proof. Let $x \in SC(\mathcal{M})$. Suppose, for a contradiction, that D is a sink SCC in G_F that does not have a cycle cover. Then there exists $a \in V(D)$ such that $x(a) \notin f_A(a)$, but as $a \in V(D)$, there exists a cycle K in D containing a. Assigning to each agent on K the endowment of his direct successor on K, it is easy to see that the vertices of K form a weakly blocking set for x. So D must have a cycle cover and $x(a) \in \omega(V(D))$ for each agent $a \in V(D)$. Hence the restriction of x to agents of $A \backslash V(D)$ is a strong core allocation for $\mathcal{M}'$.

Conversely, let $x' \in SC(\mathcal{M}')$ and let D have a cycle cover $\mathcal{K}$. It is easy to see that x' augmented by the trading cycles defined by the cycles of $\mathcal{K}$ gives a strong core allocation for $\mathcal{M}$. ■

Based on Lemma 2, it is sufficient to find any sink SCC D of the graph $G_F(\mathcal{M})$ and to test whether it admits a cycle cover. If the answer is negative, $SC(\mathcal{M}) = \emptyset$, otherwise the same procedure is invoked with the submarket $\mathcal{M} \backslash V(D)$. This is summarized in the following assertion.

Theorem 2 *In any housing market $\mathcal{M}$, the nonemptyness of the strong core can be decided in $O(L\sqrt{n})$ time.*

Proof. Computing the condensation requires $O(L)$ operations. In updating the digraph $G_F(\mathcal{M})$ to get $G_F(\mathcal{M}')$ and its condensation after a sink SCC D was deleted, it suffices to scan just the arcs that enter $V(D)$. Hence no arc will be used more than once in these computations, so in updating digraphs and their condensations no more than $O(L)$ operations will be used. Deciding the existence of a cycle cover in a component with $n_i \leq n$ vertices and $L_i \leq L$ arcs, using the approach described in [2] needs $O(L_i\sqrt{n_i})$ operations, if the bipartite matching algorithm of [8] and [10] is used. Overall, this gives $O(L\sqrt{n})$ time complexity. ■

Next we extend Theorem 2 from [15] to housing markets with duplicate houses.

Theorem 3 *In any housing market $\mathcal{M}$, $SC(\mathcal{M}) \subseteq Q(\mathcal{M})$.*

Proof. Let $x \in SC(\mathcal{M})$ be arbitrary. We will show that for a suitably defined price function p, the pair (x, p) is an economic equilibrium of $\mathcal{M}$.

Let us take the digraph $G(H, x) = (H, E_x^{\sim} \cup E_x^{\succ})$ where

$$(h_k, h_\ell) \in E_x^{\sim} \quad \text{if there exist } i, j \text{ such that } \omega(i) = h_k, \omega(j) = h_\ell \text{ and } h_\ell \sim_i x(i)$$
$$(h_k, h_\ell) \in E_x^{\succ} \quad \text{if there exist } i, j \text{ such that } \omega(i) = h_k, \omega(j) = h_\ell \text{ and } h_\ell \succ_i x(i).$$

Arcs from $E_x^{\sim}$ will be called *weak* arcs, those from $E_x^{\succ}$ are *strong* arcs. As x is a strong core allocation, any cycle K in $G(H, x)$ consists exclusively of weak arcs, otherwise the agents corresponding to the arcs of K would form a weakly blocking set for x. Let us now take the condensation $G^* = (V^*, E^*)$ of $G(H, x)$. Strong arcs (now defined accordingly) connect different vertices of G^*, so taking any topological order $\gg$ of V^*, the prices of house types can be defined by

$$p(h_i) > p(h_j) \text{ if and only if } \; h_i^* \gg h_j^*.$$

It is easy to see that for the pair (x, p) we have:

(i) $p(\omega(i)) = p(x(i))$ for each agent, as any trading cycle of x is within a SCC of $G(H, x)$, so each agent i can afford house $x(i)$;

(ii) if agent i prefers a house type $\omega(j)$ to $x(i)$, then $p(\omega(j)) > p(\omega(i))$, hence no agent i can afford a house that he prefers to $x(i)$

and so (x, p) is an economic equilibrium for $\mathcal{M}$. ∎

Pareto optimality for housing markets is studied in [1]. A polynomial algorithm for finding a strongly Pareto optimal matching and some results concerning the structure of strongly Pareto optimal matchings are described, however, all is done under the assumption of strict preferences. If ties and/or duplicate houses are present, one can use a different approach. Let x be any allocation in $\mathcal{M}$. Let us take the digraph $G^{\succ}(A, x) = (A, E_x^{\succ})$ where $(i, j) \in E_x^{\succ}$ if $\omega(j) \succ_i x(i)$ (this digraph is similar to the one used in the proof of Theorem 3, but now its vertices are agents, not house types). x is not Pareto optimal if and only if $G_x^{\succ}$ admits a cycle cover. On the other hand, x is not strongly Pareto optimal if and only of in the digraph $G^{\succeq}(A, x) = (A, E_x^{\sim} \cup E_x^{\succ})$ with $(i, j) \in E_x^{\sim}$ if $\omega(j) \sim_i x(i)$ a cycle cover exists hitting $E_x^{\succ}$. (We described an approach for checking this in Section 2.) If no hitting cycle cover exists, x is strongly Pareto optimal. Otherwise x can be upgraded to x' along a hitting cycle cover. It is easy to see that $G^{\succeq}(A, x')$ is a proper subgraph of $G^{\succeq}(A, x)$, so if we take a hitting cycle cover that gives some agent his best possible improvement, this agent will not improve any more. So we have at most n upgrades. This gives an $O(n^{3/2}L)$ algorithm (at most n times computing a maximum cardinality matching and condensation in a bipartite graph with $2n$ vertices and no more than L edges) and we believe that further improvements can be achieved by a careful implementation and the choice of a suitable starting allocation x. Summarizing:

Theorem 4 *In any housing market $\mathcal{M}$, Pareto optimal as well as strongly Pareto optimal allocations exist and both can be found in polynomial time.*

4 Housing markets with strict preferences

Let $\mathcal{M}$ be a housing market with strict preferences. We define a digraph $G_H = (H, E)$, called the *house-type digraph*, with arcs corresponding to agents in such a way that $e(a) = (h_i, h_j) \in E$ if $\omega(a) = h_i$ and $h_j \in f_A(a)$. Notice that since preferences are strict, the correspondence between agents and arcs is one-to-one.

Lemma 3 *Let $\mathcal{M}$ be a market with strict preferences where an economic equilibrium with price function p exists and let D be a sink SCC in G_H. Then*

(i) $p(h_i) = p(h_j)$ for each $h_i, h_j \in V(D)$;

(ii) D is Eulerian and

(iii) if $(h_i, h_j) \in E$ for some $h_i \notin V(D)$, $h_j \in V(D)$, then $p(h_i) < p(h_j)$.

Proof. Let the economic equilibrium in $\mathcal{M}$ be (x, p).

(i) Suppose that $V(D)$ is partitioned into two nonempty disjoint sets V^1 and V^2 in such a way that $p(h_i) > p(h_j)$ for each $h_i \in V^1$ and each $h_j \in V^2$. As D is strongly connected, there exists an agent a such that $e(a) = (h_i, h_j)$ for some $h_i \in V^1$, $h_j \in V^2$. Then, since h_j is the only most preferred house type for a in A and agent a can afford it, we have $x(a) = h_j$ – a contradiction with Lemma 1. Therefore the prices of all the house types in $V(D)$ are equal.

(*ii*) It follows from (*i*) and the definition of D that all the agents correspondig to arcs of D trade only among themselves, i.e. the arc set of D can be partitioned into several arc-disjoint directed cycles (possibly loops) and so D is Eulerian.

(*iii*) If $e(a) = (h_i, h_j) \in E$ for $h_i \notin V(D)$, $h_j \in V(D)$ and $p(h_i) \geq p(h_j)$, then agent a can afford house type h_j and since this is his only most preferred house type in $\mathcal{M}$, he must be in x on a trading cycle containing house types from $V(D)$, but this is a contradiction with the proof of (*ii*). ■

Notice that if a sink SCC of the house-type digraph consists of a single vertex, then it contains a loop, so it is trivially Eulerian.

Lemma 4 *Let $\mathcal{M} = (A, H, \omega, \mathcal{P})$ be a market with strict preferences and let D be a sink SCC in G_H. Then an economic equilibrium exists in $\mathcal{M}$ if and only if D is Eulerian and in the reduced market $\mathcal{M}' = \mathcal{M} \backslash E(D)$ an economic equilibrium exists.*

Proof. Let (x, p) be an equilibrium for $\mathcal{M}$. Lemma 3 implies that $x(a) \in V(D)$ for each agent $a \in E(D)$, that D is Eulerian and no agent a with $\omega(a) \notin V(D)$ can afford a house type in $V(D)$. Then the restriction of (x, p) to $\mathcal{M}'$ is an economic equilibrium for $\mathcal{M}'$.

Conversely, let $(x', p') \in Q(\mathcal{M}')$. Then we construct an economic equilibrium (x, p) for $\mathcal{M}$ in the following way. For each $a \in A \backslash E(D)$ set $x(a) = x'(a)$, for each $h \in H \backslash V(D)$ let $p(h) = p'(h)$. Now take an arbitrary constant $\xi > max\{p(h); h \in H \backslash V(D)\}$ and set $p(h) = \xi$ for all $h \in V(D)$. Clearly, houses in $V(D)$ are now outside the budget set for all agents in $A \backslash E(D)$, so these agents still have their most preferred affordable houses in (x, p). Each agent $a \in E(D)$ corresponds to an arc $e(a) = (h_i, h_j)$, so we set $x(a) = h_j$ for this agent. As D is Eulerian, the supply equals the demand for the houses in D, thus (x, p) is an economic equilibrium for $\mathcal{M}$. ■

Now Lemmas 3 and 4 directly imply the following simple algorithm. For a given market $\mathcal{M}$, create the house-type digraph G_H and take any sink SCC D in G_H. If D is not Eulerian, $\mathcal{M}$ does not admit any economic equilibrium. If D is Eulerian, the agents and house types corresponding to D are deleted from $\mathcal{M}$ and the whole procedure continues for the obtained submarket. As the number of arcs in the house-type digraph is n and in each reduction at least one house-type is deleted, we get the bound $O(mn)$, which could again be improved by a careful implementation. Summarizing:

Theorem 5 *If the preferences of all agents in a housing market $\mathcal{M}$ are strict, then the existence of an economic equilibrium for $\mathcal{M}$ can be decided in $O(mn)$ time.*

Notice that if a market admits an economic equilibrium, then the equilibrium allocation is unique, although it may be supported by several different price functions and not even the linear order of the prices is uniquelly determined.

Now we turn our attention to the strong core. We derived in Section 3 a condition that enables to decide the nonemptyness of the strong core for any housing market in polynomial time, but here we give an alternative characterization for markets with strict preferences using the house-type digraph.

Lemma 5 *Let $\mathcal{M}$ be a market with strict preferences and D a sink SCC in the condensation of G_H.*

(i) If $SC(\mathcal{M}) \neq \emptyset$, then D is Eulerian.

(ii) $SC(\mathcal{M}) \neq \emptyset$ if and only if $SC(\mathcal{M} \backslash E(D)) \neq \emptyset$ and D is Eulerian.

Proof. (i) Let us suppose that for a strong core allocation x an agent $a \in E(D)$ exists such that $x(a) \notin V(D)$. This means that a did not receive in x his most preferred house. Further, since D is nontrivial, there exists a cycle K in D, containing arc $e(a)$. Let us assign to each agent b associated with an arc $e(b) \in K$ the house type corresponding to the endvertex of $e(b)$. It is easy to see that $E(K)$ is a weakly blocking coalition – a contradiction with x being a strong core allocation. Now we use the same argument as in the proof of Lemma 3 (ii).

(ii) If $x \in SC(\mathcal{M})$ then clearly its restriction belongs to $SC(\mathcal{M} \backslash E(D))$ as any weakly blocking set in $\mathcal{M} \backslash E(D)$ is weakly blocking in $\mathcal{M}$ too. Conversely, if $x' \in SC(\mathcal{M} \backslash E(D))$, we extend allocation x' to an allocation x of $\mathcal{M}$ in such a way that an agent a corresponding to the arc $e(a) = (h_i, h_j) \in E(D)$ will get $x(a) = h_j$, for other agents $x(a) = x'(a)$. Suppose now that x is weakly blocked by a set Z with allocation y. Then neither $Z \subseteq E \backslash E(D)$ nor $Z \subseteq E(D)$ is possible (the former because x' is a strong core allocation for the submarket, the latter because no agent from $E(D)$ can strictly improve), but then necessarily Z contains an agent a, for whom $\omega(a) \in V(D)$ and $y(a) \notin V(D)$, hence a is worse off in allocation y, a contradiction. ■

Now it is easy to see that the following assertion holds also for markets with duplicate houses:

Corollary 1 *A housing market with strict preferences admits a strong core allocation if and only if it admits an economic equilibrium.*

5 Trichotomous markets

First we derive a simple condition for the nonemptyness of the strong core for any housing markets with trichotomous preferences. Let $G_T(\mathcal{M}) = (A, E_T)$ where $(i, j) \in E_T$ if $\omega(j) \succ_i \omega(i)$, i.e. if $\omega(j) \in H'(i)$. An allocation x in $\mathcal{M}$ corresponds to a collection of vertex-disjoint trading cycles and some single vertices in $G_T(\mathcal{M})$. It is easy to see that allocation x admits no weakly blocking set, if each agent contained in a cycle in $G_T(\mathcal{M})$ is trading in x. This implies

Theorem 6 *In a housing market $\mathcal{M}$ with trichotomous preferences, $SC(\mathcal{M}) \neq \emptyset$ if and only if each nontrivial SCC of $G_T(\mathcal{M})$ admits a cycle cover.*

In a sharp contrast with the above result is the following assertion.

Theorem 7 *In a housing market with trichotomous preferences it is NP-complete to decide whether an economic equilibrium exists.*

Proof. In the polynomial reduction we shall use the problem ONE-IN-THREE-SAT, see [6, Problem LOG4]. Here a Boolean formula B in conjunctive normal form with variables $v_1, v_2, \ldots, v_n$ and clauses $C_1, C_2, \ldots, C_m$ is given. Each clause contains exactly three literals, no variable is negated in B and the question is whether there exists a truth assignment f such that there is exactly one true literal in each clause.

Let us construct a market $\mathcal{M}(B)$ for each formula B. The set of agents contains one agent c_i for each clause C_i and for each variable v_j there are agents $\phi_j^1, \phi_j^2, \ldots \phi_j^{n_j}$

where n_j is the number of occurences of variable v_j in B. The set of houses is $\{C_1, \dots, C_m, v_1, \dots, v_n\}$ (although the same symbols are used for the elements of the formula as well as for the houses in the market, no confusion should arise) and the endowments are defined by

$$\begin{aligned} \omega(c_i) &= C_i \text{ for each } i, \\ \omega(\phi_j^k) &= v_j \text{ for each } j, k. \end{aligned}$$

Further, $H'(c_i) = \{v_{j1}, v_{j2}, v_{j3}\}$ where v_{j1}, v_{j2}, v_{j3} are the literals present in C_i and $H'(\phi_j^k)$ is equal to the set of all houses C_i such that variable v_j occurs in clause C_i.

Now suppose that the truth assignment f with the required properties exists. We shall define the pair (x, p) in the following way:

$$\begin{aligned} p(C_i) &= 1 \text{ for each } i \\ p(v_j) &= \begin{cases} 1 & \text{if } f(v_j) = \text{true and} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Further, for each clause agent we set $x(c_i) = v_j$ if the only true literal in C_i is variable v_j and

$$x(\phi_j^k) = \begin{cases} C_i & \text{if } \phi_j^k \text{ corresponds to a true literal in clause } C_i \text{ and} \\ v_j & \text{otherwise.} \end{cases}$$

It is easy to see that (x, p) is an economic equilibrium for $\mathcal{M}(B)$.

Conversely, let (x, p) be an economic equilibrium for $\mathcal{M}(B)$. First let us realize that any trading cycle in x contains alternately players c_i and ϕ_j^k, so if all the ϕ-agents endowed with the same house v_j are trading, then they use up all the houses C_i corresponding to clauses containing variable v_j. This implies that if some agent c_i is not trading, then there exists also some agent ϕ_j^k with variable v_j contained in clause C_i who also is not trading. This is however impossible, since if $p(C_i) \geq p(v_j)$ then x does not assign to agent c_i the best house he can afford; and if $p(C_i) < p(v_j)$ then agent ϕ_j^k is not assigned the best house in his budget set – both is in a contradiction with the definition of an economic equilibrium. Further, if any of the agents ϕ_j^k, $k = 1, 2, \dots, n_j$ is trading, then so are all of them. For a contradiction suppose that e.g. $x(\phi_j^1) = C_i$ and $x(\phi_j^2) = v_j$. Then $p(C_i) = p(v_j)$ and agent ϕ_j^2 can afford house C_i, but is assigned a worse house – again a contradiction. So we set v_j to be true if all the agents ϕ_j^k, $k = 1, 2, \dots, n_j$ are trading in x and set v_j to be false if none of them is trading. This will be a consistent truth assignment in which each clause contains exactly one true literal. ■

6 Conclusion

In this paper we studied housing markets with duplicate houses. We extended several results known for the classical case and narrowed down the border line between easy and hard cases in the housing markets with duplicate houses: the existence of an economic equilibrium can be decided in polynomial time if the preferences of all agents are strict (in contrast with [5]), but it remains NP-complete if all the acceptable house-types are tied in the preference list of each agent. Because of this intractability result, we propose

Definition 1 *A pair* (x, p) *is an* α-defficient equilibrium *for housing market* $\mathcal{M}$ *with trichotomous preferences, if* x *is an allocation on* A*,* $p : H \to \mathbb{R}^+$ *is a price function and*

$$|\{a \in A; B_a(p) \cap H'(a) \neq \emptyset \ \& \ x(a) = \omega(a)\}| \leq \alpha.$$

Defficiency *of a housing market with trichotomous preferences is the minimum such* α *for which an* α*-defficient equilibrium exists.*

For a further research the following problem might be interesting:

PROBLEM **1** *Given a housing market* $\mathcal{M}$ *with trichotomus preferences, compute its defficiency.*

Although we know that the core of each housing market is nonempty, still little is known about its structure. As the number of agents that receive in a core allocation their first choice house (let us say that such an agent is *happy* in the allocation) may be different in different core allocations, one may consider the following problem:

PROBLEM **2** *For a given housing market* $\mathcal{M}$ *find a core allocation that maximizes the number of happy agents.*

Notice that a core allocation making all the agents happy exists if and only if the digraph whose vertex set is the agents set and $(i, j) \in E$ if $\omega(j) \in f_A(i)$ admits a cycle cover. If this is not the case, nothing is known for PROBLEM 2. Similarly, one may wish to maximize the number of trading and/or happy agents in Pareto optimal or strongly Pareto optimal permutation.

References

[1] D. Abraham, K. Cechlárová, D. Manlove and K. Mehlhorn, *Pareto optimality in house allocation problems*, Lecure Notes in Comp. Sci. **3341**, Algorithms and Computation, ISAAC 2004, Hong Kong, December 2004, Eds. R. Fleischer, G. Trippen, 3-15 (2004).

[2] D. Abraham, A. Blum, T. Sandholm, *Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges*, EC'07, June 11–15, 2007, San Diego, California.

[3] K. Cechlárová and J. Hajduková, *Computational complexity of stable partitions with* $\mathcal{B}-$*preferences*, Int. J. Game Theory 31 (2002) 3, 353–364.

[4] K. Cechlárová and A. Romero Medina, *Stability in coalition formation games*, International Journal of Game Theory 29, 2001, 487-494.

[5] S. Fekete, M. Skutella, G. Woeginger, *The complexity of economic equilibria for house allocation markets*, Inform. Process. Lett. 88/5 (2003), 219–223.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, San Francisco (1979).

[7] D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, Foundations of Computing, MIT Press, Cambridge (1989).

[8] J.E. Hopcroft and R.M. Karp, *An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs* SIAM Journal of Computing **2**: 225–231, (1973).

[9] K. Iwama, D. Manlove, S. Miyazaki and Y. Morita, *Stable marriage with incomplete lists and ties*, Proceedings of ICALP '99 (J. Wiedermann, P. Van Emde Boas, M. Nielsen Eds.), Lecture Notes in Computer Science Vol 1644, 443–452 (1999).

[10] A.V. Karzanov, *On finding maximum flows in network with special structure and some applications* (Russian), Matematicheskije Voprosy Upravlenija Proizvodstva **5**: 81–94; Moscow (1973).

[11] B. Korte, J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, Springer, Berlin (2008).

[12] T. Quint and J. Wako, *On houseswapping, the strict core, segmentation and linear programming*, Math. Oper. Research **29**(4): 861–877 (2004).

[13] A. E. Roth and A. Postlewaite, *Weak versus strong domination in a market with indivisible goods*, J. Math. Econ. **4**:131–137 (1977).

[14] L. Shapley and H. Scarf, *On cores and indivisibility*, J. Math. Econ. **1**:23–37 (1974).

[15] J. Wako, *A note on the strong core of a market with indivisible goods*, J. Math. Econ. **13**:189–194 (1984).

Better and simpler approximation algorithms for the stable marriage problem

Zoltán Király*

June 10, 2008

Abstract

We first consider the problem of finding a maximum stable matching if incomplete lists and ties are both allowed, but ties only for one gender. For this problem we give a simple, linear time 3/2-approximation algorithm, improving on the best known approximation factor 5/3 of Irving and Manlove [5]. Next, we show how this extends to the Hospitals/Residents problem with the same ratio if the residents have strict orders. We also give a simple linear time algorithm for the general problem with approximation factor 5/3, improving the best known 15/8-approximation algorithm of Iwama, Miyazaki and Yamauchi [7]. For the cases considered in this paper it is NP-hard to approximate with a factor of less than 21/19 by the result of Halldórsson et al. [3].

Our algorithms do not only give better approximation ratios than the cited ones, but are much simpler and run significantly faster. Also we may drop a restriction used in [5] and the analysis is substantially more moderate.

Keywords: stable matching, Hospitals/Residents problem, approximation algorithms

1 Introduction

An instance of the stable marriage problem consists of a set U of N men, a set V of N women, and a preference list for each person, that is a weak linear order (ties are allowed) on some members of the opposite gender. A pair $(m \in U,\ w \in V)$ is called acceptable if m is on the list of w and w is on the list of m. We model acceptable pairs with a bipartite graph $G = (U, V, E)$, (where E is the set of acceptable pairs; we may assume that if w is not on the list of m then m is also missing from the list of w). A matching in this graph consists of mutually disjoint acceptable pairs. A matching M is *stable* if there is no blocking pair, where an acceptable pair is *blocking*, if they strictly prefer each other to their current partners (the exact definition is given below). It is well-known that a stable matching always exists and can be found in linear time. An interesting problem, motivated by applications, is to find a stable matching of maximum size. This problem is known to be NP-hard for even very restricted cases [6, 8]. Moreover, it is APX-hard [2] and cannot be approximated within a factor of $21/19 - \delta$, even if ties occur only in the preference lists of one gender, furthermore if every list is either totally ordered or consists of a single tied pair [3]. As the applications of this problem are important, researchers started to develop good approximation algorithms in the last decade. We say that an algorithm is approximating with factor r if it gives a stable matching M with size $|M| \geq (1/r) \cdot |M_{\text{opt}}|$ where M_{opt} is a stable matching of maximum size. It is easy to give a 2-approximating algorithm, as any stable matching is maximal. The first non-trivial approximation

*Department of Computer Science and Communication Networks Laboratory, Eötvös University, Pázmány Péter sétány 1/C Budapest, Hungary H-1117. Research is supported by EGRES group (MTA-ELTE), OTKA grants NK 67867, K 60802 and T 046234, and by Hungarian National Office for Research and Technology programme NKFP072-TUDORKA7. E-mail: `kiraly@cs.elte.hu`

algorithm was given by Halldórsson et al. [3] and was recently improved by Iwama, Miyazaki and Yamauchi [7] to a 15/8-approximation. This was later improved for the special case, where ties are allowed for only one gender and only at the ends of the lists, by Irving and Manlove [5]. (We must emphasize that the second restriction is not needed for our results.) They gave a 5/3-approximating algorithm for this special case. Their algorithm also applies for the Hospitals/Residents problem (see later) if residents have strictly ordered lists. If, moreover, ties are of size 2, Halldórsson et al. [3] gave an 8/5-approximation and in [4] they described a randomized algorithm for this special case with expected factor of 10/7. The paper of Irving and Manlove [5] also gives a detailed list of known and possible applications that motivate investigating approximation algorithms.

We store the lists as priorities. For an acceptable pair (m, w) let $\text{pri}(m, w)$ be an integer from 1 up to N representing the priority of w for m. We say that $m \in U$ strictly prefers $w \in V$ to $w' \in V$ if $\text{pri}(m, w) > \text{pri}(m, w')$. Ties are represented by the same number, e.g., if m equally prefers w_1, w_2 and w_3 then $\text{pri}(m, w_1) = \text{pri}(m, w_2) = \text{pri}(m, w_3)$. Of course, $\text{pri}(m, w)$ is not related to $\text{pri}(w, m)$. We represent these priorities by writing $\text{pri}(m, w)$ and $\text{pri}(w, m)$ close to the corresponding end of edge mw.

Let M be a matching. If m is *matched* in M, or in other words m is not *single*, we denote m's partner by $M(m)$. Similarly we use $M(w)$ for the partner of woman w. A pair (m, w) is *blocking*, if $mw \in E \setminus M$ (they are an acceptable pair and they are not matched) and

- m is either single or $\text{pri}(m, w) > \text{pri}(m, M(m))$, and
- w is either single or $\text{pri}(w, m) > \text{pri}(w, M(w))$.

The famous algorithm of Gale and Shapley [1] for finding a stable matching is the following. Initially every man is active and makes any strict order of acceptable women according to the priorities (higher priority comes before lower).

Each active man m proposes to the next woman w on his strict list if w exists, otherwise (if he has processed the whole list) m inactivates himself. If the proposal was (temporarily) accepted then m inactivates himself, otherwise, if m was rejected, m keeps on proposing to the next woman from his list.

Each woman w who got some proposals keeps the best man as a partner and rejects all other men. More precisely, the first man m who proposed to w will be her first partner ($M(w) := m$). Later if w gets a new proposal from another man m', she rejects m' if $\text{pri}(w, m') \leq \text{pri}(w, M(w))$; otherwise w rejects $M(w)$, then $M(w)$ is re-activated, and finally w keeps $M(w) := m'$ as a new partner. The algorithm finishes if every man is inactive (either has a partner or has searched over his strict list). This algorithm runs in $O(|E|)$ time if G is given by edge-lists and sorting is done by bucket sort (we may suppose that every person has a non-empty list).

Theorem 1 (Gale-Shapley) *Algorithm GS defined above always ends in a stable matching M.*

PROOF: Let $mw \in E \setminus M$. If m never made a proposal to w then in the end he has a partner w' who precedes w on m's strict list, consequently $\text{pri}(m, w') \geq \text{pri}(m, w)$. Otherwise, w rejected m at some point, when w had a partner m' not worse than m. Observe that after w received a proposal, she will always have a partner. Moreover, when w changes partner, she always chooses a (strictly) better one. Thus in the end $\text{pri}(w, M(w)) \geq \text{pri}(w, m') \geq \text{pri}(w, m)$, so mw is not blocking. □

In what follows, we will use not only the statement of this theorem (as most of the previous results do), but the Algorithm GS itself with some modifications/extensions.

In the *Hospitals/Residents* problem the roles of women are played by hospitals and the roles of men are played by residents. Moreover, each hospital w has a positive integer capacity $c(w)$ (the number of free positions). Instead of matchings we consider *assignments*, that is a subgraph F of G, such that all residents have degree one in F, and each hospital w has degree at most $c(w)$. For a resident m who is assigned, $F(m)$ denotes the corresponding hospital. For a hospital w, $F(w)$ denotes the set of residents assigned to it. We say that w is full, if $|F(w)| = c(w)$ and otherwise under-subscribed. Here a pair (m, w) is *blocking*, if $mw \in E \setminus F$ (they are an acceptable pair and they are not assigned to each other) and

- m is either single or $\text{pri}(m, w) > \text{pri}(m, F(m))$, and
- w is either under-subscribed or $\text{pri}(w, m) > \text{pri}(w, m')$ for at least one resident $m' \in F(w)$.

An assignment is *stable* if there is no blocking pair. It is easy to modify Algorithm GS to give a stable assignment for the Hospitals/Residents problem. Each hospital w manages to keep a set of buckets indexed by integers up to N, containing each assigned resident m in the bucket indexed by $\text{pri}(w, m)$; and w also stores the number of assigned residents and a pointer to the first non-empty bucket. If hospital w gets a new proposal from resident m then it accepts him if w is under-subscribed or if $\text{pri}(w, m) > \text{pri}(w, m')$ for the worst assigned resident m'. Apart from this, the algorithm is the same. It clearly gives a stable assignment, and it is easy to see that also runs in $O(|E|)$ time (rejections can be decided in constant time as well as updating the data when accepting). We call this modified GS algorithm HRGS. As before, we are interested in giving a maximum size assignment, i.e., a stable assignment F with maximum number of edges (that is a maximum number of assigned residents).

In the next section we consider the special case of the maximum stable marriage problem, where each man's list is strictly ordered. We allow arbitrary number of arbitrarily long ties for each woman. We give a simple algorithm running in time $O(|E|)$. First we run Algorithm GS, then we give extra scores to single men, that raise their priorities. These men are re-activated and start making proposals from the beginning of their lists. A simple proof shows that this slightly modified algorithm gives a 3/2-approximation to the maximum stable marriage problem.

In Section 3 we show that this algorithm applies to the Hospitals/Residents problem as well, 3/2-approximating the maximum assignment in time $O(|E|)$.

Section 4 contains a slightly more complicated algorithm for the general case. First we run the algorithm of Section 2, then change the roles of men and women. In the second phase women get extra scores and make proposals to men. This algorithm still runs in linear-time, and gives a 5/3-approximation. Finally we propose some open problems.

2 Men have strictly ordered lists

In this section we suppose that the lists of men are strictly ordered. We are going to define extra scores, $\pi(m)$ for every man with the following properties. Initially $\pi(m) = 0$ and at any time $0 \leq \pi(m) < 1$ for each man. We also define adjusted priorities: $\text{pri}'(m, w) := \text{pri}(m, w)$ and $\text{pri}'(w, m) := \text{pri}(w, m) + \pi(m)$ for each acceptable pair (m, w). It is straightforward to see that if M is stable with respect to pri' then it is also stable with respect to pri.

We define a modification of Algorithm GS, that is called rmGS (reduced men-proposal GS), as follows. This algorithm starts with a stable matching, given extra scores and a set of active men. Run the original GS algorithm (active men make proposals; at the beginning of the algorithm they start from the beginning of their strict lists), where women use pri' to decide rejections. Stop when every man is inactive.

If some men with zero extra score remained single, we increase the score of those men to ε and re-activate them. In the next round they start making proposals from the beginning of their strict list. At any time let SM denote the set of single men, and $\Pi_0 := \{m \in U : \pi(m) = 0\}$. We fix $\varepsilon = 1/2$.

Our approximation algorithm is as follows:

Algorithm gsa1
run GS
FOR $m \in U \quad \pi(m) := 0$
WHILE $SM \cap \Pi_0 \neq \emptyset$

FOR $m \in SM \cap \Pi_0$
    $\pi(m) := \varepsilon$
    re-activate m
run rmGS

This simple algorithm runs in $O(|E|)$ time, as there are at most $2|E|$ proposals altogether. It is easy to see that Algorithm GSA1 gives a stable matching M with respect to the adjusted priority, hence M is stable for the original problem as well.

Let M_{opt} denote any maximum size stable matching (stable for the original priorities).

Theorem 2 *If men have strictly ordered preference lists, M is the output of Algorithm GSA1 and M_{opt} is a maximum size stable matching then*

$$|M_{\text{opt}}| \leq \frac{3}{2} \cdot |M|.$$

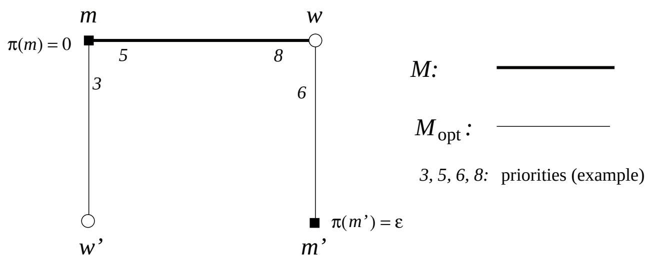


Figure 1: A path of length three in $M \cup M_{\text{opt}}$

Proof: We use an idea of Iwama, Miyazaki and Yamauchi [7]. Take the union of M and M_{opt}. We consider common edges as a two-cycle. Each component of $M \cup M_{\text{opt}}$ is either an alternating cycle (of even length) or an alternating path. It is enough to prove that in each component there are at most 3/2 times as many M_{opt}-edges as M-edges. This is clearly true for each component except for alternating paths of length three with the M-edge mw in the middle (see Figure 1).

We claim that such a component cannot exist. Suppose that $M(m) = w$, $M_{\text{opt}}(m) = w' \neq w$, $M_{\text{opt}}(w) = m' \neq m$ and that m' and w' are single in M. Observe first that w' never got a proposal during Algorithm GSA1. Consequently $\pi(m) = 0$ at the end, as otherwise he would have proposed to each acceptable woman. We may also conclude that $\text{pri}(m, w) > \text{pri}(m, w')$ because there are no ties in the men's lists. When the algorithm finishes, $\pi(m') = \varepsilon$, and m' proposed to every acceptable woman with this extra score, but w rejected him. This means that $\text{pri}'(w, m) \geq \text{pri}'(w, m') = \text{pri}(w, m') + \varepsilon$ consequently $\text{pri}(w, m) > \text{pri}(w, m')$. However, in this case edge mw blocks M_{opt}, a contradiction. □

We have an example (see Figure 2) showing that for our algorithm this bound is tight (a possible order of proposals and extra score increases is the following: $mw, m'w, m'w'', m''w''$, $\pi(m'') = \varepsilon$, $m''w''$).

Note: for open questions please see the section "Open Problems".

3 Hospitals/Residents with strictly ordered residents' lists

We consider the Hospitals/Residents problem with the restriction that residents have strict orders on acceptable hospitals. Note, that for real-life applications of this scheme, this assumption is realistic. Here, as appropriate, residents get extra scores. The adjusted priorities are defined as in Section 2.

For a reader familiar with this topic it is straightforward that after "cloning" of hospitals the previous algorithm runs with the same approximation ratio. However, we describe an algorithm for

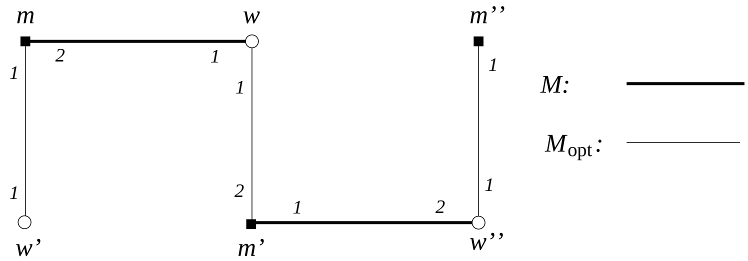


Figure 2: An example where GSA1 gives $|M| = (2/3) \cdot |M_{\text{opt}}|$

this problem in some detail for not only to newcomers, but for three more reasons: (i) the cloning is not well defined in the literature, (ii) we give a linear time algorithm, and (iii) for showing an example and a theorem at the end of this section.

We modify GSA1 by replacing GS by HRGS and define rmHRGS as a modification of HRGS analogously to the derivation of rmGS from GS. Here SM denotes the set of unassigned residents and again $\Pi_0 := \{m \in U : \pi(m) = 0\}$.

Algorithm hrgsa1
run HRGS
FOR $m \in U \quad \pi(m) := 0$
WHILE $SM \cap \Pi_0 \neq \emptyset$
    FOR $m \in SM \cap \Pi_0$
        $\pi(m) := \varepsilon$
        re-activate m
    run rmHRGS

Algorithm HRGSA1 also runs in time $O(|E|)$ (hospitals need to have $2N$ buckets), and gives a stable assignment F.

Theorem 3 *If residents have strictly ordered preference lists, F is the output of Algorithm HRGSA1 and F_{opt} is any maximum size stable assignment then*

$$|F_{\text{opt}}| \leq \frac{3}{2} \cdot |F|.$$

Proof: We suppose that positions at hospital w are numbered by $1 \ldots c(w)$. For the proof we make an auxiliary bipartite graph $G' = (U, V', E')$ and new preference lists as follows. The set U of residents remains unchanged. The set V' consists of the positions, i.e., $V' = \{w^i : w \in V,\ 1 \leq i \leq c(w)\}$. An edge connects resident m and position w^i if (m, w) was an acceptable pair (if hospital w was acceptable to m then all positions at w are acceptable to m). Each position w^i inherits the preference list of hospital w. For resident m we have to make a new (and also strict) preference list. Take the original list, and replace each w by $w^1 < w^2 < \ldots < w^{c(w)}$ (thus if w_1 was preferred by m to w_2 then all positions of w_1 will be preferred to all positions of w_2). If F is an assignment in G then it defines a matching M in G' by distributing edges of F incident to a hospital w to distinct positions $w^1, w^2, \ldots, w^{d_F(w)}$. And, conversely, any matching M of G' defines an assignment in G. The crucial observation is that if assignment F is stable in G then the associated matching M is stable in G', and if matching M is stable in G then the associated assignment F is stable in G. Moreover, if we imagine running Algorithm GSA1 on G', the resulting matching M corresponds to the assignment F given by Algorithm HRGSA1. Using these observations Theorem 2 implies this theorem. □

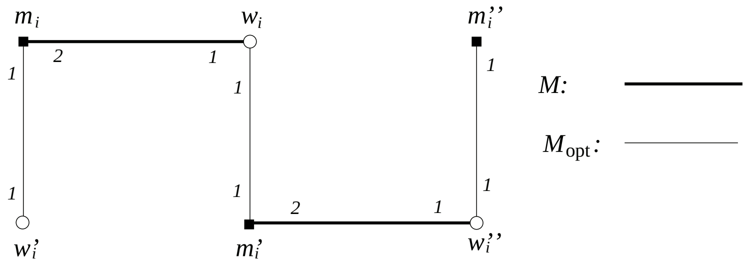


Figure 3: A building block of the example where HRGSA1 gives$|F| = (2/3) \cdot |F_{\text{opt}}|$

We note that the example on Figure 2 can be easily modified to show that this algorithm cannot achieve better approximation ratio than 3/2, not even if all hospitals have large capacities and if each hospital has an absolutely unordered list (i.e., $\text{pri}(w, m) = 1$ for every resident m).

We make c copies of the example shown in Figure 3, one for each $i = 1 \dots c$. Then glue together the c copies of w_i, the c copies of w_i' and the c copies of w_i''. Assign capacity c to each hospital (w, w' and w''). The following is a possible run of Algorithm HRGSA1 yielding an assignment F with $|F| = 2c$, while $|F_{\text{opt}}| = 3c$. First every resident m_i'' proposes to hospital w''. Next, every resident m_i proposes to hospital w; now hospitals w and w'' are full. Then every resident m_i' proposes first to w'' and then to w, but they are always rejected. So every resident m_i' gets an extra score. They proposes again to hospital w'' and they succeed. Now every resident m_i'' gets an extra score, and proposes again to w'' but they are rejected.

However, with a different type of restriction we are able to prove a stronger theorem. For a hospital w let $\tau(w)$ denote the length of the longest tie for w, and let $\lambda := \max_{w \in V} \tau(w)/(2c(w))$.

Theorem 4 *Algorithm HRGSA1 gives approximation ratio not worse than*

$$\frac{3}{2} - \frac{1}{6} \cdot \frac{1-\lambda}{1+\lambda}$$

PROOF: The proof is very technical, so we only sketch the idea of it. Every component of $M \cup M_{\text{opt}}$ (in G') that is a 5-path has a middle hospital-position w^i such that hospital w is full. Each such hospital has at most $\tau(w)/2$ positions in such a bad component and $c(w) - \tau(w)/2 \geq \frac{1-\lambda}{2\lambda}\tau(w)$ other positions lying in a good component (where the ratio of F-edges against the F_{opt}-edges is at least 3/4). In the "worst case" this component is a 7-path that can contain at most 3 such hospital-positions. □

4 General stable marriage

Now we consider the general maximum stable marriage problem. First we run the algorithm of Section 2, then change the roles of men and women. In the second phase women get extra scores and make proposals to men.

Accordingly, we also use extra scores $\pi(w)$ for women: initially $\pi(w) = 0$ and at any time $0 \leq \pi(w) < 1$ for each woman w. We also re-define adjusted priorities: $\text{pri}'(m, w) := \text{pri}(m, w) + \pi(w)$ and $\text{pri}'(w, m) := \text{pri}(w, m) + \pi(m)$ for each acceptable pair (m, w). It is straightforward to see that if M is stable with respect to pri′ then it is also stable with respect to pri.

In the first phase we run Algorithm GSA1, women do not get extra scores in this phase. Next, in the second phase we change the roles of men and women, in this phase we increase extra scores of women only. At the beginning of the second phase each woman makes any strict order of acceptable men according to the adjusted priorities (higher priority comes before lower).

We define Algorithm rwGS (reduced woman-proposal GS) similarly to Algorithm rmGS. The algorithm starts with a stable matching, given extra scores and a set of active women. Run the original GS algorithm with interchanged roles: active women make proposals, and men use pri′ to decide rejections. But here we have a major difference. If a woman w with $\pi(w) = 0$ is rejected by her actual partner at any time during the process then she gets $\pi(w) := \varepsilon/2$ extra scores, activates herself, and starts making proposals *from the beginning of her strict list.* Stop when every woman is inactive.

If some women with less than ε extra score remained single, we increase the score of those women to ε and re-activate them. In the next round they start making proposals from the beginning of their strict list. At any time let SW denote the set of single women and $\Pi := \{w \in V : \pi(w) \leq \varepsilon/2\}$. We also use $\varepsilon = 1/2$.

Our approximation algorithm is as follows.

Algorithm gsa2
Phase 1
run GSA1
Phase 2
FOR $w \in V$ $\pi(w) := 0$
WHILE $SW \cap \Pi \neq \emptyset$
FOR $w \in SW \cap \Pi$
$\pi(w) := \varepsilon$
re-activate w
run rwGS

First we claim that the algorithm runs in time $O(|E|)$. To see this we must consider two things. In Phase 2, every woman processes her strict list at most twice, so there are at most $2|E|$ proposals in the second phase. The strict lists of women can be calculated in $O(|E|)$ time altogether using bucket sort with $2N$ buckets.

Next we claim that the matching M given by the algorithm is stable with respect to pri′ consequently is stable with respect to pri. It is not hard to see this fact, we leave the standard and technical proof to the full version.

Theorem 5 *If M is the output of Algorithm GSA2 and M_{opt} is any maximum size stable matching then*

$$|M_{\text{opt}}| \leq \frac{5}{3} \cdot |M|.$$

Proof: First we need a technical lemma. Let M' denote the matching given at the end of Phase 1. Consider components of $M \cup M_{\text{opt}}$ as before.

Lemma 6 *Suppose $M \cup M_{\text{opt}}$ has a component that is an alternating path of length three, with the M-edge mw in the middle. Then $w' = M_{\text{opt}}(m)$ is matched in M'.*

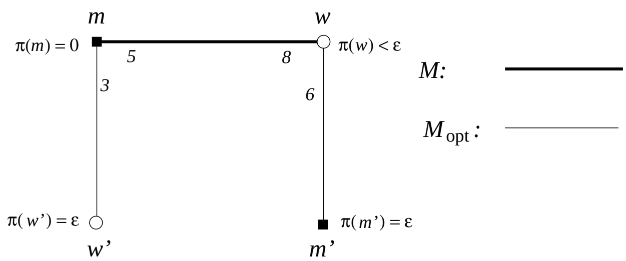


Figure 4: A path of length three in $M \cup M_{\text{opt}}$

Proof: Let $m' = M_{\rm opt}(w)$ (see Figure 4) and suppose w' was single at the end of Phase 1 (i.e., w' is single in M'). As this is a component of $M \cup M_{\rm opt}$, clearly both m' and w' are single in M, and moreover, as matched men never become single in Phase 2, m' is also single in M'.

First we observe that as w' is single in M', m did not propose to her during Phase 1, so $\pi(m) = 0$ (as $\pi(m)$ could only be positive after m searched over his strict list). However m' remained single, so $\pi(m') = \varepsilon$ at the end of the algorithm.

In Phase 2 w did not propose to m' (m' remained single, thus he did not receive any proposals), so $\pi(w) \le \varepsilon/2$. Next we use $M(w) = m$, and we consider two cases. If $M'(w) = m$ then in Phase 1, when w rejected m' the last time, she had $\mathrm{pri}'(w, m) \ge \mathrm{pri}'(w, m') = \mathrm{pri}(w, m') + \varepsilon$, so that in this case $\mathrm{pri}(w, m) > \mathrm{pri}(w, m')$. Otherwise, if $M'(w) \ne m$ then in Phase 2 w started to make proposals from the beginning of her strict list (that was made with respect to pri$'$ after Phase 1), but she did not propose to m', so $\mathrm{pri}'(w, m) \ge \mathrm{pri}'(w, m')$ also implying $\mathrm{pri}(w, m) > \mathrm{pri}(w, m')$.

At the beginning of Phase 2, $\pi(w')$ was set to ε, and w' remained single. This means that w' proposed to m and m rejected her. Consequently $\mathrm{pri}'(m, w) \ge \mathrm{pri}'(m, w')$, thus $\mathrm{pri}(m, w) > \mathrm{pri}(m, w')$. These arguments show that mw is blocking for $M_{\rm opt}$, a contradiction. □

We continue the proof of the theorem. Let SM denote the set of single men at the end of the algorithm, and $\widehat{SM} \subseteq SM$ denote the set of those single men who are matched in $M_{\rm opt}$. First note, that men in $\widehat{SM}$ were also single after Phase 1, since in Phase 2 men's positions do not decline. Observe that for each man $m \in \widehat{SM}$, woman $M_{\rm opt}(m)$ exists and is matched in both M' and M (at the end of any Phase at least one person in any acceptable pair is matched). We further partition $\widehat{SM}$ as follows. Let SM_1 consist of each man $m \in \widehat{SM}$, for whom man $M(M_{\rm opt}(m))$ is matched in $M_{\rm opt}$; and $SM_2 := \widehat{SM} \setminus SM_1$. Let $SM_1^1 := \{m \in SM_1 : M_{\rm opt}(M(M_{\rm opt}(m)))$ is matched in $M\}$ and $SM_1^2 := SM_1 \setminus SM_1^1$. By Lemma 6, for every man m in SM_1^2 woman $M_{\rm opt}(M(M_{\rm opt}(m)))$ is matched in M' (i.e., at the end of Phase 1). The next lemma plays a crucial role in the proof of the theorem.

Lemma 7

$$|SM_1| \le \frac{2}{3} \cdot |M|$$

Proof:

Case 1 $|SM_1^1| \ge |SM_1|/2$.

We form clubs, every club is led by a man in SM_1 and has one or two other men who are matched in M. For every man $m \in SM_1$ the second member of his club is $M(M_{\rm opt}(m))$. For each man $m \in SM_1^1$, his club contains a third member: $M(M_{\rm opt}(M(M_{\rm opt}(m))))$. We claim that these clubs are pairwise disjoint.

We formed one club for each man in SM_1 so it is enough to prove that any man m' who is matched in M belongs to at most one club. If $M(m')$ is single in $M_{\rm opt}$ then m' is not a member of any club. If $m = M_{\rm opt}(M(m')) \in SM$, then either $m \in SM_1$ and m' belongs to m's club or otherwise m' has no club at all. Otherwise m' belongs to the club of $m^* = M_{\rm opt}(M(M_{\rm opt}(M(m'))))$ as a third member, if m^* exists and $m^* \in SM_1^1$; and m' has no club otherwise.

Let MM denote the set men who are matched in M. We have

$$|M| = |MM| \ge |SM_1| + |SM_1^1| \ge \frac{3}{2} \cdot |SM_1|.$$

Case 2 $|SM_1^2| > |SM_1|/2$.

In this case we form different clubs, here the non-leader members will be men matched in M'. For every man $m \in SM_1$ the second member of his club is $M'(M_{\rm opt}(m))$. For each man $m \in SM_1^2$, his club contains a third member: $M'(M_{\rm opt}(M(M_{\rm opt}(m))))$. We claim that these clubs are also pairwise disjoint.

If $M'(m')$ is single in $M_{\rm opt}$ then m' is not a member of any club. If $m = M_{\rm opt}(M'(m')) \in SM$, then either $m \in SM_1$ and m' belongs to m's club or otherwise m' has no club at all. Otherwise

m' belongs to the club of $m^* = M_{\text{opt}}(M(M_{\text{opt}}(M'(m'))))$ as a third member, if m^* exists and $m^* \in SM_1^2$; and m' has no club otherwise.

Let MM' denote the set of men who are matched in M'. As men matched after Phase 1 remain matched till the end, we have

$$|M| = |MM| \geq |MM'| \geq |SM_1| + |SM_1^2| \geq \frac{3}{2} \cdot |SM_1|. \qquad \square$$

We are ready to finish the proof of the theorem. Let MM_{opt} denote the set of men who are matched in M_{opt}. We claim that $|MM \cap MM_{\text{opt}}| \leq |MM| - |SM_2|$. This is true because $|SM_2|$ is the number of components of $M \cup M_{\text{opt}}$ isomorphic to a path with two edges and with a woman in the middle, and for each such path the M-matched man is single in M_{opt}.

$$|M_{\text{opt}}| = |MM_{\text{opt}}| = |MM \cap MM_{\text{opt}}| + |SM \cap MM_{\text{opt}}| \leq$$
$$\leq (|MM| - |SM_2|) + (|SM_1| + |SM_2|) \leq |M| + \frac{2}{3} \cdot |M| = \frac{5}{3} \cdot |M|. \qquad \square$$

5 Open Problems

Open Problem 1 *Is it possible to improve the performance of GSA1 if we use smaller ε, increase extra scores more than once, and give extra scores to not only single men, but also to partners of each woman who is a neighbor of a single man?*

Open Problem 2 *Is it possible to improve the performance of GSA1 if we use the method of Irving and Manlove [5]* after *GSA1?*

Open Problem 3 *Is it possible to improve the performance of GSA2 if we use smaller ε, increase extra scores more than once, alternately for men and women? And if giving extra scores to not only persons remained single?*

Open Problem 4 *Is it possible to improve the performance of GSA2 if we use the method of Halldórsson et al. [3], or the method of Iwama, Miyazaki and Yamauchi [7]* after *GSA2?*

5.1 Acknowledgement

I am grateful to Tamás Fleiner for his valuable advice. This paper was also submitted to ESA'08. I am also indebted to the referees of ESA'08 for pointing out many typos and shortages in the first version of this paper.

References

[1] D. Gale, L. S. Shapley, College admissions and the stability of marriage *Amer. Math. Monthly* **69** (1962) pp. 9–15.

[2] M. M. Halldórsson, R. W. Irving, K. Iwama, D. F. Manlove, S. Miyazaki, Y. Morita, S. Scott, Approximability results for stable marriage problems with ties *Theor. Comput. Sci.* **306** (2003) pp. 431–447.

[3] M. M. Halldórsson, K. Iwama, S. Miyazaki, H. Yanagisawa, Improved approximation results for the stable marriage problem *ACM Trans. Algorithms* **3** No. 3, (2007) Article 30.

[4] M. M. Halldórsson, K. Iwama, S. Miyazaki, H. Yanagisawa, Randomized approximation of the stable marriage problem *Theor. Comput. Sci.* **325** (2004) pp. 439–465.

[5] R. W. Irving, D. F. Manlove, Approximation algorithms for hard variants of the stable marriage and hospitals/residents problems *Journal of Combinatorial Optimization* (2007) DOI: 10.1007/s10878-007-9133-x

[6] K. Iwama, D. F. Manlove, S. Miyazaki, Y. Morita, Stable marriage with incomplete lists and ties *Proceedings of the 26th International Colloquium on Automata, Languages and Programming* (1999) **LNCS 1644** pp. 443–452.

[7] K. Iwama, S. Miyazaki, N. Yamauchi, A 1.875-approximation algorithm for the stable marriage problem *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (2007) pp. 288–297.

[8] D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, Y. Morita, Hard variants of stable marriage *Theor. Comput. Sci.* **276** (2002) pp. 261–279.

Optimal Popular Matchings

Telikepalli Kavitha Meghana Nasre

Indian Institute of Science, Bangalore, India.
{kavitha, meghana}@csa.iisc.ernet.in

Abstract. In this paper we consider the problem of computing an "optimal" popular matching. We assume that our input instance $G = (\mathcal{A} \cup \mathcal{P}, E_1 \dot{\cup} \cdots \dot{\cup} E_r)$ admits a popular matching and we are asked to return not any popular matching but an *optimal* popular matching, where the definition of optimality is given as a part of the problem statement; for instance, the optimality criterion could be *fairness*, in which case the problem is to return a *fair* popular matching, which is a maximum cardinality popular matching that matches as few applicants to their rank r posts as possible, subject to this constraint, matches as few applicants to their rank $r-1$ posts as possible, and so on. We show an $O(N^2 + m)$ algorithm for the problem of computing an optimal popular matching, assuming that the preference lists are strict, where m is the number of edges and N is the number of applicants in G.

1 Introduction

In this paper we consider the problem of computing an *optimal* popular matching in a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, \mathcal{E})$ with one-sided preference lists. Optimality is described succinctly as a part of the problem statement, for instance, *rank-maximality*, or *fairness*, or *min-cost* (where each $(a,p) \in \mathcal{E}$ has a cost $c(a,p)$ associated with it) can be considered as optimality. More generally, we assume a partial order $\leq_P$ on matchings that obeys the following natural properties:

(A) if M_1, M_2 are two matchings that contain an edge e and $M_1 \leq_P M_2$, then $M_1 \setminus \{e\} \leq_P M_2 \setminus \{e\}$,
(B) if $M_1 \leq_P M_2$ and neither M_1 nor M_2 contains edge e and $M_1 \cup \{e\}$ and $M_2 \cup \{e\}$ are matchings, then $M_1 \cup \{e\} \leq_P M_2 \cup \{e\}$.

In this paper we consider the problem of computing a popular matching in G that is an optimal element (maximal/minimal as the case may be) with respect to $\leq_P$ among all popular matchings in G. We first describe below the popular matching problem.

The popular matching problem. An instance of the *popular matching problem* is a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, \mathcal{E})$ and a partition $\mathcal{E} = E_1 \dot{\cup} E_2 \cdots \dot{\cup} E_r$ of the edge set. The vertices of $\mathcal{A}$ are called *applicants* and the vertices of $\mathcal{P}$ are called *posts*. For each $1 \leq i \leq r$, the elements of E_i are called the edges of rank i. If $(a,p) \in E_i$ and $(a,p') \in E_j$ with $i < j$, we say that a prefers p to p'. This ordering of posts adjacent to a is called a's preference list. For any applicant a and any rank i, where $1 \leq i \leq r$, we assume that there is at most one post p such that $(a,p) \in E_i$, that is, we assume that preference lists are strictly ordered.

A *matching* M of G is a set of edges such that no two edges share an endpoint. We denote by $M(a)$ the post to which applicant a is matched in M. We say that an applicant a prefers matching M' to M if (i) a is matched in M' and unmatched in M, or (ii) a is matched in both M' and M, and a prefers $M'(a)$ to $M(a)$. M' is *more popular than* M, denoted by $M' \succ M$, if the number of applicants that prefer M' to M exceeds the number of applicants that prefer M to M'. A matching M is *popular* if and only if there is no matching M' that is more popular than M.

The *popular matching problem* is to determine if a given instance admits a popular matching, and to find such a matching, if one exists. The popular matching problem was considered in [1] and efficient algorithms were given to determine if G admits a popular matching and to compute a maximum cardinality popular matching. Note that popular matchings may have different sizes, and a largest such matching may be smaller than a maximum-cardinality matching.

1.1 Problem Definition

In this paper we assume that the input instance G admits a popular matching and here we are not content returning any popular matching or any maximum cardinality popular matching. Our goal is to compute an *optimal* popular matching, where the definition of optimality is given succinctly as a part of the problem definition. For instance, the problem description could state *fairness*, or *rank-maximality*, or min-cost of matched edges as optimality, which means that, among all popular matchings in G, we have to return that popular matching which is the most optimal with respect to fairness, or rank-maximality, or has the least cost, as the case may be. We define the terms fair and rank-maximal below.

The fair matching problem. The fair matching problem in a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E_1 \dot\cup E_2 \cdots \dot\cup E_r)$ asks for a matching M that satisfies the properties below:
(i) M is a maximum-cardinality matching in G, and
(ii) among all maximum cardinality matchings in G, M matches the least number of applicants to their rank r posts, subject to this constraint, matches the least number of applicants to their rank $r-1$ posts, subject to this constraint, matches the least number of applicants to their rank $r-2$ posts, and so on.

Currently, there are no combinatorial algorithms known for computing a fair matching. In this paper we consider the *fair popular* matching problem. A fair popular matching M is defined as follows:
(i) M is a maximum cardinality popular matching in G, and
(ii) among all maximum cardinality popular matchings in G, M matches the least number of applicants to their rank r posts, subject to this constraint, matches the least number of applicants to their rank $r-1$ posts, subject to this constraint, matches the least number of applicants to their rank $r-2$ posts, and so on.

Rank-maximal matchings. A rank-maximal matching M in a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E_1 \dot\cup \cdots \dot\cup E_r)$ is a matching that matches the maximum number of applicants to their rank one posts, subject to this constraint, matches the maximum number of applicants to their rank two posts, and so on. There are efficient combinatorial algorithms known for computing a rank-maximal matching [5]. However, there are no efficient combinatorial algorithms known for computing a *maximum cardinality* matching that is the most rank-maximal among all maximum cardinality matchings. Here we consider the problem of computing a popular matching that is the most rank-maximal among all popular matchings.

Min-cost popular matchings. Another natural definition of optimality is the following: we assume that each edge (a, p) has a non-negative cost $c(a, p)$ associated with it. $G = (\mathcal{A} \cup \mathcal{P}, E_1 \dot\cup \cdots \dot\cup E_r)$

admits a popular matching and we want to compute that popular matching M in G such that $\sum_{a \in \mathcal{A}} c(a, M(a))$ is the minimum among all popular matchings.

The problem of computing a fair popular matching/rank-maximal popular matching/min-cost popular matching can be solved by assigning suitable costs to the edges of an appropriate bipartite graph H derived from $G = (\mathcal{A} \cup \mathcal{P}, E_1 \dot\cup \cdots \dot\cup E_r)$ and computing a min-cost or max-cost perfect matching in H. It is easy to see that the order $\leq_P$ on matchings defined with respect to fairness, rank-maximality, or min-cost satisfies properties (A) and (B) defined at the beginning of this section.

Here we present a simple combinatorial algorithm that runs in $O(N^2 + m)$ time for the problem of computing an optimal popular matching in G, where m is the number of edges and N is the number of applicants in G. We assume that given two matchings M_1 and M_2, we can determine if $M_1 \leq_P M_2$ or $M_2 \leq_P M_1$ in $O(N)$ time, which is a reasonable assumption (and is indeed the case for fairness, rank-maximality, or min-cost).

1.2 Related Results

The notion of popular matchings was originally introduced by Gardenfors [3] in the context of the stable marriage problem with two-sided preference lists. It is well known that every stable marriage instance admits a weakly stable matching (one for which there is no pair who strictly prefer each other to their partners in the matching). In fact, there can be an exponential number of weakly stable matchings, and so Gardenfors considered the problem of finding one with additional desirable properties, such as popularity. Gardenfors showed that when preference lists are strictly ordered, every stable matching is popular. He also showed that when preference lists contain ties, there may be no popular matching.

Abraham et al. in [1] presented an $O(m+n)$ algorithm (for the case of strictly ordered preference lists) to determine if the input instance G on m edges and n vertices admits a popular matching and compute one, if it exists. For the case when the preference lists need not be strictly ordered, they showed an $O(m\sqrt{n})$ algorithm. Manlove and Sng [7] generalized the algorithms of [1] to the case where each post has an associated *capacity*, indicating the number of applicants that it can accommodate. (They described this in the equivalent context of the house allocation problem.) They gave an $O(\sqrt{C}n_1 + m)$ time algorithm for the strictly ordered preference lists case, and an $O((\sqrt{C} + n_1)m)$ time algorithm for the case with ties in preference lists, where n_1 is the number of applicants, and C is the total capacity of all of the posts. In [8] Mestre designed an efficient algorithm for the *weighted* popular matching problem, where each applicant is assigned a priority or weight, and the definition of popularity takes into account the priorities of the applicants. Mestre's algorithm for the case of strictly ordered preference lists has $O(n + m)$ complexity and for the case with ties in preference lists, the complexity is $O(\min(k\sqrt{n}, n)m)$, where k is the number of distinct weights assigned to applicants. Assuming that the input instance G admits a popular matching, Abraham and Kavitha [2] considered the problem of computing a shortest-length *voting path*[1] given a starting matching M_0 in G. They gave an $O(m + n)$ algorithm for this problem when the preference lists are strictly ordered and an $O(m\sqrt{n})$ algorithm for the case of ties in preference lists.

In the case of two-sided preference lists where the two sides of the bipartite graph are considered *men* and *women*, a stable matching is considered the ideal answer to what is a desirable matching.

[1] A voting path of length k is a sequence of matchings $\langle M_0, M_1, \ldots, M_k \rangle$ such that M_k is popular and $M_k \succ M_{k-1} \cdots \succ M_0$

However there is a wide spectrum of stable matchings ranging from men-optimal stable matchings to women-optimal stable matchings. Irving, Leather, and Gusfield [6] considered the problem of computing a stable matching that is optimal under some more equitable criterion of optimality. In fact, much work has been done in the two-sided preference lists setting on finding stable matchings that satisfy additional criteria (see [4] for an overview). In the same vein, assuming that the input instance G admits a popular matching, here we ask for an *optimal* popular matching where optimality is defined as a part of the problem statement.

2 Preliminaries

In this section we review the algorithmic characterisation for computing a popular matching from [1]. Since our problem is restricted to the case where preference lists do not have ties, we will present the characterisation from [1] of popular matchings for strictly ordered preference lists. For convenience, a dummy post $\ell(a)$ is added at the end of a's preference list, for each applicant a, and the edge $(a, \ell(a))$ is assigned rank $r+1$. Thus henceforth, the edge set $\mathcal{E} = E_1 \dot{\cup} \cdots \dot{\cup} E_{r+1}$ and any unmatched applicant a will be assumed to be matched to $\ell(a)$.

For each applicant a, define a *first choice post* for a, denoted by $f(a)$, and a *second choice post* for a, denoted by $s(a)$, as follows. The post $f(a)$, is one that occurs at the top of a's preference list, that is, it is a's most preferred post. The post $s(a)$ is the most preferred post on a's list that is *not* $f(a')$ for any applicant a'. Note that by above the definition, f-posts are disjoint from s-posts. For each applicant a, $f(a)$ is guaranteed to exist if its preference list is non-empty. Note that the dummy post $\ell(a)$ added at the end of a's preference list ensures that $s(a)$ always exists for each applicant a.

The following lemma from [1] characterises a popular matching.

Lemma 1. *A matching M is popular if and only if*
(1) every f-post is matched in M,
(2) for each applicant a, $M(a) \in \{f(a), s(a)\}$.

Let G' denote the graph in which each applicant a has exactly two edges, $(a, f(a))$ and $(a, s(a))$ incident to it. From Lemma 1, it is immediate that the input instance G admits a popular matching if and only if the graph G' defined above admits an $\mathcal{A}$-perfect matching. Thus the problem of determining if G admits a popular matching is now easy to solve. Algorithm 2.1 contains the main idea and [1] presents an efficient implementation of this idea that runs in $O(m+n)$ time (where m and n are the number of edges and vertices in G).

Algorithm 2.1 A simple framework from [1] to compute a popular matching in G

for each $a \in \mathcal{A}$ **do**
– determine the posts $f(a)$ and $s(a)$
end for
Construct the graph G' on vertex set $\mathcal{A} \cup \mathcal{P}$ and edge set $\cup_{a\in\mathcal{A}}\{(a, f(a)), (a, s(a))\}$.
if G' admits an $\mathcal{A}$-perfect matching **then**
return an $\mathcal{A}$-perfect matching M in G' that matches all f-posts.
else
return "G admits no popular matching"
end if

3 Our Algorithm

In this section we describe our algorithm to compute an *optimal* popular matching in G with respect to the optimality criterion specified as a part of the input. We know that any popular matching can match an applicant either to its f-post or to its s-post. Since our problem is to compute a matching that is necessarily popular, we can delete all posts p from G where p is not an f-post or an s-post. Thus G becomes the graph $(\mathcal{A} \cup \mathcal{P}, E')$ where E' consists of edges $(a, f(a))$ and $(a, s(a))$ for each $a \in \mathcal{A}$.

An optimal popular matching, by virtue of being a popular matching, has to match every f-post. So if a is an applicant such that it is the only applicant that considers $f(a)$ as its top post, we know that such an applicant a has to be matched to $f(a)$ by any popular matching. So we include all such pairs $(a, f(a))$ in our matching that we will return and delete such vertices $a, f(a)$ from G. So from now on we can assume that every applicant in G has degree exactly 2. Further we know that G admits an $\mathcal{A}$-perfect matching since any popular matching is an $\mathcal{A}$-perfect matching in $(\mathcal{A} \cup \mathcal{P}, E')$.

Let N be the number of applicants in G. We will order the applicants $a_1, \ldots, a_N$ in an arbitrary manner. We will work with the graphs H_k, for $1 \leq k \leq N$, where H_k is the graph on vertex set $\{a_1, \ldots, a_k\} \cup \{f(a_1), \ldots, f(a_k), s(a_1), \ldots, s(a_k)\}$ and edges $\cup_{j=1}^{k}\{(a_j, f(a_j)), (a_j, s(a_j))\}$. Any popular matching in the graph G restricted to applicants $\{a_1, \ldots, a_k\}$ is a matching of size k in H_k that matches all the posts $f(a_1), \ldots, f(a_k)$. We present a simple iterative strategy for computing an optimal popular matching in G: for each $1 \leq k \leq N$, we will compute a matching M_k that satisfies the following 2 properties.

(1) M_k is a matching of size k in H_k that matches all the posts $f(a_1), \ldots, f(a_k)$.
(2) among all the matchings that satisfy (1), M_k is optimal with respect to $\leq_P$.

We will compute such an M_k iteratively. Say we have already computed the desired matching M_{k-1}; in the current step, we add to the graph H_{k-1} the applicant a_k and the posts $f(a_k), s(a_k)$ (if they do not yet belong to H_{k-1}) and the edges $(a_k, f(a_k))$ and $(a_k, s(a_k))$ to form the graph H_k. We will show that M_k can be computed by *augmenting* M_{k-1} appropriately.

M_k has size k in H_k, thus it has to match each of the applicants $a_1, \ldots, a_k$. Due to the fact that we augment M_{k-1} in H_k, each of $a_1, \ldots, a_{k-1}$ remains matched (to either their f-post or s-post). Also since M_k needs to match a_k, either $(a_k, f(a_k))$ or $(a_k, s(a_k))$ has to belong to M_k. Our algorithm tries both the options:

(1) it tries to find augmenting paths p_k and q_k with respect to M_{k-1} in H_k in order to match a_k to $f(a_k)$ and to $s(a_k)$, respectively. It is easy to show that at least one of p_k, q_k has to exist.
(2) If p_k does not exist, then $M_k = M_{k-1} \oplus q_k$ and if q_k does not exist, then $M_k = M_{k-1} \oplus p_k$. If both p_k and q_k exist, then the more optimal of $M_{k-1} \oplus p_k$ and $M_{k-1} \oplus q_k$ is chosen as M_k.

We present our algorithm as Algorithm 3.1 and show that this simple method suffices.

Theorem 1 proves the correctness of our algorithm.

Theorem 1. *The matching M_N returned by our algorithm is a popular matching that is optimal with respect to $\leq_P$.*

It is easy to see that the matching M_N returned by our algorithm is popular. Note that, for each i, M_i is a maximum cardinality matching in H_i. Thus M_N is a maximum cardinality matching

Algorithm 3.1 Our algorithm to compute an optimal popular matching

for each $a \in \mathcal{A}$ **do**
  – determine the posts $f(a)$ and $s(a)$.
end for
– Set any order among the applicants so that the applicants can be labelled $a_1, a_2, \ldots, a_N$.
– Let H_1 be the graph on vertex set $\{a_1\} \cup \{f(a_1), s(a_1)\}$ and edge set $\{(a_1, f(a_1)), (a_1, s(a_1))\}$); let M_1 be the matching $\{(a_1, f(a_1))\}$.
– Initialize $i = 2$.
while $i \leq N$ **do**
  Update H_{i-1} to H_i by adding the applicant a_i and posts $f(a_i), s(a_i)$ (if they do not already exist) to the vertex set and the edges $(a_i, f(a_i))$ and $(a_i, s(a_i))$ to the edge set.
  if $f(a_i)$ is newly added **then**
    $M_i = M_{i-1} \cup \{(a_i, f(a_i))\}$.
  else
    find an augmenting path p_i with respect to M_{i-1} in H_{i-1} that begins with the edge $(a_i, f(a_i))$
    find an augmenting path q_i with respect to M_{i-1} in H_{i-1} that begins with the edge $(a_i, s(a_i))$
    if p_i (similarly, q_i) does not exist **then**
      $M_i = M_i \oplus q_i$ (resp., $M_i \oplus p_i$).
    else if both p_i and q_i exist **then**
      $M_i =$ the more optimal of $M_i \oplus p_i$ and $M_i \oplus q_i$.
    end if
  end if
  $i = i + 1$.
end while
– Return M_N.

in H_N; we know that $H_N = (\mathcal{A} \cup \mathcal{P}, E')$ admits an $\mathcal{A}$-perfect matching since the input instance admits a popular matching. Thus M_N is an $\mathcal{A}$-perfect matching. Also, by construction, we never let an f-post remain unmatched. Thus, M_N is an $\mathcal{A}$-perfect matching in $(\mathcal{A} \cup \mathcal{P}, E')$ that matches all f-posts. Thus M_N is a popular matching in the input instance.

We now need to show that among all popular matchings, M_N is optimal with respect to $\leq_P$. We will prove this by induction: we will show that for each i, M_i is a matching of size i in H_i that matches all posts $f(a_1), \ldots, f(a_i)$ and amongst all such matchings, M_i is optimal.[2]

We will now show that for all $1 \leq i \leq N$, M_i is optimal in H_i subject to the constraint that M_i has to match all of $a_1, \ldots, a_i$ and $f(a_1), \ldots, f(a_i)$. The base case $i = 1$ is trivial. By induction hypothesis, we assume that M_{k-1} is optimal in H_{k-1} subject to the constraint that it has to match all of $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$. Using this hypothesis, we will show that M_k is optimal in H_k subject to the constraint that it has to match all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$.

We consider two cases: (i) $f(a_k)$ is not present in H_{k-1} and (ii) $f(a_k)$ is present in H_{k-1}. Lemma 3 deals with the first case and Lemmas 4 and 5 deal with the second case.

Lemma 2. *Let M_{k-1} be an optimal matching in H_{k-1} subject to the constraint that the vertices $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$ have to be matched. If $f(a_k)$ is not present in H_{k-1}, then $M_k = M_{k-1} \cup \{(a_k, f(a_k))\}$ is optimal in H_k subject to the constraint that all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$ have to be matched.*

[2] Note that we can compare two matchings M, M' of H_i with respect to $\leq_P$ by extending each of M, M' to $\{a_1, \ldots, a_N\}$ by considering $\{a_{i+1}, \ldots, a_N\}$ as unmatched and we can compare the extended matchings with respect to $\leq_P$.

Proof. It is immediate from the definitions of M_{k-1} and M_k that M_k matches all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$. What remains to prove that M_k is a most optimal such matching.

Suppose not, let O_k be a more optimal such matching in H_k. We know that $f(a_k)$ is not an f-post for any applicant in $\{a_1, \ldots, a_{k-1}\}$ (by virtue of the fact that $f(a_k)$ is not present in H_{k-1}). Since O_k has to satisfy the constraint that all f-posts in H_k are matched, it follows that $O_k(a_k) = f(a_k)$. Thus O_k and M_k agree on the edge $e = (a_k, f(a_k))$.

Now consider the matching $O_k - \{e\}$. This is a matching in H_{k-1} that matches all the vertices $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$ since O_k matches all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$. We know that M_{k-1} is a most optimal such matching in H_{k-1}, implying that (let us assume that optimality is given by *maximality* with respect to $\leq_P$):

$$O_k - \{e\} \leq_P M_{k-1}$$

Adding the edge e to both the matchings above, we have $O_k \leq_P M_{k-1} \cup \{e\} = M_k$, contradicting our assumption that O_k is more optimal than M_k. □

We now deal with the case when $f(a_k)$ is present in H_{k-1}. In this case, we try to find augmenting paths p_k and q_k in H_k. Note that at least one of p_k, q_k has to exist since H_k admits a matching of size k (any $\mathcal{A}$-perfect matching of $(\mathcal{A} \cup \mathcal{P}, E')$ restricted to $a_1, \ldots, a_k$ is a matching of size k in H_k) - thus there has to exist an augmenting path with respect to the $(k-1)$-sized matching M_{k-1} in H_k. Say, p_k exists and q_k does not exist. Then we show the following.

Lemma 3. *$M_{k-1} \oplus p_k$ is an optimal matching in H_k subject to the constraint that it has to match all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$.*

Proof. It is easy to see that $M_k = M_{k-1} \oplus p_k$ matches all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$. We need to show that M_k is a most optimal such matching.

Suppose not and let O_k be a more optimal such matching in H_k. The fact that q_k does not exist implies that any matching that matches all of $a_1, \ldots, a_k$ in H_k has to match a_k to $f(a_k)$. This forces O_k to match the applicant that was matched by M_{k-1} to $f(a_k)$ to be matched to its s-post. In fact, *every* edge in p_k that is present in M_k has to be present in O_k. Thus O_k and M_k contain the same subset of edges of p_k. Call these edges $e_1, \ldots, e_t$ (refer to Figure 1).

Now consider the matching $O_k \oplus p_k$. This is a matching in H_{k-1} that contains the same edges as O_k outside p_k and the edges of p_k present in this matching are $p_k - \{e_1, \ldots, e_t\}$ (the bold edges of Figure 1) - all these bold edges are present in M_{k-1}. Since O_k and M_{k-1} match all the vertices $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$, it follows that $O_k \oplus p_k$ matches all the vertices $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$. Thus $O_k \oplus p_k$ is a matching in H_{k-1} that matches all the vertices $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$. Since M_{k-1} is an optimal such matching, we have

$$M_{k-1} \geq_P O_k \oplus p_k.$$

Now deleting the edges $p_k - \{e_1, \ldots, e_t\}$ from both the matchings M_{k-1} and $O_k \oplus p_k$, and adding the edges $e_1, \ldots, e_t$ to both these matchings, we get $M_{k-1} \oplus p_k \geq_P O_k$. In other words, $M_k \geq_P O_k$, contradicting O_k to be more optimal than M_k. □

The case when q_k exists and p_k does not exist is absolutely similar to above lemma. The only case that we are left with is the case when both p_k and q_k exist. In this case our algorithm computes both $M_{k-1} \oplus p_k$ and $M_{k-1} \oplus q_k$ and chooses the more optimal of the two matchings to be M_k. We now have to show that M_k is what we desire.

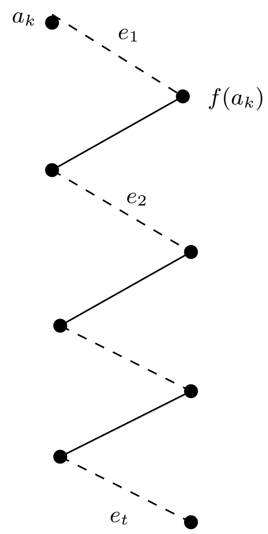


Fig. 1. The path p_k: the bold edges are present in M_{k-1} and the dashed edges are in M_k and in O_k.

Lemma 4. *M_k, the more optimal of $M_{k-1} \oplus p_k$ and $M_{k-1} \oplus q_k$, is a most optimal matching in H_k that matches all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$.*

Proof. It is obvious that M_k matches all of $a_1, \ldots, a_k$ and $f(a_1), \ldots, f(a_k)$. Suppose M_k is not an optimal such matching, let O_k be a more optimal such matching in H_k. The matching O_k has to match a_k to either $f(a_k)$ or to $s(a_k)$. We will show the following:

Claim 1. If $O_k(a_k) = f(a_k)$, then $O_k \leq_P M_{k-1} \oplus p_k$.
Claim 2. If $O_k(a_k) = s(a_k)$, then $O_k \leq_P M_{k-1} \oplus q_k$.

We know that either $O_k(a_k) = f(a_k)$ or $O_k(a_k) = s(a_k)$, which implies by Claims 1 and 2 that either $O_k \leq_P M_{k-1} \oplus p_k$ or $O_k \leq_P M_{k-1} \oplus q_k$. Thus M_k, which is the more optimal of $M_{k-1} \oplus p_k$ and $M_{k-1} \oplus q_k$ is at least as optimal as O_k, contracting our assumption that O_k is more optimal than M_k.

Hence, what we need to show are Claims 1 and 2.

Proof of Claim 1. If $O_k(a_k) = f(a_k)$, then as we argued in the proof of Lemma 4, it follows that $M_{k-1} \oplus p_k$ and O_k contain the same subset of edges of p_k. Now consider $O_k \oplus p_k$: this is a matching in H_{k-1} that matches all of $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$. Since M_{k-1} is an optimal matching in H_{k-1} that matches all of $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$, it follows that $O_k \oplus p_k \leq_P M_{k-1}$, or equivalently, $O_k \leq_P M_{k-1} \oplus p_k$. This finishes the proof of Claim 1.

Proof of Claim 2. This proof is absolutely similar to the proof of Claim 1. If $O_k(a_k) = s(a_k)$, then as we argued in the proof of Lemma 4, it follows that $M_{k-1} \oplus q_k$ and O_k contain the same subset of edges of q_k. Now consider $O_k \oplus q_k$: this is a matching in H_{k-1} that matches all of $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$. Since M_{k-1} is an optimal matching in H_{k-1} that matches all of $a_1, \ldots, a_{k-1}$ and $f(a_1), \ldots, f(a_{k-1})$, it follows that $O_k \oplus q_k \leq_P M_{k-1}$, or equivalently, $O_k \leq_P M_{k-1} \oplus q_k$. This finishes the proof of Claim 2. □

This completes the proof of Theorem 1. We will now analyse the running time of Algorithm 3.1. The f and s-posts of all applicants can be computed in $O(m+N)$ time. The main while loop of Algorithm 3.1 runs for N iterations and each iteration takes $O(N)$ time to construct the augmenting paths p_i, q_i and to compare $M_{i-1} \oplus p_i$ and $M_{i-1} \oplus q_i$. Thus our algorithm runs in $O(N^2+m)$ time.

4 Discussion

In this paper we gave an $O(N^2+m)$ algorithm for computing an *optimal popular* matching for instances with strict preference lists. This raises the question of extending our algorithm for the case when ties are allowed in the preference lists. Note that our assumption of *strict* preferences was critical in proving the correctness of our algorithm. When preference lists are allowed to have ties, an applicant can have more than one post as its f-post and similarly, it can have more than one post as its s-post.

Recall that our iterative algorithm, in case of strict preference lists, updates the current optimal matching M_{k-1} along one of the two augmenting paths p_k or q_k to get an optimal popular matching M_k. The fact that every applicant has degree exactly 2 in H_k ensures that we need to consider only two augmenting paths at each step of our algorithm. Further, if an optimal matching O_k matches applicant a_k to $f(a_k)$, then O_k is forced to agree with M_k for all applicants that appear on the augmenting path p_k. This reduces the problem of comparing O_k and M_k to comparing $O_k \oplus p_k$ and $M_k \oplus p_k$ and we use our induction hypothesis here to show that $O_k \oplus p_k$ cannot be more optimal than $M_{k-1} = M_k \oplus p_k$.

The difficulty in case of ties, is precisely in the part of extending M_{k-1} to M_k. We leave it as an open question to extend our combinatorial algorithm to the case of ties in preference lists.

References

1. D.J. Abraham, R.W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. SIAM Journal on Computing, Vol.37, No.4, pp. 1030–1045, 2007. (Preliminary version in *Proc. of 16th SODA*, pages 424-432, 2005.)
2. D. J. Abraham and T. Kavitha. Dynamic matching markets and Voting paths. In *Proceedings of SWAT 2008, the 10th Scandinavian Workshop on Algorithm Theory*, (LNCS 4059), pages 65-76, Springer-Verlag, 2006.
3. P. Gardenfors. *Match making: assignments based on bilateral preferences.* Behavioural Sciences, 20:166–173, 1975.
4. D. Gusfield and R.W. Irving. The Stable Marriage Problem: Structure and Algorithms MIT Press, 1989.
5. R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. *Rank-maximal matchings.* In Proceedings of SODA '04: the 15th ACM-SIAM Symposium on Discrete Algorithms, pages 68–75. ACM-SIAM, 2004.
6. Robert W. Irving, Paul Leather, and Dan Gusfield. *An Efficient Algorithm for the "Optimal" Stable Marriage.* Journal of the ACM, 34(3): 532-543, 1987.
7. D.F. Manlove and C. Sng. Popular matchings in the capacitated house allocation problem. In *Proceedings of ESA 2006, the 14th Annual European Symposium on Algorithms*, (LNCS 4168), pages 492-503, Springer-Verlag, 2006.
8. J. Mestre. *Weighted popular matchings.* In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, (LNCS 4051), pages 715–726, 2006.

The Hospitals/Residents Problem with Quota Lower Bounds

Koki Hamada[1], Kazuo Iwama[1], and Shuichi Miyazaki[2]
[1]Graduate School of Informatics, Kyoto University
[2]Academic Center for Computing and Media Studies, Kyoto University
{khamada, iwama}@kuis.kyoto-u.ac.jp, shuichi@media.kyoto-u.ac.jp

Abstract

The Hospitals/Residents problem (HR for short) is a many-to-one extension of the stable marriage problem. In an instance of HR, each hospital specifies a quota, i.e., an upper bound on the number of positions it provides, and a feasible matching must satisfy the condition that the number of residents assigned to each hospital is up to its quota. It is well-known that in any instance, there exists at least one stable matching, and finding one can be done in polynomial time. In this paper, we consider an extension where each hospital specifies upper and *lower* bounds on its number of positions, namely, in a feasible matching, the number of residents assigned to each hospital is at most its upper bound quota and at least its lower bound quota. Now, some instance admits no stable matching, but it is easy to see that the problem of asking if there is a stable matching is solved in polynomial time. We consider an optimization version of this problem, that is, the problem of finding a feasible solution with minimum number of blocking pairs. We show that it is hard to approximate within $(|H| + |R|)^{1-\varepsilon}$ for any positive constant ε, where H and R are the sets of hospitals and residents, respectively. This inapproximability result holds even if all preference lists are complete and strict, and all hospitals have the same preference list (known as the "master list"). We further consider the restriction that, in addition to the above, all residents have the same preference list, and show that this can be solved in polynomial time.

1 Introduction

The *stable marriage problem* is a widely known problem first studied by Gale and Shapley [5]. We are given sets of men and women, and each person's preference list that orders the members of the other sex according to his/her preference. The question is to find a *stable matching*, that is, a matching containing no pair of man and woman who prefer each other to their partners. Such a pair is called a *blocking pair*. They proved that any instance admits at least one stable matching, and proposed an algorithm to find one, known as the Gale-Shapley algorithm.

In the same paper [5], they also proposed a many-to-one extension of the stable marriage problem, which is currently known as the *Hospitals/Residents problem* (*HR* for short). In HR, two sets corresponding to men and women are residents and hospitals, respectively. Each hospital specifies its *quota*, which means that it can accept at most this number of residents. Hence in a feasible matching, the number of residents assigned to each hospital is up to its quota. Generally, in HR, preference lists need not be complete, namely, each resident orders a subset of hospitals, and each hospital orders a subset of residents. A blocking pair is now defined as a pair of hospital h and resident r that include each other in their preference lists such that (i) the number of residents assigned to h is less than its quota or h prefers r to at least one resident assigned to h and (ii) r is unassigned or prefers h to his assigned hospital. Most properties of the stable marriage problem applies to HR, e.g., any instance admits a stable matching, and one can be found by the appropriately modified Gale-Shapley algorithm.

As the name of HR suggests, it has many real-world applications in assignment systems between residents and hospitals based on their preferences, which is known as NRMP in the U.S. [7, 15], CaRMS in Canada [4], SPA (which is now called SFAS) in Scotland [8, 9], and JRMP in Japan [12]. It is also used in assigning students to schools in Singapore [18]. However, since there are special requirements in reality, some useful extensions were proposed. For example, in the NRMP in the U.S., a non-neglectable number of married couples apply for the matching who want to be matched to hospitals within a close distance one another. This extension was formally modeled [17], and an NP-completeness result [16] and some properties [3] were obtained. Another example appears in assigning students to projects in a university. Each lecturer provides one or more projects, each of which has its quota. Also, each lecturer has his/her own quota, which means the maximum number of students he/she can accept. An assignment is a mapping from students to projects so that quotas of projects and lecturers are all satisfied. Abraham et al. modeled this problem as the Student-Project Allocation problem and gave efficient algorithms to find a stable matching [2].

Our Contribution. In this paper, we propose another extension of HR where each hospital declares not only an upper bound but also a lower bound on the number of residents it accepts. Consequently, a feasible matching must satisfy the condition that the number of residents assigned to each hospital is at most its upper bound quota and at least its lower bound quota. This restriction seems quite relevant in several situations. For example, many hospitals have some sort of expectations in each year that at least this number of residents are coming, who are an important labor force for the hospital. Also, many departments of Japanese universities, especially in engineering schools, where students are again considered to be an important labor force for experiments, the number of students assigned to each professor should be somehow balanced. In this setting, we require to find a stable matching, where the stability definition is the same as that of HR. We call this problem *HR with Minimum Quota* (*HRMQ* for short). Note that if we allow incomplete lists, there can be instances with no feasible solution even if we ignore the stability condition. So, we consider only complete lists. Also, for the same reason, we assume that the number of residents is at least the sum of the lower bound quotas of all hospitals.

First of all, it is easy to see that the existence of a stable matching can be easily determined since in HR the number of students each hospital receives is the same in any stable matching [6]. Namely, if those numbers satisfy upper and lower bound conditions, it is a desirable matching, and otherwise, no stable matching exists. When there is no stable matching, we want to find a matching "as stable as possible", i.e., we consider the problem of finding a matching that minimizes the number of blocking pairs. Our first result is that this problem is NP-hard and cannot be approximated within the ratio of $(|H| + |R|)^{1-\varepsilon}$ for any positive constant ε unless P=NP, where H and R denote the sets of hospitals and residents, respectively. This is almost tight because we can show that the problem is approximable within $|H|+|R|$. Further, this inapproximability result holds even if all preference lists are complete and strict, all hospitals have the same preference list (which is known as the *master list* [10]), and all hospitals have upper bound quota 1 (which means that this result holds for the stable marriage case). We then consider stronger restriction that in both of hospitals and residents sides, all preference lists are identical. This restriction seems to be too strict. Nevertheless, we show that it is non-trivial by showing an instance whose optimal solution is against our intuition. We then give a polynomial-time algorithm for this problem.

Related Results. Classical results on the Hospitals/Residents problem (HR) can be found in [7]. Research on HR is still active recently [13, 11]. As mentioned previously, there are some extensions of HR, namely, HR with couples [17, 16, 3] and the Student-Project Allocation problem [2]. The concept of master lists can be found in [10].

There are a couple of problems that involve finding a matching with minimum number of blocking pairs. Khuller et al. [14] introduced an online version of the stable marriage problem, and proved that there is no competitive online algorithm. Abraham et al. [1] proposed the problem of finding a matching with the fewest number of blocking pairs, and showed that it is hard to approximate.

2 Preliminaries

2.1 The Hospitals/Residents Problem with Minimum Quota

An instance of the *Minimum Blocking Pair Hospitals/Residents Problem with Minimum Quota* (*Min-BP HRMQ* for short) consists of the set of residents R, the set of hospitals H, and each member's preference list that orders all members of the other party. Also, each hospital h has lower and upper bounds of quota, p and q (such that $p \leq q$), respectively. We sometimes say that the quota of h is $[p, q]$. For simplicity, we sometimes write the name of hospital with quota bounds, such as $h[p, q]$.

A *matching* is a mapping from residents to hospitals so that the number of residents assigned to each hospital $h[p, q]$ is between p and q. Let $M(r)$ be the hospital to which resident r is assigned under M (if it exists), and $M(h)$ be the set of residents assigned to hospital h. For a matching M and a hospital $h[p, q]$, if $|M(h)| = q$ (including the case that $p = q = |M(h)|$), we say that h is *full* under M, and if $|M(h)| < q$, we say that h is *under-subscribed* under M. We further partition under-subscribed hospitals. If h is under-subscribed and $|M(h)| = p$, we say that h is *critical*. Otherwise, namely if $p < |M(h)| < q$, we say that h is *intermediate*.

For a matching M, we say that a pair comprising a resident r and a hospital h forms a *blocking pair* if the following two conditions are met: (i) r is either unassigned or prefers h to $M(r)$. (ii) h is under-subscribed or prefers r to one of residents in $M(h)$. Min-BP HRMQ is the problem of finding a matching with the minimum number of blocking pairs.

Min-BP 1ML-HRMQ (Min BP 1 Master List-HRMQ) is the restriction of Min-BP HRMQ so that in a given instance, the preference lists of all hospitals are identical. *Min-BP 2ML-HRMQ* is defined similarly, namely, in an instance, all hospitals have the same preference list, and all residents have the same preference list.

2.2 Approximation Ratios

We say that an algorithm A is an $r(n)$-approximation algorithm if it satisfies $\max\{A(x)/opt(x)\}$ over all instances x of size n, where $opt(x)$ and $A(x)$ are the size of the optimal and the algorithm's solution, respectively.

3 Inapproximability Result for Min-BP 1ML-HRMQ

In this section, we show that Min-BP 1ML-HRMQ is NP-hard, and even hard to approximate.

Theorem 3.1 *For any positive constant ε, there is no polynomial-time $(|H|+|R|)^{1-\varepsilon}$-approximation algorithm for Min-BP 1ML-HRMQ unless P=NP.*

Proof. We show a gap introducing reduction from a well-known NP-complete problem Minimum Vertex Cover (VC for short).

Instance: Graph $G = (V, E)$ and a positive integer K.

Question: Does G contain a vertex cover of size at most K? (A vertex cover is a subset $V_c \subseteq V$ such that for every edge $e \in E$, at least one endpoint of e is in V_c).

Now, let $I_0 = (G_0, K_0)$ be an instance of VC, where $G_0 = (V_0, E_0)$ and K_0 is a positive integer. Define $n = |V_0|$. Without loess of generality, we may assume that $K_0 \leq n$. For a constant ε, define $c = \lceil \frac{8}{\varepsilon} \rceil$, $B_1 = n^c$ and $B_2 = n^c - |E_0|$.

We then construct the instance I of Min-BP 1ML-HRMQ from I_0. The set of residents is $R = C \cup F \cup S$, and the set of hospitals is $H = V \cup T \cup X$. Each set is defined as follows:

$$
\begin{aligned}
C &= \{c_i | 1 \leq i \leq K_0\} \\
F &= \{f_i | 1 \leq i \leq n - K_0\} \\
S^{i,j} &= \{s^{i,j}_{0,a} | 1 \leq a \leq B_2\} \cup \{s^{i,j}_{1,a} | 1 \leq a \leq B_2\} \\
S &= \bigcup_{(v_i,v_j) \in E_0, i<j} S^{i,j} \\
V &= \{v_i | 1 \leq i \leq n\} \\
T^{i,j} &= \{t^{i,j}_{0,a} | 1 \leq a \leq B_2\} \cup \{t^{i,j}_{1,a} | 1 \leq a \leq B_2\} \\
T &= \bigcup_{(v_i,v_j) \in E_0, i<j} T^{i,j} \\
X &= \{x_i | 1 \leq i \leq B_1\}
\end{aligned}
$$

Each hospital in X has a quota bound [0,1], and other hospitals have a quota bound [1,1]. Note that $|C|+|F| = |V|$ and $|S| = |T| (= 2|E_0|B_2)$. Since any hospital in $V \cup T$ has a quota bound [1,1], any matching is a one-to-one matching between R and $V \cup T$, and every hospital in X must be unassigned. Intuitively, each member of V corresponds to each vertex in V_0. If $v \in V_0$ is selected in a vertex cover, the corresponding hospital $v' \in V$ is matched with a member in C, otherwise, v' is matched with a member in F. Note that $|H| = n + 2|E_0|B_2 + B_1$ and $|R| = n + 2|E_0|B_2$; hence $|H| + |R| = 2n + 4|E_0|B_2 + B_1 = 2n - 4|E_0|^2 + (4|E_0| + 1)n^c < n^2 + 4n^{c+2} + n^c \leq 6n^{c+2}$, which is polynomial in n.

Next, we construct preference lists. Fig. 1 shows preference lists of residents, where [[V]] (resp. [[X]]) denotes a total order of elements in V (resp. X) in an increasing order of indices. "$\cdots$" denotes a total order of other hospitals in an arbitrary order.

Preference lists of hospitals are identical and are obtained from a master list given in Fig. 2. Here, [[C]] and [[F]] are as before a total order of all hospitals in C and F, respectively, in an increasing order of indices. [[S]] is a total order of [[$S^{i,j}$]] ($(v_i, v_j) \in E_0, i < j$) in any order, where

$$[[S^{i,j}]] = s^{i,j}_{1,1} \ s^{i,j}_{0,1} \ s^{i,j}_{0,2} \ \cdots \ s^{i,j}_{0,B_2} \ s^{i,j}_{1,2} \ \cdots \ s^{i,j}_{1,B_2}.$$

Now the reduction is completed. Before showing the correctness proof, we will see some properties of the reduced instance. For a resident r and a hospital h, if h appears to the right of [[X]]-part of r's list, we call (r, h) a *prohibited pair*.

$$
\begin{array}{llllllll}
c_i & : & [[V]] & [[X]] & \cdots & & & (1 \le i \le K_0) \\
f_i & : & [[V]] & [[X]] & \cdots & & & (1 \le i \le n - K_0) \\
s^{i,j}_{1,1} & : & t^{i,j}_{0,2} & v_j & t^{i,j}_{1,2} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j) \\
s^{i,j}_{0,1} & : & t^{i,j}_{0,1} & v_i & t^{i,j}_{1,1} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j) \\
s^{i,j}_{0,2} & : & t^{i,j}_{0,2} & v_i & t^{i,j}_{0,3} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j) \\
 & & \vdots & & & & & \\
s^{i,j}_{0,B_2-1} & : & t^{i,j}_{0,B_2-1} & v_i & t^{i,j}_{0,B_2} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j) \\
s^{i,j}_{0,B_2} & : & t^{i,j}_{0,B_2} & v_i & t^{i,j}_{0,1} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j) \\
s^{i,j}_{1,2} & : & t^{i,j}_{1,2} & v_j & t^{i,j}_{1,3} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j) \\
 & & \vdots & & & & & \\
s^{i,j}_{1,B_2-1} & : & t^{i,j}_{1,B_2-1} & v_j & t^{i,j}_{1,B_2} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j) \\
s^{i,j}_{1,B_2} & : & t^{i,j}_{1,B_2} & v_j & t^{i,j}_{1,1} & [[X]] & \cdots & ((v_i, v_j) \in E_0, i < j)
\end{array}
$$

Figure 1: Preference lists of residents

$$\text{ML} : \quad [[C]] \quad [[S]] \quad [[F]]$$

Figure 2: Master preference list of hospitals

Lemma 3.2 *If a matching M contains a prohibited pair, then the number of blocking pairs in M is at least B_1.*

Proof. Suppose that a matching M contains a prohibited pair (r, h). By the definition of prohibited pairs, r prefers any hospital $x \in X$ to h. On the other hand, recall that any hospital $x \in X$ is unassigned in any feasible matching, and hence, under-subscribed. So, (r, x) is a blocking pair for every $x \in X$. Since $|X| = B_1$, the proof is completed. $\square$

Now, recall that for each edge $(v_i, v_j) \in E_0$ $(i < j)$, there are the set of residents $S^{i,j}$ and the set of hospitals $T^{i,j}$. We call this pair of sets a (v_i, v_j)-*gadget*, and write it as $g_{i,j} = (S^{i,j}, T^{i,j})$. For each gadget $g_{i,j}$, let us define two perfect matchings between $S^{i,j}$ and $T^{i,j}$ as follows:

$$
\begin{aligned}
M^0_{i,j} = \{ \ & (s^{i,j}_{0,1}, t^{i,j}_{0,1}), (s^{i,j}_{0,2}, t^{i,j}_{0,2}), \ldots, (s^{i,j}_{0,a}, t^{i,j}_{0,a}), \ldots, (s^{i,j}_{0,B_2-1}, t^{i,j}_{0,B_2-1}), (s^{i,j}_{0,B_2}, t^{i,j}_{0,B_2}), (s^{i,j}_{1,1}, t^{i,j}_{1,2}), \\
& (s^{i,j}_{1,2}, t^{i,j}_{1,3}), \ldots, (s^{i,j}_{1,a}, t^{i,j}_{1,a+1}), \ldots, (s^{i,j}_{1,B_2-1}, t^{i,j}_{1,B_2}), (s^{i,j}_{1,B_2}, t^{i,j}_{1,1}) \ \} \\
M^1_{i,j} = \{ \ & (s^{i,j}_{0,1}, t^{i,j}_{1,1}), (s^{i,j}_{0,2}, t^{i,j}_{0,3}), \ldots, (s^{i,j}_{0,a}, t^{i,j}_{0,a+1}), \ldots, (s^{i,j}_{0,B_2-1}, t^{i,j}_{0,B_2}), (s^{i,j}_{0,B_2}, t^{i,j}_{0,1}), (s^{i,j}_{1,1}, t^{i,j}_{0,2}), \\
& (s^{i,j}_{1,2}, t^{i,j}_{1,2}), \ldots, (s^{i,j}_{1,a}, t^{i,j}_{1,a}), \ldots, (s^{i,j}_{1,B_2-1}, t^{i,j}_{1,B_2-1}), (s^{i,j}_{1,B_2}, t^{i,j}_{1,B_2}) \ \}
\end{aligned}
$$

Fig. 3 shows $M^0_{i,j}$ and $M^1_{i,j}$ on preference lists of $S^{i,j}$.

Lemma 3.3 *For a gadget $g_{i,j} = (S^{i,j}, T^{i,j})$, $M^0_{i,j}$ and $M^1_{i,j}$ are the only perfect matchings between $S^{i,j}$ and $T^{i,j}$ that do not include a prohibited pair. Furthermore, each of $M^0_{i,j}$ and $M^1_{i,j}$ contains only one blocking pair (r, h) such that $r \in S^{i,j}$ and $h \in T^{i,j}$. (Hereafter, we simply state this as a "blocking pair between $S^{i,j}$ and $T^{i,j}$".)*

Proof. Construct a bipartite graph $G_{i,j}$, where each vertex set is $S^{i,j}$ and $T^{i,j}$, and there is an edge between $r (\in S^{i,j})$ and $h (\in T^{i,j})$ if and only if (r, h) is not a prohibited pair. One can see that

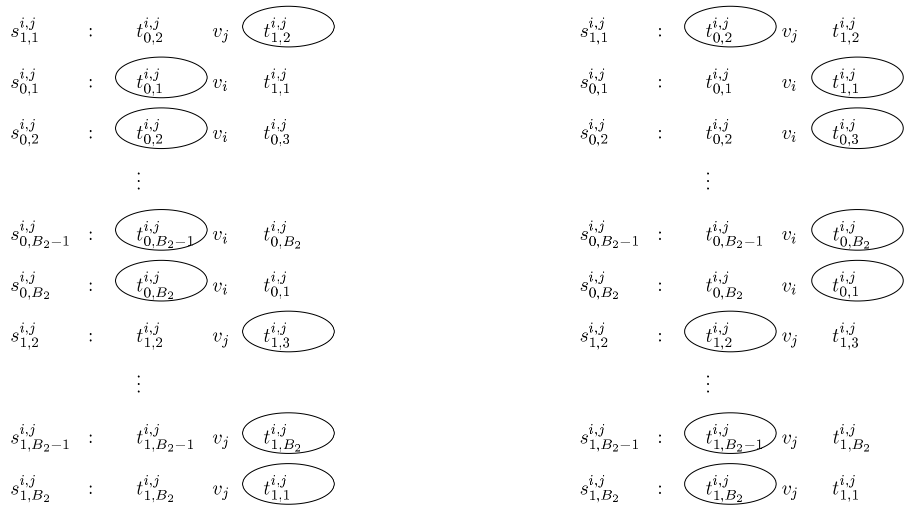


Figure 3: Matchings $M^0_{i,j}$ (left) and $M^1_{i,j}$ (right)

$G_{i,j}$ is a cycle of length $4B_2$. Hence there are only two perfect matchings between $S^{i,j}$ and $T^{i,j}$, and they are actually $M^0_{i,j}$ and $M^1_{i,j}$. Also, it is easy to check that $M^0_{i,j}$ contains only one blocking pair $(s^{i,j}_{1,1}, t^{i,j}_{0,2})$, and $M^1_{i,j}$ contains only one blocking pair $(s^{i,j}_{0,1}, t^{i,j}_{0,1})$. $\square$

We are now ready to show the gap.

Lemma 3.4 *If I_0 is an "yes" instance of VC, then I has a solution with at most $n^2 + |E_0|$ blocking pairs.*

Proof. Suppose that G_0 has a vertex cover of size at most K_0. If its size is less than K_0, add arbitrary vertices to make the size exactly K_0, which is, of course, still a vertex cover. Let this vertex cover be $V_{0c}(\subseteq V_0)$, and let $V_{0f} = V_0 \setminus V_{0c}$. For convenience, we use V_{0c} and V_{0f} also for the sets of corresponding hospitals.

We construct a matching M of I according to V_{0c}. First, match each resident in C with each hospital in V_{0c}, and each resident in F with each hospital in V_{0f}, in an arbitrary way. Since $|C \cup F| = |V| = n$, there are at most n^2 blocking pairs between $C \cup F$ and V.

For each gadget $g_{i,j} = (S^{i,j}, T^{i,j})$ $((v_i, v_j) \in E_0, i < j)$, we use one of two matchings in Lemma 3.3. Since V_{0c} is a vertex cover, either v_i or v_j is included in V_{0c}. If v_i is in V_{0c}, use $M^1_{i,j}$, otherwise, use $M^0_{i,j}$. It is then easy to see that there is no blocking pair between $S^{i,j}$ and $H \setminus T^{i,j}$ or $R \setminus S^{i,j}$ and $T^{i,j}$. Also, as proved in Lemma 3.3, there is only one blocking pair between $S^{i,j}$ and $T^{i,j}$ in either case.

So, the number of blocking pairs is at most n^2 between $C \cup F$ and V, and exactly $|E_0|$ within each gadget, and hence $n^2 + |E_0|$ in total, which completes the proof. $\square$

Lemma 3.5 *If I_0 is a "no" instance of VC, then any solution of I has at least B_1 blocking pairs.*

Proof. Suppose that I admits a matching M with less than B_1 blocking pairs. We show that I_0 has a vertex cover of size K_0.

First of all, recall that any matching must be a one-to-one matching between R and $V \cup T$. Also, by Lemma 3.2, if M contains a prohibited pair, there are at least B_1 blocking pairs, contradicting the assumption. So, M does not contain a prohibited pair. Since $|C \cup F| = |V|$ and any resident $r \in C \cup F$ includes only V to the left of [[X]]-part in the preference list, M must include a perfect matching between $C \cup F$ and V.

Then, consider a gadget $g_{i,j} = (S^{i,j}, T^{i,j})$ and observe the preference lists of $S^{i,j}$. Since v_i and v_j are matched with residents in $C \cup F$, for M to contain no prohibited pairs, all residents in $S^{i,j}$ must be matched with hospitals in $T^{i,j}$. As we have seen before, there are only two possibilities, namely, $M^0_{i,j}$ and $M^1_{i,j}$. Here, note that by Lemma 3.3, each gadget creates one blocking pair, and hence there are $|E_0|$ blocking pairs in total.

Suppose that the matching between $S^{i,j}$ and $T^{i,j}$ is $M^0_{i,j}$. Then, if the hospital v_j is matched with a resident in F, there are B_2 blocking pairs between v_j and $s^{i,j}_{1,1}, \ldots, s^{i,j}_{1,B_2}$. Then, we have $|E_0| + B_2 = B_1$ blocking pairs, contradicting the assumption. So, v_j must be matched with a resident in C. On the other hand, suppose that the matching for $g_{i,j}$ is $M^1_{i,j}$. If the hospital v_i is matched with a resident in F, again there are B_2 blocking pairs, between v_i and $s^{i,j}_{0,1}, \ldots, s^{i,j}_{0,B_2}$. So, v_i must be matched with a resident in C. Namely, for each edge (v_i, v_j), either v_i or v_j is matched with a resident in C. Note that this happens for an arbitrary edge. Hence, the collection of vertices whose corresponding hospitals are matched with residents in C is a vertex cover of size K_0. This completes the proof. □

Finally, we estimate the gap obtained by Lemmas 3.4 and 3.5. As observed previously, $|H| + |R| \leq 6n^{c+2}$. Hence, $B_1/(n^2 + |E_0|) \geq n^c/2n^2 = 8n^{c+2}2^{-4}n^{-4} \geq 8n^{c+2}n^{-8} > (|H| + |R|)^{1-\frac{8}{c}} \geq (|H| + |R|)^{1-\varepsilon}$. □

Remark 1. Instances obtained in the above proof are too artificial since the number of hospitals with quota bound $[1, 1]$ is the same as the number of residents. Hence all hospitals with quota bound $[0, 1]$ are empty despite having an upper quota bound of 1. However, by modifying the reduction, we can show that the inapproximability result holds for natural instances, for example, where all hospitals have the same quota bound $[p, q]$ and the number of residents is around the middle of the upper and lower bounds, namely, nearly $\frac{1}{2}(p+q)|H|$. The modification is much more involved and hence omitted in this paper.

Remark 2. The above inapproximability result is almost tight. The following simple algorithm achieves an approximation ratio of $|H|+|R|$ (not only for Min-BP 1ML-HRMQ but also for general Min-BP HRMQ). Given an instance I of Min-BP HRMQ, consider it as an instance of HR by ignoring quota lower bounds. Then, apply the resident-oriented Gale-Shapley algorithm to I. In the resulting matching, let k be the number of deficiencies, i.e., the sum of $\max\{p_i - x_i, 0\}$ over all hospitals $h_i[p_i, q_i]$, where x_i is the number of residents assigned to h_i by the Gale-Shapley algorithm. We then move k residents from hospitals with surplus to hospitals with deficiencies in an arbitrary way, to fill all the deficiencies.

We can prove that the above procedure creates at most $|H| + |R|$ blocking pairs per resident movement, and hence there are at most $k(|H| + |R|)$ blocking pairs in the resulting solution. On the other hand, we can also prove that if there are k deficiencies in the matching obtained by the Gale-Shapley algorithm, an optimal solution contains at least k blocking pairs. These observations give $(|H| + |R|)$-approximation upper bound.

4 Polynomial-time Solvability of Min-BP 2ML-HRMQ

In this section, we consider Min-BP HRMQ where both residents and hospitals have master lists. One may think that this problem is trivial, since intuitively, it seems reasonable to assign residents with higher priority to hospitals with higher priority as much as possible. However, as shown in the following example, this is not the case. Consider the following instance I consisting of four residents 1, 2, 3, and 4, and four hospitals $a[0,2]$, $b[1,2]$, $c[1,1]$, $d[1,1]$.

1:	a	b	c	d	$a[0,2]$:	1	2	3	4
2:	a	b	c	d	$b[1,2]$:	1	2	3	4
3:	a	b	c	d	$c[1,1]$:	1	2	3	4
4:	a	b	c	d	$d[1,1]$:	1	2	3	4

It is necessary to assign one resident to each of b, c, and d. Then, since c and d are full, the remaining one must be assigned to a or b. Since both 1 and a are most preferred, the matching $\{(1,a),(2,b),(3,c),(4,d)\}$ seems good, which creates five blocking pairs $(2,a),(3,a),(4,a),(3,b)$ and $(4,b)$. However, the matching $\{(1,b),(2,b),(3,c),(4,d)\}$ is optimal which creates four blocking pairs $(1,a),(2,a),(3,a)$ and $(4,a)$. In the optimal matching, the most preferred hospital results in empty even though there is a room for assigning a resident without breaking quota restriction.

Now, we show a polynomial-time algorithm. Let $R = \{r_i | 1 \le i \le |R|\}$ be the set of residents and $H = \{h_i | 1 \le i \le |H|\}$ be the set of hospitals, where a resident (hospital) with a smaller index is higher in the master preference list. If $i < j$, we say that hospital h_i is *higher* than h_j, and h_j is *lower* than h_i. We use the same expression for residents.

To see the property of the problem, we consider the following restriction. For each hospital $h_i[p_i, q_i]$, restrict the number of residents assigned to h_i to x_i where $p_i \le x_i \le q_i$ and $\Sigma_{i=1}^{|H|} x_i = |R|$. We call such predetermined assignment numbers a *number assignment*. Note that, in this restricted problem, the definition of blocking pairs is unchanged, namely, whether a hospital is full or under-subscribed is determined in terms of $[p_i, q_i]$.

Consider the following simple greedy algorithm (call Greedy) for this restricted problem. It matches each resident in turn from r_1 to $r_{|R|}$. For each resident r, match r to the highest hospital which still has a position to accept.

Lemma 4.1 Greedy *produces a matching with minimum number of blocking pairs.*

Proof. Since the number assignment is predetermined, whether a hospital is full or not does not depend on a given matching. Now, for a hospital h_i, let a_i be the number of residents assigned to one of the hospitals lower than h_i. For any matching, the following statements are true: If h_i is under-subscribed (namely, $x_i < q_i$), the number of blocking pairs containing h_i is a_i (there are a_i residents that prefer h_i to the assigned hospital, and h_i is under-subscribed). If h_i is full, there may be or may not be a blocking pair containing h_i.

However, Greedy creates no blocking pair of the second type. This completes the proof. □

By Lemma 4.1, our task is to find an optimal number assignment. Here is one important corollary obtained from the proof of Lemma 4.1.

Corollary 4.2 *For the above restricted problem, the number of blocking pairs containing a hospital h_i in a matching obtained by* Greedy *is 0 if h_i is full and a_i (the number of residents that are assigned to hospitals lower than h_i) if h_i is under-subscribed.*

The following lemma gives a crucial property of an optimal solution of Min-BP 2ML-HRMQ.

Lemma 4.3 *If an optimal solution of Min-BP 2ML-HRMQ contains an intermediate hospital, then all hospitals higher than it are full.*

Proof. Let a be an intermediate hospital, and let b be an under-subscribed hospital that is higher than a. Move any resident from a to b. Then the following statements (1) through (4) hold: (1) For each hospital h lower than a or higher than b, the number of blocking pairs containing h is unchanged. (2) For each hospital h higher than a and lower than b, the number of blocking pairs containing h does not increase. (3) The number of blocking pairs containing a is unchanged (note that a was already under-subscribed before modification). (4) The number of blocking pairs containing b decreases (since b was under-subscribed before modification, and the number of residents assigned to hospitals lower than b decreases by one). From (1)–(4), the number of blocking pairs decreases, contradicting the optimality. □

The following corollary is immediate from Lemma 4.3.

Corollary 4.4 *An optimal solution for Min-BP 2ML-HRMQ contains at most one hospital that is intermediate. If it contains an intermediate hospital h_i, each hospital higher than h_i is full, and each hospital lower than h_i is either full or critical.*

Now, we concentrate on the problem of finding the number assignment that produces an optimal solution. We solve it using dynamic programming. In the following, we may discuss the number of blocking pairs for number assignments (not for actual matchings). In such a case, it means the number of blocking pairs in a matching obtained from the number assignment by Greedy. For convenience, we write the ith lowest hospital as h_{-i}, namely, $h_{-i} = h_{|H|-i+1}$. This notation is also used for residents.

Meanwhile, assume that an optimal solution does not include an intermediate hospital. (We will later discuss the other case.) For each i and j such that $1 \le i \le |H|$ and $0 \le j \le |R|$, we consider the subproblem of assigning the j lowest residents $(r_{-j}, r_{-(j-1)}, \dots, r_{-1})$ to the i lowest hospitals $(h_{-i}, h_{-(i-1)}, \dots, h_{-1})$, subject to the restriction that each hospital is either full or critical. Define $b[i, j]$ be the *minimum* number of blocking pairs (containing these i hospitals) among all possible number assignments. If there is no such number assignment, let $b[i, j] = \infty$. Then, what we want to compute is $b[|H|, |R|]$.

For computing $b[i, j]$, we show the following property: For a fixed j, let $X_{-i} = \{x_{-i}, x_{-(i-1)}, \dots, x_{-1}\}$ be an optimal number assignment of the j lowest residents to the i lowest hospitals $h_{-i}, h_{-(i-1)}, \dots, h_{-1}$ so that each hospital is either full or critical. Here, "optimal" means that it creates the minimum number of blocking pairs. Then, $X_{-(i-1)} = \{x_{-(i-1)}, \dots, x_{-1}\}$ is an optimal number assignment of $j - x_{-i}$ residents to the $i-1$ lowest hospitals so that each hospital is either full or critical. For, if there is a better number assignment $X'_{-(i-1)} = \{x'_{-(i-1)}, \dots, x'_{-1}\}$, then $X'_{-i} = \{x_{-i}, x'_{-(i-1)}, \dots, x'_{-1}\}$ is a number assignment better than X_{-i}. (Note that by Corollary 4.2, the number of blocking pairs containing h_{-i} is the same for both number assignments X_{-i} and X'_{-i}.)

By the above observation, $b[i, j]$ $(1 \le i \le |H|, 0 \le j \le |R|)$ can be inductively defined as follows: Note that the number assignment for h_{-i} is either full or critical, namely, q_{-i} or p_{-i}. If h_{-i} is full, there is no blocking pair containing h_{-i}, and hence the minimum possible number is $b[i-1, j-q_{-i}]$. When h_{-i} is critical, the number of blocking pairs containing h_{-i} is $j-p_{-i}$, and hence the minimum possible number is $b[i-1, j-p_{-i}] + j - p_{-i}$. Hence, $b[i, j] = \min(b[i-1, j-q_{-i}], b[i-1, j-p_{-i}] + j - p_{-i})$.

Since $b[0,0] = 0$ and $b[0,j] = \infty$ $(j \neq 0)$, $b[i,j]$ can be defined in the following formula:

$$b[i,j] = \begin{cases} 0 & (i = 0, j = 0) \\ \infty & (i = 0, j \neq 0) \\ \min(b[i-1, j-q_{-i}], b[i-1, j-p_{-i}] + j - p_{-i}) & (1 \leq i \leq |H|, 0 \leq j \leq |R|) \end{cases}$$

$b[i,j]$ can be computed by the following Procedure 1. It also computes $s[i,j]$, which stores the number assignment to h_{-i} when j residents are assigned to h_{-i} and lower hospitals, so that we can trace back to compute the actual optimal number assignment. The 7th line is for the case that h_{-i} is critical and the 8th line for h_{-i} is full.

Procedure 1 Computing $b[i,j]$ and $s[i,j]$

1: $b[0,0] \leftarrow 0$; $s[0,0] \leftarrow 0$
2: **for** $j = 1$ **to** $|R|$ **do**
3: $b[0,j] \leftarrow \infty$; $s[0,j] \leftarrow 0$
4: **end for**
5: **for** $i = 1$ **to** $|H|$ **do**
6: **for** $j = 0$ **to** $|R|$ **do**
7: **if** $j < p_{-i}$ **then** $z \leftarrow \infty$ **else** $z \leftarrow b[i-1, j-p_{-i}] + (j - p_{-i})$
8: **if** $j < q_{-i}$ **then** $f \leftarrow \infty$ **else** $f \leftarrow b[i-1, j-q_{-i}]$
9: **if** $f \leq z$ **then** $b[i,j] \leftarrow f$; $s[i,j] \leftarrow q_{-i}$ **else** $b[i,j] \leftarrow z$; $s[i,j] \leftarrow p_{-i}$
10: **end for**
11: **end for**

Procedure 1 computes each cell of tables $b[i,j]$ and $s[i,j]$ $(0 \leq i \leq H, 0 \leq j \leq R)$, each of which is of size $O(|H||R|)$. Since each cell can be computed in time $O(1)$, the running time of Procedure 1 is $O(|H||R|)$.

Using $s[i,j]$, the following Procedure 2 computes the number assignment $\{x_{-i}, \ldots, x_{-1}\}$ to the i lowest hospitals that realizes $b[i,j]$. The running time of Procedure 2 is $O(i)$.

Procedure 2 recover(i,j)

return **if** $i \leq 0$
$x_{-i} \leftarrow s[i,j]$
recover($i-1$,$j - x_{-i}$)

By Corollary 4.4, an optimal solution contains at most one intermediate hospital. If an optimal solution does not contain intermediate hospital, then the optimal cost is $b[|H|, |R|]$ by the discussion so far. If an optimal solution contains an intermediate hospital h_i, all hospitals higher than h_i are full, by Lemma 4.3. Hence, by Corollary 4.2, none of these hospitals is included in a blocking pair. If the number of residents assigned to h_i is m $(p_i < m < q_i)$, then the number of blocking pairs that include h_i is $|R| - \sum_{\ell=1}^{i-1} q_\ell - m$. The minimum possible number of blocking pairs including hospitals lower than h_i is $b[|H| - i, |R| - \sum_{\ell=1}^{i-1} q_\ell - m]$. Hence, it suffices to compute $|R| - \sum_{\ell=1}^{i-1} q_\ell - m + b[|H| - i, |R| - \sum_{\ell=1}^{i-1} q_\ell - m]$ for all possible i and m, and take the minimum of these values and $b[|H|, |R|]$. Hence the running time of Procedure 3 is $O(|H||R|)$. In Procedure 3, the variable u stores the index of the hospital which is intermediate in an optimal solution; if $u = 0$, then there is no intermediate hospital. a is used for the number of residents assigned to h_u in an optimal solution when $u \neq 0$.

The final part goes as follows: If $u = 0$, then there is no intermediate hospital. Hence, it can be computed by *recover*$(|H|, |R|)$. Otherwise, i.e., if $u \neq 0$, the hospital h_u is intermediate in an

Procedure 3 Computing the optimal cost

$best \leftarrow b[|H|, |R|]$; $u \leftarrow 0$; $a \leftarrow 0$
$r \leftarrow |R|$
for $i = 1$ **to** $|H|$ **do**
  for $m = p_i + 1$ **to** $\min(r, q_i - 1)$ **do**
    $cost \leftarrow b[|H| - i, r - m] + (r - m)$
    if $best > cost$ **then** $best \leftarrow cost$; $u \leftarrow i$; $a \leftarrow m$
  end for
  $r \leftarrow r - q_i$
  break if $r \leq 0$
end for

optimal solution. Thus, an optimal number assignment can be obtained by making all $h_1, \ldots, h_{u-1}$ full, assigning a to h_u, and executing $recover(|H| - u, |R| - \sum_{\ell=1}^{u-1} q_\ell - a)$. In either case, the computation time is $O(|H|)$.

From the above discussion, we obtain the following Algorithm 1. After the execution of Algorithm 1, x_i $(1 \leq i \leq |H|)$ stores an optimal number assignment. The computation time of Algorithm 1 is $O(|H||R|)$; $O(|H||R|)$ for Procedure 1, $O(|H||R|)$ for Procedure 3, and $O(|H|)$ for the 3rd through 11th lines of Algorithm 1.

Algorithm 1 Computing the optimal number assignment

1: Execute Procedure 1.
2: Execute Procedure 3.
3: **if** $u = 0$ **then**
4:   $recover(|H|, |R|)$
5: **else**
6:   **for** $i = 1$ **to** $u - 1$ **do**
7:     $x_i \leftarrow q_i$
8:   **end for**
9:   $x_u \leftarrow a$
10:   $recover(|H| - u, |R| - \sum_{\ell=1}^{u-1} q_\ell - a)$
11: **end if**

As mentioned previously, an optimal assignment can be obtained by executing Greedy to the solution of Algorithm 1. Since Greedy runs in time $O(|H| + |R|)$, we have the following theorem:

Theorem 4.5 *An optimal solution for Min-BP 2ML-HRMQ can be computed in $O(|H||R|)$ time.*

5 Conclusions

In this paper, we have considered an extension of the Hospitals/Residents problem where each hospital has upper and lower bounds on a quota. We have shown that the problem of finding a matching with minimum number of blocking pairs is hard to approximate even if hospitals have a master list, while it can be solved in polynomial time when both residents and hospitals have master lists.

References

[1] D. J. Abraham, P. Biró and D. F. Manlove, " "Almost stable" matchings in the roommates problem," *Proc. WAOA 2005*, LNCS 3879, pp. 1–14, 2005.

[2] D. J. Abraham, R. W. Irving and D. F. Manlove, "Two algorithms for the Student-Project Allocation problem," *J. Discrete Algorithms*, Vol.5, no 1, pp. 73–90, 2007.

[3] B. Aldershof and O. M. Carducci, "Stable matchings with couples," *Discrete Applied Mathematics*, Vol. 68, pp. 203–207, 1996.

[4] Canadian Resident Matching Service (CaRMS), `http://www.carms.ca/`

[5] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, Vol. 69, pp. 9–15, 1962.

[6] D. Gale and M. Sotomayor, "Some remarks on the stable matching problem," *Discrete Applied Mathematics*, Vol. 11, pp. 223–232, 1985.

[7] D. Gusfield and R. Irving, "The Stable Marriage Problem: Structure and Algorithms," MIT Press, Boston, MA, 1989.

[8] R. Irving, "Matching medical students to pairs of hospitals: a new variation on an old theme," *Proc. ESA 98*, LNCS 1461, pp. 381–392, 1998.

[9] R. W. Irving, D. F. Manlove and S. Scott, "The hospital/residents problem with ties," *Proc. SWAT 2000*, LNCS 1851, pp. 259–271, 2000.

[10] R. W. Irving, D. F. Manlove and S. Scott, "The stable marriage problem with master preference lists," Discrete Applied Mathematics, to appear.

[11] R.W. Irving, D.F. Manlove and S. Scott, "Strong stability in the hospitals/residents problem," *Proc. STACS 2003*, LNCS 2607, pp. 439–450, 2003.

[12] Japanese Resident Matching Program (JRMP), `http://www.jrmp.jp/`

[13] T. Kavitha, K. Mehlhorn, D. Michail and K. Paluch, "Strongly stable matchings in time $O(nm)$ and extension to the hospitals-residents problem," *Proc. STACS 2004*, pp. 222–233, 2004.

[14] S. Khuller, S. G. Mitchell and V. V. Vazirani, "On-Line algorithms for weighted bipartite matching and stable marriages," *Theoretical Computer Science*, Vol. 127, No 2, pp. 255–267, 1994.

[15] T. Le, V. Bhushan and C. Amin, "First aid for the match, second edition," McGraw-Hill, Medical Publishing Division, 2001.

[16] E. Ronn,"NP-complete stable matching problems," *J. Algorithms*, Vol. 11, pp. 285–304, 1990.

[17] A. E. Roth, "The evolution of the labor market for medical interns and residents: a case study in game theory", *J. Political Economy*, Vol. 92, No. 6, pp. 991–1016, 1984.

[18] C.P. Teo, J.V. Sethuraman and W.P. Tan, "Gale-Shapley stable marriage problem revisited: Strategic issues and applications," *Proc. IPCO 99*, pp. 429–438, 1999.

Circular Stable Matching and 3-way Kidney Transplant

Chien-Chung Huang

Dartmouth College
villars@cs.dartmouth.edu

Abstract. We consider the following version of the stable matching problem. Suppose that men have preferences for women, women have preferences for dogs, and dogs have preferences for men. The goal is to organize them into family units so that no three of them have incentive to desert their assigned family members to join in a new family. This problem is called circular stable matching, allegedly originated by Knuth. We also investigate a generalized version of this problem, in which every participant has preference among all others. The goal is similarly to partition them into oriented triples so that no three persons have incentive to deviate from the assignment. This problem is motivated by recent innovations in kidney exchange, and we call it the 3-way kidney transplant problem. We report complexity, structural and counting results on these two problems.

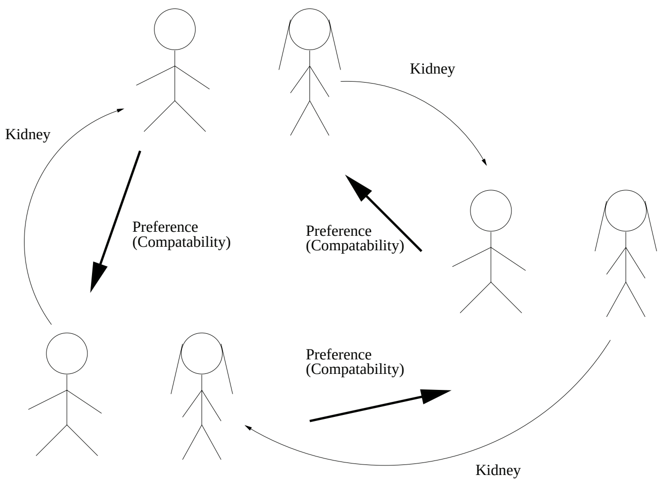
1 Introduction

Stable matching problems were introduced by Gale and Shapley in their seminal paper [5]. Knuth asked whether the stable matching problem can be extended to the case of three parties [15], say we have women, men and dogs. This fairly general problem allows several formulations. One possibility is that every player expresses her/his/its preference among the *combinations* of the other two parties. In this formulation, Ng and Hirschberg [16] proved the existence of stable matchings is NP-complete. Similar NP-completeness results have been shown in [10, 21].

Ng and Hirschberg mentioned that the reviewers of their paper suggested another formulation, and they attributed it to Knuth, for the 3-party stable matchings—the CIRCULAR STABLE MATCHING problem that we will consider in this paper: women have preferences for dogs, dogs have preferences for men, and men have preferences for women. The goal is to organize them into stable family units so that people/dogs have no incentive to desert their assigned family members to join in a new family. This problem can be seen as a natural generalization of the well-known 2-party STABLE MARRIAGE problem and has been investigated in [2, 4].

A generalized version of the CIRCULAR STABLE MATCHING problem allows each participant to express preference among all others. The goal is to partition $3n$ persons into oriented triples so that no three of them have reasons to deviate from the assignment. Again, this problem can be regarded as a generalization of the STABLE ROOMMATES problem [5]. This generalized problem has practical interest in the kidney exchange that has received much attention recently [1, 3, 7, 12, 17, 19, 18, 20]. The "preference" here can be interpreted as degrees of compatibility between recipients and donors. Figure 1 gives a more visual way of seeing the connection between circular matching and kidney exchange. In this paper, we call this problem the 3-WAY KIDNEY TRANSPLANT problem. For ease of presentation, we will refer to all participants in both problems generically as "players."

The two problems require a proper definition of stability. In the two-party STABLE MARRIAGE and STABLE ROOMMATES, a matching is stable if there is no *blocking pair*: two persons who strictly prefer each other to their assigned partners. Naturally, one would extend blocking pairs into *blocking triples* to define the stability of matchings. However, a blocking triple here is more tricky. To see why this is so, consider the following.



[h]

Fig. 1. An illustration of kidney exchange with compatibility as preference.

- In CIRCULAR STABLE MATCHING, suppose that we have a matching $\{(m_1,w_1,d_1),(m_2,w_2,d_2),(m_3,w_3,d_3)\}$. If m_1 prefers w_2 to w_1, w_2 prefers d_3 to d_2, and d_3 prefers m_1 to m_3, then (m_1,w_2,d_3) is clearly a blocking triple. But it may also be the case that w_2 prefers d_1 to d_2. Then (m_1,w_2,d_1) can also be regarded as a (weaker) blocking triple, since only m_1 and w_2 are really better off in such a triple, while d_1 is indifferent.
- In 3-WAY KIDNEY TRANSPLANT, a matching is composed of oriented triples. Here we write such a triple as $\overrightarrow{(k_1,k_2,k_3)}$ to express that k_2,k_3,k_1 are the successors of k_1,k_2,k_3, respectively. Moreover, here k_1 represents a couple (often a married couple) consisting of a person needing a new kidney and a potential kidney donor. If k_2 follows k_1 in a triple, then the donor from the couple k_2 will be passing a kidney to the recipient of k_1. Thus, it is k_1's preference (degree of compatibility) that is at issue. Note that an oriented couple $\overrightarrow{(k_1,k_2,k_3)}$ can be a blocking triple itself $\overrightarrow{(k_1,k_3,k_2)}$, if k_1 prefers k_3 to k_2, k_3 prefers k_2 to k_1, and k_2 prefers k_1 to k_3. Such phenomena may appear somehow surprising for researchers long familiar with stable matching literature.

We allow players to express their indifferences in the form of ties in the preference lists. Now we say a blocking triple is of degree i if i players are strictly better off in such a triple than in a given matching, while the remaining $3-i$ players are indifferent. Note that the indifference can be either because the involved player is still matched to the same partner (or still having the same successor in the oriented triple), or because the involved player has a partner/successor who is tied with her/his/its current assignment. We define a hierarchy of stabilities (which is similar to the one defined by Irving [11] in the 2-party matching) as follows.

- Super Stable Matching: a matching not allowing blocking triples of degree 1 or 2 or 3.
- Strong Stable Matching: a matching not allowing blocking triples of degree 2 nor those of degree 3.
- Weak Stable Matching: a matching not allowing blocking triples of degree 3.

Contributions of the Paper

Complexity: We prove the following existence problems are NP-complete: super/strong stable matchings in CIRCULAR STABLE MATCHING; super/strong/weak stable matchings in 3-WAY KIDNEY TRANSPLANT. Therefore, it is unlikely that we can design efficient algorithms to solve these problems. The complexity of weak stable matchings in CIRCULAR STABLE MATCHING remains open. However, there is empirical evidence indicating that it probably does not belong to the class of NP-complete problems. We shall discuss this issue later.

Structural Results: It is well-known that stable matchings in 2-party stable marriage and stable roommates have rich structures and sophisticated algorithms have been designed to exploit them [8, 15]. It turns out that strong stable matchings in CIRCULAR STABLE MATCHING and 3-WAY KIDNEY TRANSPLANT have parallel (but even richer) structures. Briefly, we show that the set of strong stable matchings in the former problem forms a union of distributive lattices and in the latter problem forms a union of meet-semilattices.

Counting Results: We prove that counting strong stable matchings in both problems is #P-complete. Moreover, the number of strong and weak stable matchings in both problems can be exponential.

Notation and Paper Roadmap In the paper, we use $\mathcal{M}, \mathcal{W}, \mathcal{D}$ to denote the collections of men and women and dogs in CIRCULAR STABLE MATCHING. Whatever the problem instance, we will always assume that they are of the same cardinality. Similarly, $\mathcal{K}$ means the set of players in 3-way kidney transplant. $P(p)$ denotes the preference list of player p. The notation $\succ$ indicates the preference order in the list. The braces denote a tie. For example, $P(m) = \{w_1, w_2\} \succ w_3$ means that man m prefers both w_1 and w_2 to w_3 while he is indifferent between the former two. In general, we use μ to denote a 3-dimensional matching (consisted of triples). We will need to consider the induced two-party matching of μ. For example, we write $\mu|_{\mathcal{M},\mathcal{W}}$ to denote the induced men-women matching by dropping all dogs from the triples of μ. Finally, $\pi_r(X)$ denotes an arbitrary permutation of the members in the set X.

Section 2 presents complexity results; Section 3 reports structural results of stable matching; Section 4 concerns the counting of stable matchings. Finally, Section 5 draws conclusions.

2 NP-completeness of Strong Stable Matchings

The reductions we will present share similar ideas to those used in [10]. The main difference lies in the design of "guard players" (to be explained below).

2.1 Existence Problem of Super Stable Matchings is NP-complete

To prove that the existence of super stable matchings is NP-complete in circular stable matching, we present a reduction from 3-DIMENSIONAL MATCHING, one of Karp's 21 NP-complete problems [14]. The problem instance is given in the form $\Upsilon = (\mathcal{M}, \mathcal{W}, \mathcal{D}, \mathcal{T})$, where $\mathcal{T} \subseteq \mathcal{M} \times \mathcal{W} \times \mathcal{D}$. The goal is to decide whether a perfect matching $\mu \subseteq \mathcal{T}$ exists. This problem remains NP-complete even if every player in $\mathcal{M} \cup \mathcal{W} \cup \mathcal{D}$ appears exactly 2 or 3 times in the triples of $\mathcal{T}$ [6].

We first explain the intuition behind our reduction. Supposing that man m_i appears in three triples $(m_i, w_{ia}, d_{ia}), (m_i, w_{ib}, d_{ib}), (m_i, w_{ic}, d_{ic})$ in $\mathcal{T}$, we create three *dopplegangers*, m_{i1}, m_{i2}, m_{i3} in the derived circular stable matching instance with ties Υ'. We also create four garbage collectors, $w^g_{i1}, d^g_{i1}, w^g_{i2}, d^g_{i2}$. The aim of our design is that in the derived instance Υ', in a super stable matching, exactly one doppleganger will be matched to a woman-dog pair with whom m_i shares a triple in $\mathcal{T}$, while the other two dopplegangers will be paired off with garbage collectors. In the case that there are only two triples in $\mathcal{T}$ containing man m_j, we create only 2 dopplegangers m_{j1}, m_{j2} and two garbage collectors w^g_{j1}, d^g_{j1}. Similarly, the intent is to make sure that in a super stable matching, exactly one doppleganger will be matched to a woman-dog pair with whom m_j shares a triple in $\mathcal{T}$ while the other is matched to the garbage collectors.

Now, we will refer to the set of dopplegangers as $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$, the set of garbage collectors as $\mathcal{W}^g_1, \mathcal{W}^g_2, \mathcal{D}^g_1, \mathcal{D}^g_2$ and the original set of real women and real dogs as $\mathcal{W}, \mathcal{D}$. Collectively, we refer to them as *major players*

$\Sigma = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3 \cup \mathcal{W}_1^g \cup \mathcal{W}_2^g \cup \mathcal{W} \cup \mathcal{D}_1^g \cup \mathcal{D}_2^g \cup \mathcal{D}$ and their preferences are summarized in the left column of Table 1.

To restrict the possible partners of major players in Σ, we introduce a set of gadgets called *guard players*. They are denoted as $m^\star(p), w^\star(p), d^\star(p)$, for $p \in \Sigma$ and their preferences are shown in the right column of Table 1. Their purpose is to ensure that player p, say $p = m_{i1}$, will never get a partner ranking lower than his associated guard player $w^\star(m_{i1})$ in a super stable matching. How guard players and major players interact is captured by the following lemma.

Table 1. The preference lists of all players in the derived instance Υ'. Recall that $\{\}$ denotes a tie in the preferences. Note also that real women $\mathcal{W}$ and real dogs $\mathcal{D}$ only list real dogs and dopplegangers, respectively, with whom they share triples in $\mathcal{T}$, at the top of their lists.

Major Players	Preference Lists	Guard Players	Preference Lists
$m_{i1} \in \mathcal{M}_1$	$\{w_{i1}^g, w_{i2}^g, w_{ia}\} \succ w^\star(m_{i1}) \succ \cdots$	$m^\star(m^\dagger), m^\dagger \in \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3$	$w^\star(m^\dagger) \succ \cdots$
$m_{i2} \in \mathcal{M}_2$	$\{w_{i1}^g, w_{i2}^g, w_{ib}\} \succ w^\star(m_{i2}) \succ \cdots$	$w^\star(m^\dagger), m^\dagger \in \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3$	$d^\star(m^\dagger) \succ \cdots$
$m_{i3} \in \mathcal{M}_3$	$\{w_{i1}^g, w_{i2}^g, w_{ic}\} \succ w^\star(m_{i3}) \succ \cdots$	$d^\star(m^\dagger), m^\dagger \in \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3$	$\{m^\dagger, m^\star(m^\dagger)\} \succ \cdots$
$w \in \mathcal{W}$	$\{d \mid (*, w, d) \in \mathcal{T}\} \succ d^\star(w) \succ \cdots$	$m^\star(w^\dagger), w^\dagger \in \mathcal{W}_1^g \cup \mathcal{W}_2^g \cup \mathcal{W}$	$\{w^\dagger, w^\star(w^\dagger)\} \succ \cdots$
$d \in \mathcal{D}$	$\{m_{ij} \mid (m_i, w, d) \in \mathcal{T}, w \succ_{m_{ij}} w^\star(m_{ij})\} \succ m^\star(d) \succ \cdots$	$w^\star(w^\dagger), w^\dagger \in \mathcal{W}_1^g \cup \mathcal{W}_2^g \cup \mathcal{W}$	$d^\star(w^\dagger) \succ \cdots$
$w_{i1}^g \in \mathcal{W}_1^g$	$d_{i1}^g \succ d^\star(w_{i1}^g) \succ \cdots$	$d^\star(w^\dagger), w^\dagger \in \mathcal{W}_1^g \cup \mathcal{W}_2^g \cup \mathcal{W}$	$m^\star(w^\dagger) \succ \cdots$
$w_{i2}^g \in \mathcal{W}_2^g$	$d_{i2}^g \succ d^\star(w_{i2}^g) \succ \cdots$	$m^\star(d^\dagger), d^\dagger \in \mathcal{D}_1^g \cup \mathcal{D}_2^g \cup \mathcal{D}$	$w^\star(d^\dagger) \succ \cdots$
$d_{i1}^g \in \mathcal{D}_1^g$	$\{m_{i1}, m_{i2}, m_{i3}\} \succ m^\star(d_{i1}^g) \succ \cdots$	$w^\star(d^\dagger), d^\dagger \in \mathcal{D}_1^g \cup \mathcal{D}_2^g \cup \mathcal{D}$	$\{d^\star(d^\dagger), d^\dagger\} \succ \cdots$
$d_{i2}^g \in \mathcal{D}_2^g$	$\{m_{i1}, m_{i2}, m_{i3}\} \succ m^\star(d_{i2}^g) \succ \cdots$	$d^\star(d^\dagger), d^\dagger \in \mathcal{D}_1^g \cup \mathcal{D}_2^g \cup \mathcal{D}$	$m^\star(d^\dagger) \succ \cdots$

Lemma 1. *In the derived instance Υ', if a super stable matching exists, then in such a matching, (1) all major players in Σ will be matched to other major players ranking higher than her/his/its associated guard players, (2) the set of guard players $m^\star(p), w^\star(p), d^\star(p)$, where $p \in \Sigma$ are matched to one another, and (3) the garbage collectors created for a particular man m_i will be matched to one another and the two dopplegangers of m_i (or just one if m_i only appears twice in the triples of the given 3-dimensional matching instance Υ.)*

Proof. Without loss of generality, consider the major player $p = m_{i1}$. In a super stable matching, if m_{i1} is matched to a woman ranking below $w^\star(m_{i1})$, then $(m_{i1}, w^\star(m_{i1}), d^\star(m_{i1}))$ is a blocking triple of degree at least 1, a contradiction. If m_{i1} is matched to $w^\star(m_{i1})$, then $(m^\star(m_{i1}), w^\star(m_{i1}), d^\star(m_{i1}))$ is a blocking triple of degree at least 1, again a contradiction.

For the second part, by the above discussion, we know that all major players must be matched to one another. Hence, if $(m^\star(p), w^\star(p), d^\star(p))$ is not part of a super stable matching, they form a blocking triple of degree at least 1.

The third part follows straightforwardly from the previous two. □

Lemma 2. *The given instance $\Upsilon = (\mathcal{M}, \mathcal{W}, \mathcal{D}, \mathcal{T})$ contains a perfect matching if and only if the derived instance Υ' allows a super stable matching.*

Proof. (Sufficiency) If the derived instance Υ' allows a super stable matching, then by the third part of Lemma 1, it is easy to see that Υ contains a perfect matching.

(Necessity) Suppose that μ is a perfect matching in Υ. We construct a super stable matching μ' for the derived instance Υ' as follows. Assuming that $(m_i, w_x, d_y) \in \mu$, we choose the doppleganger m_{ij} who ranks w_x higher than his guard player $w^\star(m_{ij})$ and make (m_{ij}, w_x, d_y) a triple in μ'. Further, the other two dopplegangers of m_i are matched to (w_{i1}^g, d_{i1}^g) and (w_{i2}^g, d_{i2}^g) respectively. (If there are only two dopplegangers of m_i, then the other doppleganger $m_{ij'} \neq m_{ij}$ is matched to w_{i1}^g, d_{i1}^g). Finally, let the three guard players created for a particular major player be matched to one another. By this construction, it can be verified that we only allow blocking triples of degree 0, which are permissible for a super stable matching. □

Theorem 1. *Deciding whether a super stable matching exists in a circular stable matching problem with ties in the preferences is NP-complete. This is true even if all ties are of size at most 3 and they are at the front of the preference lists.*

To prove the existence of strong stable matching is NP-complete, we can use the same reduction as above with just one alteration: we need a different set of guard players for each major player. Note that in the proof of Lemma 1, we rely on blocking triples of degree 1; those are not counted as blocking triples based on the definition of strong stable matching.

The design of guard players for the reduction of strong stable matching is similar to those used in a reduction in Section 3, so we omit the details here.[1]

2.2 Strong/Super Stability in 3-way Kidney Transplant

We now present a reduction from a circular stable matching instance $\Upsilon = (\mathcal{M}, \mathcal{W}, \mathcal{D}, \mathcal{L})$ (with or without ties in the preferences) to a 3-way kidney transplant instance Υ'. Suppose that $m \in \mathcal{M}, w \in \mathcal{W}, d \in \mathcal{D}$ have preferences $P(m), P(w), P(d)$, respectively. In Υ', their preferences are transformed into

- $P'(m) = P(m) \succ \pi_r(\mathcal{D}) \succ \pi_r(\mathcal{M} - \{m\})$
- $P'(w) = P(w) \succ \pi_r(\mathcal{M}) \succ \pi_r(\mathcal{W} - \{w\})$
- $P'(d) = P(d) \succ \pi_r(\mathcal{W}) \succ \pi_r(\mathcal{D} - \{d\})$

To prove this is a valid reduction, we have to argue that strong/super stable matchings exist in Υ if and only if they exist in Υ'. It is straightforward to show one direction (from Υ to Υ'), but the other direction takes some argument.

Lemma 3. *If a strong/super stable matching μ' exists in Υ', the following holds*

- *Every oriented triple contains exactly one man, one woman, and one dog.*
- *Given a triple $t \in \mu'$, t's orientation must be $t = \overrightarrow{(m, w, d)}$.*

Proof. For the first part, without loss of generality, assume that a triple $t \in \mu'$ contains at least two men. There are three possible cases and all lead to contradiction.

1. Suppose that $t = \overrightarrow{(m, m', m'')}$. Then there exist two triples t' and t'', which contain two women and two dogs, respectively. As a result, a woman $w \in t'$ and a dog $d \in t''$ have as successors a woman, and a dog, respectively. Similarly, there is a man $m \in t$ whose successor is another man. Then $\overrightarrow{(m, w, d)}$ is a blocking triple of degree 3, violating the stability of μ'.
2. Suppose that $t = \overrightarrow{(m, m', w)}$. Then there exists a triple t' containing two dogs. At least one dog $d \in t'$ has another dog as successor. Then $\overrightarrow{(m, w, d)}$ is a blocking triple of degree 3, blocking μ'.
3. Suppose that $t = \overrightarrow{(m, m', d)}$. Then the argument is analogous to the previous case.

For the second part, if $t = \overrightarrow{(m, d, w)} \in \mu'$, then the reverse triple $\overrightarrow{(m, w, d)}$ is a blocking triple of degree 3. □

[1] However, in our reduction, ties are allowed. We leave it open whether the NP-completeness holds when all preferences are strictly-ordered.

By Lemma 3, the following theorem is immediate.

Theorem 2. *It is NP-complete to decide whether a strong/super stable matching exists in the 3-way kidney transplant problem.*

3 Weak Stability in 3-way Kidney Transplant

The reduction we are presenting in this section shares similar basic ideas to those we used in Section 2.1: reduction from a 3-dimensional matching problem instance $\Upsilon = (\mathcal{M}, \mathcal{W}, \mathcal{D}, \mathcal{L})$, creating dopplegangers $\mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3$ and garbage collectors $\mathcal{W}_1^g \cup \mathcal{W}_2^g \cup \mathcal{D}_1^g \cup \mathcal{D}_2^g$, and using sets of guard players to restrict the potential partners (successors in triples) of the major players. The key difference is the design of the guard players' preferences.

We introduce the following gadget for each major player $k \in \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3 \cup \mathcal{W}_1^g \cup \mathcal{W}_2^g \cup \mathcal{D}_1^g \cup \mathcal{D}_2^g$. (Note that real women $\mathcal{W}$ and real dogs $\mathcal{D}$ do not need them.) Let Υ_k be a 3-way kidney transplant instance that has the following three properties: (1) It contains 7 players, $k_i^\#, 1 \leq i \leq 7$, (2) it does not allow any weak stable matching, and (3) if one player, $k_1^\#$, is *removed* from Υ_k, then the remaining 6 players' preferences allow at least one weak stable matching. Such an instance Υ_k can be found in [9]. Our plan is to "embed" instances Υ_k into the intended 3-way kidney transplant instance Υ'.

We now explain in more detail what we mean by embedding of Υ_k into Υ'. For illustration, we first show the preferences of m_{i1} and his six associated guard players in Υ'.

- $P_{\Upsilon'}(m_{i1}) = w_{i2}^g \succ w_{i1}^g \succ w_{ia} \succ L_{\Upsilon_{m_{i1}}}(m_{i1,1}^\#) \succ \cdots$, where $P_{\Upsilon_{m_{i1}}}(m_{i1,1}^\#)$ is the preference list of $m_{i1,1}^\#$ in the instance $\Upsilon_{m_{i1}}$.
- $P_{\Upsilon'}(m_{i1,j}^\#) = L_{\Upsilon_{m_{i1}}}(m_{i1,j}^\#) \succ \cdots$, where $2 \leq j \leq 7$ and $P_{\Upsilon_{m_{i1}}}(m_{i1,j}^\#)$ is the preference list of $m_{i1,j}^\#$ in the instance $\Upsilon_{m_{i1}}$.

In words, as $k = m_{i1}$, we let m_{i1} "play the role" of $k_1^\#(= m_{i1,1}^\#)$. His associated six guard players in $\Upsilon_k(= \Upsilon_{m_{i1}})$ are added into Υ' and, in their new preferences, they still put one another on top of their lists. By this arrangement, if m_{i1} can be matched to some woman ranking higher than his associated guard players, then in this sense, $m_{i1,1}^\#(= m_{i1})$ is removed from the problem instance $\Upsilon_{m_{i1}}$; on the other hand, if he is not, then $\Upsilon_{m_{i1}}$ will engender at least a blocking triple, disrupting the stability of the matching in Υ'.

Lemma 4. *In a weak stable matching μ' in Υ', the successor of m_{i1} ranks at least as high as w_{ia}. Moreover, the six guard players of m_{i1} must be matched to one another.*

Proof. If m_{i1} is matched to someone ranking lower than w_{ia}, then whatever the oriented triples of μ' involving the six guard partners of m_{i1} and m_{i1} himself, the situation is identical to one where we have a matching μ^ϕ for the problem instance $\Upsilon_{m_{i1}}$, which by design, involves at least one blocking triple of degree 3 to block μ^ϕ, and also μ'. The second part of the lemma follows from the first part and the way we chose the gadget $\Upsilon_k(= \Upsilon_{m_{i1}})$. □

The detailed preferences of major players can be found in Table 2. Note that Lemma 4 also applies to other major players who have associated guard players. Thus, in a weak stable matching, they will get a successor ranking strictly higher than their guard players.

Theorem 3. *Deciding whether a weak stable matching exists in a 3-way kidney transplant problem is NP-complete.*

Table 2. The preference lists of major players in the derived problem instance Υ'.

Players	Preference Lists	Players	Preference Lists
$m_{i1} \in \mathcal{M}_1$	$w^g_{i2} \succ w^g_{i1} \succ w_{ia} \succ L_{\Upsilon_{m_{i1}}}(m^{\#}_{i1,1}) \succ \cdots$	$w^g_{i1} \in \mathcal{W}^g_1$	$d^g_{i1} \succ L_{\Upsilon_{w^g_{i1}}}(w^{g,\#}_{i1,1}) \succ \cdots$
$m_{i2} \in \mathcal{M}_2$	$w^g_{i2} \succ w^g_{i1} \succ w_{ib} \succ L_{\Upsilon_{m_{i2}}}(m^{\#}_{i2,1}) \succ \cdots$	$w^g_{i2} \in \mathcal{W}^g_2$	$d^g_{i2} \succ L_{\Upsilon_{w^g_{i2}}}(w^{g,\#}_{i2,1}) \succ \cdots$
$m_{i3} \in \mathcal{M}_1$	$w^g_{i2} \succ w^g_{i1} \succ w_{ic} \succ L_{\Upsilon_{m_{i3}}}(m^{\#}_{i3,1}) \succ \cdots$	$d^g_{i1} \in \mathcal{D}^g_1$	$m_{i1} \succ m_{i2} \succ m_{i3} \succ L_{\Upsilon_{d^g_{i1}}}(d^{g,\#}_{i1,1}) \succ \cdots$
$w \in \mathcal{W}$	$\pi_r(\{d \vert (*,w,d) \in \mathcal{T}\}) \succ \cdots$	$d^g_{i2} \in \mathcal{D}^g_2$	$m_{i1} \succ m_{i2} \succ m_{i3} \succ L_{\Upsilon_{d^g_{i2}}}(d^{g,\#}_{i2,1}) \succ \cdots$
$d \in \mathcal{D}$	$\pi_r(\{m_{ij} \vert m_{ij} \in \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3\}) \succ \cdots$		

Proof. By Lemma 4, if μ' is a weak stable matching in Υ', we can throw away triples involving guard players of Υ', along with the garbage collectors (and the dopplegangers matched to them). Replace the doppleganger m_{ij} with the real man m_i gives the desired perfect matching μ in Υ.

For the other direction, we will construct a weak stable matching μ' in Υ' based on a perfect matching μ in Υ. Suppose that $\overrightarrow{(m_i, w_x, d_y)} \in \mu$. In μ', we insert three triples, $\overrightarrow{(m_{ij}, w_x, d_y)}$, where m_{ij} is the doppleganger of m_i who ranks w_x higher than his guard players, and $\overrightarrow{(m_{ij'}, w^g_{i1}, d^g_{i1})}$ and $\overrightarrow{(m_{ij''}, w^g_{i2}, d^g_{i2})}$. (Or we only add the first two triples, provided that m_i only appears twice in the triples of $\mathcal{T}$.) It can be observed that μ' involves only blocking triples of degree at most 2, which are allowed because of the definition of weak stable matchings. □

4 Structures of Strong Stable Matchings

We first review the definitions of distributive lattices and meet-semilattices.

Definition 1. *Let $(\mathcal{E}, \preceq)$ be a poset. Such a poset is a distributive lattice if it fulfills the following three properties:*

1. *Each pair of elements $a, b \in \mathcal{E}$ has an infinum, called meet, denoted as $a \wedge b \in \mathcal{E}$, such that $a \wedge b \preceq a, a \wedge b \preceq b$, and there is no element $c \in \mathcal{E}$ such that $c \preceq a, c \preceq b$, and $a \wedge b \succ c$.*
2. *Each pair of elements $a, b \in \mathcal{E}$ has a supremum, called join, denoted as $a \vee b \in \mathcal{E}$, such that $a \preceq a \vee b, b \preceq a \vee b$, and there is no element $c \in \mathcal{E}$ such that $a \preceq c, b \preceq c$, and $c \succ a \vee b$.*
3. *Given any three elements, $a, b, c \in \mathcal{E}$, the distributive law holds, i.e., $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$, and $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$.*

A poset $(\mathcal{E}, \preceq)$ is a meet-semilattice if it only fulfills the first property.

Note that in this section, we assume that all preference lists are strictly ordered.

4.1 Strong Stable Matchings in Circular Stable Matching

Our major finding regarding the structure of strong stable matchings in CIRCULAR STABLE MATCHING is that they are a collection of distributive lattices. In particular, consider the subset of strong stable matchings in which all players in one group (men,women, or dogs) have the same partners. Such a subset is a distributive lattice. The following theorem gives a more precise statement.

Theorem 4. *Let $\Upsilon = (\mathcal{M}, \mathcal{W}, \mathcal{D}, \mathcal{P})$ be a circular stable matching instance and the set of strong stable matchings in Υ be denoted as Ω. Further, given any two-party matching $N_{\mathcal{P},\mathcal{Q}} = \{(p_{i1}, q_{i1}), (p_{i2}, q_{i2}), \cdots, (p_{in}, q_{in})\}$ where $p_{ij} \neq p_{ij'}, q_{ij} \neq q_{ij'}, p_{ij} \in \mathcal{P}, q_{ij} \in \mathcal{Q}, \mathcal{P}, \mathcal{Q} \in \{\mathcal{M}, \mathcal{W}, \mathcal{D}\}, \mathcal{P} \neq \mathcal{Q}$. Then, the subset of strong stable matchings $\Omega_{N_{\mathcal{P},\mathcal{Q}}} = \{\mu \vert \mu \in \Omega, \mu\vert_{\mathcal{P},\mathcal{Q}} = N_{\mathcal{P},\mathcal{Q}}\}$ is a distributive lattice.*

We make two remarks here. First, when we consider a non-empty subset $\Omega_{N_{\mathcal{P},\mathcal{Q}}} = \Omega_{N_{\mathcal{M},\mathcal{W}}}$ of strong stable matchings. We impose a partial order on the elements based on the welfare of one particular group, which, in this case, is $\mathcal{W}$. (Note that all men $\mathcal{M}$ are doing the same in all strong stable matchings in $\Omega_{N_{\mathcal{M},\mathcal{W}}}$). Thus, if $\mu, \mu' \in \Omega_{N_{\mathcal{M},\mathcal{W}}}$, then $\mu \succ \mu'$ if and only if all women in $\mathcal{W}$ are getting dogs in μ ranking at least as high as those they get in μ'. Second, if $\Omega_{N_{\mathcal{P},\mathcal{Q}}} = \emptyset$, we are assuming that it is (vacuously) a distributive lattice as well.

Lemma 5. *Let μ and μ' be two strong stable matchings in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$ and man m and woman w belong to the same triple in μ but not so in μ'. Then one of them prefers μ while the other prefers μ'.*

Proof. Let $\mathcal{X}, \mathcal{Y}$ be the sets of men and women preferring μ respectively; analogously, let $\mathcal{X}', \mathcal{Y}'$ be the set of men and women preferring μ' respectively.

We claim that if $m \in \mathcal{X}$, then his partner w in μ must be a member of $\mathcal{Y}'$. If this is not so, then (m, w, d) blocks μ', where d is the dog that has m as a partner in both μ and μ'. Thus, we have $|\mathcal{X}| \leq |\mathcal{Y}'|$. By an analogous argument, every man m in $\mathcal{X}'$ must have a woman $w \in \mathcal{Y}$ as a partner in μ', otherwise, (m, w, d) blocks μ, where d is the dog that has m as a partner in both μ and μ'. So we have $|\mathcal{X}'| \leq |\mathcal{Y}|$

By the fact that in both μ and μ', all dogs have the same partners, so the number of men and women having different partners must be equal: $|\mathcal{X}| + |\mathcal{X}'| = |\mathcal{Y}| + |\mathcal{Y}'|$. This, combined with the previous two facts, $|\mathcal{X}| \leq |\mathcal{Y}'|$ and $|\mathcal{X}'| \leq |\mathcal{Y}|$, implies that $|\mathcal{X}| = |\mathcal{Y}'|$, $|\mathcal{X}| = |\mathcal{Y}'|$. Now if every man in $\mathcal{X}$ has a woman in $\mathcal{Y}'$ as a partner in μ, then every man in $\mathcal{X}'$ must have a woman in $\mathcal{Y}$ in μ. This gives us the lemma. □

Lemma 6. *Let μ and μ' be two strong stable matchings in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$. If all men are given the better partners in the two matching μ and μ', then the resultant matching, denoted as $\mu \wedge \mu'$, is also a strong stable matching in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$.*

Proof. We first need to argue that $\mu \wedge \mu'$ is really a matching. Suppose, for a contradiction, that both m and m' are matched to w in $\mu \wedge \mu'$. Without loss of generality, let m and m' be matched to w in μ and μ', respectively. By Lemma 5, since m prefers matching μ, then w must prefer μ'. This, combined with the fact that m' also prefers w to his partner in μ, implies that (m', w, d'), where dog d' always has m' as a partner in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$ is a blocking triple of degree 2 in μ, a contradiction.

We now argue the stability of $\mu \wedge \mu'$. Suppose that (m, w, d) is a blocking triple of degree 3. Without loss of generality, let m' be the man who gets w as a partner in μ and he prefers (or is indifferent to) μ. In μ, w also strictly prefers d to her assigned dog partner d', who always has m' as a partner in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$, in μ. It is easy to see that man m and dog d prefers w and m, respectively, to their assigned partner in both μ and μ'. Therefore, (m, w, d) is a blocking triple of degree 3 in μ, a contradiction.

Finally, suppose (m, w, d) is a blocking triple of degree 2 to $\mu \wedge \mu'$. There are three cases to consider and their arguments are similar. We consider only one case. Suppose m is the player who is indifferent. Let μ be the matching in which m is matched to w and m prefers (or is indifferent to) μ. Then (m, w, d) is also a blocking triple of degree 2 in μ, a contradiction. □

The lemma below follows analogous arguments to those in the preceding one.

Lemma 7. *Let μ and μ' be two strong stable matchings in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$. If all women are given the better partners in the two matching μ and μ', then the resultant matching, denoted as $\mu \vee \mu'$, is a strong stable matching in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$.*

Now, armed with Lemma 6 and Lemma 7, we can introduce the lemma that establishes the distributive law of the lattice.

Lemma 8. *Let μ, μ' and μ'' be three strong stable matchings in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$. Then $\mu \wedge (\mu' \vee \mu'') = (\mu \wedge \mu') \vee (\mu \wedge \mu'')$ and $\mu \vee (\mu' \wedge \mu'') = (\mu \vee \mu') \wedge (\mu \vee \mu'')$*

Proof. Lemma 6 and Lemma 7 establish that meet and join operations result in a strong stable matching in $\Omega_{N_{\mathcal{D},\mathcal{M}}}$. The distributive law can be easily verified. □

The correctness of Theorem 4 follows from Lemma 6, Lemma 7 and Lemma 8.

4.2 Strong Stable Matchings in 3-way Kidney Transplant

In the context of 3-way kidney transplant, the set of the strong stable matchings forms a union of meet-semilattices. Consider the subset of strong stable matchings in which each of a fixed set of *n* players has the same successor and they all belong to different triples. Such a subset is equivalent to a meet-semilattice.

Theorem 5. *Let $\Upsilon = (\mathcal{K}, \mathcal{P})$ be a 3-way kidney transplant instance and the set of strong stable matchings in Υ be denoted as Ω. Furthermore, let a two-party matching be $N = \{(k_{i1}, k_{i2}), (k_{i3}, k_{i4}), \cdots, (k_{i(2n-1)}, k_{i(2n)})\}$ where $k_{ij} \in \mathcal{K}, k_{ij} \neq k_{ij'}$. Then, the subset of strong stable matchings $\Omega_N = \{\mu | \mu \in \Omega; if\ (k_{ij}, k_{i(j+1)}) \in N,\ then\ \exists t = \overrightarrow{(k_{ij}, k_{i(j+1)}, k^{\dagger})} \in \mu\}$ is a meet-semilattice.*

It takes three lemmas to prove Theorem 5, whose correctness arguments are mostly extended from Section 4.3 in Gusfield and Irving's book [8].

Lemma 9. *Let Ω_N be the subset of strong matchings based on a two-party matching N. Given any two matchings $\mu, \mu' \in \Omega_N$. Suppose k_y is k_x's successor in μ but not in μ'; moreover, k_y has different successors in μ and μ'. Then one of them prefers μ while the other prefers μ'.*

Proof. Let $\mathcal{B}, \mathcal{V}, \mathcal{S} \subset \mathcal{K}$ be the set of players who are getting better, worse, the same successors in μ. Consider a triple $\overrightarrow{(k_x, k_y, k_z)} \in \mu$ and suppose that $k_x \in \mathcal{B}$ and k_y gets different successors in μ and μ'. Then it follows that $k_z \in \mathcal{S}$. (Recall that exactly one player in (k_x, k_y, k_z) has the same successor in all the matchings in Ω_N.) We claim that $k_y \in \mathcal{V}$, otherwise, $\overrightarrow{(k_x, k_y, k_z)} \in \mu$ is a blocking triple of degree 2 in μ'. So we have that $|\mathcal{B}| \leq |\mathcal{V}|$.

Conversely, let $\overrightarrow{(k_x, k_y, k_z)} \in \mu'$, $k_x \in \mathcal{V}$ and k_y gets different successors in μ and μ'. It follows that $k_z \in \mathcal{S}$. Again, k_y must be in $\mathcal{B}$, otherwise, $\overrightarrow{(k_x, k_y, k_z)}$ is a blocking triple of degree 2 in μ. This gives us $|\mathcal{V}| \leq |\mathcal{B}|$.

Combining the above two facts, we have that $|\mathcal{V}| = |\mathcal{B}|$. Moreover, given any triple in μ or in μ', if a player and his successor have different successors in μ and in μ', one of them prefers μ while the other prefers μ'. □

Lemma 9 is useful in proving the following lemma.

Lemma 10. *Let Ω_N be the subset of strong matchings. Given any three matchings $\mu, \mu', \mu'' \in \Omega_N$, if every player k_i is given the median choice of his successors in the three matchings, we derive another strong stable matching $\overline{\mu} \in \Omega_N$.*

Proof. We first need to argue that $\overline{\mu}$ is really a matching. Recall that in all the matchings in Ω_N, the set of players $(k_{i1}, k_{i3}, \cdots, k_{i(2n-1)})$ have fixed successors and belong to different triples. Hence, in $\overline{\mu}$, they must still have the same successors $(k_{i2}, k_{i4}, \cdots, k_{i(2n)})$. Now consider a member $k_{i(2s)}$ from the latter group and denote his successor in $\overline{\mu}$ as $k^{\dagger}$. If, among at least two out of $\mu, \mu', \mu'' \in \Omega_N$, $k_{i(2s)}$ has $k^{\dagger}$ as a successor, then $k^{\dagger}$ necessarily has $k_{i(2s-1)}$ as successor, guaranteeing that $\overrightarrow{(k_{i(2s-1)}, k_{i(2s)}, k^{\dagger})}$ to be an oriented triple in $\overline{\mu}$. On the other hand, if in the three matchings $\{\mu, \mu', \mu''\}$, $k_{i(2s)}$ has all different successors and he prefers μ in μ' in μ'', then by Lemma 9, $k^{\dagger}$ prefers μ'' in μ' in μ, again ensuring that $\overrightarrow{(k_{i(2s-1)}, k_{i(2s)}, k^{\dagger})}$ is really an oriented triple in $\overline{\mu}$.

Now we argue the strong stability of $\overline{\mu}$. Suppose there exists a blocking triple $\overrightarrow{(k^{\phi 1}, k^{\phi 2}, k^{\phi 3})}$. If it is of degree 3, then $k^{\phi 1}$ strictly prefers $k^{\phi 2}$ to his successor in at least 2 out of the three matchings in $\{\mu, \mu', \mu''\}$. Analogous argument applies to $k^{\phi 2}$ and $k^{\phi 3}$, respectively, implying $\overrightarrow{(k^{\phi 1}, k^{\phi 2}, k^{\phi 3})}$ blocks at least one matching in $\{\mu, \mu', \mu''\}$.

If, on the other hand, $\overrightarrow{(k^{\phi 1}, k^{\phi 2}, k^{\phi 3})}$ is a blocking triple of degree 2. Let $k^{\phi 1}$ be the player who is indifferent. Then $k^{\phi 2}$ must be his successor in $\overline{\mu}$. So $k^{\phi 1}$ *either is indifferent to, or strictly prefers* $k^{\phi 2}$ in two out of the three matchings in $\{\mu, \mu', \mu''\}$. For $k^{\phi 2}$ and $k^{\phi 3}$, they strictly prefer $k^{\phi 3}$ and $k^{\phi 1}$ in at least two out the three matchings in $\{\mu, \mu', \mu''\}$. So, $\overrightarrow{(k^{\phi 1}, k^{\phi 2}, k^{\phi 3})}$ is a blocking triple of degree *at least* 2 in one of the matchings in $\{\mu, \mu', \mu''\}$. This contradiction completes the stability proof. □

Given a strong stable matching $\mu \in \Omega_N$, we define a function T that maps μ to a collection of pairs. To be precise, $(k_1, k_2) \in T(\mu)$ if k_2 ranks at least as high as k_1's successor in the matching μ. Moreover, we can choose an arbitrary *pivot matching* $\mu_0 \in \Omega_N$ and define a function T_0 so that $T_0(\mu) = T(\mu) \bigoplus T(\mu_0)$. And then we can use the following lemma establish Theorem 5.

Lemma 11. *The sets $T_0(\mu)$, taken over all strong stable matchings $\mu \in \Omega_N$, are closed under intersection, and so can be regarded as a meet-semilattice in which $T_0(\mu_0)$ is the minimal element, where $\mu_0 \in \Omega_N$ is the pivot matching.*

Proof. Suppose that we are given two strong stable matchings $\{\mu, \mu'\} \subseteq \Omega_N$ and suppose that the median matching of μ, μ', μ_0 (using the operation of Lemma 10) is $\overline{\mu}$. We will show that $T_0(\mu) \cap T_0(\mu') = T_0(\overline{\mu})$, thereby establishing that the sets $T_0(\mu)$, for all $\mu \in \Omega_N$ are closed under intersection.

Let k be a fixed player. We consider all possible cases.

- If k has the same successor in μ and in μ_0, or the same successor in μ' and in μ_0, then there is no pair (k, k') in $T_0(\mu) \cap T_0(\mu')$; also, there will not be any pair (k, k') in $T_0(\overline{\mu})$, since k's successor in $\overline{\mu}$ will be the same as the one that he has in μ_0.
- If k has the same successor in μ and in μ', but he has a different successor in μ_0, then all the pairs (k, k') exist in $T_0(\mu), T_0(\mu')$, and $T_0(\overline{\mu})$. The reason is because k will have the same successor in $\overline{\mu}$ as in μ and μ'.
- If k has three different successors in μ, μ', μ_0, we then consider further the subcases.
  - If k has the same successor in $\overline{\mu}$ and μ_0, then there is no pair (k, k') in $T_0(\mu) \cap T_0(\mu')$, nor in $T_0(\overline{\mu})$.
  - If k has the same successor in $\overline{\mu}$ and μ, then the pairs (k, k') in $T_0(\mu) \cap T_0(\mu')$ are exactly those in $T_0(\mu)$. Moreover, they are also those in $T_0(\overline{\mu})$.
  - If k has the same successor in $\overline{\mu}$ and μ', then the argument is analogous to the preceding case.

By the above case analysis, we establish the closure under intersection. It is obvious that $T_0(\mu_0)$ is the minimal element, as it is an empty set. □

By Lemma 11, we prove Theorem 5.

5 #P-completeness of Strong Stable Matchings

In this section, we present a reduction from the 2-party STABLE MARRIAGE problem to the 3-WAY KIDNEY TRANSPLANT problem. Counting the number of stable matchings in a stable marriage instance is #P-complete, a fact established by Irving and Leather [13].

To build up some intuition, we first show how to "embed" a STABLE MARRIAGE instance $\Upsilon = (\mathcal{M}, \mathcal{W}, \mathcal{P})$ into a CIRCULAR STABLE MATCHING instance $\Upsilon' = (\mathcal{M}', \mathcal{W}', \mathcal{D}', \mathcal{P}')$. For each player $p \in \mathcal{M} \cup \mathcal{W}$, we create a player p' and add her/him/it into the derived instance Υ'. Suppose a man $m'_i \in \mathcal{M}'$ is created based on $m_i \in \mathcal{M}$. We let him have the same preference as m_i. Precisely, supposing that $P(m_i) = w_{i1} \succ w_{i2} \succ \cdots \succ w_{in}$, let $P'(m'_i) = w'_{i1} \succ w'_{i2} \succ \cdots \succ w'_{in}$. Furthermore, for each man $m'_i \in M'$, we create a dog d'_i and add it into $\mathcal{D}'$ with preference $P'(d'_i) = m'_i \succ \cdots$. For a woman $w'_i \in \mathcal{W}'$, her preference is now for dogs, moreover, in her new preference, *the indices are kept the same*. To be precise, if $P(w_i) = m_{i1} \succ m_{i2} \succ \cdots \succ m_{in}$, we make $P(w'_i) = d'_{i1} \succ d'_{i2} \succ \cdots \succ d'_{in}$.

By this construction, it is easy to observe that the matching $\mu = \{(m_{j1}, w_{j1}), (m_{j2}, w_{j2}), \cdots, (m_{jn}, w_{jn})\}$ is stable in Υ if and only if the matching $\mu' = \{(m'_{j1}, w'_{j1}, d'_{j1}), (m'_{j2}, w'_{j2}, d'_{j2}), \cdots, (m'_{jn}, w'_{jn}, d'_{jn})\}$ is strongly stable in Υ'. A blocking pair (m_{jk}, w_{jl}) in the former implies a blocking triple $(m'_{jk}, w'_{jl}, d'_{jk})$ of degree 2 in the latter. Conversely, there cannot be a blocking triple of degree 3 in μ' (since every dog is matched to its top-ranked man). A blocking triple $(m'_{jk}, w'_{jl}, d'_{jk})$ of degree 2 implies that (m_{jk}, w_{jl}) blocks μ as well.

From the fact that the number of stable matchings in STABLE MARRIAGE can be exponential (see Knuth's book [15]), the fact that weak stable matchings are a superset of strong stable matchings, and the reduction given in Section 2.2, we establish:

Theorem 6. *The number of weak and strong stable matchings in circular stable matching and 3-way kidney transplant problems can be exponential.*

Unfortunately, the above construction of Υ' is not a reduction, instead, it is merely an embedding. There is no guarantee that some other strong stable matching (in which dogs are not always matched to their top-ranked men) will not arise in Υ'. To prove the #P-completeness, we need one more twist.

We transform Υ' into a 3-WAY KIDNEY TRANSPLANT INSTANCE $\Upsilon'' = (\mathcal{K}'', \mathcal{L}'')$ as follows. We first make a copy of every player in $\mathcal{M}' \cup \mathcal{W}' \cup \mathcal{D}'$ and add it into $\mathcal{K}''$. For each dog $d''_i \in \mathcal{K}''$, we create a set of guard players to restrict its possible successors in a strong stable matching. The idea here is similar to the one we used in the reduction of Section 2.3. We need an instance $\Upsilon_{d''_i} = (\mathcal{K}_{d''_i}, \mathcal{L}_{d''_i})$ which has the properties: (1) it has four players, $k^{\#}_{d''_i,j}, 1 \leq j \leq 4$, and (2) it does not allow strong stable matching itself (see [9] for such an instance).

We embed $\Upsilon_{d''_i}$ into Υ'' by altering the preferences of d''_i and its associated three guard players as follows.

- $P''(d''_i) = m''_i \succ P_{\Upsilon_{d''_i}}(k^{\#}_{d''_i,1}) \succ \cdots$, where $P_{\Upsilon_{d''_i}}(k^{\#}_{d''_i,1})$ is the preference list of $k^{\#}_{d''_i,1}$ in the instance $\Upsilon_{d''_i}$
- $P''(k^{\#}_{d''_i,j}) = P_{\Upsilon_{d''_i}}(k^{\#}_{d''_i,j}) \succ \cdots$, where $2 \leq j \leq 4$ and $P_{\Upsilon_{d''_i}}(k^{\#}_{d''_i,j})$ is the preference list of $k^{\#}_{d''_i,j}$ in the instance $\Upsilon_{d''_i}$.

The intent here is to try to remove one player, d''_i (who plays the role of $k^{\#}_{d''_i,1}$) from $\Upsilon_{d''_i}$ to prevent a potential blocking triple in $\Upsilon_{d''_i}$ from blocking a strong stable matching in Υ''.

After adding the $3n$ guard players into $\mathcal{K}''$, we also have to update the preferences of the men and women who are the copies of those in $\mathcal{M}' \cup \mathcal{W}'$. Such a player, say, m''_i, replaces each woman $w'_j \in \mathcal{W}''$ with w''_j in his list and attaches other players to the end of his list.

It can be checked that in all strong stable matchings in Υ'', dogs have their top-ranked men as successors. Moreover, a matching $\mu = \{(m_{j1}, w_{j1}), (m_{j2}, w_{j2}), \cdots, (m_{jn}, w_{jn})\}$ is stable in Υ if and only if a matching $\mu'' = \{\overrightarrow{(m''_{j1}, w''_{j1}, d''_{j1})}, \overrightarrow{(m''_{j2}, w''_{j2}, d''_{j2})}, \cdots, \overrightarrow{(m''_{jn}, w''_{jn}, d''_{jn})}\}$ is strongly stable in Υ''. Therefore, the reduction from Υ to Υ'' is correct. Using a similar and slightly more complicated gadget (of guard players), it is also possible to have a reduction from Υ to an instance of CIRCULAR STABLE MATCHING. We omit it here.

We conclude this section with the following theorem.

Theorem 7. *It is #P-complete to count the number of strong stable matchings in both circular stable matching and 3-way kidney transplant problems.*

6 Conclusion

We have left a complexity issue unanswered: existence of a weak stable circular matching. We were unable to come up with a reduction, for there is no similar gadget (a small instance allowing no weak stable matchings) to the one we used in Section 3. Indeed, the reason may go deeper. Empirical evidence indicates that the number of weak stable circular matchings grows extraordinarily fast with the problem size. Eriksson, Sjöstrand and Strimling [4] conjectured that weak stable matchings always exist. This is why we remarked previously that finding one is probably not NP-complete. Is there a technique, combinatorial or otherwise, to prove their perennial existence?

Acknowledgments

I thank my advisor Peter Winkler for many helpful discussions.

References

1. David Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *ACM E-commerce*, 2007.
2. Endre Boros, Vladimir Gurvich, Steven Jaslar, and Daniel Krasner. Stable matchings in three-sided systems with cyclic preferences. *Discrete Mathematics*, 289(1-3):1–10, 2004.
3. Katarína Cechlárová, Tamás Fleiner, and David Manlove. The kidney exchange game. In *The 8th International Symposium on Operations Research in Slovenia*, pages 77–83, 2005.
4. Kimmo Eriksson, Jonas Sjöstrand, and Pontus Strimling. Three-dimensional stable matching with cyclic preferences. *Mathematical Social Sciences*, 52(1):77–87, 2006.
5. David Gale and Lloyd Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.
6. Michael Garey and David Johnson. *Computers and Intractablility*. Freeman, 1979.
7. Sommer Gentry, Dorry Segev, and Robert Montgomery. A comparison of populations served by kidney paired donation and list paired donation. *American Journal of Transplantation*, 5(8):1914–1921, 2004.
8. Dan Gusfield and Robert Irving. *The Stable Marriage Problem*. The MIT Press, 1989.
9. Chien-Chung Huang. Ph.D. thesis.
10. Chien-Chung Huang. Two's company, three's a crowd: Stable family and threesome roommates problems. In *15th Annual European Symposium on Algorithms (ESA)*, 2007.
11. Robert Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48:261–272, 1994.
12. Robert Irving. The cycle roommates problem: a hard case of kidney exchange. *Information Processing Letters*, 103:1–4, 2007.
13. Robert Irving and Paul Leather. The complexity of counting stable marriages. *SIAM Journal on Computing*, 15:655–667, 1986.
14. Richard Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.
15. Donald Knuth. *Mariages stables et leurs relations avec d'autre problèmes combinatoires*. Les Presses de l'université de Montréal, 1976.
16. Cheng Ng and Daniel Hirschberg. Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics*, 4(2):245–252, 1991.
17. Alvin Roth, Tayfun Sonmez, and M. Utku Unver. Kidney exchange. *Quarterly Journal of Economics*, 119(2):457–488, May 2004.
18. Alvin Roth, Tayfun Sonmez, and M. Utku Unver. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation*, 81(5):773–782, March 2006.
19. Alvin Roth, Tayfun Sonmez, and M. Utku Unver. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97(3):828–851, June 2007.
20. Dorry Segev, Sommer Gentry, Daniel Waren, Brigitte Reeb, and Robert Montgomery. Kidney paired donation and optimizing the use of live donor organs. *Journal of the American Medical Association*, 293(15):1883–1890, April 2005.
21. Ashok Subramanian. A new approach to stable matching problems. *SIAM Journal on Computing*, 23(4):671–700, 1994.

Three-sided stable matchings with cyclic preferences*

Péter Biró[†] and Eric McDermid

Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK
Email: `pbiro@dcs.gla.ac.uk`, `mcdermid@dcs.gla.ac.uk`.

Abstract

Knuth [14] asked whether the stable matching problem can be generalised to three dimensions i. e., for families containing a man, a woman and a dog. Subsequently, several authors considered the three-sided stable matching problem with cyclic preferences, where men care only about women, women only about dogs, and dogs only about men. In this paper we prove that if the preference lists may be incomplete, then the problem of deciding whether a stable matching exists, given an instance of three-sided stable matching problem with cyclic preferences is NP-complete. Considering an alternative stability criterion, strong stability, we show that the problem is NP-complete even for complete lists. These problems can be regarded as special types of stable exchange problems, therefore these results may have relevance in some real applications, such as kidney exchange programs.

Keywords: stable marriage problem, three-dimensional matching, cyclic preferences, computational complexity

1 Introduction

An instance of the Stable Marriage problem (SM) comprises a set of n men $a_1, \dots, a_n$ and a set of n women $b_1, \dots, b_n$. Each person has a complete preference list consisting of the members of the opposite sex. If b_j precedes b_k on a_i's list then a_i is said to *prefer* b_j to b_k. The problem is to find a matching that is *stable* in the sense that no man and woman both prefer each other to their current partner in the matching. The Stable Marriage problem was introduced by Gale and Shapley [9]. They constructed a linear time algorithm that always finds a stable matching for an SM instance.

Considering the Stable Marriage problem with Incomplete Lists (SMI), the only difference is that the numbers of men and women are not necessarily equal and each preference list consist of a subset of the members of the opposite sex, i.e., each person lists his or her *acceptable partners*. Here, a matching $\mathcal{M}$ is a set of acceptable pairs, and $\mathcal{M}$ is stable if for every pair $(a_i, b_j) \notin \mathcal{M}$, either a_i prefers his matching partner $\mathcal{M}(a_i)$ to b_j or b_j prefers her matching partner $\mathcal{M}(b_j)$ to a_i. We can model this problem by a bipartite graph $G = (A \cup B, E)$, where the sets of vertices, A and B, correspond to the sets of men and women, respectively, and the set of edges, E represents the acceptable pairs. An extended version of the Gale–Shapley algorithm always produces a stable matching for this setting too.

*This work was supported by EPSRC grant EP/E011993/1.
[†]Supported by OTKA grant K69027.

In an instance of the Stable Marriage problem with Ties and Incomplete Lists (SMTI) it is possible that an agent is indifferent between some acceptable agents from the opposite set; in such a case, when these agents appear together in a *tie* in the preference list. Here, a matching $\mathcal{M}$ is stable if there is no blocking pair $(a_i, b_j) \notin \mathcal{M}$ such that a_i is either unmatched or prefers b_j to $\mathcal{M}(a_i)$, and simultaneously b_j is either unmatched or prefers a_i to $\mathcal{M}(b_j)$. Manlove et al. [15] proved that the problem of finding a stable matching of maximum cardinality for an instance of SMTI, the so-called MAX SMTI problem, is NP-hard.

The Three-Dimensional Stable Matching problem (3DSM), also referred to as the Three Gender Stable Marriage problem, was introduced by Knuth [14]. Here, we have three sets of agents: men, women and dogs, say, and each agent has preference over all pairs from the two other sets. A *matching* is a set of disjoint *families* i.e., triples of the form (man, woman, dog). A matching is *stable* if there exists no blocking family that is preferred by all its members to their current families in the matching.

Alkan [2] gave the first example of an instance of 3DSM where no stable matching exists. Ng and Hirschberg [17] proved that the problem of deciding whether a stable matching exists, given an instance of 3DSM, is NP-complete; later Subramanian [26] gave an alternative proof for this. Recently, Huang [10] proved that the problem remains NP-complete even if the preference lists are "consistent". (A preference list is inconsistent if, for example, man m ranks (w_1, d_1) higher than (w_2, d_1), but he also ranks (w_2, d_2) higher than (w_1, d_2), so he does not consistently prefer woman w_1 to woman w_2.)

As an open problem, Ng and Hirschberg [17] mentioned the cyclic 3DSM, defined formally in Section 2, where men only care about women, women only care about dogs and dogs only care about men. Boros et al. [5] showed that if the number of agents n, is at most 3 in every set, then a stable matching always exists. Eriksson et al. [8] proved that this also holds for $n = 4$ and conjectured that a stable matching exists for every instance of cyclic 3DSM.

In Section 2, we study the cyclic 3DSM problem with Incomplete Lists (cyclic 3DSMI). Here, each preference list may consist of a subset of the members of the next gender, i.e. his, her or its *acceptable partners*, and the cardinalities of the sets are not necessarily the same, a matching is a set of acceptable families. Thus cyclic 3DSMI is obtained via a natural generalisation of cyclic 3DSM in a way analogous to the extension SMI of SM. First we give an instance of cyclic 3DSMI for $n = 6$ where no stable matching exists. Then, by using this instance as a gadget, we show that the problem of deciding whether a stable matching exists in an instance of cyclic 3DSMI is NP-complete. We reduce from MAX SMTI.

In Section 3, we study the cyclic 3DSM problem under *strong stability*. A matching is strongly stable if there exists no *weakly blocking family*. This is a family not in the matching that is weakly preferred by all its members (i.e. no member prefers his original family to the new blocking family). We show that the problem of deciding whether a strongly stable matching exists in an instance of cyclic 3DSM is NP-complete.

In Section 4, we describe the correspondence between the cyclic 3DSMI problem and the so-called stable exchange problem with restrictions, defined in Section 4. More precisely, we show that the 3-way stable 3-way exchange problem for tripartite cyclic graphs is equivalent to cyclic 3DSMI. Therefore, the complexity result for cyclic 3DSMI applies also to the 3-way stable 3-way exchange problem, which is an important model for the kidney exchange problem (this application is described in further detail in Section 4).

2 Cyclic 3DSMI is NP-complete

Problem definition

We consider three sets of agents: M, W, D (men, women and dogs). Every man has a strict preference list over the women that are acceptable to him. Analogously, every woman has a strict preference list over her acceptable dogs, and every dog has a strict preference list over its acceptable men. The list of an agent x is denoted by $P(x)$. A *matching* $\mathcal{F}$ is a set of disjoint families, i.e., triples from $M \times W \times D$, such that for each family $(m, w, d) \in \mathcal{F}$, w is acceptable to m, d is acceptable to w and m is acceptable to d. Formally, if $(m, w, d) \in \mathcal{F}$, then we say that $\mathcal{F}(m) = w$, $\mathcal{F}(w) = d$ and $\mathcal{F}(d) = m$, thus in a matching, $\mathcal{F}(x) \in P(x) \cup \{x\}$ holds for every agent x, where $\mathcal{F}(x) = x$ means that agent x is unmatched in $\mathcal{F}$. Note that agent x prefers y to being unmatched if $y \in P(x)$.

A matching $\mathcal{F}$ is said to be *stable* if there exists no *blocking family*, that is a triple $(m, w, d) \notin \mathcal{F}$ such that m prefers w to $\mathcal{F}(m)$, w prefers d to $\mathcal{F}(w)$ and d prefers m to $\mathcal{F}(d)$.

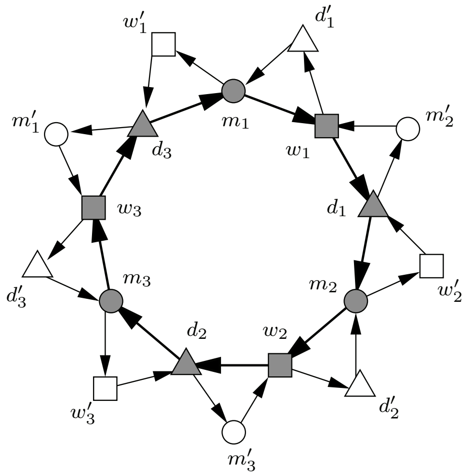
We define the underlying directed graph $D_I = (V, A)$ of an instance I of cyclic 3DSMI as follows. The vertices of D_I correspond to the agents, so $V(D_I) = M \cup W \cup D$, and we have an arc (x, y) in D_I if $y \in P(x)$. This type of directed graph where $A(D_I) \subseteq (M \times W) \cup (W \times D) \cup (D \times M)$ is called a *tripartite cyclic digraph*. Therefore, a matching of I corresponds to a disjoint packing of directed 3-cycles in D_I.

An unsolvable instance of cyclic 3DSMI

We give an instance of cyclic 3DSMI with $n = 6$, denoted by $R6$, where no stable matching exists.

Example 1. *The preference lists and underlying graph of $R6$ are as shown below. Here, the thickness of arrows correspond to preferences.*

$m_1 : w_1, w'_1$	$w_1 : d_1, d'_1$	$d_1 : m_2, m'_2$
$m_2 : w_2, w'_2$	$w_2 : d_2, d'_2$	$d_2 : m_3, m'_3$
$m_3 : w_3, w'_3$	$w_3 : d_3, d'_3$	$d_3 : m_1, m'_1$
$m'_1 : w_3$	$w'_1 : d_3$	$d'_1 : m_1$
$m'_2 : w_1$	$w'_2 : d_1$	$d'_2 : m_2$
$m'_3 : w_2$	$w'_3 : d_2$	$d'_3 : m_3$



We refer to the agents $\{m_i, w_i, d_i : 1 \leq i \leq 3\} = I$ as the *inner agents* of $R6$ and the agents $\{m'_i, w'_i, d'_i : 1 \leq i \leq 3\} = O$ as the *outer agents* of $R6$.

Lemma 1. *The instance $R6$ of cyclic 3DSMI admits no stable matching.*

Proof. By inspection of the underlying graph of $R6$, we can observe that the only acceptable families are of the form (m_i, w'_i, d_{i-1}), (m_i, w_i, d'_i) and (m'_i, w_{i-1}, d_{i-1}), so that any acceptable family contains exactly two inner agents. It is clear that for any matching $\mathcal{F}$, it must be the case that at least one inner agent is unmatched in $\mathcal{F}$. By the symmetry of the instance we may suppose without loss of generality that the inner agent m_1 is unmatched in $\mathcal{F}$. Then, the family (m_1, w'_1, d_3) is a blocking family for $\mathcal{F}$. □

We note that the 9 acceptable families of $R6$ have a natural cyclic order, the same order that the directed 9-cycle has which is formed by the 9 inner agents in the underlying graph, such that if an acceptable family is not in a stable matching $\mathcal{F}$ then the successor family must be in $\mathcal{F}$. For example, if $(m_1, w_1, d'_1) \notin \mathcal{F}$ then $(m'_2, w_1, d_1) \in \mathcal{F}$, since (m_1, w_1, d'_1) would be blocking otherwise. This argument gives an alternative proof for the above Lemma.

The instance created by removing the inner agent m_1 from $R6$, denoted by $R6 \setminus m_1$, becomes solvable, since $\mathcal{F}^* = \{(m'_2, w_1, d_1), (m_2, w_2, d'_2), (m_3, w'_3, d_2), (m'_1, w_3, d_3)\}$ is a stable matching for $R6 \setminus m_1$. In fact, $\mathcal{F}^*$ is the unique stable matching for $R6 \setminus m_1$, so we denote it by $\mathcal{F}_{R6\setminus m_1}$. This is because in $R6 \setminus m_1$ we have 7 acceptable families in a row with the property discussed above: if an acceptable family is not in a stable matching $\mathcal{F}$ then the subsequent family must be in $\mathcal{F}$. We state this claim formally below; its proof follows from the symmetry of the instance.

Lemma 2. *Let a_i be an inner agent of $R6$. Then, $R6 \setminus a_i$ admits a unique stable matching, denoted by $\mathcal{F}_{R6\setminus a_i}$.*

The instance $R6$ will also be of use to us as a gadget in the NP-completeness proofs of the subsequent sections.

The NP-completeness proof

In [15], Manlove et al. proved that determining if an instance of SMTI admits a complete stable matching is NP-complete, even if the ties appear only on the women's side, and each woman's preference list is either strictly ordered or consists entirely of a tie of size two (these conditions holding simultaneously).

We refer to the MAX SMTI problem under the above restrictions as Restricted SMTI. The underlying graph $G = (A \cup B, E)$ of a Restricted SMTI instance is such that the set $A = \{a_1, a_2, \dots, a_n\}$ consists of men a_i, all of whom have strictly ordered preference lists, while the set B of women can be partitioned into two sets $B_1 \cup B_2 = \{b_1, \dots, b_{n_1}\} \cup \{b^T_1, \dots, b^T_{n_2}\}$ where $n_1 + n_2 = n$, each woman $b_j \in B_1$ has a strictly ordered preference list, and each woman $b^T_j \in B_2$ has a preference list consisting solely of a tie of length 2. We denote a woman who can either be a member of B_1 or B_2 by $b^{(T)}_i$.

In the remainder of this section we describe a polynomial-time reduction from Restricted SMTI to cyclic 3DSMI. Let I be an instance of Restricted SMTI with the underlying graph $G = (A \cup B, E)$. We construct an instance I' of cyclic 3DSMI with sets M, W, and D of men, women, and dogs as follows.

The sets of men and women of I' are created in direct correspondence to the men and women in I, so let $M = \{m_1, \dots, m_n\}$ and $W = W_1 \cup W_2 = \{w_1, \dots, w_{n_1}\} \cup \{w^T_1, \dots, w^T_{n_2}\}$. The set of dogs of I' consists of two parts $D_1 \cup D_2 = D$, defined by creating a dog $d_{j,i}$ in D_1 if $a_i \in P(b_j)$, and creating a dog d^T_j in D_2 if $b^T_j \in B_2$.

Let us now describe the construction of the strictly ordered preference lists of I'. We let $P(x)[l]$ denote the lth entry in agent x's preference list, and a tie in the preference list of an agent is indicated by parentheses. The preference lists of I' are defined by the following cases:

1. If $P(a_i)[l] = b_j^{(T)}$ then let $P(m_i)[l] = w_j^{(T)}$ ($1 \le l \le r$, where r is the length of a_i's list).

2. If $P(b_j)[l] = a_i$ then let $P(w_j)[l] = d_{j,i}$ and $P(d_{j,i}) = m_i$ ($1 \le l \le r$, where r is the length of b_j's list).

3. If $P(b_j^T) = (a_p, a_q)$ then let $P(w_j^T) = d_j^T$ and $P(d_j^T) = m_p m_q$ (in arbitrary order).

This is the *proper part* of the instance. Next we construct the *additional part* of the instance by creating $n = |M|$ copies of $R6$, such that the t-th copy of $R6$ consists of inner agents $\{m_{t_i}, w_{t_i}, d_{t_i} : 1 \le i \le 3\}$ and outer agents $\{m'_{t_i}, w'_{t_i}, d'_{t_i} : 1 \le i \le 3\}$ with preference lists as described in Example 1. We add these n copies of $R6$ to the instance in the following way. In the t-th added copy of $R6$, denoted by $R6_t$, replace the inner agent m_{t_1} in $R6_t$ with man $m_t \in M$ by replacing each occurrence of m_{t_1} in the preference lists of each agent in $R6_t$ with m_t. Also, let m_{t_1}'s acceptable partners in $R6_t$, namely w_{t_1} and w'_{t_1} be appended in this order to the end of m_t's list. The final preference list of man m_t along with $R6_t$ is shown below. The portion of m_t's preference list consisting of women from the proper part of the instance is denoted by P_t.

m_t	:	$P_t\ w_{t_1}\ w'_{t_1}$	w_{t_1}	:	$d_{t_1}\ d'_{t_1}$	d_{t_1}	:	$m_{t_2}\ m'_{t_2}$
m_{t_2}	:	$w_{t_2}\ w'_{t_2}$	w_{t_2}	:	$d_{t_2}\ d'_{t_2}$	d_{t_2}	:	$m_{t_3}\ m'_{t_3}$
m_{t_3}	:	$w_{t_3}\ w'_{t_3}$	w_{t_3}	:	$d_{t_3}\ d'_{t_3}$	d_{t_3}	:	$m_t\ m'_{t_1}$
m'_{t_1}	:	w_{t_3}	w'_{t_1}	:	d_{t_3}	d'_{t_1}	:	m_t
m'_{t_2}	:	w_{t_1}	w'_{t_2}	:	d_{t_1}	d'_{t_2}	:	m_{t_2}
m'_{t_3}	:	w_{t_2}	w'_{t_3}	:	d_{t_2}	d'_{t_3}	:	m_{t_3}

This ends the reduction, which plainly can be computed in polynomial time. Now, we prove that there is a one-to-one correspondence between the complete stable matchings in I and the stable matchings in I'.

First we show that there is a one-to-one correspondence between the matchings of I and the matchings in the proper part of I'. This comes from the natural one-to-one correspondence between the edges of I and the families in the proper part of I'. More precisely, if $\mathcal{M}$ is a matching in I, then the corresponding matching $\mathcal{F}_p$ in the proper part of I is created as follows: $(a_i, b_j) \in \mathcal{M} \iff (m_i, w_j, d_{j,i}) \in \mathcal{F}_p$ and $(a_i, b_j^T) \in \mathcal{M} \iff (m_i, w_j^T, d_j^T) \in \mathcal{F}_p$. To prove this, it is enough to observe that two edges in I are disjoint if and only if the two corresponding families in I' are also disjoint. Next, we show that stability is preserved by this correspondence.

Lemma 3. *A matching $\mathcal{M}$ of I is stable if and only if the corresponding matching $\mathcal{F}_p$ in the proper part of I' is stable.*

Proof. It is enough to show that an edge (a_i, b_j) is blocking in I if and only if the corresponding family $(m_i, w_j, d_{j,i})$ is also blocking in I'; and similarly, an edge (a_i, b_j^T) is blocking in I if and only if the corresponding family (m_i, w_j^T, d_j^T) is also blocking in I'.

Suppose first that (a_i, b_j) is blocking in I, which means that a_i is either unmatched or prefers b_j to $\mathcal{M}(a_i)$ and b_j is either unmatched or prefers a_i to $\mathcal{M}(b_j)$. This implies that m_i prefers w_j to $\mathcal{F}_p(m_i)$, w_j prefers $d_{j,i}$ to $\mathcal{M}(w_j)$, and $d_{j,i}$ is unmatched in $\mathcal{F}_p$, i.e.

$(m_i, w_j, d_{j,i})$ is blocking in I'. Similarly, if (a_i, b_j^T) is blocking then a_i is either unmatched or prefers b_j^T to $\mathcal{M}(a_i)$ and b_j^T is unmatched in $\mathcal{M}$. This implies that m_i prefers w_j^T to $\mathcal{F}_p(m_i)$, w_j^T and d_j^T are both unmatched in $\mathcal{F}_p$, and hence (m_i, w_j^T, d_j^T) is blocking in I'.

In the other direction, if $(m_i, w_j, d_{j,i})$ is blocking in I', then m_i prefers w_j to $\mathcal{F}_p(m_i)$, w_j prefers $d_{j,i}$ to $\mathcal{F}_p(w_j)$, and $d_{j,i}$ is unmatched in $\mathcal{F}_p$. This implies that a_i is either unmatched or prefers b_j to $\mathcal{M}(a_i)$ and b_j is either unmatched or prefers a_i to $\mathcal{M}(b_j)$, so (a_i, b_j) is blocking in I. Similarly, if (m_i, w_j^T, d_j^T) is blocking in I', then w_j^T and d_j^T are both unmatched in $\mathcal{F}_p$ and m_i prefers w_j^T to $\mathcal{F}_p(m_i)$. This implies that a_i is either unmatched or prefers b_j^T to $\mathcal{M}(a_i)$ and b_j^T is unmatched in $\mathcal{M}$, so (a_i, b_j^T) is blocking in I. □

Furthermore, if the matching $\mathcal{M}$ is complete, then we can enlarge the corresponding matching to the additional part of I' by matching every $R6_t \setminus m_t$ in the unique stable way, so by adding $\mathcal{F}_{R6_t \setminus m_t}$ to $\mathcal{F}_p$ for every t. This leads to the following one-to-one correspondence between the complete stable matchings of I and the stable matching of I'.

Lemma 4. *The instance I admits a complete stable matching $\mathcal{M}$ if and only if the reduced instance I' admits a stable matching $\mathcal{F}$, where $\mathcal{F}$ is the corresponding matching of $\mathcal{M}$.*

Proof. The stability of $\mathcal{M}$ implies that $\mathcal{F}$ is stable in the proper part of I' by Lemma 3. The completeness of $\mathcal{M}$ and Lemma 2 implies that $\mathcal{F}$ is also stable in the additional part of I'.

In the other direction, if $\mathcal{F}$ is stable then every man in M must be matched in a proper family, since otherwise, if a proper man m_t does not have a proper partner in $\mathcal{F}$ then $R6_t$ would contain a blocking family, by Lemma 1. This implies that the corresponding matching $\mathcal{M}$, defined in Lemma 3, is complete. The stability of $\mathcal{M}$ is a consequence of Lemma 3. Finally, we note that the additional part has a unique stable matching, since every $R6_t \setminus a_t$ must be matched in the unique stable way indicated by Lemma 2, which implies the one-to-one correspondence. □

The following Theorem is a direct consequence of Lemma 4.

Theorem 1. *Determining the existence of a stable matching in a given instance of cyclic* 3DSMI *is NP-complete.*

3 Cyclic 3DSM under strong stability is NP-complete

Problem definition

For an instance of cyclic 3DSM, a matching $\mathcal{F}$ is *strongly stable* if there exists no *weakly blocking family*, that is a family $(m, w, d) \notin \mathcal{F}$ such that m prefers w to $\mathcal{F}(m)$ or $w = \mathcal{F}(m)$, w prefers d to $\mathcal{F}(w)$ or $d = \mathcal{F}(w)$, and d prefers m to $\mathcal{F}(d)$ or $m = \mathcal{F}(d)$. We note that in a weakly blocking family at least two members obtain a better partner, since the preference lists are strictly ordered.

An unsolvable instance

We firstly show that, by completing the preference lists of $R6$ in an arbitrary way (i.e. by appending agents not on the lists in an arbitrary order to the tail of the original lists), the resulting instance of cyclic 3DSM, denoted by $\overline{R6}$, does not admit any strongly stable matching. The subinstance $R6$ of $\overline{R6}$ is called the *suitable part* of $\overline{R6}$, the original entries of an agent x in $R6$ are the *suitable partners of* x and the families of $R6$ are called *suitable families.*

Lemma 5. *The instance* $\overline{R6}$ *of cyclic 3DSM admits no strongly stable matching.*

Proof. Suppose for contradiction that $\mathcal{F}$ is a strongly stable matching. As the 9 inner agents form a 9-cycle in the underlying directed graph, the 9 suitable families have a natural cyclic order. We show that if a suitable family, say (m_1, w_1, d_1') is not in $\mathcal{F}$, then the successor suitable family (m_2', w_1, d_1) must be in $\mathcal{F}$, which would imply a contradiction given that the number of these suitable families is odd. If $(m_1, w_1, d_1') \notin \mathcal{F}$ then $\mathcal{F}(w_1) = d_1$, since otherwise (m_1, w_1, d_1') would be weakly blocking. Similarly, $(m_2', w_1, d_1) \notin \mathcal{F}$ implies $\mathcal{F}(d_1) = m_2$. But this means that $(m_2, w_1, d_1) \in \mathcal{F}$, so (m_2, w_2', d_1) is weakly blocking. □

Recall that $\mathcal{F}_{R6\setminus a_t}$ is the unique stable matching for $R6 \setminus a_t$. Let $\overline{R6} \setminus a_t$ denote the instance created by removing an inner agent a_t from $\overline{R6}$. We denote by $C_{R6\setminus a_t}$ the subset of agents of $\overline{R6} \setminus a_t$ that are covered by $\mathcal{F}_{R6\setminus a_t}$, and by $U_{R6\setminus a_t}$ those who are uncovered by $\mathcal{F}_{R6\setminus a_t}$, respectively.

Lemma 6. *Let* a_t *be an inner agent of* $\overline{R6}$*. For every matching* $\mathcal{F}^* \supseteq \mathcal{F}_{R6\setminus a_t}$ *of* $\overline{R6} \setminus a_t$*, no suitable family can be weakly blocking, and therefore no agent from* $C_{R6\setminus a_t}$ *can be involved in a weakly blocking family. For any other matching, at least one suitable family is weakly blocking.*

Proof. It is straightforward to verify that $\mathcal{F}_{R6\setminus a_t}$ is a strongly stable matching for $R6 \setminus a_t$, so no suitable family in $\overline{R6} \setminus a_t$ can weakly block $\mathcal{F}^* \supseteq \mathcal{F}_{R6\setminus a_t}$. Moreover, no agent x of $C_{R6\setminus a_t}$ can be involved in a non-suitable weakly blocking family either, since x has a suitable partner in $\mathcal{F}^*$.

Suppose that $\mathcal{F}'$ is a matching of $\overline{R6} \setminus a_t$ which is not a superset of $\mathcal{F}_{R6\setminus a_t}$. As in the proof of Lemma 5, we use the fact that if a suitable family is not in $\mathcal{F}'$, then the successor suitable family is either in $\mathcal{F}'$ or weakly blocking. Therefore, if we do not include four from the seven suitable families of $\overline{R6} \setminus a_t$ in a matching then one of them would be weakly blocking. □

The NP-completeness proof

The reduction we describe in this section again begins with an instance of Restricted SMTI, only we assume without loss of generality the role of the men and women of the instance to be "reversed". To be precise, we assume a given instance of Restricted SMTI I that its vertex set $((A_1 \cup A_2) \cup B)$ consists of a set $A_1 = \{a_1, a_2, \ldots, a_{n_1}\}$ of men with strictly ordered preference lists, and $A_2 = \{a_1^T, a_2^T, \ldots, a_{n_2}^T\}$ of men with preference lists consisting of a single tie of length 2, and $n_1 + n_2 = n$. The set $B = \{b_1, b_2, \ldots, b_n\}$ consists entirely of women with strictly ordered preference lists.

Given an instance I of Restricted SMTI as defined above, we create an instance I' of cyclic 3DSM. First we create a *proper instance* I_p' of cyclic 3DSMI as a subinstance of I' with agents $M_p \cup W_p \cup D_p$ in the following way.

First we create a set W_p of n women $\{w_1, w_2, \ldots, w_n\}$ such that the preference list of woman w_j is a single entry, dog $d_j \in D_p$. The preference list of d_j is such that if $P(b_j)[l] = a_i$, then $P(d_j)[l] = m_i$, otherwise if $P(b_j)[l] = a_i^T$, then $P(d_j)[l] = m_{i,j}'$ for $1 \leq l \leq r$, where r is the length of b_j's list. So the preference list of dog d_j is essentially the "same" as that of woman b_j, only with men in M_p rather than A.

For each man $a_i \in A_1$, create a man $m_i \in M_p$, such that if $P(a_i)[l] = b_j$, then let $P(m_i)[l] = w_j$ for $1 \leq l \leq r$, where r is the length of a_i's list. So the preference list of man m_i is essentially the "same" as that of man a_i. For each man $a_i^T \in A_2$, with a preference list consisting of a single tie of length two, say (b_r, b_s), we create five men $m_i^T, m'_{i,r}, m''_{i,r}, m'_{i,s}, m''_{i,s}$, four women $w'_{i,r}, w''_{i,r}, w'_{i,s}, w''_{i,s}$ and four dogs $d'_{i,r}, d''_{i,r}, d'_{i,s}, d''_{i,s}$ where the preference list of m_i^T contains $w'_{i,r}$ and $w'_{i,s}$ in an arbitrary order, and the other preference lists are as shown below.

$$
\begin{array}{lll}
m'_{i,r} : w'_{i,r} \quad w_r & \qquad w'_{i,r} : d'_{i,r} \quad d''_{i,r} & \qquad d'_{i,r} : m''_{i,r} \quad m_i^T \\
m''_{i,r} : w''_{i,r} & \qquad w''_{i,r} : d''_{i,r} \quad d'_{i,r} & \qquad d''_{i,r} : m'_{i,r} \quad m''_{i,r} \\
m'_{i,s} : w'_{i,s} \quad w_s & \qquad w'_{i,s} : d'_{i,s} \quad d''_{i,s} & \qquad d'_{i,s} : m''_{i,s} \quad m_i^T \\
m''_{i,s} : w''_{i,s} & \qquad w''_{i,s} : d''_{i,s} \quad d'_{i,s} & \qquad d''_{i,s} : m'_{i,s} \quad m''_{i,s}
\end{array}
$$

We also add these agents to M_p, W_p and D_p, respectively. Note that in I'_p every set of agents has the same cardinality: $n_p = |M_p| = |W_p| = |D_p| = n + 4n_2$. The notions of *proper agent, proper partner* and *proper family* are defined in the obvious way.

The *additional part* of instance I' contains three subinstances. The *suitable part* of I' is the disjoint union of $3n_p$ copies of $R6$, such that the ith copy of $R6$, denoted $R6_i$, incorporates the ith agent of I'_p, as described in the previous reduction in the proof of Theorem 1 (we omit the full description of this process again). The new agents are referred to as *additional agents*.

Let $\mathcal{F}_s = \cup_{i \in \{1,\dots 3n_p\}} \mathcal{F}_{R6_i \setminus a_i}$ be the so-called *suitable matching* of the additional part, where a_i is the proper agent of $R6_i$. We call the set $C = \cup_{i \in \{1,\dots 3n_p\}} C_{R6_i \setminus a_i}$ *covered additional agents*, as these additional agents are covered by $\mathcal{F}_s$, and we call the set $U = \cup_{i \in \{1,\dots 3n_p\}} U_{R6_i \setminus a_i}$ *uncovered additional agents*, as these additional agents are not covered by $\mathcal{F}_s$.

The *fitting part* of I' is constructed on U as follows. Note that U has equal numbers of men, women and dogs. The fitting part consists of disjoint families that covers U, so that every agent has exactly one agent in his/her/its list, i.e. the fitting part is a complete matching of U, denoted by $\mathcal{F}_f$.

Finally, the *dummy part* is obtained by an arbitrary extension of the preference lists, so that by putting together the four subinstances, the proper and the three additional parts, we get the complete instance I'. The preferences of the agents over the partners in different parts respect the order in which we defined these parts: the list of a proper agent contains the proper partners first, then the suitable partners, and finally the dummy partners; the list of a covered additional agent contains the suitable partners first, then the dummy partners; the list of an uncovered additional agent contains the suitable partners first, then the fitting partner, and finally the dummy partners.

First we show that there is a one-to-one correspondence between the complete stable matchings of I and the complete strongly stable matchings of I'_p. The stability is preserved via the following one-to-one correspondence between the complete matchings of I and complete matchings of I':

$$(a_i, b_j) \in \mathcal{M} \iff (m_i, w_j, d_j) \in \mathcal{F}_p$$

$$(a_i^T, b_s) \in \mathcal{M} \iff (m_i^T, w'_{i,s}, d'_{i,s}), (m''_{i,s}, w''_{i,s}, d''_{i,s}), (m'_{i,s}, w_s, d_s) \in \mathcal{F}_p$$

$$(a_i^T, b_s) \notin \mathcal{M} \iff (m'_{i,s}, w'_{i,s}, d''_{i,s}), (m''_{i,s}, w''_{i,s}, d'_{i,s}) \in \mathcal{F}_p$$

Lemma 7. *A complete matching $\mathcal{M}$ of I is stable if and only if the corresponding complete matching $\mathcal{F}_p$ of I'_p is strongly stable.*

Proof. As a man a_i^T cannot belong to a blocking pair in I, it may be verified that his corresponding copy m_i^T cannot belong to a weakly blocking family in I_p either. Therefore, it is enough to show that a pair (a_i, b_j) is blocking for $\mathcal{M}$ if and only if the corresponding family (m_i, w_j, d_j) is blocking for $\mathcal{F}_p$. But this is obvious, because the preference lists of a_i and m_i are essentially the same, and the preference lists of b_j and d_j are also essentially the same. □

Now, given a matching $\mathcal{M}$ of I let us create the corresponding matching $\mathcal{F}$ of I' by adding $\mathcal{F}_s$ and $\mathcal{F}_f$ to $\mathcal{F}_p$, so $\mathcal{F} = \mathcal{F}_p \cup \mathcal{F}_s \cup \mathcal{F}_f$.

Lemma 8. *The instance I admits a complete stable matching $\mathcal{M}$ if and only if the reduced instance I' admits a strongly stable matching $\mathcal{F}$, where $\mathcal{F}$ is the corresponding matching of $\mathcal{M}$.*

Proof. Suppose that we have a complete stable matching $\mathcal{M}$ of I, and $\mathcal{F}$ is the corresponding matching in I'. Lemma 7 implies that every proper agent has a proper partner in $\mathcal{F}$ and no proper family is weakly blocking. Therefore, no proper agent can be involved in any weakly blocking family either. By construction of $\mathcal{F}_s$, every covered additional agent has a suitable partner in $\mathcal{F}$ and by Lemma 6, no suitable family is weakly blocking. Therefore, no such agent can be part of any weakly blocking family. Finally, every uncovered additional agent has a fitting partner in $\mathcal{F}$, so these agent cannot form a weakly blocking family either, since an uncovered additional agent prefers only suitable partners to fitting partners, which cannot be involved in a weakly blocking family. Hence $\mathcal{F}$ is strongly stable.

In the other direction, suppose that $\mathcal{F}$ is a strongly stable matching of I'. Every proper agent must have a proper partner, since otherwise if a_t had no proper partner in $\mathcal{F}$, then $\overline{R6}_t$ would contain a suitable weakly blocking family by Lemma 5. So the corresponding matching $\mathcal{M}$ in I is complete. The stability of $\mathcal{M}$ is a consequence of Lemma 7. Finally, we note that the additional agents must be matched in the unique strongly stable way in $\mathcal{F}$, namely, the covered additional agents must be covered by matching $\mathcal{F}_s$ by Lemma 6, and the uncovered additional agents must be covered by $\mathcal{F}_f$, since otherwise a fitting family would weakly block $\mathcal{F}$. Therefore, we have a one-to-one correspondence as was claimed. □

Theorem 2. *Determining the existence of a strongly stable matching in a given instance of cyclic* 3DSM *is NP-complete.*

4 Stable exchanges with restrictions

Problem definition

Given a simple digraph $D = (V, A)$, where V is the set of agents, suppose that each agent has exactly one indivisible good, and $(i, j) \in A$ if the good of agent j is suitable for agent i. An *exchange* is a permutation π of V such that, for each $i \in V$, $i \neq \pi(i)$ implies $(i, \pi(i)) \in A$. Alternatively, an exchange can be considered as a disjoint packing of directed cycles in D.

Let each agent have strict preferences over the goods, that are suitable for him. These orderings can be represented by preference lists. In an exchange π, the agent i receives the good of his *successor*, $\pi(i)$; therefore the agent i prefers an exchange π to another exchange σ if he prefers $\pi(i)$ to $\sigma(i)$. An exchange π is *stable* if there is no *blocking coalition* B, i.e. a set B of agents and a permutation σ of B where every agent $i \in B$ prefers σ to π. An exchange is *strongly stable* is there exists no *weakly blocking coalition* B with a permutation σ of B where for every agent $i \in B$, either $\sigma(i) = \pi(i)$ or i prefers σ to π, and $\sigma(i) \neq \pi(i)$ for at least one agent $i \in B$.

Complexity results about stable exchanges

Shapley and Scarf [25] showed that the stable exchange problem is always solvable and a stable exchange can be found in polynomial time by the Top Trading Cycle (TTC) algorithm, proposed by Gale. Moreover, Roth and Postlewaite [18] proved that the exchange obtained by the TTC algorithm is strongly stable and this is the only such solution. We note that they considered this problem as a so-called *houseswapping game*, where a *core element* corresponds to a stable solution. (For further details about these connections with Game Theory, see [3].)

In some applications the length of the possible cycles is bounded by some constant l. In this case we consider an *l-way exchange problem*. Furthermore, the size of the possible blocking coalitions can also be restricted. We say that an exchange is *b-way stable* if there exists no blocking coalition of size at most b. Because of some applications, the most relevant problems are for constants 2 and 3. Henceforth we also refer to "2-way" as "pairwise" in the context of cycle lengths and blocking coalitions sizes.

For $l = b = 2$, the pairwise stable pairwise exchange problem is in fact, equivalent to the *stable roommates problem*. Therefore, a stable solution may not exist [9], but there is a polynomial-time algorithm that finds a stable solution if one does exist [11] or reports that none exists. For $l = b = 3$, the 3-way stable 3-way exchange problem is NP-hard, even for three-sided directed graphs, as is stated by the following theorem.

Theorem 3. *The 3-way stable 3-way exchange problem for tripartite directed graphs is equivalent to the cyclic 3DSMI problem, and is therefore NP-complete.*

Finally, we note that Irving [12] proved recently that the stable pairwise exchange and the 3-way stable pairwise exchange problems are NP-hard. The pairwise stable 3-way exchange problem is open. This particular problem can be a relevant regarding the application of kidney exchanges, next described.

Kidney exchange problem

Living donation is the most effective treatment that is currently known for kidney failure. However a patient who requires a transplant may have a willing donor who cannot donate to them for immunological reasons. So these incompatible patient-donor pairs may want to exchange kidneys with other pairs. Kidney exchange programs have already been established in several countries such as the Netherlands [13] and the USA [20].

In most of the current programs the goal is to maximise the number of patients that receive a suitable kidney in the exchange [21, 22, 23, 1] by regarding only the eligibility of the grafts. Some more sophisticated variants consider also the difference between suitable kidneys. Sometimes the "total benefit" is maximised [24], whilst other models [19, 6, 7, 4] require first the stability of the solution under various criteria.

The length of the cycles in the exchanges is bounded in the current programs, because all operations along a cycle have to be carried out simultaneously. Most programs allow only pairwise exchanges. But sometimes 3-way exchanges are also possible, like in the New England Program [16] and in the National Matching Scheme of the UK [27]. In these kind of applications, if one considers stability as the first priority of the solution, then we obtain a 3-way stable 3-way exchange problem, where the incompatible patient-donor pairs are the agents and their preferences are determined according to the special parameters of the suitable kidneys.

5 Further questions

For cyclic 3DSMI, the smallest instance that admits no stable matching given here satisfies $n = 6$. Is there an even smaller counterexample?

The main questions that remain unsolved are (i) whether there exists an instance of cyclic 3DSM that admits no stable matching, and (ii) whether there is a polynomial-time algorithm to find such a matching or report that none exists, given an instance of cyclic 3DSM.

References

[1] D. J. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter-exchange markets: Enabling nationwide kidney exchanges. *In Proceedings of ACM-EC 2007: the Eighth ACM Conference on Electronic Commerce*, 2007.

[2] A. Alkan. Non-existence of stable threesome matchings. *Mathematical Social Sciences*, 16:207–209, 1988.

[3] P. Biró. *The stable matching problem and its generalizations: an algorithmic and game theoretical approach.* PhD thesis, Budapest University of Technology and Economics, 2007.

[4] P. Biró and K. Cechlárová. Inapproximability of the kidney exchange problem. *Inf. Process. Lett.*, 101(5):199–202, 2007.

[5] E. Boros, V. Gurvich, S. Jaslar, and D. Krasner. Stable matchings in three-sided systems with cyclic preferences. *Discrete Mathematics*, 289:1–10, 2004.

[6] K. Cechlárová, T. Fleiner, and D. F. Manlove. The kidney exchange game. *Proc. SOR'05*, Eds. L. Zadik-Stirn, S. Drobne:77–83, 2005.

[7] K. Cechlárová and V. Lacko. The kidney exchange game: How hard is to find a donor? *IM Preprint*, 4/2006, 2006.

[8] K. Eriksson, J. Sjöstrand, and P. Strimling. Three-dimensional stable matching with cyclic preferences. *Math. Social Sci.*, 52(1):77–87, 2006.

[9] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *Amer. Math. Monthly*, 69(1):9–15, 1962.

[10] Chien-Chung Huang. Two's company, three's a crowd: stable family and threesome roommate problems. In *Algorithms—ESA 2007*, volume 4698 of *Lecture Notes in Comput. Sci.*, pages 558–569. Springer, Berlin, 2007.

[11] R.W. Irving. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms*, 6:577–595, 1985.

[12] R.W. Irving. The cycle-roommates problem: a hard case of kidney exchange. *Information Processing Letters*, 103:1–4, 2007.

[13] K. M. Keizer, M. de Klerk, B. J. J. M. Haase-Kromwijk, and W. Weimar. The Dutch algorithm for allocation in living donor kidney exchange. *Transplantation Proceedings*, 37:589–591, 2005.

[14] D.E. Knuth. *Mariages Stables.* Les Presses de L'Université de Montréal, 1976.

[15] D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002.

[16] New England Program for Kidney Exchange. *http://www.nepke.org.*

[17] C. Ng and D. S. Hirschberg. Three-dimensional stable matching problems. *SIAM J. Discrete Math.*, 4(2):245–252, 1991.

[18] A. E. Roth and A. Postlewaite. Weak versus strong domination in a market with indivisible goods. *J. Math. Econom.*, 4(2):131–137, 1977.

[19] A. E. Roth, T. Sönmez, and U. M. Ünver. Kidney exchange. *J. Econom. Theory*, 119:457–488, 2004.

[20] A. E. Roth, T. Sönmez, and U. M. Ünver. A kidney exchange clearinghouse in New England. *American Economic Review, Papers and Proceedings*, 95(2):376–380, 2005.

[21] A. E. Roth, T. Sönmez, and U. M. Ünver. Pairwise kidney exchange. *J. Econom. Theory*, 125(2):151–188, 2005.

[22] A. E. Roth, T. Sönmez, and U. M. Ünver. Coincidence of wants in markets with compatibility based preferences. *American Economic Review*, 97(3):828–851, 2007.

[23] S. L. Saidman, A. E. Roth, T. Sönmez, U. M. Ünver, and S. L. Delmonico. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation*, 81(5):773–782, 2006.

[24] S. L. Segev, S. E. Gentry, D. S. Warren, B. Reeb, and R. A. Montgomery. Kidney paired donation and optimizing the use of live donor organs. *J. Am. Med. Assoc.*, 293:1883–1890, 2005.

[25] L. S. Shapley and H. E. Scarf. On cores and indivisibility. *J. Math. Econom.*, 1(1):23–37, 1974.

[26] A. Subramanian. A new approach to stable matching problems. *SIAM Journal on Computing*, 23(4):671–700, 1994.

[27] UK Transplant. *http://www.uktransplant.org.uk.*

Approximating Matches Made in Heaven

Ning Chen* Nicole Immorlica† Anna R. Karlin*
Mohammad Mahdian‡ Atri Rudra§

Abstract

Motivated by applications in online dating and kidney exchange, we study a stochastic matching problem in which we have a random graph G given by a node set V and probabilities $p(i,j)$ on all pairs $i,j \in V$ representing the probability that edge (i,j) exists. Additionally, each node has an integer weight $t(i)$ called its patience parameter. Nodes represent agents in a matching market with dichotomous preferences, i.e., each agent finds every other agent either *acceptable* or *unacceptable* and is indifferent between all acceptable agents. The goal is to maximize the welfare, or produce a matching between acceptable agents of maximum size. Preferences must be solicited based on probabilistic information represented by $p(i,j)$, and agent i can be asked at most $t(i)$ questions regarding his or her preferences.

A stochastic matching algorithm iteratively probes pairs of nodes i and j with positive patience parameters. With probability $p(i,j)$, an edge exists and the nodes are irrevocably matched. With probability $1-p(i,j)$, the edge does not exist and the patience parameters of the nodes are decremented. We give a simple greedy strategy for selecting probes which produces a matching whose cardinality is, in expectation, at least a quarter of the size of this optimal algorithm's matching. We additionally show that variants of our algorithm (and our analysis) can handle more complicated constraints, such as a limit on the maximum number of rounds, or the number of pairs probed in each round.

1 Introduction

Matching is a fundamental primitive of many markets including job markets, commercial markets, and even dating markets [3, 4, 5, 14, 15, 16]. While matching is a well understood graph-theoretic concept, its stochastic variants are considerably less well-developed. Yet stochastic variants are precisely the relevant framework for most markets which incorporate a degree of uncertainty regarding the preferences of the agents. In this paper we study a

*University of Washington, Seattle, WA. Email: {ning,karlin}@cs.washington.edu
†CWI, Amsterdam, Netherlands. Email: nickle@alum.mit.edu
‡Yahoo! Research, Santa Clara, CA. Email: mahdian@alum.mit.edu
§University at Buffalo, State University of New York, Buffalo, NY. Email: atri@cse.buffalo.edu

stochastic variant of matching motivated by applications in the kidney exchange and online dating markets, or more generally, for matching markets with dichotomous preferences in which each agent finds every other agent either *acceptable* or *unacceptable* and is indifferent between acceptable agents (see, e.g., [6]). The basic stochastic matching problem, which is the main focus of this paper, can be stated as follows:

> Let G be a random undirected graph given by a node set V (representing agents in the matching market) and a probability $p(i, j)$ on any pair i, j of nodes, representing the probability that an edge exists between that pair of nodes (i.e., the probability that the corresponding agents find each other acceptable). Whether or not there is an edge between a pair of nodes is not revealed to us unless we *probe* this pair (solicit the preference information from the relevant agents). Upon probing a pair, if there is an edge between them, they are matched and are removed from the graph. In other words, when a pair (i, j) is probed, a coin is flipped with probability $p(i, j)$. Upon heads, the pair is matched and leaves the system. Also, for every vertex i, we are given a number $t(i)$ called the *patience parameter* of i, which specifies the maximum number of failed probes i is willing to participate in.
>
> The goal is to maximize the welfare, i.e., design a probing strategy to maximize the expected number of matches.

The above formulation of the problem is similar in nature to the formulation of other stochastic optimization problems such as stochastic shortest path [10, 7] and stochastic knapsack [8]. The stochastic matching problem is an exponential-sized Markov Decision Process (MDP) and hence has an optimal dynamic program, also exponential. Our goal is to approximate the expected value of this dynamic program in polynomial time. We show that a simple non-adaptive greedy algorithm that runs in near-linear time is a 4-approximation (Section 3). The algorithm simply probes edges in order of decreasing probability. Our algorithm is practical, intuitive, and near-optimal. Interestingly, the algorithm need not even know the patience parameters, but just which edges are more probable.

It is easy to see that the above greedy algorithm is a good approximation when the patience parameters are all one or all infinite: when the patience parameters are all one, the optimal algorithm clearly selects a maximum matching and so the maximal matching selected by the greedy algorithm is a 2-approximation; when the patience parameters are all infinite, for any instantiation of the coin flips, the greedy algorithm finds a maximal matching and hence is a 2-approximation to the (ex-post) maximum matching. To prove that the greedy algorithm is a constant approximation in general, we can no longer compare our performance to the expected size of the maximum matching as the gap between the expected size of the maximum matching and the expected value of the optimum algorithm may be larger than any constant. Instead, we compare the decision tree of the greedy algorithm to the decision tree of the optimum algorithm. Using induction on the graph as well as a careful charging scheme, we are able to see that the greedy algorithm is a 4-approximation for general patience parameters.

We also show that our algorithm and analysis can be adapted to handle more complicated constraints (Section 4). In particular, if probes must be performed in a limited number of rounds, each round consisting of probing a matching, a natural generalization of the greedy algorithm gives a 6-approximation in the uniform probability case. We can also generalize the algorithm to a case where we are given a bound on the maximum number of edges probed in each round.

1.1 Motivation

In addition to being an innately appealing and natural problem, the stochastic matching problem has important applications. We outline here two applications to kidney exchange and online dating.

Kidney Exchange. Currently, there are 98,167 people in need of an organ in the United States. Of these, 74,047 patients are waiting for a kidney.[1] Every healthy person has two kidneys, and only needs one kidney to survive. Hence it is possible for a living friend or family of the patient to donate a kidney to the patient. Unfortunately, not all patients have compatible donors. At the recommendation of the medical community [12, 13], in year 2000 the United Network for Organ Sharing (UNOS) began performing *kidney exchanges* in which two incompatible patient/donor pairs are identified such that each donor is compatible with the other pair's patient. Four simultaneous operations are then performed, exchanging the kidneys between the pairs in order to have two successful transplants.

To maximize the total number of kidney transplants in the kidney exchange program, it is important to match the maximum number of pairs. This problem can be phrased as that of maximum matching on graphs in which the nodes represent incompatible pairs and the edges represent possible transplants based on medical tests [15, 16]. There are three main tests which indicate the likelihood of successful transplants. The first two tests, the blood-type test and the antibody screen, compare the blood of the recipient and donor. The third test, called *crossmatching* combines the recipient's blood serum with some of the donor's red blood cells and checks to see if the antibodies in the serum kill the cells. If this happens (the crossmatch is *positive*), then the transplant can not be performed. If this doesn't happen (the crossmatch is *negative*), then the transplant may be performed.[2]

Of course, the feasibility of a transplant can only be determined after the final crossmatch test. As this test is time-consuming and must be performed close to the surgery date [2, 1], it is infeasible to perform crossmatch tests on all nodes in the graph. Furthermore, due to incentives facing doctors, it is important to perform a transplant as soon as a pair with negative crossmatch tests is identified. Thus the edges are really stochastic; they only reflect the *probability*, based on the initial two tests and related factors, that an exchange is possible.

[1] Data retrieved on November 19th, 2007 from United Network for Organ Sharing (UNOS) — The Organ Procurement and Transplantation Network (OPTN), `http://www.optn.org/data`.

[2] Recent advances in medicine actually allow positive crossmatch transplants as well, but these are significantly more risky.

Based on this information alone, edges must be selected and, upon a negative crossmatch test, the surgery performed. Hence the matching problem is actually a stochastic matching problem. The patience parameters in the stochastic matching problem can be used to model the unfortunate fact that patients will eventually die without a successful match.

Online Dating. Another relevant marketplace for stochastic matching is the online dating scene, the second-largest paid-content industry on the web, expected to gross around $600 million in 2008 [9]. In many online dating sites, most notably eHarmony and Just A Lunch, users submit profiles to a central server. The server then estimates the compatibility of a couple and sends plausibly compatible couples on blind dates (and lately, possibly virtual blind dates). The purported goal of these sites is to create as many happily married couples as possible.

Again, this problem may be modeled as a stochastic matching problem. Here, the people participating in the online match-making program are the nodes in the graph. From the personal characteristics of these individuals, the system deduces for each pair a probability that they are a good match. Whether or not a pair is actually successful can only be known if they are sent on a date. In this case, if the pair is a match, they will immediately leave the program. Also, each person is willing to participate in at most a given number of unsuccessful dates before he/she runs out of patience and leaves the match-making program. The online dating problem is to design a schedule for dates to maximize the expected number of matched couples.

2 Preliminaries

The stochastic matching problem can be represented by a random graph $G = (V, E)$, where for each pair (α, β) of vertices, there is an undirected edge between α and β with a probability $p(\alpha, \beta) \in [0, 1]$.[3] For the rest of the paper, w.l.o.g. we will assume that E contains exactly the pairs that have positive probability. These probabilities are all independent. Additionally, for each vertex $\gamma \in V$ a number $t(\gamma)$ called the *patience parameter* of γ is given. The existence of an edge between a pair of vertices of the graph is only revealed to us after we *probe* this pair. When a pair (α, β) is probed, a coin is flipped with probability $p(\alpha, \beta)$. Upon heads, the pair is matched and is removed from the graph. Upon tails, the patience parameter of both α and β are decremented by one. If the patience parameter of a node reaches 0, this node is removed from the graph. This guarantees that each vertex γ can be probed at most $t(\gamma)$ times. The problem is to design (possibly adaptive) strategies to probe pairs of vertices in the graph such that the expected number of matched pairs is maximized.

An instance of our problem is thus a tuple (G, t). For a given algorithm ALG, let $\mathbf{E}_{\texttt{ALG}}(G, t)$ (or $\mathbf{E}_{\texttt{ALG}}(G)$ for simplicity, when t is clear from the context) be the expected number of pairs

[3]Note that here we do not impose any constraint that the graph G should be bipartite. In settings such as heterosexual dating where such a constraint is natural, it can be imposed by setting the probabilities between vertices on the same side to zero.

matched by ALG, where the expectation is over the realizations of probes and (possible) coin tosses of the algorithm itself.

Decision Tree Representation. For any deterministic algorithm ALG and any instance (G, t) of the problem, the entire operation of ALG on (G, t) can be represented as an (exponential-sized) *decision tree* $T_{\texttt{ALG}}$. The root of $T_{\texttt{ALG}}$, r, represents the first pair $e = (\alpha, \beta) \in E$ probed by ALG. The *left* and the *right* subtrees of r represent *success* and *failure* for the probe to (α, β), respectively. In general, each node of this tree corresponds to a probe and the left and the right subtrees correspond to the respective success or failure.

For each node $v \in T_{\texttt{ALG}}$, a corresponding sub-instance (G_v, t_v) of the problem can be defined recursively as follows: The root r corresponds to the initial instance (G, t). If a node v that represents a probe to a pair (α, β) corresponds to (G_v, t_v),

- the left child of v corresponds to $(G_v \setminus \{\alpha, \beta\}, t_v)$, and
- the right child of v corresponds to $(G_v \setminus \{(\alpha, \beta)\}, t'_v)$, where $G_v \setminus \{(\alpha, \beta)\}$ denotes the instance obtained from G_v by setting the probability of the edge (α, β) to zero, and $t'_v(\alpha) = t_v(\alpha) - 1$, $t'_v(\beta) = t_v(\beta) - 1$ and $t'_v(\gamma) = t_v(\gamma)$ for any other vertex γ.

For each node $v \in T_{\texttt{ALG}}$, let T_v be the subtree rooted at v. Let $T_{L(v)}$ and $T_{R(v)}$ be the left and right subtree of v, respectively. Observe that T_v essentially defines an algorithm ALG′ on the sub-instance (G_v, t_v) corresponding to v. That is, the first pair probed by ALG′ is the pair represented by v. If the probe succeeds, ALG′ goes to $T_{L(v)}$. Otherwise, ALG′ goes to $T_{R(v)}$. Define $\mathbf{E}_{\texttt{ALG}}(T_v)$ to be the expected value generated by the algorithm corresponding to ALG′, i.e. $\mathbf{E}_{\texttt{ALG}}[T_v] = \mathbf{E}_{\texttt{ALG}'}(G_v, t_v)$.

The stochastic matching problem can be viewed as the problem of computing the optimal policy in an exponential-sized Markov Decision Process (for more details on MDPs, see the textbook by Puterman [11]). The states of this MDP correspond to subgraphs of G that are already probed, and the outcome of these probes. The actions that can be taken at a given state correspond to the choice of the next pair to be probed. Given an action, the state transitions probabilistically to one of two possible states, one corresponding to a success, and the other corresponding to a failure in the probe. We denote by OPT the optimal algorithm, i.e., the solution of this MDP. Note that we can assume without loss of generality that OPT is deterministic, and therefore, a decision tree $T_{\texttt{OPT}}$ representing OPT can be defined as described above. Observe that by definition, for any node v of this tree, if the probability of reaching v from the root is non-zero, the algorithm defined by T_v must be the optimal for the instance (G_v, t_v) corresponding to v. To simplify our arguments, we assume without loss of generality that the algorithm defined by T_v is optimal for (G_v, t_v) for *every* $v \in T_{\texttt{OPT}}$, even for nodes v that have probability zero of being reached. Note that such nodes can exist in $T_{\texttt{OPT}}$, since OPT can probe edges of probability 1, in which case the corresponding right subtree is never reached.

Note that it is not even clear that the optimal strategy OPT can be described in polynomial space. Therefore, one might hope to use other benchmarks such as the optimal offline solution (i.e., the expected size of maximum matching in G) as an upper bound on OPT. However, as we show in the full version, the gap between OPT and the optimal offline solution can be larger than any constant.

3 Greedy Algorithm

We consider the following greedy algorithm.

```
GREEDY.
1.  Sort all edges in E by probabilities, say, p(e_1) ≥ p(e_2) ≥ ··· ≥ p(e_m)
      (ties are broken arbitrarily)
2.  For i = 1,...,m
3.    if the two endpoints of e_i are available, probe e_i
```

Our main result is as follows.

Theorem 3.1. *For any instance graph* (G, t)*,* GREEDY *is a 4-approximation to the optimal algorithm, i.e.* $\mathbf{E}_{\mathtt{OPT}}(G, t) \leq 4 \cdot \mathbf{E}_{\mathtt{GREEDY}}(G, t)$.

In the rest of this section, we will sketch the proof of Theorem 3.1. The proof is inductive and is based on carefully charging the value obtained at different nodes of $T_{\mathtt{OPT}}$ to $T_{\mathtt{ALG}}$. We will begin by establishing two lemmas that will be useful for the proof.

Lemma 3.1. *For any node* $v \in T_{\mathtt{OPT}}$*,* $\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right) \leq \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right) \leq 1 + \mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)$.

Proof. Let the node v in $T_{\mathtt{OPT}}$ correspond to probing the edge $e = (\alpha, \beta) \in E$. Since OPT reaches $T_{L(v)}$ if the probe to e succeeds and reaches $T_{R(v)}$ if the probe to e fails, $T_{L(v)}$ defines a valid algorithm on the instance $(G_{R(v)}, t_{R(v)})$ corresponding to $R(v)$. By the optimality of OPT on every subtree, we have $\mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right) \leq \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right)$.

On the other hand, since $T_{R(v)}$ is a valid algorithm for the sub-instance (G_v, t_v) corresponding to v,

$$\mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right) \leq \mathbf{E}_{\mathtt{OPT}}(T_v) = p(e) \cdot \left(1 + \mathbf{E}_{\mathtt{OPT}}\left(T_{L(v)}\right)\right) + (1 - p(e)) \cdot \mathbf{E}_{\mathtt{OPT}}\left(T_{R(v)}\right),$$

where the equality follows from the problem definition. The above implies that $\mathbf{E}_{\mathtt{OPT}}(T_{R(v)}) \leq 1 + \mathbf{E}_{\mathtt{OPT}}(T_{L(v)})$ as $p(e) > 0$. □

Lemma 3.2. *For any node* $v \in T_{\mathtt{OPT}}$*, assume* v *represents the edge* $e = (\alpha, \beta) \in E$*, and let* $p = p(\alpha, \beta)$ *be the probability of* e*. If we increase the probability of* v *to* $p' > p$ *in* $T_{\mathtt{OPT}}$*, then* $\mathbf{E}_{\mathtt{OPT}}(T_{\mathtt{OPT}})$ *will not decrease.*

Note that the claim does not mean we increase the probability of edge e in graph G. It only says for a particular probe of e in $T_{\mathtt{OPT}}$, which corresponds to node v in the claim, if the probability of e is increased, the expected value of OPT will not decrease.

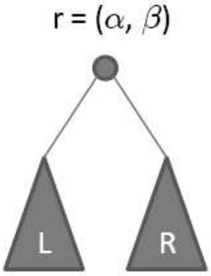


Figure 1: Greedy tree $T_{\texttt{GREEDY}}$

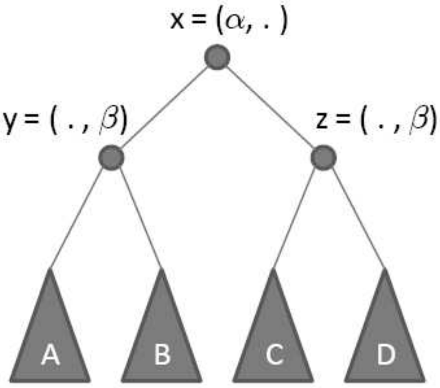


Figure 2: Optimum tree $T_{\texttt{OPT}}$

Proof of the Lemma 3.2. By Lemma 3.1 and the assumption that $p' > p$,

$$(p' - p)\mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right) \leq (p' - p)\left(1 + \mathbf{E}_{\texttt{OPT}}\left(T_{L(v)}\right)\right),$$

which implies that

$$p \cdot \left(1 + \mathbf{E}_{\texttt{OPT}}\left(T_{L(v)}\right)\right) + (1 - p) \cdot \mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right) \leq p' + p' \cdot \mathbf{E}_{\texttt{OPT}}\left(T_{L(v)}\right) + (1 - p') \cdot \mathbf{E}_{\texttt{OPT}}\left(T_{R(v)}\right).$$

The proof is complete by noting that the LHS and RHS of the above inequality corresponds to $\mathbf{E}_{\texttt{OPT}}\left(T_v\right)$ before and after the probability of v is increased to p'. □

These two lemmas provide the key ingredients of our proof. To get an idea of the proof, imagine that the first probe of the greedy algorithm is to edge (α, β) represented by node r at the root of $T_{\texttt{GREEDY}}$ as in Figure 1 and suppose that $T_{\texttt{OPT}}$ is as in Figure 2. Let p_r be the probability of success of probe (α, β). Note the algorithm ALG_1 defined by subtree A in $T_{\texttt{OPT}}$ is a valid algorithm for the left subtree of greedy (since the optimum algorithm has already matched nodes α and β upon reaching subtree A, all probes in subtree A are valid probes for the left-subtree of $T_{\texttt{GREEDY}}$). Furthermore, ALG_1 achieves the same value, in expectation, as the optimum algorithm on subtree A. Similarly the algorithm ALG_2 defined by subtree D in $T_{\texttt{OPT}}$ is a valid algorithm for the right subtree of greedy *except* ALG_2 may perform a probe to (α, β). Thus we define a secondary (randomized) algorithm ALG_2' which follows ALG_2 but upon reaching a probe to (α, β) simply flips a coin with probability p_r to decide which subtree to follow and does not probe the edge. Hence ALG_2' is a valid algorithm for the right subtree of greedy, and gets the same value as the optimum algorithm on subtree D minus a penalty of p_r for the missed probe to (α, β). The value of ALG_1 and ALG_2' on the left and right subtree of $T_{\texttt{GREEDY}}$ respectively is at most the value of the optimum algorithm on those subtrees and so, by the inductive hypothesis, at most four times the value of the greedy algorithm on those subtrees. By Lemma 3.2, we can assume the probes at nodes x, y, and z in $T_{\texttt{OPT}}$ have probability p_r of success. Furthermore, we can use Lemma 3.1 to bound the value of the optimum algorithm in terms of the left-most subtree A and the right-most subtree D. With a slight abuse of notation, we use A to denote the expected value of the optimum algorithm on subtree A (and similarly, B, C, and D). Summarizing the above observations, we then get:

$$\begin{aligned}
\mathbf{E}_{\mathtt{OPT}}(G,t) &\le p_r^2(A+2)+p_r(1-p_r)(B+1)+p_r(1-p_r)(C+1)+(1-p_r)^2D \\
&\le 2p_r+p_r^2A+p_r(1-p_r)(A+1)+p_r(1-p_r)D+(1-p_r)^2D \\
&= 3p_r-p_r^2+p_rA+(1-p_r)D \\
&\le 4p_r+p_rA+(1-p_r)(D-p_r) \\
&= 4\cdot\left(p_r(1+\mathbf{E}_{\mathtt{ALG}_1})+(1-p_r)\mathbf{E}_{\mathtt{ALG}_2'}\right) \\
&\le 4\mathbf{E}_{\mathtt{GREEDY}}(G,t)
\end{aligned}$$

where the first inequality is by Lemma 3.2, the second inequality is by Lemma 3.1, and the fourth inequality is by the inductive hypothesis.

The above sketch represents the crux of the proof. To formalize the argument, we must account for generic structures of $T_{\mathtt{OPT}}$. We do this by considering "frontiers" in $T_{\mathtt{OPT}}$ representing initial probes to α and β, and then follow the general accounting scheme suggested above via slightly more complicated algebraic manipulations. The complete proof is deferred to the full version.

4 Multiple Rounds Matching

In this section, we consider a generalization of the stochastic matching problem defined in Section 2. In this generalization, the algorithm proceeds in rounds, and is allowed to probe a set of edges (which have to be matching) in each round. The additional constraint is a bound, k, on the maximum number of rounds. We show in the full version of the paper that finding the optimal strategy in this new model is NP-hard. Note that when k is large enough, the problem is equivalent to the model discussed in previous sections.

In the rest of this section, we will study approximation algorithms for the problem. By looking at the probabilities as the weights on edges, we have the following natural generalization of the greedy algorithm.

```
GREEDY_k.
1.  For each round i = 1,...,k
2.      compute the maximum weighted matching in the current graph
3.      probe all edges in the matching
```

Let $\mathtt{OPT}_k$ be the optimal algorithm under this setting. We would like to compare $\mathbf{E}_{\mathtt{GREEDY_k}}$ against $\mathbf{E}_{\mathtt{OPT_k}}$. Unfortunately, as the following example shows, with no restriction on the instance, $\mathtt{GREEDY}_k$ can be arbitrarily bad.

Example 4.1. *Consider a bipartite graph $G=(A,B;E)$ where $A=\{\alpha_1,\dots,\alpha_n\}$ and $B=\{\beta_1,\dots,\beta_n\}$. Let $\epsilon=1/n^3$. Let $p(\alpha_1,\beta_j)=\epsilon$ and $p(\alpha_i,\beta_1)=\epsilon$ for $i,j=1,\dots,n$,* *and* $p(\alpha_i, \beta_i) = \frac{\epsilon}{n-2}$ *for* $i = 2, \ldots, n$. *There are no other edges in the graph. Further, define patience* $t(\alpha_1) = t(\beta_1) = \infty$ *and* $t(\alpha_i) = t(\beta_i) = 1$ *for* $i = 2, \ldots, n$. *Consider any given* $k \leq n-1$. *Now, a maximum matching in this example is* $\{(\alpha_1, \beta_2), (\alpha_2, \beta_1), (\alpha_3, \beta_3), \ldots, (\alpha_n, \beta_n)\}$. *The expected value that* $\texttt{GREEDY}_k$ *obtains by probing this matching in the first round is* 3ϵ. *After these probes, in the next round, due to patience restriction, in the best case only edge* (α_1, β_1) *will remain. Thus,* $\texttt{GREEDY}_k$ *obtains another* ϵ, *which implies that the total expected value is at most* 4ϵ. *On the other hand, consider another algorithm which probes edges* (α_1, β_{i+1}) *and* (α_{i+1}, β_1) *for any round* $i = 1, \ldots, k$. *The revenue generated by the algorithm is at least* $2k\epsilon - 2\binom{n}{2}\epsilon^2 = \Omega(k\epsilon)$. *Thus,* $\texttt{GREEDY}_k$ *can be as bad as a factor of* $\Omega(k)$. *Note that* $\texttt{GREEDY}_k$ *is trivially a factor* k *approximation algorithm.*

However, we can still prove that $\texttt{GREEDY}_k$ is a constant-factor approximation algorithm in two important special cases: when all nodes have infinite patience, and when nodes have arbitrary patience but all non-zero probability edges of G have the same probability. Furthermore, we observe that the latter result can be used to give a logarithmic approximation for the general case of the problem. All proofs are deferred to the full version.

Theorem 4.1. *For any graph* $G = (V, E)$, $\mathbf{E}_{\texttt{OPT}_k}[G] \leq 4 \cdot \mathbf{E}_{\texttt{GREEDY}_k}[G]$, *when the patience of all vertices are infinity.*

Theorem 4.2. *Let* (G, t) *be an instance such that for all pairs* α, β *of vertices,* $p(\alpha, \beta)$ *is either 0 or* p. *Then* $\mathbf{E}_{\texttt{OPT}_k}[G] \leq 6 \cdot \mathbf{E}_{\texttt{GREEDY}_k}[G]$.

The general case. In this section, we sketch how Theorem 4.2 can be used to obtain an approximation algorithm for the general case of the multi-round stochastic matching problem with an approximation factor of $O(\log n)$. Given an instance (G, t), denote the maximum probability of an edge in this instance by $p_{\max}$. First, we note that $\mathbf{E}_{\texttt{OPT}}(G, t) \geq p_{\max}$, and therefore removing all edges that have probability less than $p_{\max}/n^2$ cannot decrease the value of $\mathbf{E}_{\texttt{OPT}}(G, t)$ by more than a constant factor. Next, we partition the set of edges of G into $O(\log n)$ classes, depending on which of the intervals $(p_{\max}/2, p_{\max}]$, $(p_{\max}/4, p_{\max}/2]$, $\ldots$, $(p_{\max}/2^{2\lceil \log_2 n \rceil}, p_{\max}/2^{2\lceil \log_2 n \rceil} - 1]$, the probability of the edge falls into. Let E_i denote the i'th class of this partition. We can write the value of $\texttt{OPT}$ as the sum of $O(\log n)$ terms, where the i'th term corresponds to the value that $\texttt{OPT}$ extracts from edges in E_i. One of these values, say the value corresponding to E_{i^*}, should be at least $\Omega(1/\log n)$ times the value of $\texttt{OPT}$. Now, we define our algorithm as follows: For every i, simulate $\texttt{GREEDY}_k$ on the instance obtained by restricting the edges of G to E_i and changing the probability of all these edges to $p_{\max}/2^i$; by running these simulations many times and computing the average, we obtain an estimate of the value of $\texttt{GREEDY}_k$ on each of these instances (this follows from concentration inequalities; details of the argument are omitted here). Pick the i that achieves the maximum value, and follow $\texttt{GREEDY}_k$ on E_i.

A Further Extension. We now briefly consider the following extension of the k-rounds model. In each round, an algorithm is only allowed to probe a matching of size at most C, where $1 \leq C \leq \lfloor |V|/2 \rfloor$ is another parameter (V is the set of vertices in the graph). Note that till now we have only considered the cases $C = 1$ and $C = \lfloor |V|/2 \rfloor$. We now very briefly sketch how the results we have seen in this section so far can be extended to this new model.

Obviously finding an optimal solution in the new model is still NP-hard. The extension of $\texttt{GREEDY}_k$ in this model is also straightforward: in each round probe the maximum weighted matching (with the constraint that the matching uses at most C edges). Theorems 4.1 and 4.2 also hold in this model. Further, it can be shown that for the arbitrary patience and probability case, $\texttt{GREEDY}_k$ is a $\Theta(\min(k, C))$-approximation algorithm.

5 Conclusion

We studied natural greedy algorithms for the stochastic matching problem with patience parameters and proved that these algorithms are constant factor approximations. A natural question to ask is if designing the optimal strategy is computationally hard (this is even unknown for infinite patience parameters). In the full version we show the following two variants are NP-hard: (i) The algorithm can probe a matching in at most k rounds (the model we studied in Section 4) and (ii) the order in which the edges need to be probed are fixed (and the algorithm just needs to decide whether to probe an edge or not). In terms of positive results, it is well known that the greedy algorithm in Section 3 for the special cases of (i) all probabilities being 1 and (ii) all patience parameters being infinity is a 2-approximation. However, we proved that the greedy algorithm is a factor 4-approximation. We conjecture that the greedy algorithm is in fact a 2-approximation even for the general stochastic matching problem.

Another interesting variant of the problem is when edges also have weights associated with them and the objective is to maximize the (expected) total weight of the matched edges. In the full version, we give an example that shows that the natural greedy algorithm has an unbounded approximation ratio. The greedy algorithm considered in Section 3 is *non-adaptive*, that is, the order of edges to probe are decided before the first probe. A natural question to ask is if there is a "gap" between the non-adaptive and *adaptive* optimal values? In the full version, we show that the adaptive optimal is strictly larger than the non-adaptive optimal.

References

[1] http://www.aakp.org/aakp-library/transplant-compatibility/index.cfm.

[2] http://www.online-family-doctor.com/medical-tests/crossmatching.html.

[3] A. Abdulkadiroglu, P.A. Pathak, and A.E. Roth. The new york city high school match. *American Economic Review*, 95(2):364–367, May 2005.

[4] A. Abdulkadiroglu, P.A. Pathak, A.E. Roth, and T. Sonmez. The boston public school match. *American Economic Review*, 95(2):368–371, May 2005.

[5] A. Abdulkadiroglu and T. Sonmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, June 2003.

[6] A. Bogomolnaia and H. Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004.

[7] M. Brand, J. Kelner, M. Mitzenmacher, and E. Nikolova. Stochastic shortest paths via quasi-convex maximization. In *Proceedings of 2006 European Symposium of Algorithms (ESA)*, 2006.

[8] B. Dean, M. Goemans, and J. Vondrak. Approximating the stochastic knapsack: the benefit of adaptivity. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 208–217, 2004.

[9] R. Epstein. The truth about online dating. *Scientific American*, February 2007.

[10] C.H. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, July 1991.

[11] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley Series in Probability and Statistics, 2005.

[12] F.T. Rapaport. The case for a living emotionally related international kidney donor exchange registry. *Transplantation Proceedings*, XVIII:5–9, 1986.

[13] L.F. Ross, D.T. Rubin, M. Siegler, M.A. Josephson, J.R. Thistlethwaite, and E.S. Woodle. Ethics of a paired-kidney-exchange program. *The New England Journal of Medicine*, CCCXXXVI:1752–1755, 1997.

[14] A.E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Economic Theory*, 92:991–1016, 1984.

[15] A.E. Roth, T. Sonmez, and U. Unver. Kidney exchange. *Quarterly Journal of Economics*, 119(2):457–488, May 2004. Originally published as NBER Paper w10002, September 2003.

[16] A.E. Roth, T. Sonmez, and U. Unver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188, December 2005. Originally published as NBER Paper w10698, August 2004.

Smith and Rawls Share a Room: Stability and Medians*

Bettina Klaus† Flip Klijn‡

Contribution to the workshop "Match-up: Matching under Preferences"
Reykjavik, July 2008

Abstract

We consider one-to-one, one-sided matching (roommate) problems in which agents can either be matched as pairs or remain single. We introduce a so-called bi-choice graph for each pair of stable matchings and characterize its structure. Exploiting this structure we obtain as a corollary the "lonely wolf" theorem and a decomposability result. The latter result together with transitivity of blocking leads to an elementary proof of the so-called stable median matching theorem, showing how the often incompatible concepts of stability (represented by the political economist Adam Smith) and fairness (represented by the political philosopher John Rawls) can be reconciled for roommate problems. Finally, we extend our results to two-sided matching problems.

JEL classification: C62, C78.

Keywords: fairness, matching, median, stability.

1 Introduction

Gale and Shapley (1962, Example 3) introduced the so-called roommate problems as follows: "An even number of boys wish to divide up into pairs of roommates." A very common extension is to allow also for odd numbers of agents and to consider the formation of pairs and singletons (rooms can be occupied either by one or by two agents). Therefore, an outcome for a roommate problem, a matching, is a partition of agents in pairs (of matched agents) and singletons. The

*We thank Elena Molis, participants of the SISL mini-conference on matching (CalTech), and a reviewer for useful comments and discussions. This paper is a shortened version of our working paper with the same name which is available at `http://ideas.repec.org/p/dgr/umamet/2008009.html` (Meteor Research Memorandum/08-009). B. Klaus thanks the Netherlands Organisation for Scientific Research (NWO) for its support under grant VIDI-452-06-013. F. Klijn's research was supported through the Spanish *Plan Nacional I+D+I* (SEJ2005-01690) and the *Generalitat de Catalunya* (SGR2005-00626 and the Barcelona Economics Program of XREA).

†Department of Economics, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; e-mail: `b.klaus@algec.unimaas.nl`

‡*Corresponding author*: Institute for Economic Analysis (CSIC), Campus UAB, 08193 Bellaterra (Barcelona), Spain; e-mail: `flip.klijn@iae.csic.es`

class of roommate problems include as special cases the well-known marriage problems (Gale and Shapley, 1962).[1]

In a roommate problem, each agent has preferences over being matched with any of the other agents and remaining unmatched. A key property of a matching is stability: a matching is stable if it satisfies individual rationality and no pair of agents that are not matched with one another prefer to be so. Gale and Shapley (1962) exhibited an unsolvable roommate problem, i.e., a roommate problem in which there is no stable matching. Instead of focusing on necessary and/or sufficient conditions on the preferences for the existence of a stable matching,[2] we directly study the class of solvable roommate problems. For a solvable roommate problem there are typically multiple stable matchings. Our quest is to single out particularly appealing stable matchings. However, before dealing with this selection problem, we introduce for any two stable matchings a so-called bi-choice graph and characterize its structure (Lemma 1). This graphical tool will allow us to provide elementary and illustrative proofs of various well-known results.

A first idea to find appealing stable matchings is to select stable matchings that maximize the number of matched agents. However, Gusfield and Irving (1989) have shown that an agent who is single at any stable matching is also single at all other stable matchings. Using the bi-choice graph structure, we provide an elementary graphical proof of this "lonely wolf" theorem (Theorem 1). The bi-choice graph structure can also be used to derive an elementary proof of the so-called decomposability lemma (Lemma 2), which turns out to be crucial for later results.

Then, since no selection can be based on the set of matched students, we try to find a stable matching that will be perceived as fair by the agents. Imagine that we ask each agent to rank all his matches in the stable matchings according to his preferences. Note that since an agent might be matched to the same agent in several stable matchings, this ranking is not strict. Clearly, we cannot always give the best match to every agent, but can we implement fairness by finding a matching that matches each agent to his l-th ranked match for some natural number l? We show that this idea of fairness or compromise is feasible if there is an odd number of stable matchings: the so-called median matching that assigns to each agent his median (ranked) match is well-defined and stable (Theorem 2). In a similar fashion, fairness is "almost" feasible if there is an even number of stable matchings (Corollary 1). Hence, stability (represented by the political economist Adam Smith[3]) and fairness (represented by the political philosopher John Rawls[4]) can be reconciled for solvable roommate problems and "Smith and Rawls (almost) share a room." This result was already stated by Sethuraman and Teo (2001). Here we provide an elementary proof that does not resort to linear programming tools.

In the second part of the paper we turn to two-sided matching problems. A marriage problem is a roommate problem where the set of agents is exogenously partitioned into two sets such that each agent can only be matched with an agent of the other set. A further generalization of marriage problems are college admissions problems (with responsive preferences[5]), where one

[1]There is a large literature on the marriage problem; see, for instance, Roth and Sotomayor (1990) and the two-sided matching bibliography on Al Roth's game theory, experimental economics, and market design page. In comparison, relatively few papers and books deal with roommate problems; some of the key references concerning roommate problems are Gusfield and Irving (1989); Tan (1991); Chung (2000); Diamantoudi et al. (2004).

[2]See, for instance, Tan (1991) and Chung (2000).

[3]Adam Smith (1723–1790) propagated the view that individuals even though interested only in their own gains will still advance public interest (Smith, 1796).

[4]John Rawls (1921–2002) discussed important aspects of fairness and justice particularly suited for economic applications (Rawls, 1971).

[5]By responsiveness (Roth, 1985), a college's preference relation over sets of students is related to its ranking of

side of the market has several positions (seats or slots). Unlike roommate problems, marriage and college admissions problems (with responsive preferences) always allow for stable matchings.

Using linear programming tools, Teo and Sethuraman (1998) and Sethuraman et al. (2006) established the existence of natural compromise mechanisms for marriage and college admissions problems, respectively. Specifically, they showed that "generalized median matchings" are well-defined and stable. More formally, if all agents order again their matches in the, say, k stable matchings from best to worst, then the map that assigns to each agent of one side of the market its l-th best match and to each agent of the other side its $(k-l+1)$-st best match constitutes a stable matching. Fleiner (2002) and Klaus and Klijn (2006) independently provided alternative, short proofs for the above mentioned generalized median result for college admissions problems based on the so-called lattice structure of the set of stable matchings. A slight adaptation of our elementary proof of the "Smith and Rawls share a room" theorem immediately leads to the stronger result of stable generalized median matchings for the (more specific) class of marriage problems (Theorem 3).[6] Essentially the same proof is also valid for college admissions problems (Theorem 4). Unfortunately, in this case the extended proof can no longer be considered elementary because the decomposability lemma for college admissions problems (Lemma 3) is not an elementary result and in addition we have to establish a (non-elementary) "transitivity of blocking" property for college admissions problems (Lemma 4).

Our paper is organized as follows. Sections 2 and 3 contain the results as discussed in detail above for roommates and two-sided matching problems, respectively. We conclude in Section 4.

2 Roommate Problems

A *roommate problem* (Gale and Shapley, 1962) is a pair $(N, (\succeq_i)_{i \in N})$ where N is a finite set of agents and, for each $i \in N$, $\succeq_i$ is a complete, transitive, and strict preference relation over N. For each $i \in N$, we interpret $\succeq_i$ as agent i's preferences over sharing a room with any of the agents in $N \backslash \{i\}$ and having a room by himself (or consuming an outside option such as living off-campus). Preferences are strict, i.e., $k \succeq_i j$ and $j \succeq_i k$ if and only if $j = k$. The indifference and strict preference relation associated with $\succeq_i$ are denoted by $\sim_i$ and $\succ_i$, respectively. A solution to a roommate problem, a *matching* μ, is a partition of N into pairs and singletons. Alternatively, we describe a matching by a function $\mu : N \to N$ of order two, i.e., for all $i \in N$, $\mu(\mu(i)) = i$. Agent $\mu(i)$ is agent i's *match*, i.e., the agent with whom he is matched to share a room (possibly himself). If $\mu(i) = i$ then we call i a single.

A *marriage problem* (Gale and Shapley, 1962) is a roommate problem $(N, (\succeq_i)_{i \in N})$ such that N is the union of two disjoint sets M and W (men and women), and each agent in M (respectively W) prefers being alone to being matched with any other agent in M (respectively W).

A matching μ is *blocked by a pair* $\{i, j\} \subseteq N$ (possibly $i = j$) if $j \succ_i \mu(i)$ and $i \succ_j \mu(j)$. If $\{i, j\}$ blocks μ, then $\{i, j\}$ is called a blocking pair for μ. A matching is *individually rational* if there is no blocking pair $\{i, j\}$ with $i = j$. A matching is *stable* if there is no blocking pair. A roommate problem is *solvable* if the set of stable matchings is non-empty. Gale and Shapley

single students in the following way: the college always prefers to add an acceptable student to any set of students (provided this does not violate the capacity constraint) and it prefers to replace any student by a better student.

[6]Cheng (2008) showed that the problem of actually *finding* certain generalized median stable matchings is NP-hard.

(1962) showed that all marriage problems are solvable and provided an unsolvable roommate problem (Gale and Shapley, 1962, Example 3).

The following is a simplified version of Gale and Shapley's example with three agents: $2 \succ_1 3 \succ_1 1$, $3 \succ_2 1 \succ_2 2$, and $1 \succ_3 2 \succ_3 3$. Clearly, the matching where all agents are singles is not stable (any two agents can block). So, assume two agents share a room. Then, the single agent is the best roommate for one of these two agents and hence a blocking pair can be formed. Tan (1991) provided a necessary and sufficient condition for the existence of a stable matching.

Smith and Rawls: Stability and Fairness

The starting point of our analysis is a solvable roommate problem. Typically there are multiple stable matchings and with choice comes the opportunity to select a particularly appealing stable matching. Before we explore this choice in order to address fairness in addition to stability, we introduce a so-called bi-choice graph for each pair of stable matchings and characterize its structure.

2.1 Bi-Choice Graphs

Let μ and μ' be stable matchings. We consider the following bi-choice graph $G(\mu, \mu') = (V, E)$. The set of vertices is the set of agents, i.e., $V = N$. The set E consists of three types of edges. Let $i, j \in N$. Then,

E1. there is a continuous directed edge from i to j, denoted by $i \bullet\!\!\longrightarrow\!\!\bullet\, j$ if $j = \mu(i) \succ_i \mu'(i)$, i.e., agent i strictly prefers his match $j = \mu(i)$ under μ to his match under μ';

E2. there is a discontinuous directed edge from i to j, denoted by $i \bullet\!\!\dashrightarrow\!\!\bullet\, j$ if $j = \mu'(i) \succ_i \mu(i)$, i.e., agent i strictly prefers his match $j = \mu'(i)$ under μ' to his match under μ;

E3. there is a (continuous) undirected edge between i and j, denoted by $i \bullet\!\!-\!\!\!-\!\!\bullet\, j$ if $j = \mu(i) \sim_i \mu'(i)$, i.e., agent i is indifferent between his match $j = \mu(i)$ under μ and his match under μ'. Note that for $j = i = \mu(i) \sim_i \mu'(i)$ we allow for an undirected edge from i to himself; we call such an edge a loop: $i \bullet\!\!\bigcirc$.

We make the following observations about the bi-choice graph $G(\mu, \mu')$.

O1. An edge $i \bullet\!\!\longrightarrow\!\!\bullet\, j$ implies $j \neq \mu'(i)$. An edge $i \bullet\!\!\dashrightarrow\!\!\bullet\, j$ implies $j \neq \mu(i)$.

O2. An edge $i \bullet\!\!-\!\!\!-\!\!\bullet\, j$ implies that $\mu'(i) = \mu(i) = j$, and therefore that there is no other edge adjacent to i or j.

O3. For each agent i, there is either (i) a unique undirected edge to which i is adjacent or (ii) a unique outgoing directed edge from i.

O4. An edge $i \bullet\!\!\longrightarrow\!\!\bullet\, j$ or $i \bullet\!\!\dashrightarrow\!\!\bullet\, j$ implies that $j \neq i$.

O5. An edge $i \bullet\!\!\longrightarrow\!\!\bullet\, j$ or $i \bullet\!\!\dashrightarrow\!\!\bullet\, j$ implies that there is no directed edge from j to i.

O6. An edge $i \bullet\!\!\longrightarrow\!\!\bullet\, j$ implies $j \bullet\!\!\dashrightarrow\!\!\bullet\, k$ for some $k \neq i, j$.
Similarly, an edge $i \bullet\!\!\dashrightarrow\!\!\bullet\, j$ implies $j \bullet\!\!\longrightarrow\!\!\bullet\, k$ for some $k \neq i, j$.

O7. Each vertex has at most 1 incoming directed edge.

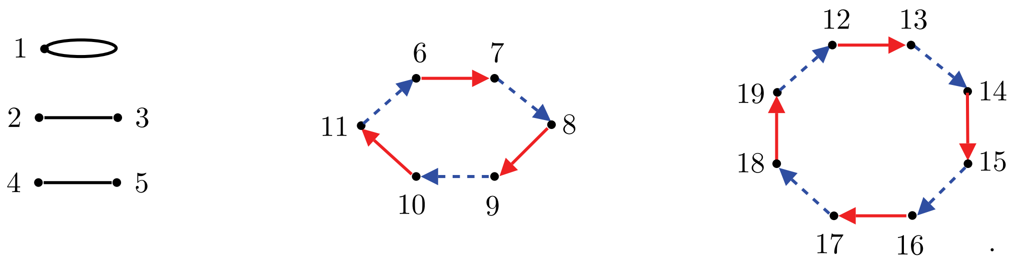
The following lemma follows immediately from the observations.

Lemma 1 (**Bi-choice graph components**).
Let μ and μ' be stable matchings. Let $i \in N$. Then, agent i's component of $G(\mu, \mu')$ either

(a) *equals i •——• j for some agent j (i.e., i ⬭ if $j = i$), or*

(b) *is a directed even cycle, i.e., there is a directed path starting from i that induces a closed cycle $c_i = (i_1, i_2, i_3, \ldots, i_p)$ consisting of an even number $p \geq 4$ of agents (with $i \in \{i_1, \ldots, i_p\}$ and $i_r \neq i_s$ for all $r \neq s$) where continuous and discontinuous edges alternate.*

An example of a bi-choice-graph is



.

2.2 Lonely Wolves, Medians, and Compromise

We now return to our quest of choosing a particularly appealing stable matching for a solvable roommate problem with multiple stable matchings. A first idea is to select a stable matching that maximizes the number of matched pairs. It turns out that no such selection is possible because an agent who is single at any stable matching is also single at all other stable matchings (Gusfield and Irving, 1989, Theorem 4.5.2). According to Roth and Sotomayor (1990, p. 50), the first statement of this theorem for the (sub)class of marriage problems can be found in McVitie and Wilson (1970) for the case when all men and women are mutually acceptable and in Roth (1984) for college admissions problems. Here we provide a new and elementary proof based on the structure of bi-choice graphs.

Theorem 1 (**The lonely wolf theorem**).
Let μ and μ' be stable matchings. Then, μ and μ' have the same set of single agents, i.e., $\mu(i) = i$ if and only if $\mu'(i) = i$.

Proof. Suppose w.l.o.g. $\mu(i) = i$ but $\mu'(i) \neq i$. Consider $G(\mu, \mu')$. Since preferences are strict and μ' is individually rational, $\mu'(i) \succ_i i = \mu(i)$. Let $j = \mu'(i)$. Thus, by $E2$, i •- - -▸• j is a directed edge in $G(\mu, \mu')$ and by Lemma 1 (b), agent i is part of a (directed) cycle, where his predecessor and successor are his two different matches under μ and μ'. In particular, $\mu(i) \neq i$, contradicting the initial assumption. □

Since no selection can be based on the set of matched agents, we next try to find a stable matching that will be perceived as fair by the agents. Imagine that we ask each agent to rank all stable matchings according to his preferences. We extend agents' preferences over the set of agents to matchings as follows. For any agent $i \in N$ and any two (stable) matchings μ and μ', $\mu \succeq_i \mu'$ if and only if $\mu(i) \succeq_i \mu'(i)$. Note that since an agent might be matched to the same agent in different matchings, this ranking is not strict. The indifference and strict preference relation (over matchings) associated with $\succeq_i$ are denoted by $\sim_i$ and $\succ_i$, respectively.

Clearly, we cannot always give the best match to every agent, but can we implement fairness by finding a matching that matches each agent with his l-th ranked match for some natural number l? It is not difficult to show that this idea of fairness or compromise is not always feasible if there is an even number of stable matchings. Next, we show that for roommate problems with an odd number of stable matchings a compromise matching where each agent is matched to a match of the same rank is possible. In fact, we prove that for any odd number of stable matchings, a stable matching in which each agent is matched to his "median" match always exists. Thus, for roommate problems with an odd number of stable matchings Adam Smith (who stands for stability) and John Rawls (who stands for fairness) represent compatible criteria and hence "can share a room."

The next lemma, which appeared in Gusfield and Irving (1989, Lemma 4.3.9),[7] facilitates the proof of our main result.

Lemma 2 (Decomposability).
Let μ and μ' be stable matchings. Let $i \in N$. Suppose $\mu'(i) \neq \mu(i) = j$ for some $j \in N$. Then, $j, \mu'(i) \neq i$. Moreover,

(a) *$\mu(i) \succ_i \mu'(i)$ implies $\mu'(j) \succ_j \mu(j)$;*

(b) *$\mu'(i) \succ_i \mu(i)$ implies $\mu(j) \succ_j \mu'(j)$.*

Proof. By Theorem 1, $j, \mu'(i) \neq i$. To prove (a), assume $j = \mu(i) \succ_i \mu'(i)$. In terms of the bi-choice graph $G(\mu, \mu')$, $i \longrightarrow j$. By O6, $j \dashrightarrow k$ where $k = \mu'(j) \neq i$. Hence, $\mu'(j) \succ_j \mu(j)$. To prove (b), assume $\mu'(i) \succ_i \mu(i)$. In terms of the bi-choice graph $G(\mu, \mu')$, $i \dashrightarrow \mu'(i)$. By Lemma 1 (b), j is i's predecessor in $G(\mu, \mu')$ and there is a continuous directed edge from j to i. Hence, $\mu(j) = i \succ_j \mu'(j)$. □

Our main result, Theorem 2, extends Theorem 4.3.5 in Gusfield and Irving (1989) from three to any odd number of stable matchings.

Let $\mu_1, \ldots, \mu_{2k+1}$ be an odd number of (possibly non-distinct) stable matchings. Let each agent rank these matchings according to his preferences. We define agent i's median match as its $(k+1)$-st ranked match, and denote it by $\text{med}\{\mu_1(i), \ldots, \mu_{2k+1}(i)\}$.

Theorem 2 (The median matching theorem or Smith and Rawls share a room). *Let $\mu_1, \ldots, \mu_{2k+1}$ be an odd number of (possibly non-distinct) stable matchings. Then, $\mu^* : N \to N$ defined by*

$$\mu^*(i) := \text{med}\{\mu_1(i), \ldots, \mu_{2k+1}(i)\} \text{ for all } i \in N$$

is a well-defined stable matching. We call μ^* the median matching *of $\mu_1, \ldots, \mu_{2k+1}$.*

In the context of the linear programming approach to so-called bistable matching problems, Sethuraman and Teo (2001, Theorem 3.2) mentioned this result (without proof) as an interesting structural property of stable roommate matchings. Here we provide an elementary proof. In the first part of the proof we use the decomposability result (Lemma 2) to show that the median

[7] According to Roth and Sotomayor (1990, p. 50), the first statement of this lemma for the (sub)class of marriage problems can be found in Knuth (1976).

matching is well-defined. In the second part of the proof we use "transitivity of blocking" to show that the median matching is stable.

For any agent $i \in N$ we define partial preferences over sets of matchings as follows. Let U and V be two sets of matchings. If for all $\mu^U \in U$ and $\mu^V \in V$, $\mu^U(i) \succ_i \mu^V(i)$, then $U \succ_i V$. Furthermore, if for all $\mu^U \in U$ and $\mu^V \in V$, $\mu^U(i) \sim_i \mu^V(i)$, then $U \sim_i V$.

Proof. First, we show that μ^* is a well-defined matching, i.e., μ^* is of order two. Let $i \in N$ with $\mu^*(i) \neq i$. Let $j := \mu^*(i)$. We have to prove that $\mu^*(j) = i$. W.l.o.g. $\mu_1(i) \succeq_i \mu_2(i) \succeq_i \cdots \succeq_i \mu_{2k}(i) \succeq_i \mu_{2k+1}(i)$. Then, $\mu^*(i) = \mu_{k+1}(i) = j$ and $\mu_{k+1}(j) = i$. Let $\Sigma_1 := \{\mu_1, \ldots, \mu_k\}$ and $\Sigma_2 := \{\mu_{k+2}, \ldots, \mu_{2k+1}\}$. Define

$$\begin{array}{lcll} SA_i & := & \{\mu \in \Sigma_1 \mid \mu \succ_i \mu_{k+1}\} & \text{(matchings that are "strictly above" } \mu_{k+1}\text{)},\\ IA_i & := & \{\mu \in \Sigma_1 \mid \mu \sim_i \mu_{k+1}\} & \text{(matchings that are "indifferent above" } \mu_{k+1}\text{)},\\ SB_i & := & \{\mu \in \Sigma_2 \mid \mu \prec_i \mu_{k+1}\} & \text{(matchings that are "strictly below" } \mu_{k+1}\text{), and}\\ IB_i & := & \{\mu \in \Sigma_2 \mid \mu \sim_i \mu_{k+1}\} & \text{(matchings that are "indifferent below" } \mu_{k+1}\text{)}. \end{array}$$

For notational convenience we denote the singleton set $\{\mu_{k+1}\}$ by μ_{k+1}. Then,

$$SA_i \succ_i IA_i \sim_i \mu_{k+1} \sim_i IB_i \succ_i SB_i.$$

Note that

$$\mu \in IB_i \cup IA_i \Rightarrow \mu(i) = \mu_{k+1}(i) = j. \qquad (1)$$

By decomposability (Lemma 2, a) and $\mu_{k+1} \succ_i SB_i$, $SB_i \succ_j \mu_{k+1}$. By decomposability (Lemma 2, b) and $SA_i \succ_i \mu_{k+1}$, $\mu_{k+1} \succ_j SA_i$. By (1), $IB_i \sim_j \mu_{k+1} \sim_j IA_i$. Summarizing,

$$SB_i \succ_j IB_i \sim_j \mu_{k+1} \sim_j IA_i \succ_j SA_i.$$

Since $SA_i \cup IA_i = \Sigma_1$ and $SB_i \cup IB_i = \Sigma_2$, we have $|SA_i \cup IA_i| = |IB_i \cup SB_i| = k$ and therefore, $\mu^*(j) = \mu_{k+1}(j) = i$. Hence, μ^* is a well-defined matching.

We now prove that μ^* is stable. By definition, μ^* is individually rational. Suppose there is a blocking pair $\{i, j\}$ with $i \neq j$ for μ^*, i.e., $j \succ_i \mu^*(i)$ and $i \succ_j \mu^*(j)$. Then, i prefers j to his match under at least $k+1$ stable matchings in $\Sigma := \{\mu_1, \ldots, \mu_{2k+1}\}$. Similarly, j prefers i to his match under at least $k+1$ stable matchings in Σ. Since Σ contains only $2k+1$ matchings, for at least one matching $\mu \in \Sigma$, both $j \succ_i \mu(i)$ and $i \succ_j \mu(j)$ (it is here that we apply transitivity of blocking[8]). Hence, $\{i, j\}$ is a blocking pair for μ. This however contradicts stability of μ (the set Σ only contains stable matchings). Therefore, there is no blocking pair for μ^*. Hence, μ^* is stable. □

Note that the "median operator" is not closed, i.e., the resulting median matching need not be one of the stable matchings that were used to calculate it. Moreover, we can easily extend the result of Theorem 2 to an even number of stable matchings (Sethuraman and Teo, 2001, Theorem 3.3).

Corollary 1 (Smith and Rawls (almost) share a room).
Let $\mu_1, \ldots, \mu_{2k}$ be an even number of (possibly non-distinct) stable matchings. Then, there exists a stable matching at which each agent is assigned a match of rank k or $k+1$.

[8]Note that $j \succ_i \mu^*(i) \succeq_i \ldots \succeq_i \mu(i) \succeq_i \ldots$ and $i \succ_j \mu^*(j) \succeq_j \ldots \succeq_j \mu(j) \succeq_j \ldots$ together with transitivity implies $j \succ_i \mu(i)$ and $i \succ_j \mu(j)$ (for the pairwise blocking notion we consider here this is immediate, but for more general blocking notions transitivity of blocking might not hold).

3 Two-Sided Matching and Generalized Medians

3.1 Marriage Problems

We now turn to the subclass of marriage problems. These two-sided roommate problems exhibit some additional structure and therefore we can strengthen the median matching theorem. Let $\mu_1, \ldots, \mu_k$ be k (possibly non-distinct) stable matchings for a marriage problem and assume that each agent ranks these matchings according to his preferences. Using linear programming tools, Teo and Sethuraman (1998, Theorem 2) showed that the map that assigns to each man his l-th (weakly) best match and to each woman her $(k-l+1)$-st (weakly) best match determines a well-defined and stable matching. Klaus and Klijn (2006) introduced and discussed (generalized) medians as compromise solutions for two-sided matching problems. A slight adaptation of the (elementary) proof of Theorem 2 yields this stronger result for marriage problems. For this reason, and in contrast to previous proofs that use the so-called lattice structure of the set of stable matchings (see Fleiner, 2002, Theorem 5.5, Klaus and Klijn, 2006, Theorem 3.2, and Yenmez and Schwarz, 2007, Theorem 1)[9] our proof is elementary.

Consider a marriage problem $(M \cup W, (\succeq_i)_{i \in M \cup W})$. Let $\mu_1, \ldots, \mu_k$ be k (possibly non-distinct) stable matchings. Let each agent rank these matchings according to his preferences as explained before. Formally, for each $i \in M \cup W$ there is a sequence of matchings $(\mu^i_1, \ldots, \mu^i_k)$ such that $\{\mu^i_1, \ldots, \mu^i_k\} = \{\mu_1, \ldots, \mu_k\}$ and for any $l \in \{1, \ldots, k-1\}$, $\mu^i_l(i) \succeq_i \mu^i_{l+1}(i)$. Thus, for any $l \in \{1, \ldots, k\}$, under μ^i_l agent i is assigned to his l-th (weakly) best match (among the k stable matchings).

For any $l \in \{1, \ldots, k\}$, we define the *generalized median matching* α_l as the function $\alpha_l : M \cup W \to M \cup W$ such that

$$\alpha_l(i) := \begin{cases} \mu^i_l(i) & \text{if } i \in M; \\ \mu^i_{(k-l+1)}(i) & \text{if } i \in W. \end{cases}$$

Theorem 3 (Marriage and compromise – generalized medians).
Let $\mu_1, \ldots, \mu_k$ be k (possibly non-distinct) stable matchings for a marriage problem. Then, for any $l \in \{1, \ldots, k\}$, α_l is a well-defined stable matching.

3.2 College Admissions Problems

Fleiner (2002, Theorem 5.5), Klaus and Klijn (2006, Theorem 3.2), and Sethuraman et al. (2006, Theorem 9) generalized Theorem 3 to college admissions problems (Gale and Shapley, 1962) with so-called responsive preferences in which students have to be matched to colleges based on the students' and the colleges' preferences over the other side of the market and colleges' capacity constraints. Hence, like the roommate model, the college admissions model is a generalization of the marriage model. (However, the college admissions model is not a generalization of the roommate model, nor vice versa.) Next, we show that the proof of Theorem 3 is essentially valid for the college admissions model. More precisely, we can extend Theorem 3 to college admissions problems by using the same steps as in the proof of Theorem 3 (or, in fact, Theorem 2). Unfortunately, the extended proof will no longer be elementary in the sense that in order to be

[9] Yenmez and Schwarz (2007) extend the analysis of stable median matchings from the class of marriage problems to two-sided (one-to-one) matching problems with side payments.

able to apply the key steps we need to resort to well-known but non-trivial results for college admissions problems.

We first extend the marriage model to the college admissions model by introducing the following notation. There are two finite and disjoint sets of agents: a set $S = \{s_1, \ldots, s_m\}$ of students and a set $\mathcal{C} = \{C_1, \ldots, C_n\}$ of colleges. We denote a generic student by s and a generic college by C. For each college C, there is a fixed quota q_C that represents the number of positions it offers.[10]

Each student has a complete, transitive, and strict preference relation $\succeq_s$ over the colleges and the prospect of being *unmatched*. Hence, student s's preferences are defined over the elements in $\mathcal{C} \cup \{s\}$. If $C \in \mathcal{C}$ such that $C \succ_s s$, then we call C an *acceptable college for student s*.

A set of students $S' \subseteq S$ *is feasible for college C* if $|S'| \leq q_C$. Each college has a complete and transitive preference relation $\succeq_C$ over feasible sets of students. Hence, college C's preferences are defined over the elements in $\mathcal{P}(S, q_C) := \{S' \subseteq S : |S'| \leq q_C\}$. We make two assumptions on the preferences of a college C.[11]

First, C's preferences over singleton sets of students, or equivalently over individual students, are strict. For notational convenience we denote a singleton set $\{s\}$ by s. The second assumption describes comparisons of feasible sets of students when a single student is added or replaced. If $s \in S$ is such that $s \succ_C \emptyset$, then we call s an *acceptable student for college C*. If $s, s' \in S$ are such that $s \succ_C s'$, then we call student s a *better student than student s' for college C*. We assume that each college C's preferences over feasible sets of students are based on preferences over individual students such that C always prefers to add an acceptable student and it also prefers to replace any student by a better student. More formally, we assume that C's preferences are *responsive*, i.e., for all $S' \in \mathcal{P}(S, q_C)$,

- if $s \notin S'$ and $|S'| < q_C$, then $(S' \cup s) \succ_C S'$ if and only if $s \succ_C \emptyset$ and
- if $s \notin S'$ and $t \in S'$, then $((S' \backslash t) \cup s) \succ_C S'$ if and only if $s \succ_C t$.

A *college admissions problem* is a triple $(S, \mathcal{C}, (\succeq_i)_{i \in S \cup \mathcal{C}})$. A *matching* for college admissions problem $(S, \mathcal{C}, (\succeq_i)_{i \in S \cup \mathcal{C}})$ is a function μ on the set $S \cup \mathcal{C}$ such that

- each student is either matched to exactly one college or unmatched, i.e., for all $s \in S$, either $\mu(s) \in \mathcal{C}$ or $\mu(s) = s$,
- each college is matched to a feasible set of students, i.e., for all $C \in \mathcal{C}$, $\mu(C) \in \mathcal{P}(S, q_C)$, and
- a student is matched to a college if and only if the college is matched to the student, i.e., for all $s \in S$ and $C \in \mathcal{C}$, $\mu(s) = C$ if and only if $s \in \mu(C)$.

Given matching μ, we call $\mu(s)$ *student s's match* and $\mu(C)$ *college C's match*.

Similar to the roommate and marriage model, a key property of matchings in the college admissions model is *stability*. First, we impose a voluntary participation condition. A matching μ is *individually rational* if neither a student nor a college would be better off by breaking a current match, i.e., if $\mu(s) = C$, then $C \succ_s s$ and $\mu(C) \succ_C (\mu(C) \backslash s)$. By responsiveness

[10]The marriage model is the special case where for all $C \in \mathcal{C}$, $q_C = 1$.

[11]See Roth and Sotomayor (1989) for a discussion of these assumptions.

of $\succeq_C$, the latter requirement can be replaced by $s \succ_C \emptyset$. Thus alternatively, a matching μ is individually rational if any student and any college that are matched to one another are mutually acceptable. Second, if a student s and a college C are not matched to one another at a matching μ but the student would prefer to be matched to the college and the college would prefer to either add the student or replace another student by student s, then we would expect this mutually beneficial adjustment to be carried out. Formally, a pair (s, C) *blocks* $(\mu(s), \mu(C))$ if $C \succ_s \mu(s)$ and

B1. $[\,|\mu(C)| < q_C$ and $s \succ_C \emptyset\,]$ or

B2. $[$there exists $t \in \mu(C)$ such that $s \succ_C t\,]$.[12]

A matching μ is *stable* if it is individually rational and there is no pair (s, C) that blocks $(\mu(s), \mu(C))$.[13] Gale and Shapley (1962) proved that each college admissions problem has at least one stable matching.

Consider a college admissions problem $(S, \mathcal{C}, (\succeq_i)_{i \in S \cup \mathcal{C}})$. Let $\mu_1, \ldots, \mu_k$ be k (possibly non-distinct) stable matchings. Let each student rank these matchings according to his preferences. Formally, for each $s \in S$ there is a sequence of matchings $(\mu_1^s, \ldots, \mu_k^s)$ such that $\{\mu_1^s, \ldots, \mu_k^s\} = \{\mu_1, \ldots, \mu_k\}$ and for any $l \in \{1, \ldots, k-1\}$, $\mu_l^s(s) \succeq_s \mu_{l+1}^s(s)$. Thus, for any $l \in \{1, \ldots, k\}$, under μ_l^s student s is assigned to his l-th (weakly) best match (among the k stable matchings). For any $l \in \{1, \ldots, k\}$, define the function α_l^S on the set S such that for all $s \in S$, $\alpha_l^S(s) := \mu_l^s(s)$.

By Roth and Sotomayor (1989, Theorem 3), each college can proceed similarly.[14] Formally, for each $C \in \mathcal{C}$ there is a sequence of matchings $(\mu_1^C, \ldots, \mu_k^C)$ such that $\{\mu_1^C, \ldots, \mu_k^C\} = \{\mu_1, \ldots, \mu_k\}$ and for any $l \in \{1, \ldots, k-1\}$, either $\mu_l^C(C) \succ_C \mu_{l+1}^C(C)$ or $\mu_l^C(C) = \mu_{l+1}^C(C)$. Thus, for any $l \in \{1, \ldots, k\}$, under μ_l^C college C is assigned to its l-th (weakly) best match (among the k stable matchings). For any $l \in \{1, \ldots, k\}$, define the function $\alpha_l^{\mathcal{C}}$ on the set $\mathcal{C}$ such that for all $C \in \mathcal{C}$, $\alpha_l^{\mathcal{C}}(C) := \mu_l^C(C)$.

For any $l \in \{1, \ldots, k\}$, we define the *generalized median matching* α_l by

$$\alpha_l(i) := \begin{cases} \alpha_l^S(s) & \text{if } i \in S; \\ \alpha_{k-l+1}^{\mathcal{C}}(C) & \text{if } i \in \mathcal{C}. \end{cases}$$

Theorem 4 (College admissions and compromise – generalized medians).
Let $\mu_1, \ldots, \mu_k$ be k (possibly non-distinct) stable matchings for a college admissions problem. Then, for any $l \in \{1, \ldots, k\}$, α_l is a well-defined stable matching.

The proof is similar to that of Theorem 3. It is based on the two properties that were already key in the proofs of Theorems 2 and 3: (weak) decomposability and transitivity of blocking. Unfortunately, in this case the extended proof can no longer be considered elementary because the decomposability lemma for college admissions problems (Lemma 3) is not an elementary result and in addition we have to establish a (non-elementary) "transitivity of blocking" property for college admissions problems (Lemma 4).

[12]Recall that by responsiveness $B1$ implies $(\mu(C) \cup s) \succ_C \mu(C)$ and $B2$ implies $((\mu(C) \backslash t) \cup s) \succ_C \mu(C)$.

[13]Roth and Sotomayor (1989, Proposition 1) showed that this is the "correct" concept of stability in the sense that there is no blocking *coalition* if and only if there is no blocking pair. Also note that in the special case of $q_C = 1$ (for all $C \in \mathcal{C}$) the concept coincides with the previously introduced stability concept for marriage problems.

[14]Roth and Sotomayor (1989, Theorem 3) stated that for all stable matchings μ and μ' and all $C \in \mathcal{C}$, either $\mu(C) \succ_C \mu'(C)$, $\mu'(C) \succ_C \mu(C)$, or $\mu(C) = \mu'(C)$.

Lemma 3 (**Weak decomposability**, Roth and Sotomayor, 1990, Theorem 5.33).
Let μ and μ' be stable matchings. Let $C \in \mathcal{C}$, $s \in S$, and $s \in \mu(C) \cup \mu'(C)$. Then,

(a) $\mu(C) \succ_C \mu'(C)$ *implies* $\mu'(s) \succeq_s \mu(s)$*;*

(b) $\mu(s) \succ_s \mu'(s)$ *implies* $\mu'(C) \succeq_C \mu(C)$*.*

Note that matchings μ and μ' play a symmetric role in Lemma 3. Next we extend the transitivity of blocking property (see Footnote 8) used in the proofs of Theorems 2 and 3 to college admissions problems.

Lemma 4 (**Transitivity of blocking for college admissions**).
Let μ and μ' be matchings, $C \in \mathcal{C}$, and $s \in S$. Suppose (s, C) blocks $(\mu(s), \mu(C))$. Suppose also that C is assigned groups of students $\mu(C)$ and $\mu'(C)$ under some stable *matchings.*[15] *If $\mu(s) \succeq_s \mu'(s)$ and $\mu(C) \succeq_C \mu'(C)$, then (s, C) also blocks $(\mu'(s), \mu'(C))$.*[16]

4 Conclusion

For three different matching models (roommate, marriage, and college admissions) we have shown that certain compromise stable matchings exist in the form of so-called (generalized) median matchings. For roommate problems, we prove the existence and stability of the median matching of an odd number of stable matchings (Theorem 2). For the two-sided (marriage and college admissions) matching problems we prove the existence and stability of generalized median matchings for any number of stable matchings (Theorems 3 and 4). In all proofs we use a decomposability property to show that (generalized) median matchings are well-defined and a transitivity of blocking property to show that they are stable. It is not difficult to construct examples in more general settings (for instance, college admissions with q-separable and substitutable preferences, Martínez et al., 2000, or network formation, Jackson and Watts, 2002) that demonstrate that our results can break down if decomposability or transitivity of blocking is violated.

References

Cheng, C. (2008): "The Generalized Median Stable Matchings: Finding Them is not that Easy." E.S. Laber et al. (Eds.): Proceedings of the 8th Latin American Theoretical Informatics (LATIN'08), LNCS vol. 4957, pp. 568–579. Springer-Verlag, Berlin.

Chung, K.-S. (2000): "On the Existence of Stable Roommate Matchings." *Games and Economic Behavior*, 33: 206–230.

Diamantoudi, E., Miyagawa, E., and Xue, L. (2004): "Random Paths to Stability in the Roommate Problem." *Games and Economic Behavior*, 48: 18–28.

Fleiner, T. (2002): "Some Results on Stable Matchings and Fixed Points." Technical Report TR-2002-08, Egerváry Research Group, Budapest. `http://www.cs.elte.hu/egres`.

Gale, D. and Shapley, L. S. (1962): "College Admissions and the Stability of Marriage." *American Mathematical Monthly*, 69: 9–15.

[15] That is, there are stable matchings $\bar{\mu}$ and $\bar{\mu}'$ such that $\bar{\mu}(C) = \mu(C)$ and $\bar{\mu}'(C) = \mu'(C)$.

[16] By "(s, C) also blocks $(\mu'(s), \mu'(C))$" we mean that $C \succ_s \mu'(s)$ and *B1'* [$|\mu'(C)| < q_C$ and $s \succ_C \emptyset$] or *B2'* [there exists $t' \in \mu'(C)$ such that $s \succ_C t'$].

Gusfield, D. and Irving, R. W. (1989): *The Stable Marriage Problem: Structure and Algorithms.* The MIT Press, Cambridge.

Jackson, M. O. and Watts, A. (2002): "The Evolution of Social and Economic Networks." *Journal of Economic Theory*, 106: 265–295.

Klaus, B. and Klijn, F. (2006): "Median Stable Matching for College Admissions." *International Journal of Game Theory*, 34: 1–11.

Knuth, D. E. (1976): *Marriages Stables.* Les Presses de l'Université de Montreal, Montreal.

Martínez, R., Massó, J., Neme, A., and Oviedo, J. (2000): "Single Agents and the Set of Many-to-One Stable Matchings." *Journal of Economic Theory*, 91: 91–105.

McVitie, D. G. and Wilson, L. B. (1970): "Stable Marriage Assignments for Unequal Sets." *BIT*, 10: 295–309.

Rawls, J. (1971): *A Theory of Justice.* Harvard University Press, Cambridge.

Roth, A. E. (1984): "The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory." *Journal of Political Economy*, 92: 991–1016.

Roth, A. E. (1985): "The College Admissions Problem is not Equivalent to the Marriage Problem." *Journal of Economic Theory*, 36: 277–288.

Roth, A. E. and Sotomayor, M. A. O. (1989): "The College Admissions Problem Revisited." *Econometrica*, 57: 559–570.

Roth, A. E. and Sotomayor, M. A. O. (1990): *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis.* Cambridge University Press, Cambridge.

Sethuraman, J. and Teo, C.-P. (2001): "A Polynomial-Time Algorithm for the Bistable Roommates Problem." *Journal of Computer and Systems Sciences*, 63: 486–497.

Sethuraman, J., Teo, C.-P., and Qian, L. (2006): "Many-to-One Stable Matching: Geometry and Fairness." *Mathematics of Operations Research*, 31: 581–596.

Smith, A. (1796): *An Inquiry into the Nature and Causes of the Wealth of Nations.* Oxford University Press, Oxford.

Tan, J. (1991): "A Necessary and Sufficient Condition for the Existence of a Complete Stable Matching." *Journal of Algorithms*, 12: 154–178.

Teo, C.-P. and Sethuraman, J. (1998): "The Geometry of Fractional Stable Matchings and its Applications." *Mathematics of Operations Research*, 23: 874–891.

Yenmez, M. B. and Schwarz, M. (2007): "Median Stable Matching." Mimeo. `http://ssrn.com/abstract=1031277`.

The Stability of the Roommate Problem Revisited*

E. Inarra, C. Larrea and E. Molis†

10th June 2008

Abstract

The lack of stability in some matching problems suggests that alternative solution concepts to the core might be applied to find predictable matchings. We propose absorbing sets as a solution for the class of roommate problems with strict preferences. This solution, which always exists, either gives the matchings in the core or predicts other matchings when the core is empty. Furthermore, it satisfies the interesting property of outer stability. We also determine the matchings in absorbing sets and find that in the case of multiple absorbing sets a similar structure is shared by all.

KEYWORDS: Roommate problem, core, absorbing sets.

1 Introduction

Matching markets are of great interest in a variety of social and economic environments, ranging from marriages formation, through admission of students into colleges to matching firms with workers.[1] One of the aims pursued by the analysis of these markets is to find stable matchings. There are, however, some markets for which the set of stable matchings, *i.e.* the core, is empty. For these cases, we suggest that instead of using the common approach of restricting the preferences domain to deal with nonempty core matching markets,[2] other solution concepts may be applied to find "predictable" matchings. We argue that this alternative is a step towards furthering our understanding of matching market performance.

Our approach consists of associating each matching market with an abstract system (an abstract set endowed with a binary relation) and then applying one of the existing solution concepts to determine predictable matchings. The modeling of abstract systems deals with the problem of choosing a subset from a feasible set of alternatives. Various solution concepts have been proposed for solving abstract systems, such as the core, von Neumann-Morgenstern stable sets[3] (von Neumann-Morgenstern, 1947), subsolutions (Roth, 1976), admissible sets (Kalai, Pazner and Schmeidler, 1976), and absorbing sets.

*We thank Svetoslav Ivantchev for his help in programming, Ana Saracho for her participation in the early stages of the paper and Peter Biró, Bettina Klaus, Flip Klijn, Jordi Massó, Marilda Sotomayor, Federico Valenciano and Rodrigo Velez for their comments and suggestions. Elena Molis also acknowledges the hospitality of the Department of Economics of Maastricht University.This research has been supported by the University of the Basque Country under project 1/UPV 00031.321-HA-7903/2000, project UPV/EHU Grant PIFA010502 and project GIU06/44 and by the Spanish Ministerio de Educación y Ciencia under project SEJ2006-05455, cofunded by FEDER and project BEC2000-0875. It has also benefited from the Franco-Spanish Exchange Program HF-2006-0021/EGIDE-Picasso.

†*Address:* Avda. L. Agirre 83, 48015 Bilbao, Spain. Inarra: elena.inarra@ehu.es, Larrea: mariaconcepcion.larrea@ehu.es, Molis:elena.molis@ehu.es

[1] See Roth and Sotomayor (1990) for a comprehensive survey of two-sided matching models.

[2] See for example, Roth (1985) and Kelso and Crawford (1982).

[3] Ehlers (2007) studies von Neumann-Morgenstern stable sets in two-sided matching markets.

The notion of absorbing sets, which is the solution concept selected in our work, was first introduced by Schwartz (1970) and it coincides with the elementary dynamic solution (Shenoy, 1979).

We focus our attention on one-sided matching markets where each agent is allowed to form at most one partnership. This kind of problems is known as the *roommate problem* and is a generalization of the marriage problem, see Gale and Shapley (1962). In these problems each agent in a set ranks all others (including herself) according to her preferences. In this seminal paper it is shown that this problem may not have a stable matching.

The abstract system associated with a roommate problem is the pair formed by the set of all matchings and a binary domination relation which represents the existence of a blocking pair of agents allowing transition from one matching to another. Matchings that are not blocked by any pair of agents are called stable. In this model the set of stable matchings equals the core. Roommate problems that do not admit any such matchings are called *unsolvable*. Otherwise they are said to be *solvable*.

Core stability for solvable roommate problems has been studied by Gale and Shapley (1962), Irving (1985), Tan (1991), Abeledo and Isaak (1991), Chung (2000), Diamantoudi, Miyagawa and Xue (2004) and Klaus and Klijn (2008) among others. With few exceptions, however, unsolvable roommate problems have not been so thoroughly studied. When there is no core stability, interest is rekindled in the application of other solution concepts to the class of roommate problems. Such interest is further enhanced from the empirical perspective in that as Pittel and Irving (1994) observe, when the number of agents increases, the probability of a roommate problem being solvable decreases fairly steeply.

Here we propose *absorbing sets* as a solution for the class of roommate problems with strict preferences. In this context, an absorbing set is a set of matchings that satisfies the following two conditions: (i) any two distinct matchings inside the set (directly or indirectly) dominate each other and (ii) no matching in the set is dominated by a matching outside the set. We believe that the selection of this solution concept is well justified since for a solvable roommate problem it exactly provides the matchings in the core, and for an unsolvable roommate problem it gives a nonempty set of matchings with an interesting property of stability. Thus, the solution of absorbing sets may be considered as a generalization of the core.

The notion of an absorbing set may perhaps be better understood if it is illustrated with the following description: Consider matchings derived from an unstable matching by satisfying a blocking pair of agents. This can be seen as a dynamic process in which unstable matchings are adjusted when a blocking pair of agents mutually decide to become partners. Either this change gives a stable matching or a new blocking pair of agents will generate another matching and so on. If some stable matching exists this dynamic process eventually converges to one[4]. Otherwise the process will lead to a set of matchings (an absorbing set) such that via this dynamic process (i) any matching in the set can be obtained from any other and (ii) it is impossible to abandon the absorbing set[5]. From this perspective it is easy to see that an absorbing set satisfies a property of *outer stability* in the sense that all matchings not in an absorbing set are (directly or indirectly) dominated by a matching that does belong to an absorbing set. As a result, matchings outside absorbing sets can be ruled out as reasonable matchings.

Among the scant literature on unsolvable roommate problems the papers by Tan (1990) and Abraham, Biró and Manlove (2005) are worthy of mention. The former investigates matchings with the maximum number of disjoint pairs of agents such that these pairs are "internally" stable and the latter looks at matchings with the smallest number of blocking pairs. For solvable roommate problems

[4]For marriage problems Roth and Vande Vate (1990) show that there exists a convergence domination path from any unstable matching to a stable one. This is also shown for solvable roommate problems by Diamantoudi, Miyagawa and Xue (2004).

[5]For unsolvable roommate problems Inarra, Larrea and Molis (2008) show that there is a domination path from any matching that reaches certain matchings called P-stable matchings.

both proposals give the matchings in the core, but for unsolvable ones it is easy to check that neither satisfies the outer stability property.

The contribution of this paper to the analysis of the stability of roommate problems can be summarized as follows:

First, we find that absorbing sets are determined by stable partitions. This notion, introduced by Tan (1991) as a structure generalizing the notion of a stable matching, allowed him to establish a necessary and sufficient condition for the existence of a stable matching in roommate problems. By using the relation between absorbing sets and stable partitions we also show that if a roommate problem is solvable then an absorbing set is a singleton consisting of a stable matching and the union of all absorbing sets coincides with the core.

Second, we characterize absorbing sets in terms of stable partitions. The characterization provided allows us to specify the stable partitions determining absorbing sets. A property of these partitions helps us to identify the matchings in absorbing sets.

Third, we show that all matchings in an absorbing set share some common features. Furthermore, in the case of a roommate problem with multiple absorbing sets we prove some similarities among their (corresponding) matchings. Specifically in terms of the dynamic process mentioned above, we find that any two absorbing sets have the same set of blocking agents responsible for going from matching to matching within the set, and that the other (nonblocking) agents are paired in a stable way, though this pairing is different across absorbing sets.

The rest of the paper is organized into the following sections. Section 2 contains the preliminaries. In Section 3 we study absorbing sets of a roommate problem. Those sets are determined in Section 4. We study the structure of their matchings in Section 5. Two appendixes conclude the paper.

2 Preliminaries

A *roommate problem* is a pair $(N, (\succcurlyeq_x)_{x \in N})$ where N is a finite set of agents and for each agent $x \in N$, $\succcurlyeq_x$ is a complete, transitive preference relation defined over N. Let $\succ_x$ be the strict preference associated with $\succcurlyeq_x$. In this paper we only consider roommate problems with strict preferences, which we denote by $(N, (\succ_x)_{x \in N})$.

A *matching* μ is a one to one mapping from N onto itself such that for all $x \in N$ $\mu(\mu(x)) = x$, where $\mu(x)$ denotes the partner of agent x under the matching μ. If $\mu(x) = x$, then agent x is single under μ. Given $S \subseteq N$, $S \neq \emptyset$, let $\mu(S) = \{\mu(x) : x \in \mathcal{S}\}$. That is, $\mu(S)$ is the set of partners of the agents in S under μ. Let $\mu \mid_S$ be the mapping from S to N which denotes the restriction of μ to S. If $\mu(S) = S$ then $\mu \mid_S$ is a matching in $(S, (\succ_x)_{x \in S})$.

A pair of agents $\{x, y\} \subseteq N$ (possibly $x = y$) is a *blocking pair* of the matching μ if

$$y \succ_x \mu(x) \text{ and } x \succ_y \mu(y). \qquad [1]$$

That is, x and y prefer each other to their current partners at μ. If $x = y$, [1] means that agent x prefers being alone to being matched with $\mu(x)$. An agent $x \in N$ *blocks* a matching μ if that agent belongs to some blocking pair of μ. A matching is called *stable* if it is not blocked by any pair $\{x, y\}$. Let $\{x, y\}$ be a blocking pair of μ. A matching μ' is obtained from μ by *satisfying* $\{x, y\}$ if $\mu'(x) = y$ and for all $z \in N \backslash \{x, y\}$,

$$\mu'(z) = \begin{cases} z & if\ \mu(z) \in \{x, y\} \\ \\ \mu(z) & \text{otherwise.} \end{cases}$$

That is, once $\{x, y\}$ is formed, their partners (if any) under μ are alone in μ', while the remaining agents are matched as in μ.

Tan (1991) establishes a necessary and sufficient condition for the solvability of roommate problems with strict preferences in terms of stable partitions. This notion, which is crucial in the investigation of this paper, can be formally defined as follows:[6]

Let $A = \{a_1, ..., a_k\} \subseteq N$ be an ordered set of agents. The set A is a *ring* if $k \geq 3$ and for all $i \in \{1, ..., k\}$, $a_{i+1} \succ_{a_i} a_{i-1} \succ_{a_i} a_i$.(subscript modulo k) The set A is a pair of mutually acceptable agents if $k = 2$ and for all $i \in \{1, 2\}$, $a_{i-1} \succ_{a_i} a_i$ (subscript modulo 2).[7] The set A is a singleton if $k = 1$.

A *stable partition* is a partition P of N such that:
(i) For all $A \in P$, the set A is a ring, a mutually acceptable pair of agents or a singleton, and
(ii) For any sets $A = \{a_1, ..., a_k\}$ and $B = \{b_1, ..., b_l\}$ of P (possibly $A = B$), the following condition holds:

$$\text{if } \; b_j \succ_{a_i} a_{i-1} \text{ then } b_{j-1} \succ_{b_j} a_i,$$

for all $i \in \{1, ..., k\}$ and $j \in \{1, ..., l\}$ such that $b_j \neq a_{i+1}$. Condition (ii) may be interpreted as a notion of stability over the partitions satisfying Condition (i).

Note that a stable partition is a generalization of a stable matching. To see this, consider a matching μ and a partition P formed by pairs of agents and/or singletons. Let $A = \{a_1, a_2 = \mu(a_1)\}$ and $B = \{b_1, b_2 = \mu(b_1)\}$ of P. If P is a stable partition then Condition (ii) implies that if $b_1 \succ_{a_1} a_2$ then $b_2 \succ_{b_1} a_2$, which is the usual notion of stability. Hence μ is a stable matching.

The following assertions are proven by Tan (1991).

Remark 1 *(i) A roommate problem $(N, (\succ_x)_{x \in N})$ has no stable matchings if and only if there exists a stable partition with an odd ring. (ii) Any two stable partitions have exactly the same odd rings.[8] (iii) Every even ring in a stable partition can be broken into pairs of mutually acceptable agents preserving stability.*

Throughout the paper we only consider stable partitions which do not contain even rings. By Remark 1 (iii) this does not imply a loss of generality.

Using the notion of a stable partition Inarra *et al.* (2008) introduce some specific matchings, called P-stable matchings, defined as follows:

Definition 1 *Let P be a stable partition. A P-stable matching is a matching μ such that for each $A = \{a_1, ..., a_k\} \in P$, $\mu(a_i) \in \{a_{i+1}, a_{i-1}\}$ for all $i \in \{1, ..., k\}$ except for a unique j where $\mu(a_j) = a_j$ if A is odd.*

3 Absorbing sets for the roommate problem

In this section we introduce the absorbing sets for the class of roommate problems with strict preferences. First, we find that absorbing sets are strongly related to stable partitions so that the notion of stable partition is converted into a useful tool for analyzing absorbing sets. To be specific, we show that each of these sets is determined by some stable partition. Second, by using this relation, we show that if a roommate problem is solvable then each absorbing set contains only one matching which is stable. Furthermore, the union of all of them coincides with the core. Thus, absorbing sets may be considered as a generalization of this solution concept in this framework.

An *abstract system* is a pair $(X, \mathcal{R})$ where X is a finite set of alternatives and $\mathcal{R}$ is a binary relation on X. Two of the solution concepts put forward to solve an abstract system are the core and absorbing sets. In what follows, we associate a roommate problem with strict preferences with an abstract system and define these two solution concepts in this particular setting. Let $\mathcal{M}$ denote

[6] See Biró *et al.* (2007) for a clarifying interpretation of this notion.
[7] Hereafter we omit subscript modulo k.
[8] A ring is odd (even) if its cardinality is odd (even).

the set of all matchings. Set $X = \mathcal{M}$ and define a binary relation R on $\mathcal{M}$ as follows: Given two matchings μ, $\mu' \in \mathcal{M}$, $\mu' R \mu$ if and only if μ' is obtained from μ by satisfying a blocking pair of μ. We say that μ' *directly dominates* μ if $\mu' R \mu$. Hereafter the system associated with the roommate problem $(N, (\succ_x)_{x \in N})$ is the pair $(\mathcal{M}, R)$. Let R^T denote the transitive closure of R. Then $\mu' R^T \mu$ if and only if there exists a finite sequence of matchings $\mu = \mu_0, \mu_1, ..., \mu_m = \mu'$ such that, for all $i \in \{1, ..., m\}$, $\mu_i R \mu_{i-1}$. We say that μ' *dominates* μ if $\mu' R^T \mu$.

As mentioned in the introduction, the conventional solution considered in matching problems is the *core*, which coincides with the set of stable matchings. In roommate problems, however, the core may be empty and *absorbing sets* stand out as a good candidate for an alternative solution concept. For these problems an absorbing set can be formally defined as follows:

Definition 2 *A nonempty subset $\mathcal{A}$ of $\mathcal{M}$ is an absorbing set of $(\mathcal{M}, R)$ if the following conditions hold:*
(i) For any two distinct $\mu, \mu' \in \mathcal{A}$, $\mu' R^T \mu$.
(ii) For any $\mu \in \mathcal{A}$ there is no $\mu' \notin \mathcal{A}$ such that $\mu' R \mu$.

Condition (i) means that matchings of $\mathcal{A}$ are symmetrically connected by the relation R^T. That is, every matching in an absorbing set is dominated by any other matching in the same set. Condition (ii) means that the set $\mathcal{A}$ is R-closed. That is, no matching in an absorbing set is directly dominated by a matching outside the set.

A nice property of this solution is that it always exists, although, in general, it may be not unique. Theorem 1 in Kalai *et al.* (1977) states that if X is finite then the admissible set (the union of absorbing sets) is nonempty (see also Theorem 2.5 in Shenoy (1979)). Thus either of these two results allows us to conclude that any $(\mathcal{M}, R)$ has at least one absorbing set. Absorbing sets also satisfy the property of *outer stability,* which says that every matching not belonging to an absorbing set is dominated by a matching that does belong to an absorbing set.[9]

Our first theorem establishes that stable partitions may be considered as structures generating the matchings in absorbing sets. Let P be a stable partition. We denote by $\mathcal{A}_P$ the set formed by all the P-stable matchings and those matchings that dominate them. The following result states that an absorbing set is one of these sets $\mathcal{A}_P$.

Theorem 1 *Let $(N, (\succ_x)_{x \in N})$ be a roommate problem. If $\mathcal{A}$ is an absorbing set then $\mathcal{A} = \mathcal{A}_P$ for some stable partition P.*

Proof. First, we prove that there exists a P-stable matching $\overline{\mu}$ such that $\overline{\mu} \in \mathcal{A}$. Let μ be an arbitrary matching of $\mathcal{A}$. If μ is a P-stable matching for some stable partition P then $\overline{\mu} = \mu$ and we are done. Otherwise, by Theorem 1 in Inarra *et al.* (2008), there exists a P-stable matching $\overline{\mu}$ such that $\overline{\mu} R^T \mu$ and by Condition (ii) of Definition 2 we have $\overline{\mu} \in \mathcal{A}$.

Now, we prove that $\mathcal{A} = \mathcal{A}_P$. By Lemma 2, we have $\mathcal{A}_P = \{\overline{\mu}\} \cup \{\mu \in \mathcal{M}{:}\mu R^T \overline{\mu}\}$.
($\subseteq$): Let $\mu \in \mathcal{A}$. We must show that $\mu \in \mathcal{A}_P$. If $\mu = \overline{\mu}$ and given that $\overline{\mu} \in \mathcal{A}_P$ we are done. Assume that $\mu \neq \overline{\mu}$. Since $\overline{\mu} \in \mathcal{A}$, by Condition (i) of Definition 2, we have $\mu R^T \overline{\mu}$. Hence $\mu \in \mathcal{A}_P$ as desired.
($\supseteq$): Let $\mu \in \mathcal{A}_P$. We must show that $\mu \in \mathcal{A}$. If $\mu = \overline{\mu}$ since $\overline{\mu} \in \mathcal{A}$ we are done. If $\mu \neq \overline{\mu}$ then $\mu R^T \overline{\mu}$. As $\overline{\mu} \in \mathcal{A}$, by Condition (ii) of Definition 2 it follows that $\mu \in \mathcal{A}$. ■

This result is used in proving the relation between absorbing sets and stable matchings as shown in our second theorem.

Theorem 2 *If the roommate problem $(N, (\succ_x)_{x \in N})$ is solvable then $\mathcal{A}$ is an absorbing set if and only if $\mathcal{A} = \{\mu\}$ for some stable matching μ.*

[9]This is shown in Kalai *et al.* (1976).

Proof. If $\mathcal{A}$ is an absorbing set then, by Theorem 1, $\mathcal{A} = \mathcal{A}_P$ for some stable partition P. Now, as the roommate problem is solvable, by Remark 1 (i) the stable partition P contains no odd rings. Hence there exists a unique P-stable matching μ which is stable by the stability of P. Then $\mathcal{A}_P = \{\mu\}$ and therefore $\mathcal{A} = \{\mu\}$. Conversely, if $\mathcal{A} = \{\mu\}$ for some stable matching μ, then $\mathcal{A}$ satisfies Conditions (i) and (ii) of Definition 2. Hence $\mathcal{A}$ is an absorbing set. ■

As a result of this theorem we have that the union of all absorbing sets coincides with the core.

To clarify the notion of absorbing sets we consider the following example, which is also used elsewhere in the paper to illustrate other results.

EXAMPLE 1 Consider the following 10-agent roommate problem:

$$2 \succ_1 3 \succ_1 4 \succ_1 5 \succ_1 6 \succ_1 7 \succ_1 8 \succ_1 9 \succ_1 1 \succ_1 10$$
$$3 \succ_2 1 \succ_2 4 \succ_2 5 \succ_2 6 \succ_2 7 \succ_2 8 \succ_2 9 \succ_2 10 \succ_2 2$$
$$1 \succ_3 2 \succ_3 4 \succ_3 5 \succ_3 6 \succ_3 7 \succ_3 8 \succ_3 9 \succ_3 3 \succ_3 10$$
$$7 \succ_4 8 \succ_4 9 \succ_4 5 \succ_4 6 \succ_4 1 \succ_4 2 \succ_4 3 \succ_4 4 \succ_4 10$$
$$8 \succ_5 9 \succ_5 7 \succ_5 4 \succ_5 6 \succ_5 5 \succ_5 1 \succ_5 2 \succ_5 3 \succ_5 10$$
$$9 \succ_6 7 \succ_6 8 \succ_6 4 \succ_6 5 \succ_6 6 \succ_6 1 \succ_6 2 \succ_6 3 \succ_6 10$$
$$5 \succ_7 6 \succ_7 1 \succ_7 4 \succ_7 9 \succ_7 8 \succ_7 7 \succ_7 2 \succ_7 3 \succ_7 10$$
$$6 \succ_8 4 \succ_8 5 \succ_8 7 \succ_8 9 \succ_8 8 \succ_8 1 \succ_8 2 \succ_8 3 \succ_8 10$$
$$4 \succ_9 5 \succ_9 6 \succ_9 7 \succ_9 8 \succ_9 9 \succ_9 1 \succ_9 2 \succ_9 3 \succ_9 10$$
$$2 \succ_{10} 10 \succ_{10} 1 \succ_{10} \ldots$$

There are three stable partitions: $P_1 =\{\{1,2,3\},\{4,7\},\{5,8\},\{6,9\},\{10\}\}$, $P_2 =\{\{1,2,3\},\{4,8\},\{5,9\},\{6,7\},\{10\}\}$ and $P_3 = \{\{1,2,3\},\{4,9\},\{5,7\},\{6,8\},\{10\}\}$. Consider the stable partition P_2. The associated P_2-stable matchings are: $\mu_1 =[\{1\},\{2,3\},\{4,8\},\{5,9\},\{6,7\},\{10\}]$, $\mu_2 =[\{2\},\{1,3\},\{4,8\},\{5,9\},\{6,7\},\{10\}]$ and $\mu_3=[\{3\},\{1,2\},\{4,8\},\{5,9\},\{6,7\},\{10\}]$ and the set $\mathcal{A}_{P_2} = \{\mu_1, \mu_2, \mu_3, \mu_4\}$, where $\mu_4 = [\{1,3\},\{2,10\},\{4,8\},\{5,9\},\{6,7\}]$. Notice that any of these matchings dominates any other but they are not directly dominated by any matching outside $\mathcal{A}_{P_2}$. Therefore $\mathcal{A}_{P_2}$ is an absorbing set. In addition, matching $\mu_1 = [\{1\},\{2,3\},\{4,8\},\{5,9\},\{6,7\},\{10\}]$ can be derived from the P_1-stable matching $\mu = [\{1\},\{2,3\},\{4,7\},\{5,8\},\{6,9\},\{10\}]$ by satisfying the following sequence of blocking pairs: $\{1,7\}$, $\{4,8\}$, $\{5,9\}$, $\{6,7\}$. Hence μ_1 belongs to $\mathcal{A}_{P_1}$. It is easy to verify, however, that μ does not dominate μ_1. Thus $\mathcal{A}_{P_1}$ is not an absorbing set since it does not satisfy Condition (i) of Definition 2.

4 Matchings in the absorbing sets

In the previous section we have shown the existence of a link between absorbing sets and stable partitions. This link is straightforward when the roommate problem is solvable, since each stable partition induces an absorbing set[10]. But this result is not maintained when the roommate problem is unsolvable. In this case from Theorem 1 we know that absorbing sets are determined by stable partitions but, as it is shown in Example 2, stable partitions with odd rings may not yield absorbing sets. These results suggest that we should investigate what the stable partitions determining the absorbing set are. Thus, in this section, we start by characterizing the absorbing sets in terms of stable partitions.

For the characterization pursued we define two types of agents for each stable partition P (hence, the set $\mathcal{A}_P$ is defined): "Dissatisfied" agents who move from one matching to another over the matchings in $\mathcal{A}_P$ without finding a permanent partner, and "satisfied" ones, agents who lack any incentive to change their current partner over these matchings. As we shall see, satisfied agents play a crucial

[10]Tan (1991) establishes the relation between stable matchings and stable partitions.

role in this characterization since the stable partitions determining the absorbing sets are those with the greatest number of them.

The investigation conducted proves to be useful in identifying matchings in absorbing sets. Notice that if P is the stable partition giving rise to the absorbing set $\mathcal{A}_P$ then, by Theorem 1, this set is formed by the set of P-stable matchings and by the matchings that dominate them. The results of this section are illustrated using Example 1.

Our first theorem gives a characterization for absorbing sets in terms of stable partitions. To obtain it, some additional definitions are introduced.

Given a stable partition P, let D_P denote the set of dissatisfied agents that block some matching in $\mathcal{A}_P$, and let $S_P = N \backslash D_P$ be the set of satisfied ones. In Appendix 1 we give an *iterative process* for calculating these two sets. From Remark 3 of this appendix we learn that for any set A of the stable partition P, either $A \subseteq D_P$ or $A \subseteq S_P$.

Let $P\mid_{S_P} = \{A \in P : A \subseteq S_P\}$ denote the stable partition P restricted to the set of satisfied agents S_P. Given that the elements in $P\mid_{S_P}$ are pairs and/or singletons matched in a stable manner, (see again the iterative process in Appendix 1) it is immediate that $P\mid_{S_P}$ is also a stable partition for the roommate problem $(S_P\ (\succ_x)_{x\in Sp})$. Thus, $P\mid_{S_P}$ may be interpreted as a "partial" matching for the roommate problem $(N,(\succ_x)_{x\in N})$.

We denote by $\mathcal{P} = \{P\mid_{S_P}: P$ is a stable partition$\}$ the set of all partial matchings for a roommate problem $(N,(\succ_x)_{x\in N})$. We say that $P\mid_{S_P}$ is maximal in $\mathcal{P}$ if there is not a stable partition P' such that $P\mid_{S_P} \subset P'\mid_{S_{P'}}$.

Theorem 3 *Let $(N,(\succ_x)_{x\in N})$ be a roommate problem. $\mathcal{A}$ is an absorbing set if and only if $\mathcal{A} = \mathcal{A}_P$ for some stable partition P such that $P\mid_{S_P}$ is maximal in $\mathcal{P}$.*

Proof. ($\Longrightarrow$): Let $\mathcal{A}$ be an absorbing set. Then, by Theorem 1, $\mathcal{A} = \mathcal{A}_P$ for some stable partition P. We prove that $P\mid_{S_P}$ is maximal in $\mathcal{P}$. Assume that $P\mid_{S_P}$ is not maximal, *i.e.*, there exists a stable partition P' such that $P\mid_{S_P} \subset P'\mid_{S_{P'}}$. Let μ and μ' be a P-stable matching and a P'-stable matching respectively. Thus, by Lemma 5, $\mu' R^T \mu$. Now, since $\mu \in \mathcal{A}_P$ and $\mathcal{A} = \mathcal{A}_P$ we have $\mu \in \mathcal{A}$. Hence, by Condition (ii) of Definition 2 $\mu' \in \mathcal{A}$. But then, by Condition (i), $\mu R^T \mu'$ and therefore, by Lemma 5, $P'\mid_{S_{P'}} \subseteq P\mid_{S_P}$, contradicting that $P\mid_{S_P} \subset P'\mid_{S_{P'}}$.
($\Longleftarrow$): Let P be a stable partition such that $P\mid_{S_P}$ is maximal in $\mathcal{P}$. We prove that $\mathcal{A}_P$ is an absorbing set, *i.e.*, $\mathcal{A}_P$ satisfies Conditions (i) and (ii) of Definition 2. By Lemma 2, $\mathcal{A}_P = \{\overline{\mu}\} \cup \{\mu \in \mathcal{M} : \mu R^T \overline{\mu}\}$ where $\overline{\mu}$ is a P-stable matching. Let $\mu \in \mathcal{A}_P$. If there exists $\mu' \in \mathcal{M}$ such that $\mu' R \mu$ then $\mu' R^T \overline{\mu}$. Hence $\mu' \in \mathcal{A}_P$ and Condition (ii) follows.
Now we show that $\mathcal{A}_P$ satisfies Condition (i). It suffices to prove that $\overline{\mu} R^T \mu$ for all $\mu \in \mathcal{A}_P$ such that $\mu \neq \overline{\mu}$. If μ is not a P'-stable matching for any stable partition P', by Theorem 1 in Inarra *et al.* (2007), there exists a P'-stable matching μ' such that $\mu' R^T \mu$. Since $\mu R^T \overline{\mu}$ we have $\mu' R^T \overline{\mu}$ (if μ is a P'-stable matching for some stable partition P' then $\mu' = \mu$ can be considered.) Thus, by Lemma 5, $P\mid_{S_P} \subseteq P'\mid_{S_{P'}}$ and since $P\mid_{S_P}$ is maximal in $\mathcal{P}$, it follows that $P\mid_{S_P} = P'\mid_{S_{P'}}$. But then $\overline{\mu} R^T \mu'$ and since $\mu' R^T \mu$ we conclude that $\overline{\mu} R^T \mu$ as desired. ■

As an immediate consequence of the above theorem and Lemma 6, the number of absorbing sets in a roommate problem can be determined straightforwardly.

Corollary 4 *Let $(N,(\succ_x)_{x\in N})$ be a roommate problem. The number of absorbing sets is equal to the number of distinct maximal partitions of $\mathcal{P}$.*

The following theorem specifies a property verified by some partial matchings for the roommate problem $(N,(\succ_x)_{x\in N})$. Specifically, it proves that any two stable partitions that determine two absorbing sets, have the same set of satisfied agents.

Theorem 5 *Let* $(N,(\succ_x)_{x\in N})$ *be a roommate problem. If* P *and* P' *are two stable partitions such that* $P\mid_{S_P}$ *and* $P'\mid_{S_{P'}}$ *are maximal in* $\mathcal{P}$ *then* $S_P = S_{P'}$.

Proof. Suppose, by contradiction, that $S_P \neq S_{P'}$. Then $S_P \cap D_{P'} \neq \emptyset$ or $S_{P'} \cap D_P \neq \emptyset$. We assume, without loss of generality, that $S_P \cap D_{P'} \neq \emptyset$ (otherwise, the argument will be identical except for the roles of P and P', which are interchanged). By Lemma 7, for each $A \in P$ either $A \subseteq D_{P'}$ or $A \subseteq S_{P'}$. Let $P^* = \{A \in P : A \subseteq D_{P'}\} \cup \{A' \in P' : A' \subseteq S_{P'}\}$ be a partition of N. It is easy to verify that P^* is stable. Now we prove that $D_{P^*} \subseteq D_P \cap D_{P'}$. By the iterative process described in Appendix 1, there exists a finite sequence of sets $\langle D_t^* \rangle_{t=0}^{r^*}$ such that:
(i) D_0^* is the union of all odd rings of P^*.
(ii) For $t \geq 1$, $D_t^* = D_{t-1}^* \cup D_t^*$ where $B_t^* = \{b_1^*(t), ..., b_{l_t^*}^*(t)\} \in P^*$ ($l_t^* = 1$ or 2), $B_t^* \nsubseteq D_{t-1}^*$, for which there is a set $A_t^* = \{a_1^*(t), ..., a_{k_t^*}^*(t)\} \in P^*$, $A_t^* \subseteq D_{t-1}^*$ and

$$b_j^*(t) \succ_{a_i^*(t)} a_i^*(t) \text{ and } a_i^*(t) \succ_{b_j^*(t)} b_{j-1}^*(t), \qquad [2]$$

for some $i \in \{1, ..., k_t^*\}$ and $j \in \{1, ..., l_t^*\}$.
Then, the process, $D_{P^*} = D_{r^*}^*$. We prove by induction on t that, for each $t = 0, ..., r^*$, $D_t^* \subseteq D_P \cap D_{P'}$. If $t = 0$, this is trivial. Assume that $t \geq 1$. It is suffices to prove that $B_t^* \subseteq D_P \cap D_{P'}$. By Lemma 7, we only need to show that $b_j^*(t) \in D_P \cap D_{P'}$. Since $A_t^* \subseteq D_{t-1}^*$, by the inductive hypothesis, $a_i^*(t) \in D_P \cap D_{P'}$. Clearly $b_j^*(t) \in D_{P'}$ (otherwise, $B_t^* \in P'$ and since $a_i^*(t) \in D_{P'}$, by [2], $b_j^*(t) \in D_{P'}$). So $B_t^* \in P$ and since $a_i^*(t) \in D_P$, from [2] it follows that $b_j^*(t) \in D_P$, as desired.
Finally, since $D_{P^*} \subseteq D_P \cap D_{P'}$ we have $S_{P'} \cup (S_P \cap D_{P'}) \subseteq S_{P^*}$ and therefore $P'\mid_{S_{P'}} \subset P^*\mid_{S_{P^*}}$, contradicting the maximality of $P'\mid_{S_{P'}}$. ■

The following remark, which follows immediately from Theorem 3 and Theorem 5, states that absorbing sets are determined by those stable partitions with the maximum number of satisfied agents.

Remark 2 *Let* $(N,(\succ_x)_{x\in N})$ *be a roommate problem.* $\mathcal{A}$ *is an absorbing set if and only if* $\mathcal{A} = \mathcal{A}_P$ *for some stable partition* P*, such that* $|S_P| \geq |S_{P'}|$ *for every stable partition* P'.

Therefore, if P is the stable partition yielding the absorbing set $\mathcal{A}_P$, then by Theorem 1 we know that this set is formed by the P-stable matchings and those matchings that dominate them, and from this remark we also know that the set of satisfied agents of these matchings has greater or equal cardinality than the matchings in $\mathcal{A}_{P'}$.

To conclude this section, let us illustrate the above results with the roommate problem from Example 1.

Applying the iterative process in Appendix 1 to the stable partitions of this problem gives the following information: For the stable partition P_1, we have that the sets of dissatisfied and satisfied agents are $D_{P_1} = \{1, 2...., 10\}$ and $S_{P_1} = \emptyset$ respectively. For the stable partitions P_2 and P_3 we have $D_{P_2} = D_{P_3} = \{1, 2, 3, 10\}$ and $S_{P_2} = S_{P_3} = \{4, ..., 9\}$. Hence, the partial matchings of $\mathcal{P}$ are $P_1\mid_{S_{P_1}} = \emptyset$, $P_2\mid_{S_{P_2}} = \{\{4, 8\}, \{5, 9\}, \{6, 7\}\}$ and $P_3\mid_{S_{P_3}} = \{\{4, 9\}, \{5, 7\}, \{6, 8\}\}$. Notice that $P_2\mid_{S_{P_2}}$ and $P_3\mid_{S_{P_3}}$ are the maximal partitions of $\mathcal{P}$ with the greatest set of satisfied agents. Therefore, by Theorem 3 and Corollary 4, this roommate problem has exactly two absorbing sets $\mathcal{A}$ and $\mathcal{A}'$ where $\mathcal{A} = \mathcal{A}_{P_2}$ contains the following matchings: μ_1=[{1},{2, 3},{4, 8},{5, 9},{6, 7},{10}], μ_2=[{2},{1, 3},{4, 8},{5, 9},{6, 7},{10}], μ_3=[{3},{1, 2},{4, 8},{5, 9},{6, 7},{10}] and μ_4=[{1, 3},{2, 10}, {4, 8},{5, 9},{6, 7}] and $\mathcal{A}' = \mathcal{A}_{P_3}$ containing the P_3-stable matchings which are: μ_1' = [{1}, {2, 3},{4, 9}, {5, 7},{6, 8},{10}], μ_2' = [{2}, {1, 3},{4, 9},{5, 7},{6, 8},{10}], μ_3'= [{3},{1, 2},{4, 9},{5, 7},{6, 8},{10}] and μ_4'=[{1, 3},{2, 10},{4, 9},{5, 7},{6, 8}].

5 Structure of matchings in absorbing sets

In this section we investigate the structure of the matchings of absorbing sets. First, we show that all matchings in an absorbing set share certain common features. Furthermore, in the case of a roommate problem with multiple absorbing sets we also find similarities among their matchings.

Let $\mathcal{A}_P$ be an absorbing set associated with the stable partition P and set $\mathcal{A}_P = \mathcal{A}$. Then, as in the previous section, the sets $D_{\mathcal{A}}$ and $S_{\mathcal{A}}$ will denote respectively the sets of dissatisfied and satisfied agents for the absorbing set $\mathcal{A}$.

The following theorem, easily derived from Theorem 3 and Lemma 4, proves that all matchings in an absorbing set $\mathcal{A}$ have some identical pairings formed by the satisfied agents which, in addition, are a stable matching for the roommate problem $(S_{\mathcal{A}}, (\succ_x)_{x\in S_{\mathcal{A}}})$.

Theorem 6 *Let $(N, (\succ_x)_{x\in N})$ be a roommate problem. For any absorbing set $\mathcal{A}$ such that $S_{\mathcal{A}} \neq \emptyset$ the following conditions hold:*
(i) For any $\mu \in \mathcal{A}$, $\mu(S_{\mathcal{A}}) = S_{\mathcal{A}}$ and $\mu\mid_{S_{\mathcal{A}}}$ is stable for $(S_{\mathcal{A}}, (\succ_x)_{x\in S_{\mathcal{A}}})$.
(ii) For any μ, $\mu' \in \mathcal{A}$, $\mu\mid_{S_{\mathcal{A}}} = \mu'\mid_{S_{\mathcal{A}}}$.

For an illustration of the result above see Example 1 at the end of Section 4.

Next, we investigate the structure of absorbing sets in case of multiplicity. For this purpose, some additional definitions are required. Given an absorbing set $\mathcal{A}$ such that $D_{\mathcal{A}} \neq \emptyset$, let $\mathcal{A}\mid_{D_{\mathcal{A}}} = \{\mu\mid_{D_{\mathcal{A}}} : \mu \in \mathcal{A}\}$ denote the set of "partial" matchings of the absorbing set $\mathcal{A}$ restricted to the set of dissatisfied agents $D_{\mathcal{A}}$. Analogously, if $S_{\mathcal{A}} \neq \emptyset$, let $\mathcal{A}\mid_{S_{\mathcal{A}}} = \{\mu\mid_{S_{\mathcal{A}}} : \mu \in \mathcal{A}\}$. The following theorem shows that there are similarities among matchings belonging to different absorbing sets.

Theorem 7 *Let $(N, (\succ_x)_{x\in N})$ be a roommate problem. For any two absorbing sets $\mathcal{A}$ and $\mathcal{A}'$, the following conditions hold:*
(i) $D_{\mathcal{A}} = D_{\mathcal{A}'}$ and $S_{\mathcal{A}} = S_{\mathcal{A}'}$.
(ii) $\mathcal{A}\mid_{D_{\mathcal{A}}} = \mathcal{A}'\mid_{D_{\mathcal{A}'}}$.
(iii) $\mathcal{A}\mid_{S_{\mathcal{A}}}$ and $\mathcal{A}'\mid_{S_{\mathcal{A}'}}$ are singletons consisting of a stable matching in $(S_{\mathcal{A}}, (\succ_x)_{x\in S_{\mathcal{A}}})$, where $S = S_{\mathcal{A}} = S_{\mathcal{A}'}$.

Proof. Let $\mathcal{A}$ and $\mathcal{A}'$ be two absorbing sets. Then, by Theorem 3, there are stable partitions P and P' such that $\mathcal{A} = \mathcal{A}_P$, $\mathcal{A}' = \mathcal{A}_{P'}$ where $P\mid_{S_P}$ and $P'\mid_{S_{P'}}$ are maximal in $\mathcal{P}$.
(i) Since $S_{\mathcal{A}} = S_P$ and $S_{\mathcal{A}'} = S_{P'}$ and, by Theorem 5, $S_P = S_{P'}$, then $S_{\mathcal{A}} = S_{\mathcal{A}'}$. Therefore $D_{\mathcal{A}} = D_{\mathcal{A}'}$.
(ii) It is very easy to verify that $\mathcal{A}\mid_{D_{\mathcal{A}}}$ and $\mathcal{A}'\mid_{D_{\mathcal{A}'}}$ are absorbing sets in $(D, (\succ_x)_{x\in D})$ where $D = D_{\mathcal{A}} = D_{\mathcal{A}'}$ such that $\mathcal{A}\mid_{D_{\mathcal{A}}} = \mathcal{A}_{P\mid_{D_P}}$ and $\mathcal{A}'\mid_{D_{\mathcal{A}'}} = \mathcal{A}_{P'\mid_{D_{P'}}}$. Since $S_{P\mid_{D_P}} = S_{P'\mid_{D_{P'}}} = \emptyset$, from Lemma 6, we conclude that $\mathcal{A}\mid_{D_{\mathcal{A}}} = \mathcal{A}'\mid_{D_{\mathcal{A}'}}$.
(iii) This follows directly from Theorem 6. ■

Thus, for a roommate problem $(N, (\succ_x)_{x\in N})$, all its absorbing sets have the following coincidences: (i) The set of dissatisfied agents is the same for all matchings across all absorbing sets and so is the set of satisfied agents. (ii) The roommate problem of the dissatisfied agents $(D, (\succ_x)_{x\in D})$ has a unique absorbing set. (iii) Satisfied agents form stable matchings for the roommate problem $(S, (\succ_x)_{x\in S})$. Hence, the two absorbing sets $\mathcal{A}$ and $\mathcal{A}'$ only differ in how the satisfied agents are matched.

The three conditions above provide all absorbing sets of a roommate problem with strict preferences with a similar structure, as illustrated in Figure 1.

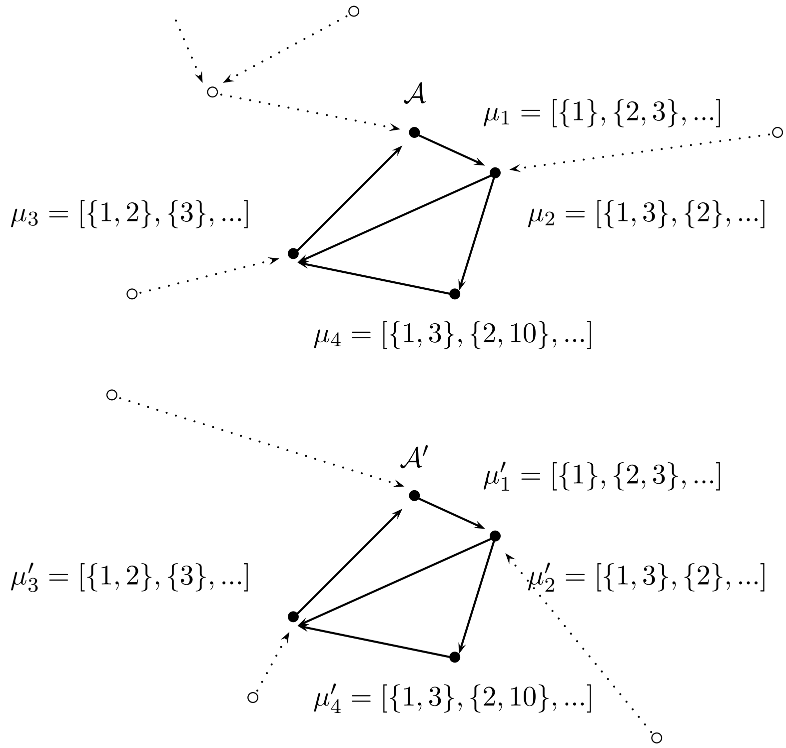


Figure 1.- The two absorbing sets of the roommate problem in Example 1.

To explain this last result, consider the two absorbing sets from Example 1: $\mathcal{A} = \mathcal{A}_{P_2}$ and $\mathcal{A}' = \mathcal{A}_{P_3}$. Since $D_{\mathcal{A}} = D_{\mathcal{A}'} = \{1,2,3,10\}$ we have $\mathcal{A} = \{\mu_1,\mu_2,\mu_3,\mu_4\}$ and $\mathcal{A}' = \{\mu'_1,\mu'_2,\mu'_3,\mu'_4\}$ where μ_1,μ_2,μ_3 are the P_2-stable matchings and μ'_1,μ'_2,μ'_3 are the P_3-stable matchings (see Figure 1). Additionally, $\mathcal{A}\mid_{D_{\mathcal{A}}} = \{\mu_1\mid_{D_{\mathcal{A}}},\mu_2\mid_{D_{\mathcal{A}}},\mu_3\mid_{D_{\mathcal{A}}},\mu_4\mid_{D_{\mathcal{A}}}\}$ and $\mathcal{A}'\mid_{D_{\mathcal{A}'}} = \left\{\mu'_1\mid_{D_{\mathcal{A}'}},\mu'_2\mid_{D_{\mathcal{A}'}},\mu'_3\mid_{D_{\mathcal{A}'}},\mu'_4\mid_{D_{\mathcal{A}'}}\right\}$ where $\mu_1\mid_{D_{\mathcal{A}}} = \mu'_1\mid_{D_{\mathcal{A}'}} = [\{1\},\{2,3\}]$, $\mu_2\mid_{D_{\mathcal{A}}} = \mu'_2\mid_{D_{\mathcal{A}'}} = [\{1,3\},\{2\}]$, $\mu_3\mid_{D_{\mathcal{A}}} = \mu'_3\mid_{D_{\mathcal{A}'}} = [\{1,2\},\{3\}]$ and $\mu_4\mid_{D_{\mathcal{A}}} = \mu'_4\mid_{D_{\mathcal{A}'}} = [\{1,3\},\{2,10\}]$. Furthermore, $\mathcal{A}\mid_{S_{\mathcal{A}}}$ and $\mathcal{A}'\mid_{S_{\mathcal{A}'}}$ are respectively singletons consisting of the stable matchings $\mu = [\{4,8\},\{5,9\},\{6,7\}]$ and $\mu' = [\{4,9\}, \{5,7\}, \{6,8\}]$ in $(S,(\succ_x)_{x\in S})$ where $S = \{4,5,6,7,8,9\}$.

Appendix 1

An iterative process to determine the sets of dissatisfied and satisfied agents

Given a stable partition P (hence the set $\mathcal{A}_P$ is immediately defined) the process determines the set D_P, which is formed by those dissatisfied agents that block some matching in $\mathcal{A}_P$ and the set S_P formed by those satisfied agents that do not block any matching in $\mathcal{A}_P$.

The set D_P can be determined by an iterative process in a finite number of steps. To that end, we define inductively a sequence of sets $\langle D_t\rangle_{t=0}^{\infty}$ as follows:

(i) for $t = 0$, D_0 is the union of all odd rings of P.
(ii) for $t \geq 1$, $D_t = D_{t-1} \cup B_t$ where $B_t = \{b_1(t),...,b_{l_t}(t)\} \in P$ ($l_t = 1$ or 2), $B_t \nsubseteq D_{t-1}$, and there is a set $A_t = \{a_1(t),...,a_{k_t}(t)\} \in P$ such that $A_t \subseteq D_{t-1}$ and

$$b_j(t) \succ_{a_i(t)} a_i(t) \text{ and } a_i(t) \succ_{b_j(t)} b_{j-1}(t), \qquad [3]$$

for some $i \in \{1,...,k_t\}$ and $j \in \{1,...,l_t\}$.[11]

[11] If no such set exists then $D_t = D_{t-1}$.

Given that P contains a finite number of sets, then $D_t = D_{t-1}$ for some t. Let r be the minimum number such that $D_{r+1} = D_r$. Then $D_r = D_P$[12].

From this iterative process the following remark easily follows.

Remark 3 *For any set $A \in P$, either $A \subseteq D_P$ or $A \subseteq S_P$.*

To illustrate the iterative process above, consider the stable partition $P_1 = \{\{1,2,3\},\{4,7\},\{5,8\},\{6,9\},\{10\}\}$ of Example 1. Note that P_1 contains a unique odd ring. Then $D_0 = \{1,2,3\}$. Let $B_1 = \{10\}$ and $A_1 = \{1,2,3\}$. Since $10 \succ_2 2$ and $2 \succ_{10} 10$, then $D_1 = D_0 \cup B_1 = \{1,2,3,10\}$. Let $B_2 = \{4,7\}$ and $A_2 = \{1,2,3\}$. Since $7 \succ_1 1$ and $1 \succ_7 4$, then $D_2 = D_1 \cup B_2 = \{1,2,3,10,4,7\}$. Consider now the sets $B_3 = \{5,8\}$ and $A_3 = \{4,7\}$. As $8 \succ_4 4$ and $4 \succ_8 5$, then $D_3 = D_2 \cup B_3 = \{1,2,3,10,4,7,5,8\}$. Finally, let $B_4 = \{6,9\}$ and $A_4 = \{5,8\}$. Since $9 \succ_5 5$ and $5 \succ_9 6$, then $D_4 = D_3 \cup B_4 = \{1,2,3,10,4,7,5,8,6,9\}$ and the process is completed. Hence $D_{P_1} = D_4$. Repeating the process for $P_2 = \{\{1,2,3\},\{4,8\},\{5,9\},\{6,7\},\{10\}\}$ and $P_3 = \{\{1,2,3\}, \{4,9\},\{5,7\},\{6,8\},\{10\}\}$ we have $D_{P_2} = D_{P_3} = \{1,2,3,10\}$. Therefore the sets of satisfied agents are $S_{P1} = \emptyset$ and $S_{P_2} = S_{P_3} = \{4,5,6,7,8,9\}$

Appendix 2[13]

Lemma 1 *Given a stable partition P. For any two distinct P-stable matchings μ and μ', $\mu' R^T \mu$.*

Lemma 2 *Let P be a stable partition and $\overline{\mu}$ be a P-stable matching. Then, $\mathcal{A}_P = \{\overline{\mu}\} \cup \{\mu \in \mathcal{M} : \mu R^T \overline{\mu}\}$.*

Lemma 3 *Let P be a stable partition. Then, there exists $\mu^* \in \mathcal{A}_P$ such that*

$$\mu^*(x) = \begin{cases} x & \text{if } x \in D_P \backslash D_0 \\ \\ \overline{\mu}(x) & \text{otherwise,} \end{cases}$$

where $\overline{\mu}$ is a P-stable matching.

Lemma 4 *Let P be a stable partition such that $S_P \neq \emptyset$. The following conditions hold:*
(i) For any $\mu \in \mathcal{A}_P$, $\mu(S_P) = S_P$ and $\mu\mid_{S_P}$ is stable for $(S_P, (\succ_x)_{x \in S_P})$.
(ii) For any μ, $\mu' \in \mathcal{A}_P$, $\mu\mid_{S_P} = \mu'\mid_{S_P}$.

Lemma 5 *Let P and P' be two distinct stable partitions and let μ and μ' be a P-stable matching and a P'-stable matching respectively. Then, $\mu' R^T \mu$ if and only if $P\mid_{S_P} \subseteq P'\mid_{S_{P'}}$.*

Lemma 6 *Let P and P' be two stable partitions. $\mathcal{A}_P = \mathcal{A}_{P'}$ if and only if $P\mid_{S_P} = P'\mid_{S_{P'}}$.*

Lemma 7 *Let P and P' be two stable partitions. Then for each $A \in P$ either $A \subseteq D_{P'}$ or $A \subseteq S_{P'}$.*

6 References

ABELEDO, H., and G. ISAAK (1991): "A Characterization of Graphs that Ensure the Existence of Stable Matchings." *Mathematical Social Sciences*, 22, 93-96.

ABRAHAM, D., BIRÒ, P., and D. MANLOVE (2005): "Almost Stable Matchings in the Roommates Problem." Paper ID: 7953. In Proceedings of WAOA 2005: the 3rd Workshop on Approximation and Online Algorithms, volume 3879 of Lecture Notes in Computer Science, 1-14.

[12] Proof upon request

[13] Proofs upon request

BIRÒ, P., CECHLÀROVÀ, K., and T. FLEINER (2008): "The Dynamics of Stable Matchings and Half-matchings for the Stable Marriage and Roommates Problems." *International Journal of Game Theory*, 36, 333-352.
CHUNG, K-S. (2000): "On the Existence of Stable Roommate Matchings." *Games and Economic Behaviour*, 33, 206-230.
DIAMANTOUDI, E., MIYAGAWA, E., and L. XUE (2004): "Random Paths to Stability in the Roommate Problem." *Games and Economic Behaviour*, 48, 18-28.
DIAMANTOUDI, E., and L. XUE (2003): "Farsighted Stability in Hedonic Games." *Social Choice and Welfare*, 21, 39-61.
EHLERS, L. (2007): "Von Neumann -Morgenstern Stable Sets in Matching Problems." *Journal of Economic Theory*, 134, 537-547.
GALE, D., and L. SHAPLEY (1962): "College Admissions and the Stability of Marriage". *American Mathematical Monthly*: 69, 9-15.
INARRA, E., LARREA, C., and E. MOLIS (2008): "Random Paths to P-stability in the Roommate Problem." *International Journal of Game Theory*, 36, 461-471.
IRVING, R. (1985): "An Efficient Algorithm for the Stable Roommates Problem." *Journal of Algorithms,* 6, 577-595.
KALAI, E., PAZNER, E. A., and D. SCHMEIDLER (1976): "Admissible Outcomes of Social Bargaining Processes as Collective Choice Correspondence." *Econometrica*, 63, 299-325.
KALAI, E., and D. SCHMEIDLER (1977): "An Admissible Set Occurring in Various Bargaining Situations." *Journal of Economic Theory*, 14, 402-411.
KELSO, A. S., and V. CRAWFORD (1982): "Job Matching, Coalition Formation and Gross Substitutes." *Econometrica*, 50, 1483-1504.
KLAUS, B., and F. KLIJN (2008): "Smith and Rawls Share a Room: Stability and Medians." Mimeo (March).
PAGE, F., WOODERS, M., and S. KAMAT (2005): "Networks and Farsighted Stability" *Journal of Economic Theory*, 120, 257-261.
PITTEL, B. G., and R. W. IRVING (1994): "An Upper Bound for the Solvability Probability of a Random Stable Roommates Instance." *Random Structures and Algorithms*, 5, 465-486.
ROTH, A. (1976): "Subsolutions and the Supercore of Cooperative Games." *Mathematics of Operations Research*, 1, 43-49.
ROTH, A. (1985): "The College Admissions Problem is not Equivalent to the Marriage Problem." *Journal of Economic Theory*, 36, 277-288.
ROTH, A., and M. SOTOMAYOR (1990): Two-Sided Matching: A Study in Game-theoretic Modeling and Analysis. Econometric Society Monograph Series. New York: Cambridge University Press.
ROTH, A., and J. H. VANDE VATE (1990): "Random Paths to Stability in Two-Sided Matching." *Econometrica*, 58, 1475-1480.
SHENOY, P. (1979): "On Coalition Formation: A Game-Theoretical Approach." *International Journal of Game Theory*, 8, 133-164.
SCHWARTZ, T. (1970): "On the Possibility of Rational Policy Evaluation." *Theory and Decision*, 1, 89-106.
TAN, J. J. M. (1990): "On a maximum stable matching problem." *BIT,* 29, 631-640.
TAN, J. J. M. (1991): "A Necessary and Sufficient Condition for the Existence of a Complete Stable Matching." *Journal of Algorithms*, 12, 154-178.
VON NEUMANN, J. and O. MORGENSTERN (1947): Theory of Games and Economic Behavior. Princeton: Princeton University Press, NJ.

An algorithm for a super-stable roommates problem

Tamás Fleiner* Robert W. Irving† David F. Manlove‡

June 6, 2008

Abstract

In this paper we describe an efficient algorithm that decides if a stable matching exists for a generalized stable roommates problem, where, instead of linear preferences, agents have partial preference orders on potential partners. Furthermore, we may forbid certain partnerships, that is, we are looking for a matching such that none of the matched pairs is forbidden, and yet, no blocking pair (forbidden or not) exists.

To solve the above problem, we generalize the first algorithm for the ordinary stable roommates problem.

1 Introduction

The study of stable matching problems were initiated by Gale and Shapley [3] who introduced the *stable marriage problem*. In this problem each of n men and n women have a linear preference order on the members of the opposite gender. We ask if there exists a marriage scheme in which no man and woman mutually prefer one another to their eventual partners. The authors prove that the so called deferred acceptance algorithm always finds a stable marriage scheme.

It is natural to ask the same question for a more general, nonbipartite (sometimes called: one sided) model, in which we have n agents with preference orders on all other agents. This is the so called *stable roommates problem*, and we are looking for a matching (i.e. a pairing of the agents) such that no two agents prefer one another to their eventual partners. Such a matching is called a *stable matching*. A significant difference between the stable marriage and the stable roommates problems is that for the latter, it might happen that no stable matching exists. The stable roommates problem was solved by Irving [4], with an efficient algorithm that either finds a stable matching or concludes that no stable matching exists for the particular problem. Later, Tan [8] used this algorithm to give a good characterization, that is, he proved that for any stable roommates problem, there always exists a so called *stable partition* (that can be regarded as a half integral, fractional stable matching) with the property that either it is a stable matching, or it is a compact proof for the nonexistence of a stable matching.

*Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Magyar Tudósok körútja 2, Budapest, H-1117. fleiner@cs.bme.hu Research was supported by the János Bolyai Research fellowship of the Hungarian Academy of Sciences, by the Royal Society of Edinburgh International Exchange Programme, and by the K69027 and K60802 OTKA grants.

†Department of Computing Science University of Glasgow G12 8QQ, UK. rwi@dcs.gla.ac.uk Supported by the Engineering and Physical Sciences Research Council grant EP/E011993/1.

‡Department of Computing Science University of Glasgow G12 8QQ, UK. davidm@dcs.gla.ac.uk Supported by the Royal Society of Edinburgh / Scottish Executive Personal Research Fellowship, and Engineering and Physical Sciences Research Council grant GR/R84597/01 and EP/E011993/1.

In both the stable marriage and the stable roommates problems strict preferences of the participating agents play a crucial role. However, in many practical situations, we have to deal with indifferences in the preference orders. Our model for this is that preference orders are partial (rather than linear) orders. We can extend the notion of a stable matching to this model in at least three different ways. One possibility is that a matching is *weakly stable* if no pair of agents a, b exists such that they mutually strictly prefer one another to their eventual partner. Ronn proved that deciding the existence of a weakly stable matching is NP-complete [6]. A more restrictive notion is that a matching is *strongly stable* if there are no agents a and b such that a strictly prefers b to his eventual partner and b does not prefer his eventual partner to a. Scott gave an algorithm that finds a strongly stable matching or reports if none exists in $O(m^2)$ time [7]. The most restrictive notion is that of super-stability. A matching is *super-stable* if there exist no two agents a and b such that neither of them prefers his eventual situation to being a partner of the other. In other words, a matching is super-stable, if it is stable for any linear extensions of the preference orders of the agents. For the case where indifference is transitive, Irving and Manlove gave an $O(m)$ algorithm to find a super-stable matching, if exists [5]. Interestingly, the algorithm has in two phases, just like Irving's [4], but its second phase is completely different. It is also noted there that the algorithm works without modification for the more general poset case.

The motivation of our present work is to give a direct algorithm to this kind of stable matching problem by generalizing Irving's original algorithm. This latter algorithm works in such a way that it keeps on deleting edges of the underlying graph until a (stable) matching is left. It turnes out that deleting an edge is too harsh a transformation, we need a finer one as well. For this reason, we extend our model and we also allow forbidden edges. And, instead of deleting, we will also forbid certain edges during the algorithm. Although a stable matching problem with forbidden edges is a special case of the poset problem (for each forbidden edge add a parallel copy and declare them equal in the preference orders), it is an interesting problem in itself. Dias *et al.* gave an $O(m)$ algorithm to the stable marriage problem with forbidden pairs [1].

Our present problem, the super-stable matching problem with forbidden edges is known to be polynomial-time solvable. Fleiner *et al.* exhibited a reduction of this problem to 2-SAT [2]. However, this reduction does not give much information about the structure of super-stable matchings. In particular, it is not obvious if there exists a "short proof" for the nonexistence of a super-stable matching, just like Tan's stable partition [8] works for the ordinary stable roommates problem. Our direct approach may be useful to find such a certificate.

To formalize our problem, we define a *preference model* as a triple $(G, F, \mathcal{O})$, where $G = (V, E)$ is a graph, the set F of *forbidden edges* is a subset of the edge set E of G, and $\mathcal{O} = \{<_v : v \in V\}$, where $<_v$ is a partial order on the star $E(v)$ of v (that is, the set of those edges of G that are incident with vertex v). It is convenient to think that we deal with a market situation: vertices of G are the acting agents and edges of G represent possible partnerships between them. Partial order $<_v$ is the preference order of agent v on his possible partnerships. Parallel edges are allowed in G: the same two agents may form different partnerships, that may yield different profits for them. A subset M of E is a *matching* if edges of M do not share a vertex, that is, each agent participates in at most one partnership. Matching M is *stable* (we omit the super prefix for convenience), if $M \subseteq E \setminus F$ (in other words, no edge of M is forbidden, that is, all edges of M are *free*), and if each edge e of E is *dominated by* M, that is, if $e \in M$ or there is an edge $m \in M$ and a vertex $v \in V$ such that $m <_v e$. If M is a matching and e is not dominated by M then e is a *blocking edge of* M. The *stable roommates problem with partial orders and forbidden pairs* is the decision problem on an input preference model whether it has a stable matching or not.

Note that in the standard terminology, agents have preferences on possible partners, rather than on partnerships. It is easy to see that in our approach, this corresponds to the case where graph G in the preference model is simple. We also have a slightly different way of defining stability via dominance. Traditionally, we first define the notion of blocking and then we say that a stable matching is a matching that has no blocking edge. Also note that the stable roommates problem is the special case where G is simple, $F = \emptyset$, and each order $<_v$ is linear.

2 The generalized algorithm

Let us fix a preference model $(G_0, F_0, \mathcal{O}_0)$, as the input of our algorithm. We should find a stable matching, if it exists. The algorithm works step by step. In each step, it transforms the actual model $(G_i, F_i, \mathcal{O}_i)$ to a simpler model $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ in such a way that the answer to the latter problem is a valid answer to the former one, as well. That is, after the transformation no new stable matching can emerge and if there was a stable matching in the former model, then there should also be one in the new model. We use three kind of transformations: we forbid edges, we delete forbidden edges and we restrict the model.

If e is a free edge of G_i, then *forbidding* e means that $G_{i+1} := G_i$, $F_{i+1} := F_i \cup \{e\}$ and $\mathcal{O}_{i+1} := \mathcal{O}_i$. The algorithm may forbid e if either no stable matching contains e or if e is not contained in all stable matchings. After such a forbidding, there is a stable matching in $(G_i, F_i, \mathcal{O}_i)$ if and only if there is one in $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$, and any stable matching of $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ is a stable matching of $(G_i, F_i, \mathcal{O}_i)$. Forbidding a subset E' of E means that we simultaneously forbid all edges of E'.

If e is a forbidden edge of G_i then *deleting* e means that we delete e from G_i to get G_{i+1}, $F_{i+1} := F_i \setminus \{e\}$, and the partial orders in $\mathcal{O}_{i+1}$ are the restrictions of the corresponding partial orders of $\mathcal{O}_i$, to the corresponding stars of G_{i+1}. The algorithm may delete e if there exists no matching in $(G_i, F_i, \mathcal{O}_i)$ that is blocked only by e. This implies that the set of stable matchings in $(G_i, F_i, \mathcal{O}_i)$ and in $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ is the same.

If U is a proper subset of the vertex set of G_i then *restriction to* U means that G_{i+1} is the graph we get from G_i after deleting all vertices outside U, F_{i+1} is the subset of F_i that is spanned by G_{i+1}, and the partial orders of $\mathcal{O}_{i+1}$ are the restricted partial orders of $\mathcal{O}_i$ to the corresponding stars of G_{i+1}.

We shall use different kinds of steps throughout the algorithm. There is a certain hierarchy of them: the next step of the algorithm always has the highest priority among those steps that can be executed. To describe these step types, we say that edge $e = E_i(v)$ of G_i (forbidden or not) is a *first choice edge of* v, if there is no edge $f \in E_i(v) \setminus F_i$ with $f <_v e$ (i.e., if no free edge can dominate e at vertex v). Note that there can be more than one 1st choices of v present.

0th priority (proposal) step If $e = vw$ is a 1st choice of v then orient e from v to w, and $(G_i, F_i, \mathcal{O}_i) = (G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$.

Clearly, the set of stable matchings does not change by a proposal step. We shall call the 1st choice arcs we create by the proposal steps *1-arcs*. Note that it is possible that a 1-arc is bioriented.

After the algorithm have found all 1-arcs, it looks for a

1st priority (mild rejection) step If 1-arc e of G_i points to v and $E_i(v) \ni f \not<_v e$ (that is, f is not better than e according to v in G_i) then forbid f.

Obviously, if f is in some matching M then $e \notin M$, and hence e (being a first choice at its other end) blocks M. So f cannot be in a stable matching, we can forbid it. Eventually, we have to delete edges and the algorithm does this only the following way.

2nd priority (firm rejection) step If some free 1-arc e of G_i points to v and $e <_v f \in E_i(v)$ (e is better than f according to v in G_i) then we delete f.

Note that the above f is already forbidden by a 1st priority step. Assume that f blocks matching M, hence, in particular, $e \notin M$. But e is a first choice of its other endvertex, thus e is also blocking M. So deleting f does not change the set of stable matchings of the preference model.

Note that the so called 1st phase steps in Irving's algorithm [4] for the stable roommates problem are the special cases of our proposal and firm rejection steps. It is true for the stable roommates problem that as soon as no more 1st phase steps can be executed, the preference model has the so called first-last property: if some edge $e = uv$ is a first choice of u, then e is the last choice of v. A generalization of this property holds in our setting. Assume that the algorithm cannot execute a 0th, 1st or 2nd priority step for $(G_i, F_i, \mathcal{O}_i)$. Let V_0 denote the set of those vertices of G_i that are not incident with any free edges, V_1 stand for the set of those vertices of G_i that are incicent with a bioriented free 1-arc and V_2 refer to the set of the remaining vertices of G_i. The following properties are true.

Theorem 1. *Assume that no proposal or rejection step can be made in G_i, and let V_0, V_1 and V_2 be defined as above.*

If $v \in V_1 \cup V_2$ then there is a unique 1*-arc entering v and there is a unique* 1*-arc leaving v, and all these* 1*-arcs are free. There is no edge of G_i leaving V_0. Bioriented free* 1*-arcs form a matching M_1 that covers V_1, and no more edges are incident with V_1 in G_i.*

M is a stable matching of $(G_i, F_i, \mathcal{O}_i)$ if and only if the following properties hold: (1) each vertex of V_0 is isolated and (2) $M_1 \subseteq M$ and (3) $M \setminus M_1$ is a stable matching of the model restricted to V_2.

Proof. Let $v \in V_1 \cup V_2$. By definition, there is at least one free edge incident with v, hence there is at least one free 1-arc leaving v. On the other hand, no proposal or rejection step (mild or firm) can be made in G_i, hence at most one free 1-arc enters v. By definition, no free 1-arc enter vertices of V_0, and this means that 1-arcs leaving vertices of $V_1 \cup V_2$ enter this very same vertex set. Consequently, there is a unique free 1-arc leaving and entering each vertex of $V_1 \cup V_2$. Can there be a forbidden 1-arc e incident with a vertex v of $V_1 \cup V_2$? The answer is no: such an arc cannot enter v, as otherwise v would be able to reject. So $e = uv$ is a 1-arc from $V_1 \cup V_2$ to V_0. But v is not incident with any free arcs by definition, thus vu is a 1-arc that enters vertex u of $V_1 \cup V_2$, contradiction. Hence all 1-arcs that are incident with $V_1 \cup V_2$ are free.

Let $u \in V_0$ and $e = uv$ be an edge of G_i. Clearly e is a 1-arc and e is forbidden by the definition of V_0, so $v \in V_0$ holds. This means that all edges incident with a vertex of V_0 are completely inside V_0.

If v is in V_1 then there is a unique 1-arc a that leaves v, so a must bioriented by the definition of V_1. If $e = uv$ is an edge of G_i then either $e = a$ or e is not a first choice of v, hence $a \prec_v e$ holds. But in this case v should delete e in a firm rejection step as a is a 1-arc entering v. This argument shows that edges of G_i that are incident with V_1 are all bioriented and form a matching M_1 covering V_1.

Assume now that M is a stable matching of G_i. No edge of G_i incident with a vertex of V_0 can block M, hence V_0 consists of isolated vertices. As M is not blocked by an edge of M_1, edges of M_1 all belong to M. As there is no edge of G_i that leaves V_2, edges of M in V_2 form a stable matching of the restricted model to V_2.

Let now M_2 be a stable matching of the model restricted to V_2 and assume that V_0 consists of isolated vertices. Let $M := M_2 \cup M_1$. Clearly M is a matching. If some edge e blocks M then e cannot be incident with V_0, as these vertices are isolated, and e cannot have a vertex in V_1 either, as vertices of V_1 are only incident with edges of M_1. Hence e is an edge within V_2, contradicting to the fact that M_2 is a matching. □

If some vertex of V_0 is not isolated then the algorithm stops and concludes that no stable matching exists. If this is not the case, then another possibility is that $V_2 = \emptyset$. This case the algorithm stops, and reports that there is a stable matching. To construct one, the algotithm takes M_1 and completes it to a stable matching of the original preference model with the previously listed other matchings of type M_1. Theorem 1 justifies both these terminations. If none of the above cases hold then $V_2 \neq \emptyset$ and we make a

3rd priority (restriction) step: if $V_0 \cup V_1 \neq \emptyset$ then we restrict the model to V_2. By Theorem 1, it is enough to find a stable matching for the restricted G_{i+1}: if there is such a matching M', then $M' \cup M_1$ is a stable matching of G_i. If no stable matching exists after the restriction, then there was no stable matching even before it.

Assume that in $(G_i, F_i, \mathcal{O}_i)$, the algorithm can execute no 0th, 1st or 2nd or 3rd priority step. An edge $e \in E_i(v)$ is a *second choice of* v if $e >_v f \notin F$ implies that f is *the* 1st choice of v. In other words, e is a second choice, if the only free edge that dominates e at v is the unique 1-arc leaving v. Note that every vertex v of G_i is incident with at least one free second choice edge: in the "worst case" it is the unique 1-arc pointing to v.

4th priority step If $e = vw$ is a second choice of v then (counterintuitively) orient e from w to v. Arcs created at this step are called *2-arcs*. As we do not modify the preference model ($G_{i+1} = G_i, F_{i+1} = F_i$ and $\mathcal{O}_{i+1} = \mathcal{O}_i$), the set of stable matchings does not change by a 4th priority step.

What is the meaning of a 2-arc? Let, vv' and uu' be 1-arcs and $u'v$ be a 2-arc. As vv' is the only free edge dominating $u'v$ at v, we get that if uu' is present in a stable matching M then uu' does not dominate uv', hence $vv' \in M$ follows. In other words, 2-arcs represent implications on 1-arcs. This allows us to build an implication structure on the set of 1-arcs.

In this structure, two 1-arcs e and f are called *sm-equivalent*, if there is a directed cycle D formed by 1-arcs and 2-arcs in an alternating manner such that D contains both e and f. (Note that D may use the same vertex more than once.) Sm-equivalence is clearly an equivalence relation and if C is an sm-class and M is a stable matching then either C is disjoint from M or C is contained in M.

Beyond determining sm-equivalence classes, 2-arcs yield further implications between sm-classes: if uu' is a 1-arc of sm-class C and vv' is a 1-arc of sm-class C' and $u'v$ is a 2-arc, then sm-class C "implies" sm-class C' in such a way that if C is not disjoint from stable matching M then M contains both classes C and C'. Assume that sm-class C is on the top of this implication structure, i.e. C is not implied by any other sm-class (but C may imply certain other classes). Formally, we have that

$$\text{if } vv' \text{ is a 1-arc of } C \text{ and } w'v \text{ is a 2-arc} \qquad (1)$$
$$\text{then (the unique) 1-arc } ww' \text{ is sm-equivalent to } vv'.$$

To find a top sm-class C, introduce an auxiliary digraph on the vertices of G_i, such that if uu' is a 1-arc and $u'v$ is a 2-arc, then we introduce an arc uv of the auxiliary graph. It is well known that by depth first search, we can find a source strong component of the auxiliary graph in linear time. If it contains vertices $u_1, u_2, \ldots, u_k$ then it determines a top sm-class $C = \{u_1u_1', u_2u_2', \ldots, u_ku_k'\}$ formed by 1-arcs. Note that it is possible here that $u_i = u_j'$ for different i and j.

5th priority step If for 1-arcs $u_iu_i', u_ju_j' \in C$ there are 2-arcs vu_i and vu_j with $vu_i \not<_v vu_j$ then forbid vu_i.

To justify this step, assume that $vu_i \in M$ for some stable matching M of G_i. As vu_i does not dominate vu_j, vu_j has to be dominated at u_j by $u_ju_j' \in M$. As u_iu_i'

and $u_j u'_j$ are sm-equivalent, this means that $u_j u'_j$ also belongs to M, a contradiction. So vu_i does not belong to any stable matching and after forbidding it, the set of stable matchings does not change. Note that after we take a 5th priority step, new 2-arcs may be created so we might continue with a 4th priority step.

6th priority step Forbid all edges of C in $(G_i, F_i, \mathcal{O}_i)$.

To justify this kind of step, we check two cases. Case 1 is that C is not a matching, that is, $u_i = u'_j$ for some $i \neq j$. As a subset of a matching is a matching, no matching (hence no stable matching) can contain C. So by sm-equivalence, C is disjoint from any stable matching of G_i, and forbidding C is not changing the set of stable matchings.

Case 2 is that C is a matching. Each u_i is adjacent to at least two free edges: the incoming and the outgoing 1-arcs. So each u_i receives at least one free 2-arc. This free 2-arc must come from some u'_j by property (1). Let C' denote the set of free 2-arcs of the form $u'_j u_i$. As we have seen, each u_i receives at least one arc of C', hence $|C'| \geq k$. As we cannot execute any more 5th priority steps in $(G_i, F_i, \mathcal{O}_i)$, from each u'_j there is at most one arc of C' leaving, implying $|C'| \leq k$. This means that $|C'| = k$ and each u_i receives exactly one arc of C' and each u'_i sends exactly one arc of C'. As sets $\{u_1, u_2, \ldots, u_k\}$ and $\{u'_1, u'_2, \ldots, u'_k\}$ are disjoint, this means that set C' forms a perfect matching on vertices $u_1, u'_1, u_2, u'_2, \ldots, u_k, u'_k$.

Let M be a stable matcing of $(G_i, F_i, \mathcal{O}_i)$. If M is disjoint from C then M is stable in $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$ as well. Otherwise, by sm-equivalence, M contains all edges of C and disjoint from C'. We claim that $M' := M \setminus C \cup C'$ is another stable matching of $(G_i, F_i, \mathcal{O}_i)$ and hence it is a stable matching of $(G_{i+1}, F_{i+1}, \mathcal{O}_{i+1})$, as well.

Indeed: M' is a matching, as C and C' cover the same set of vertices. Each edge $u_i u'_i$ is dominated at u'_i by M' by Theorem 1. Each forbidden 2-arc of type $u'_j u_i$ is dominated at u'_j by the 5th priority step. For the remaining edges, if some edge e does not have a vertex u_i then e is dominated the same way in M' as in M. Otherwise, if u_i is a vertex of e then e is neither a first nor a second choice of u_i as we have already checked these edges. This means that the free 2-arc pointing to u_i is dominating e, so C' and thus M' also dominates e at u_i.

Clearly, this 6th priority step corresponds to the so called rotation elimination of Irving's algorithm [4], where $C \cup C'$ is the generalization of a rotation.

If the algorithm does not stop after some 2nd priority step with the conclusion that no stable matching exists then it keeps on forbidding and deleting edges. Sooner or later it cannot do this any more, so no further step can be made. Pick a vertex v of the actual G_i. As no 3rd priority step is possible, there is a free edge adjacent to v. So v sends a free 1-arc, and it also receives a free 1-arc. Again by the 3rd priority step, these arcs are different, hence there is a 2-arc pointing to v. This implies that a 5th or a 6th priority step can be executed, a contradiction. So the algorithm always terminates before a 3rd priority step either by concluding that no stable matching exists or by constructing a stable matching.

To convince ourselves about the polynomial time complexity of the algorithm let us calculate the cost of deleting or forbidding an edge. Clearly, the most time consuming is the 6th priority deletion step. For this we check every edge for the 1st and 2nd priority steps in $O(m)$ time (where m is the number of edges of G_0), and we check all vertices in $O(n)$ time for the 3rd priority step. (n is the number of vertices of G_0.) To check the possible 4th priority steps takes $O(m)$ time, and finding top sm-class C is a depth first search, that can be done in $O(n + m)$ time. Checking the 5th priority steps takes $O(m)$ time, and after this we can forbid C. So forbidding or deleting an edge takes altogether $O(n + m)$ time. We can delete or forbid at most $2m$ times altogether, so the total complexity of our algorithm is $O(m(n+m))$. (Note that this is a pretty rough estimate. Probably, by streamlining the algorithm, one can get a much better estimate.)

References

[1] Vânia M. F. Dias, Guilherme D. da Fonseca, Celina M. H. de Figueiredo, and Jayme L. Szwarcfiter, The stable marriage problem with restricted pairs, *Theoret. Comput. Sci.* (2003) **306**(1-3) 391–405

[2] Tamás Fleiner, Robert W. Irving, and David F. Manlove, Efficient algorithms for generalised stable marriage and roommates problems, *Theoret. Comput. Sci.* (2007) **381**(1-3) 162–176 and DCS Tech Report, TR-2005-207, `http://www.dcs.gla.ac.uk/publications/` (2005)

[3] D. Gale and L.S. Shapley, College admissions and stability of marriage, *Amer. Math. Monthly* (1962) **69**(1) 9–15

[4] Robert W. Irving, An efficient algorithm for the "stable roommates" problem, *J. Algorithms* (1985) **6**(4) 577-595

[5] Robert W. Irving and David F. Manlove, The stable roommates problem with ties, *J. Algorithms* (2002) **43**(1) 85–105

[6] Eytan Ronn, NP-complete stable matching problems, *J. Algorithms* (1990) **11**(2) 285–304

[7] Sandy Scott, The study of stable marriage problems with ties, 2005, PhD dissertation, University of Glasgow, Department of Computing Science.

[8] Jimmy J. M. Tan, A necessary and sufficient condition for the existence of a complete stable matching, *J. Algorithms* (1991) **12**(1) 154–178

Faster Algorithms For Stable Allocation Problems

Brian C. Dean
School of Computing
Clemson University
bcdean@cs.clemson.edu

Siddharth Munshi
School of Computing
Clemson University
smunshi@cs.clemson.edu

June 10, 2008

Abstract

We consider a high-multiplicity generalization of the classical stable matching problem known as the *stable allocation problem*, introduced by Baiou and Balinski in 2002. By leveraging new structural properties and sophisticated data structures, we show how to solve this problem in $O(m \log n)$ time on a bipartite instance with n vertices and m edges, improving the best known running time of $O(mn)$. Our approach simplifies the algorithmic landscape for this problem by providing a common generalization of two different approaches from the literature — the classical Gale-Shapley algorithm, and a recent algorithm of Baiou and Balinski. Building on this algorithm, we provide an $O(m \log n)$ algorithm for the non-bipartite stable allocation problem. Finally, we give a polynomial-time algorithm for solving the "optimal" variant of the bipartite stable allocation problem, as well as a 2-approximation algorithm for the NP-hard "optimal" variant of the non-bipartite stable allocation problem.

1 Introduction

The classical stable matching (marriage) problem has been extensively studied since its introduction by Gale and Shapley in 1962 [5]. Given n men and n women, each of whom submits an ordered preference list over all members of the opposite sex, we seek a matching between the men and women that is *stable* — having no man-woman pair (m, w) (known as a *blocking pair* or a *rogue couple*) where both m and w would both be happier if they were matched with each-other instead of their current partners. Gale and Shapley showed how to solve the problem optimally in $O(n^2)$ time using a simple and natural "propose and reject" algorithm, and over the years we have come to understand a great deal about the rich mathematical and algorithmic structure of this problem and its many variants (e.g., see [6, 8]).

In this paper we study a high-multiplicity variant of the stable matching problem known as the *stable allocation problem*, introduced by Baiou and Balinski in 2002 [2]. This problem follows in a long line of "many-to-many" generalizations of the classical stable matching problem. The many-to-one *stable admission* problem [9] has been used since the 1950s in a centralized national program in the USA known as the *National Residency Matching Program* (NRMP) to assign medical school graduates to residencies at hospitals; here, we have a bipartite instance with unit-sized elements (residents) on one side and capacitated non-unit-sized elements (hospitals) on the other. In 2000, Baiou and Balinski [1] studied what one could call the stable bipartite b-matching problem, where both sides of our bipartite graph contain elements of non-unit size, and each element i has a specified quota $b(i)$ governing the number of elements on the other side of the graph to which it should be matched. The stable allocation problem is a further generalization of this problem where the amount of assignment between two elements i and j is no longer zero or one, but a nonnegative real number (we will give a precise definition of the problem in a moment). The stable allocation problem is also known as the *ordinal transportation problem* since it can be viewed as a variant of the classical transportation problem where the quality of an assignment is specified in terms of ranked preference lists and stability instead of absolute numeric costs. This can be a useful model in practice since in many applications, ranked preference lists are often easy to obtain while there may not be any reasonable way to specify exact numeric assignment costs; for example, it may be obvious that it is preferable to process a certain job on machine A rather than machine B, even though there is no natural way to assign specific numeric costs to each of these alternatives.

In the literature, there are two prominent algorithms for solving the stable allocation problem. The first is a natural generalization of the Gale-Shapley (GS) algorithm that issues "batch" proposals and rejections. Although this algorithm tends to run quite fast in practice, often even in sublinear time, its worst-case running time is exponential [4]. Baiou and Balinksi (BB) propose what one could view as an "end-to-end" variant of the GS algorithm (we will describe both algorithms in detail in a moment), with worst-case running time $\Theta(mn)$ on a bipartite instance with n vertices and m edges. In this paper we develop an algorithm that generalizes both the GS and BB approaches and uses additional structural properties as well as dynamic tree data structures to achieve a worst-case running time of $O(m \log n)$, which is only a factor of $O(\log n)$ worse than the optimal linear running time we can achieve for the much simpler unit stable matching problem. Note that since the fastest known algorithms for solving high-multiplicity "flow-based" assignment problems run in $\Omega(mn)$ worst-case time, our new results now provide a significant algorithmic incentive to model assignment problems as stable allocation problems rather than flow problems.

Building on our new algorithm, we also provide an $O(m \log n)$ algorithm for the *non-bipartite* stable allocation problem, a natural generalization of the non-bipartite unit stable matching problem (commonly called the *stable roommates* problem). In the book of Gusfield and Irving on the stable marriage problem [6], one of the open questions posed by the authors is whether or not there exists a convenient transformation from the non-bipartite stable roommates problem to the simpler bipartite stable matching problem. We show that a transformation of this flavor does indeed exist, and that it simplifies the construction of algorithms not only for stable roommates but also for the non-bipartite stable allocation problem. It also provides a simple proof of the well-known fact that although an integer-valued solution may not always exist for the stable roommates problem, a half-integral solution does always exist.

The Gale-Shapley algorithm for the unit stable matching problem finds a stable solution that is "man-optimal, woman-pessimal", where each man ends up paired with the best partner he could possibly have in any stable matching, and each woman ends up with the worst partner she could possibly have in any stable assignment (by symmetry, we obtain a "woman-optimal, man-pessimal" matching if the women propose instead of the men). In order to rectify this asymmetry, Gusfield et al. [7] developed a polynomial-time algorithm for the *optimal* stable matching problem, where we associate a cost with each (man, woman) pairing and ask for a stable matching of minimum total cost (costs are typically designed so that the resulting solution tends to be "fair" to both sexes). Bansal et al. [3] extended this approach to the optimal stable bipartite b-matching problem, and we show how to extend it further to solve the optimal stable allocation problem in polynomial time.

As a consequence, we also obtain a 2-approximation algorithm for the NP-hard "optimal" variant of the non-bipartite stable allocation problem by generalizing a similar 2-approximation algorithm for the optimal stable roommates problem.

2 Preliminaries

In order to eliminate any awkwardness associated with multiple-partner matchings involving men and women, let us assume we are matching I jobs indexed by $[I] = \{1, \ldots, I\}$ to J machines indexed by $[J] = \{1, \ldots, J\}$. Each job i has an associated processing time $p(i)$, and each machine j has a capacity $c(j)$. The jobs and machines comprise the left and right sides of a bipartite graph with $n = I + J$ vertices and m edges. Let $N(i)$ denote the set of machines to which job i is adjacent in this graph, and similarly let $N(j)$ denote the set of jobs that are neighbors of machine j. For each edge (i, j) we associate an upper capacity $u(i, j) \leq \min(p(i), c(j))$ governing the maximum amount of job i that can be assigned to machine j. Later on, we will also associate a cost $c(i, j)$ with edge (i, j). Problem data is not assumed to be integral (see [4] for further notes on the issue of integrality in stable allocation problems).

Each job i submits a ranked preference list over machines in $N(i)$, and each machine j submits a ranked preference list over jobs in $N(j)$. If job i prefers machine $j \in N(i)$ to machine $j' \in N(i)$ or if $j \in N(i)$ and $j' \notin N(i)$, then we we write $j >_i j'$; similarly, we say $i >_j i'$ if machine j prefers job i to job i'. Preference lists are strict, containing no ties. Letting $x(i, j)$ denote the amount of job i assigned to machine j, we say the entire assignment $x \in \mathbf{R}^m$ is *feasible* if it satisfies

$$\begin{array}{ll} x(i, [J]) = p(i) & \forall i \in [I] \\ x([I], j) = c(j) & \forall j \in [J] \\ 0 \leq x(i, j) \leq u(i, j) & \forall \text{ edges } (i, j), \end{array}$$

where we denote by $x(S, T)$ the sum of $x(i, j)$ over all $i \in S$ and $j \in T$. In order to ensure that a feasible solution always exists, we assume job 1 and machine 1 are both "dummy" elements with very large respective processing times and capacities, which we set so that $p(1) = c([J] - \{1\})$ and $p([I]) = c([J])$. The preference list of job 1 should contain all machines in arbitrary order, ending with machine 1, and the preference list of machine 1 should contain all jobs in an arbitrary order, ending with job 1. We can regard a job or machine that ends up being assigned to a dummy as being unassigned in our original instance.

An edge (i, j) is said to be a *blocking pair* for assignment x if $x(i, j) < u(i, j)$, there exists a machine $j' <_i j$ for which $x(i, j') > 0$, and there exists a job $i' <_j i$ for which $x(i', j) > 0$. Informally, (i, j) is a blocking pair if $x(i, j)$ has room to increase, and both i and j can be made happier by increasing $x(i, j)$ in exchange for decreasing some of their current lesser-preferred allocations. An assignment x is said to be *stable* if it is feasible and admits no blocking pairs. Note that the dummy job can never be part of a blocking pair, and neither can the dummy machine.

One can show that a stable assignment exists for any problem instance. Moreover, there always exists a unique stable assignment that is *job-optimal*, where an assignment is job-optimal if the vector describing the allocation of each job i (ordered by i's preference list) is lexicographically maximal over all possible stable assignments. By symmetry, a unique *machine-optimal* assignment always exists as well. As it turns out, a job-optimal assignment is always machine-pessimal and vice-versa. It is also a well-known fact that the dummy allocations $x(1, j)$ and $x(i, 1)$ are the same

Advance-q(i):	Advance-r(j):
While $x(i,j) = u(i,j)$ or q_i not accepting of i: Step q_i downward in i's preference list.	**While** $x(r_j, j) = 0$: Step r_j upward in j's preference list. Advance-q(r_j)

in every stable assignment.

Given any assignment x, we define r_j to be the job $i \in N(j)$ with $x(i,j) > 0$ that is least preferred by j. Job r_j is the job that j would logically choose to reject first if it were offered an allocation from a more highly-preferred job. If $i >_j r_j$, then we say machine j is *accepting* for job i, since j would be willing to accept some additional allocation from i in exchange for rejecting some of its current allocation from r_j. For each job i, we let q_i be the machine j most preferred by i such that $x(i,j) < u(i,j)$ and j is accepting for i. If i wishes to increase its allocation, q_i is the first machine it should logically ask.

The Gale-Shapely (GS) Algorithm. The GS algorithm for the stable allocation problem is a natural generalization of the well-studied GS "propose and reject" algorithm for the unit stable matching problem. The analysis of this algorithm will help us to analyze the correctness and running time of our new algorithm to follow.

Although the GS algorithm typically starts with an empty assignment, we start with an assignment x where every machine j is fully assigned to the dummy job ($x(1,j) = c(j)$), and the remaining jobs are unassigned — this simplifies matters somewhat since every machine except the dummy henceforth remains fully assigned. In each iteration of the algorithm, we select a arbitrary job i that is not yet fully assigned; let $T = p(i) - x(i,[J])$ be the amount of i's processing time that is currently unassigned. Job i "proposes" $T' = \min(T, u(i,j) - x(i,j))$ units of processing time to machine $j = q_i$, which accepts. However, if j is any machine except the dummy, then it is now overfilled by T' units beyond its capacity, so it proceeds to reject T' units, starting with job r_j. During the process, $x(r_j, j)$ may decrease to zero, in which case r_j becomes a new job higher on j's preference list and rejection continues until j is once again assigned exactly $c(j)$ units. The algorithm terminates when all jobs are fully assigned, and successful termination is ensured by the fact that each job can send all of its processing time to the dummy machine as a last resort.

Consider briefly the behavior of the q_i's and r_j's during the GS algorithm. We regard q_i as a pointer into job i's preference list that starts out pointing at i's first choice and over time scans monotonically down i's preference list according to the Advance-q procedure above, which is automatically called any time an edge (i, q_i) becomes saturated ($x(i,j) = u(i,j)$). Similarly, r_j is a pointer into machine j's preference list that starts at job 1 (the dummy, which is the least-preferred job on j's list), and over time advances up the list according to the Advance-r procedure, which is automatically called any time an edge (r_j, j) becomes empty. Note that all of this "pointer management" takes only $O(m)$ total time over the entire GS algorithm. We use exactly the same pointer management infrastructure in our new algorithm.

Lemma 1. *Irrespective of proposal order, the GS algorithm for the stable allocation problem always terminates in finite time (even with irrational problem data), and it does so with a stable assignment that is job-optimal and machine-pessimal.*

Proof. It is well-known (see, e.g., [6]) that for the classical unit stable matching problem, the

GS algorithm always terminates with a man-optimal (job-optimal) stable assignment. This result easily extends to the stable allocation problem if we have integral problem data and no upper edge capacities, since in this case the GS algorithm can be viewed as a "batch" version of the classical GS algorithm executed on the unit instance we obtain when we split each job i into $p(i)$ unit jobs and each machine j into $c(j)$ unit machines. However, this reduction no longer applies if we have irrational problem data or upper edge capacities. In this case, finite termination is shown in [4]. To show that our final assignment is stable, suppose at termination that (i, j) is a blocking pair. Since $q_i <_i j$, we know j must have rejected i at some point; however, this implies that $r_j \leq_j i$, contradicting our assumption that j has any allocation it prefers less than i. To show that our assignment is job-optimal, suppose it is not. At some point during execution, there must have been a rejection from some machine j to some job i that resulted in an assignment x with $x(i, j) < x^*(i, j)$, where x^* is a stable assignment satisfying $x(i, j') \geq x^*(i, j')$ for all $j' >_i j$. Consider the first point in time when such a rejection occurs, and let x denote our assignment right after this rejection. Since $x(i, j) < x^*(i, j)$ and since j is fully assigned in both x and x^*, there must be some i' for which $x(i', j) > x^*(i', j)$. Note that $i' >_j i$, since otherwise j would have rejected i' fully before rejecting i. Since $x(i', j) > x^*(i', j)$ and $x(i', [J]) \leq x^*(i', [J])$, there must be some machine j' such that $x(i', j') < x^*(i', j')$; let j' be the first such machine in the preference list of i'. We know $j >_{i'} j'$ since otherwise i' would have already been rejected by j', contradicting the fact that (i, j) is the earliest instance of a rejection of the type considered above. Since $x^*(i', j) < u(i', j)$, this implies that (i', j) is a blocking pair in x^*, contradicting our assumption that x^* was stable. The argument showing that our final assignment is machine-pessimal is analogous and completely symmetric to this job-optimality argument. □

Lemma 2. *For each edge (i, j), as the GS algorithm executes, $x(i, j)$ will never increase again after it experiences a decrease.*

Proof. This is also shown in [4], and it follows easily as a consequence of the monotonic behavior of the q_i and r_j pointers: $x(i, j)$ increases as long as $q_i = j$, stopping when q_i advances past j, which happens either when (i, j) becomes saturated, or when r_j advances to i. From this point on, $x(i, j)$ decreases until r_j advances past i, after which $x(i, j) = 0$ forever. □

Corollary 3. *During the execution of the GS algorithm, each edge (i, j) becomes saturated at most once, and it also becomes empty at most once.*

In practice, the GS algorithm often runs quite fast; for example, in the common case where all jobs get one of their top choices, the algorithm usually runs in sublinear time. Unfortunately, the worst-case running time can be exponential even on relatively simple problem instances [4].

3 An Improved "Augmenting Path" Algorithm

In this section, we describe our $O(m \log n)$ algorithm for the stable allocation problem and show how it generalizes and improves upon the GS algorithm and the algorithm of Baiou and Balinski (BB), which we describe shortly. Just like the GS algorithm, our algorithm starts with an assignment x in which every job but the dummy is unassigned, and every machine is fully assigned to the dummy job. As the algorithm progresses, the machines remain fully assigned and the jobs become progressively more assigned. The algorithm terminates when every job is fully assigned.

At any given point in time during the execution of the GS algorithm (say, where we have built up some partial assignment x), we define $G(x)$ to be a bipartite graph on the same set of vertices as our original instance, having edges (i, q_i) for all $i \in [I]$ and (r_j, j) for all $j \in [J] - \{1\}$. Initially $G(x)$ is a tree, containing n vertices, $n-1$ edges, and no cycles; we regard the dummy machine (the only machine j without an incident (r_j, j) edge) as the root of this tree.

Lemma 4. *For every assignment x we obtain during the course of the GS algorithm, $G(x)$ consists of a collection of disjoint components, the one containing the root vertex (the dummy machine) being a tree and each of the others containing one unique cycle.*

Proof. Consider any connected component C of $G(x)$ spanning job set I' and machine set J'. If $1 \in J'$ (i.e., if C contains the root), then C has $|I'| + |J'| - 1$ edges and must therefore be a tree. Otherwise, C has $|I'|+|J'|$ edges, so it consists of a tree plus one additional cycle-forming edge. □

We say a component in $G(x)$ is *fully assigned* if $x(i, [J]) = p(i)$ for each job i in the component. As we run our algorithm, we maintain the structure of the tree and cycle components in $G(x)$ along with a list of jobs in each component that are not yet fully assigned. In each iteration of our algorithm, we select an arbitrary component C of $G(x)$ that is not fully assigned and perform an *augmentation* within C. We terminate when every component is fully assigned.

An augmentation consists of a simultaneously-enacted series of proposals and rejections along a path or cycle that can be viewed as the "end to end" execution of a series of GS operations. In the tree component, an augmentation starts from any job i that is not yet fully assigned, and follows the unique path from i to the root (i.e.., i proposes to $j = q_i$, which rejects $i' = r_j$, which proposes to $j' = p_{i'}$, and so on, just as the GS algorithm would operate, until we reach machine 1, which is the only machine that accepts a proposal without issuing a subsequent rejection). Along our augmenting path from i to the root, we increase the assignment of each (i, q_i) edge and decrease the assignment of each (r_j, j) edge by the same amount. For a cycle component, we augment along the unique cycle within the component, increasing the assignment on (i, q_i) edges and decreasing the assignment on (r_j, j) edges by the same amount.

We define the *residual capacity* of an edge (i, j) in $G(x)$ as $r(i, j) = u(i, j) - x(i, j)$ for an (i, q_i) edge, and $r(i, j) = x(i, j)$ for an (r_j, j) edge. The residual capacity $r(\pi)$ of an augmenting path/cycle π is defined as $r(\pi) = \min\{r(i, j) : (i, j) \in \pi\}$. When we augment along an augmenting path π starting from job i, we push exactly $\min(r(\pi), p(i) - x(i, [J]))$ units of assignment along π, since this is just enough to either make i fully assigned, or to saturate or make empty one of the edges along π. When we augment along a cycle π, we push exactly $r(\pi)$ units of assignment, since this suffices to saturate or empty out some edge along π, thereby "breaking" the cycle π. When one or more edges along π become saturated or empty, this triggers any appropriate calls to our pointer management infrastructure above, resulting in a change to the structure of $G(x)$ because one or more of the q_i or r_j pointers advances. In general, any time one of these pointers advances, one edge leaves $G(x)$ and another enters: if some pointer q_i advances to q'_i, then (i, q_i) leaves G and (i, q'_i) enters, and if r_j advances to r'_j then (r_j, j) leaves and (r'_j, j) enters. The net impact of each of these modifications is either (i) the tree component splits into a tree and a cycle component, (ii) the tree component and some cycle component merge into a tree component, (iii) one cycle component splits into two cycle components, or (iv) two cycle components merge into one cycle component.

In order to augment efficiently, we store each component of $G(x)$ in a dynamic tree data structure (see [10, 11]). For cycle components, we store a dynamic tree plus one arbitrary edge along the cycle.

This allows us to find the residual capacity along an augmenting path/cycle as well as augment on the path/cycle in $O(\log n)$ time (amortized time is also fine), in much the same way dynamic trees are used push flow along augmenting paths when solving maximum flow problems. Since dynamic trees can handle *split* and *join* operations in $O(\log n)$ time, we can also efficiently maintain the structure of the components of $G(x)$ as edges are removed and added. In total, we spend $O(\log n)$ time for each edge removal (split) and edge addition (join), and since edges are removed at most once (when saturated or emptied) and added at most once, this contributes $O(m \log n)$ to our total running time. Each augmentation takes $O(\log n)$ time and either saturates an edge, empties out an edge, or fully assigns some job, all three of which can only happen once per edge/job. We therefore perform at most $2m + n = O(m)$ augmentations, for a total running time of $O(m \log n)$. Since we are performing in an aggregate fashion a set of proposals and rejections that the original GS algorithm *could* have performed, Lemma 1 tells us that our algorithm must terminate with a stable assignment that is job-optimal and machine-pessimal.

One might wish to think of our algorithm as either an "end to end" variant of the GS algorithm, or as a more sophisticated implementation of the algorithm of Baiou and Balinski [2], which performs augmentations in a similar but much slower fashion ($O(n)$ time per augmentation, leading to a worst-case running time of $\Omega(mn)$, as shown in Appendix A). The key to our approach is the use of dynamic trees to augment quickly, owing to our new structural insight involving the decomposition of the $G(x)$ graph. In addition to unifying the algorithmic landscape for the stable allocation problem, our approach also exposes a remarkable similarity between state-of-the-art approaches based on dynamic trees for solving our problem and the related maximum flow problem.

4 The Optimal Stable Allocation Problem

Once our algorithm from the previous section terminates with a stable, job-optimal assignment x, the graph $G(x)$ may still contain cycle components. The augmenting cycles in these components are known in the unit stable matching literature as *rotations*, and they generalize readily to the case of stable allocation. Rotations lie at the heart of a rich mathematical structure underlying the stable allocation problem, and they give us a means of describing and moving between all different stable assignments for an instance.

Lemma 5. *Let x be a stable assignment with a cycle component C in $G(x)$, where π_C is the unique augmenting cycle in C. We obtain another stable assignment when we augment any amount in the range $[0, r(\pi_C)]$ around π_C.*

Lemma 6. *If x is a stable assignment, then x is the machine-optimal (and job-pessimal) assignment if and only if $G(x)$ has no cycles (i.e., $G(x)$ consists of a single tree component).*

For space considerations, we leave the (fairly mechanical) proofs of these lemmas for the full version of this paper. If we augment $r(\pi_C)$ units around the rotation π_C, we say that we *eliminate* π_C, since this causes one of the edges along π_C to saturate or become empty, thereby eliminating π_C permanently from $G(x)$. The resulting structural change to $G(x)$ might *expose* new rotations that were not initially present in $G(x)$. Note that the structure of $G(x)$ only changes when we eliminate (fully apply) π_C, and not when we push less than $r(\pi_C)$ units around π_C.

Suppose we start with the job-optimal assignment and continue running the same algorithm from the previous section to eliminate all rotations we encounter, in some arbitrary order, until we finally

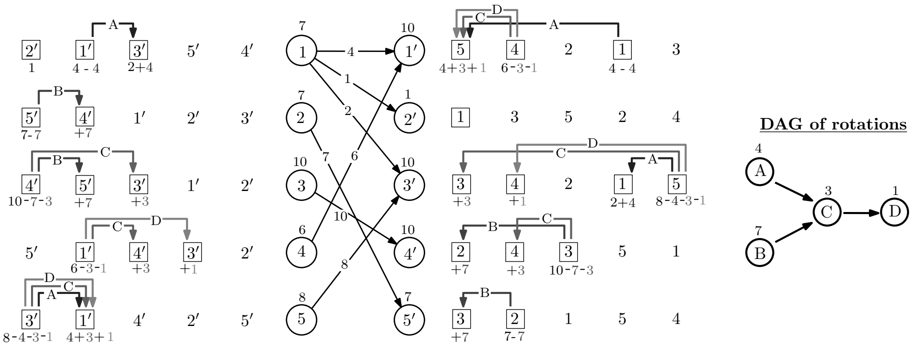


Figure 1: An example bipartite instance and its rotations. No dummy job or machine is shown, since once we reach the job-optimal assignment neither of these takes part in any further augmenting cycles (rotations). The initial job-optimal assignment is shown in the bipartite region, and rotations $A \ldots D$ and their associated multiplicities are shown in the DAG on the right. The net effect of each rotation on our assignment is shown with a set of arrows over the preference lists.

reach the machine-optimal assignment (the only stable assignment with no further exposed rotations). We call this a *rotation elimination ordering*. Somewhat surprisingly, as with the unit case, one can show that irrespective of the order in which we eliminate rotations, we always encounter *exactly the same set* of rotations along the way.

Lemma 7. *Let π be a rotation with initial residual capacity r encountered in some rotation elimination ordering. Then π appears with initial residual capacity r in every elimination ordering.*

Since each elimination saturates or empties an edge, we conclude that there are at most $2m$ combinatorially-distinct rotations that can ever appear in $G(x)$, where each such rotation π has a well-defined initial residual capacity $r(\pi)$ (we will also call this the *multiplicity* of π). Let Π denote the set of these rotations. For any $\pi_i, \pi_j \in \Pi$, we say $\pi_i \prec \pi_j$ if π_j cannot be exposed unless π_i is fully applied at an earlier point in time; that is, $\pi_i \prec \pi_j$ if π_i precedes π_j in every rotation elimination ordering. For example, if π_i and π_j share any job or machine in common, then since simultaneously-exposed rotations are vertex disjoint it must be the case that $\pi_i \prec \pi_j$ or $\pi_j \prec \pi_i$; otherwise, we could find a rotation elimination ordering in which π_i and π_j are both exposed at some point in time. We can extend this argument to show that for any job i (machine j) we must have $\pi_1 \prec \pi_2 \prec \ldots \prec \pi_k$, where $\pi_1 \ldots \pi_k$ are the rotations containing job i (machine j), appropriately ordered. Clearly, Π also contains no cycle $\pi_1 \prec \pi_2 \prec \ldots \prec \pi_k \prec \pi_1$, since otherwise none of the rotations $\pi_1 \ldots \pi_k$ could ever be exposed. Let us therefore construct a directed acyclic graph $D = (\Pi, E)$ where $(\pi_i, \pi_j) \in E$ if $\pi_i \prec \pi_j$. An example of this *rotation DAG* is shown in Figure 1.

For our purposes, it will be sufficient to compute a "reduced" rotation DAG $D' = (\Pi, E')$ with $E' \subseteq E$ whose transitive closure is D. To do this, we run our algorithm from Section 3 to obtain a job-optimal assignment, then we continue running it until we have generated the set of all rotations Π. This takes $O(m \log n)$ time, although $O(mn)$ time is needed if we actually wish to write down the structure of each rotation along the way. We then use the observation above to generate the $O(mn)$ edges in E' in $O(mn)$ time as follows: for each job i (machine j), compute the set of rotations $\pi_1 \ldots \pi_k$ containing i (j), ordered according to the order in which they were eliminated.

We then add the $k-1$ edges $(\pi_1, \pi_2) \ldots (\pi_{k-1}, \pi_k)$ to E'.

The (reduced) rotation DAG has been instrumental in the unit case (see, e.g., [6]) in characterizing the set of all stable matchings for an instance. Conveniently, we can generalize this to the stable allocation problem. Let us call the vector $y \in \mathbf{R}^{|\Pi|}$ *D-closed* if $y(\pi) \in [0, r(\pi)]$ for each rotation $\pi \in \Pi$, and $y(\pi_i) = r(\pi_i)$ if there is an edge $(\pi_i, \pi_j) \in E$ with $y(\pi_j) > 0$. The vector y tells us the extent to which we should apply each rotation in an elimination ordering that follows a topological ordering of D. The D-closed property ensures that we fully apply any rotation π_i upon which another rotation π_j depends. Note that D'-closed means the same thing as D-closed, since D and D' share the same transitive closure (we will give a more detailed discussion of this fact in the full version of this paper).

Lemma 8. *For any instance of the stable allocation problem, there is a one-to-one correspondence between all stable assignments x and all D-closed (D'-closed) vectors y.*

Consider now the *optimal* stable allocation problem: given a cost $c(i,j)$ on each edge (i,j) in our original instance, we wish to find a stable assignment x minimizing $\sum_{ij} x(i,j)c(i,j)$. Using Lemma 8, we can solve this problem in polynomial time the same way we can solve the optimal variant of the unit stable matching problem. Note that an optimal stable assignment corresponds to a subset of fully-applied rotations — that is, a D'-closed vector y with $y(\pi) \in \{0, r(\pi)\}$ for each $\pi \in \Pi$. By assigning each rotation $\pi \in \Pi$ cost indicating the net cost of fully applying π, we can transform the optimal stable allocation problem into an equivalent minimum-cost closure problem on the DAG D'; for further details, see [7].

5 The Non-Bipartite Stable Allocation Problem

In the *non-bipartite stable allocation problem*, we are given an n-vertex, m-edge graph $G = (V, E)$ where every vertex $v \in V$ has an associated size $b(v)$ and a ranked preference list over its neighbors, and every edge $e \in E$ has an associated upper capacity $u(e)$. Letting $I(v)$ denote the set of edges incident to v, our goal is to compute an assignment $x \in \mathbf{R}^m$ with $\sum_{e \in I(v)} x(e) = b(v)$ for all $v \in V$ that is stable in that it admits no blocking pair. Here, a blocking pair is an edge $e = uv \in E$ such that $x(e) < u(e)$ and both u and v would prefer to increase $x(e)$ while decreasing some of their other allocations.

In the unit case (with $b(v) = 1$ for all $v \in V$), with the added restriction that x must be integer-valued, this is known as the *stable roommates* problem, and it can be solved in $O(m)$ time. As a consequence of the integrality restriction, one can construct instances that have no stable integer-valued solution for the roommates problem. However, no such difficulties arise with the non-bipartite stable allocation problem since it is inherently a "real-valued" problem; we also ensure a solution always exists by adding uncapacitated self-loops to all vertices, and by placing each vertex last on its own preference list. Just as with the dummy job and machine in the bipartite case, we can regard a vertex assigned to itself as actually being unassigned in the original instance, and one can show that the extent to which each vertex is unassigned must be the same in every stable assignment.

In response to an open question posed by Gusfield and Irving [6] on whether or not there exists a convenient transformation from the stable roommates problem to the simpler stable matching problem, we show that a transformation of this flavor does indeed exist, and that it simplifies

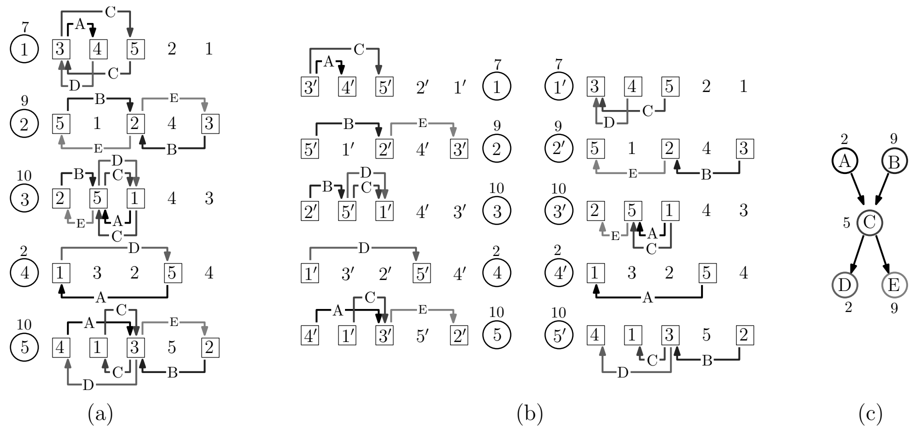


Figure 2: Transforming a non-bipartite instance (a) into a symmetric bipartite instance (b) with rotation DAG (c). Rotations in the resulting bipartite instance are shown overlaid on the non-bipartite instance. A symmetric stable assignment for the bipartite instance is obtained by eliminating A, B, and half of C.

the construction of algorithms not only for stable roommates but also for the non-bipartite stable allocation problem. Suppose we construct a symmetric bipartite instance by replicating a non-bipartite instance, as shown in Figure 2. If we can find a *symmetric* stable assignment x for this symmetric instance (with $x(u, v) = x(v, u)$ for each edge $e = uv$), then by setting $x(e) = x(u, v) = x(v, u)$ we will obtain a stable solution to the non-bipartite instance (since if there was a blocking pair in the non-bipartite solution, this would imply an analogous blocking pair in the bipartite solution). Hence, to solve the non-bipartite problem, we need only consider how to find a *symmetric solution* to a *symmetric instance* of the bipartite problem. One can always do this by carefully choosing the right combination of rotations to apply, starting from the job-optimal assignment.

Due to symmetry, rotations in our bipartite instance now tend to come in pairs. In the example shown in Figure 2(a) we have taken the left-hand-side and right-hand-side effect of each rotation and overlaid these on the original non-bipartite instance. From this, we can see that rotations A and D are mirror images, or *duals*, of each-other, as are rotations B and E. More precisely, rotations π and π' are duals if π is the symmetric analog of π' when we reverse the roles of the left-hand and right-hand sides of our bipartite instance. Note that $r(\pi) = r(\pi')$ if π and π' are duals. The rotation C is its own dual, so we call it a *self-dual* rotation.

Lemma 9. *Consider any symmetric bipartite instance. There is a one-to-one correspondence between symmetric stable assignments x and D-closed (D'-closed) vectors y where $y(\pi) + y(\pi') = r(\pi)$ for every dual pair of rotations (π, π') and $y(\pi) = r(\pi)/2$ for every self-dual rotation.*

This lemma gives another simple proof of the (previously-known) fact that a 1/2-integral solution always exists for the stable roommates problem, and it also leads us to an $O(m \log n)$ algorithm for the non-bipartite stable allocation problem: transform into a symmetric bipartite instance, compute the job-optimal stable assignment, then eliminate rotations starting from the job-optimal assignment, taking care not to eliminate the dual of any rotation previously eliminated (there are several ways to accomplish this; for example, we can store a hash of each eliminated rotation).

Finally, eliminate half of the remaining self-dual rotations, leaving a symmetric stable assignment. Complete implementation details will appear in the full version of this paper.

The "optimal" (i.e., minimum-cost) version of the non-bipartite stable allocation problem is NP-hard since it generalizes the NP-hard optimal stable roommates problem. The only difference between the two lies in the self-dual rotations. For the stable roommates problem, the existence of a self-dual rotation is precisely what prevents the existence of an integral stable solution, so we must assume there are no self-dual rotations in our instance. For the non-bipartite stable allocation problem, we are forced to take half of each self-dual rotation, thereby removing them from consideration as well. The remaining problem now looks the same in both cases: find an optimal D-closed set of rotations containing one of each dual pair. For this NP-hard problem, a 2-approximation algorithm can be obtained via a reduction to a weighted 2SAT problem [6], so the same technique also gives us a 2-approximation for the optimal bipartite stable allocation problem.

References

[1] M. Baiou and M. Balinski. Many-to-many matching: Stable polyandrous polygamy (or polygamous polyandry). *Discrete Applied Mathematics*, 101:1–12, 2000.

[2] M. Baiou and M. Balinski. Erratum: The stable allocation (or ordinal transportation) problem. *Mathematics of Operations Research*, 27(4):662–680, 2002.

[3] V. Bansal, A. Agrawal, and V. S. Malhotra. Polynomial time algorithm for an optimal stable assignment with multiple partners. *Theoretical Computer Science*, 379(3):317–328, 2007.

[4] B.C. Dean, N. Immorlica, and M.X. Goemans. Finite termination of "augmenting path" algorithms in the presence of irrational problem data. In *Proceedings of the 14th annual European Symposium on Algorithms (ESA)*, pages 268–279, 2006.

[5] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–14, 1962.

[6] D. Gusfield and R. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.

[7] R.W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the "optimal" stable marriage. *Journal of the ACM*, 34(3):532–543, 1987.

[8] D.E. Knuth. Stable marriage and its relation to other combinatorial problems. In *CRM Proceedings and Lecture Notes, vol. 10, American Mathematical Society, Providence, RI. (English translation of Marriages Stables, Les Presses de L'Université de Montréal, 1976)*, 1997.

[9] A.E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92:991–1016, 1984.

[10] D.D. Sleator and R.E. Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, 1983.

[11] D..D. Sleator and R.E. Tarjan. Self-adjusting binary search trees. *Journal of the ACM*, 32(3):652–686, 1985.

A A Hard Instance for the BB Algorithm

We describe here an n-vertex bipartite instance that causes the BB algorithm to run in $\Omega(n^3)$ time. Suppose we have $n/2$ jobs, each of whose processing time is an integer chosen independently at random from $\{n+1, \ldots, 2n\}$ except for job 1 (the dummy), with $p(1) = n^2/2$. We also have $n/2$ machines, each with capacity n except machine 1 (the dummy), whose capacity is set so that $p([n/2]) = c([n/2])$. Each job ranks the machines in order $n/2, n/2-1, \ldots, 1$, and each machine ranks the jobs in order $n/2, n/2-1, \ldots, 1$. There are no upper capacities $u(i,j)$. When applying the BB algorithm to this instance, we repeatedly augment starting from job 2 until it is fully assigned, then from job 3, and so on (recall that every machine starts out assigned to the dummy job 1).

Due to the order of the preference lists and the order in which we augment, the structure of every intermediate assignment x generated during the execution of the BB algorithm is as follows: a contiguous range of jobs $1 \ldots i_0 - 1$ will be fully assigned, with job i_0 (the job from which augmentations are currently issued) partially assigned. These jobs will be assigned to a suffix of the machines $j_0 \ldots n/2$. The graph $G(x)$ will be a tree, and the path through $G(x)$ from i_0 to the root (machine 1) visits every job from i_0 down to 1 in sequence. Intuitively, each augmentation starting from i_0 causes the entire assignment to "shift up" from the perspective of the machines.

Let us focus on execution of the BB algorithm from $i_0 = n/4+1$ onward. In this regime, there are at least $n^2/4$ units of processing time still to assign, and each augmentation takes $\Omega(n)$ time since each augmenting path has length $\Omega(n)$.

Lemma 10. *For the instance described above, with $i_0 > n/4$, each augmenting path π satisfies* $\mathbf{E}[r(\pi)] \leq 5$.

Suppose we perform $n^2/20$ augmentations (starting from $i_0 = n/4+1$). Letting X denote the number of units of processing time assigned during this process, we have $\mathbf{E}[X] \leq n^2/4$. Since $\mathbf{Pr}[X \leq n^2/4] > 0$, the probabilistic method tells us that there must be *some* instance for which $X \leq n^2/4$. For this instance, the BB algorithm performs at least $n^2/20 = \Omega(n^2)$ augmentations, each taking $\Omega(n)$ time.

Proof of Lemma 10. Consider a particular augmenting path π with $i_0 > n/4$, where x denotes the assignment immediately before augmentation on π. Consider any job $i \in [n/4]$. Note job i is assigned in x to a contiguous range of machines $j_i \ldots j_i'$, and that augmenting on π will increase $x(i, j_i)$ while decreasing $x(i, j_i')$. Since we can decrease $x(i, j_i')$ to no less than zero, $r(\pi) \leq x(i, j_i')$, and moreover $r(\pi) \leq Z$ where $Z = \min\{x(i, j_i') : i \in [n/4]\}$.

Due to the uniform machine capacities and the fact that jobs $i+1 \ldots i_0$ are assigned to a contiguous suffix of the machines, we can write $x(i, j_i') = n - ((p(\{i+1, \ldots, i_0-1\}) + x(i_0, [n/2])) \bmod n)$, which we rearrange to obtain $n - x(i, j_i') \equiv p(i+1) + K \pmod n$, where $K = p(\{i+2, \ldots, i_0-1\}) + x(i_0, [n/2])$. Irrespective of K, we see that $p(i+1) \bmod n$ is uniform in $\{0, \ldots, n-1\}$, so $x(i, j_i')$ is a uniform random number in $[n]$. Moreover, since each $p(i)$ is chosen independently, the $x(i, j_i')$'s are also independent. Using this fact, we see that Z is the minimum of a set of independent random variables each uniformly chosen from $[n]$. Hence,

$$\mathbf{E}[r(\pi)] \leq \mathbf{E}[Z] = \sum_{k=1}^{\infty} \mathbf{Pr}[Z \geq k] = \sum_{k=0}^{n-1} \left(1 - \frac{k}{n}\right)^{n/4} \leq \sum_{k=0}^{n-1} e^{-k/4} \leq 5.$$

Finding all stable pairs for the (many-to-many) Stable Matching

P. Eirinakis[1], D. Magos[2], I. Mourtos[1], P. Miliotis[1]

[1]Department of Management Science & Technology,
Athens University of Economics and Business, 76 Patission Ave., 104 34 Athens, Greece
email: {peir,mourtos,miliotis}@aueb.gr
[2]Department of Informatics, Technological Educational Institute of Athens
Ag. Spyridonos Str., 12210 Egaleo, Greece
email: dmagos@teiath.gr

Abstract

The many-to-many Stable Matching problem is defined in the context of a job market and asks for an assignment of workers (W) to firms (F) satisfying the quota of each agent and being stable, pairwise or setwise, with respect to given preference lists or relations. In this paper, we propose an algorithm identifying all stable worker-firm pairs in $O(m^2)$ steps where $m = \max\{|W|, |F|\}$. Further, we establish that this algorithm is appropriate under responsive group preferences (a) for pairwise stability and (b) for setwise stability when (in addition) workers or firms have strongly substitutable preferences. Computational results on random instances illustrate that removing non-stable pairs implies a substantial reduction in the preference lists.

1 Introduction

The *many-to-many Stable Matching* (MM) problem is naturally defined in the context of a job market in which each firm wants to hire a group of workers and each worker can be employed by several firms. Conventionally, F denotes the set of firms and W the set of workers. Let $m = \max\{|W|, |F|\}$. The *quota* q_f of a firm $f \in F$ denotes the maximum number of workers the firm can employ, while each worker $w \in W$ can be employed by at most q_w firms. A firm (worker) not fullfilling its quota is called undersubsribed. Furthermore, firms have preferences over individual workers and vice versa. These preferences are assumed to be strict and transitive, thus representable by ordered lists, called *preference lists.* We denote by $P(w)$ ($P(f)$) the preference list of worker w (firm f). For $f \in F$ ($w \in W$), the event that worker w_1 (firm f_1) ranks higher in $P(f)$ ($P(w)$) than worker w_2 (firm f_2) is denoted by $w_1 \succ_f w_2$ ($f_1 \succ_w f_2$). Preference lists need not be complete; if a firm f (worker w) is not in worker w's (firm f's) preference list, then f (w) prefers to leave a workspace empty than to employ w (f). The simplest form of the MM problem asks for a maximal set $M \subseteq W \times F$ satisfying the following two conditions:

(i) for no pair $(w, f) \in M$ there exist $\bar{w} \in W$ and $\bar{f} \in F$ such that $\bar{w}$ is more preferable than w by f *and* $\bar{f}$ is more preferable than f by w (stability condition),

(ii) the quota of each agent (i.e., worker/firm) is not exceeded (matching condition).

The MM problem is a generalization of the *Stable Admissions* (SA) problem which is, in turn, a generalization of the *Stable Marriage* (SM) problem. In the latter (former) case, also called the *one-to-one (one-to many)* Stable Matching problem, the quotas of the agents of both (one of the) sets are set to one. Both of these problems were introduced in a study of the mechanics of assigning students to colleges [7]. Besides the fact that MM is a proper generalization of SA (and SM), there are several real-life applications motivating its study. Primarily, it models several centralized markets, an example being the well-known medical interns' market in the U.K. [18], where each student must seek two positions, one in medicine and one in surgery. Also, most labor markets include certain many-to-many interacting agents, which are examined in terms of the MM model, since their study in terms of the (simpler) one-to-many framework produces significantly different results [5, Example 2].

The generalization of preferences and the stability condition, introduced in the literature, make the MM a more involved structure than SM. Under this setting, the preference lists are also generalized into *preference relations;* these represent the preferences of each firm f on groups of workers and vice versa. Thus in the MM case, the stability condition can be extended from pairs to *sets (groups)* of agents. That is, a matching M is called *setwise* (or *group*) stable if there is no subset of agents who by forming new partnerships only among themselves, possibly dissolving some partnerships of M to remain within their quotas and possibly keeping other ones, can all obtain a strictly preferred set of partners [19]. Note that preference relations include individual preferences as a group may be a singleton.

We examine three main types of preference relations that have been reported in the literature: *responsive* preferences [18], *substitutable* preferences [13] and *strongly substitutable* preferences [5]. To describe these terms, let $P^{\#}(f)$ denote the preference relation of the firm f and $M(f)$ the set of workers assigned to f in the matching M.

Definition 1 *The preference relation $P^{\#}(f)$ over sets of workers is responsive (to the preferences $P(f)$ over individual workers) if, whenever $M'(f) = M(f) \cup \{w_1\} \setminus \{w_2\}$ for $w_2 \in M(f)$ and $w_1 \notin M(f)$, the firm f prefers $M'(f)$ to $M(f)$ (denoted by $M'(f) \succ_f M(f)$) if and only if $w_1 \succ_f w_2$.*

The substitutable preferences comprise a weaker criterion. (In fact, it is the weakest condition under which the existence of stable matchings is guaranteed, according to [16].) Suppose that a firm f prefers a group of workers $\mathfrak{w}$ including a worker w. Then the firm has substitutable preferences, if, whenever some members of $\mathfrak{w}$ become unavailable, it still wants to employ a subgroup of $\mathfrak{w}$ that includes w (see [19] for a formal definition). In [5], a variant of the notion of substitutability, called *strong substitutability,* is introduced. That is, assume that hiring worker w is optimal when certain workers are available. Then, strong substitutability requires that hiring w must still be optimal even when a worse set of workers is available. Note that responsiveness implies substitutability but not strong substitutability. Figure 1 presents the relationship between responsive, substitutable and strong substitutable preferences (see examples in [5]).

The preference relations, described above, imply a partial ordering of stable matchings for each agent. This ordering depicts the preference of the agent over the matchings. In the case of individual preferences which imply *pairwise stability* (see (i) above) the standard criterion used is the so called *maxmin* criterion [2]. According to that criterion, a group of workers (firms) is preferred by a firm f (worker w) to some other group if the least preferred worker (firm) of the first group ranks higher in the preference list of firm f (worker w) to the least preferred worker (firm) of the other group. Hence, under the maxmin criterion, each agent has always a complete ordering of all stable matchings. A formal definition follows.

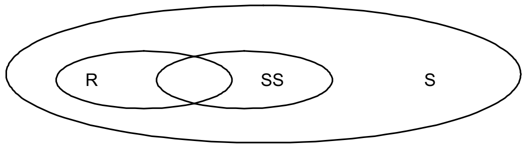


Figure 1: Relationship between responsive, substitutable and strong substitutable preferences

Definition 2 *Let M and M' be two arbitrary matchings. The firm $f \in F$ prefers M at least as M' (denoted as $M \succeq_f M'$) if either (i) $M(f) = M'(f)$ or (ii) $|M(f)| \geq |M'(f)|$ and* $\text{last}_M(f) \succ_f \text{last}_{M'}(f)$, *where* $\text{last}_M(f)$ ($\text{last}_{M'}(f)$) *denotes the least preferred worker of the group $M(f)$ ($M'(f)$) by f.*

A pair (w, f) is called *stable* if w is assigned to f in at least one stable matching. The problem studied here is the ALLSTABLEPAIRS problem, i.e. the problem of identifying all stable pairs. The usefulness of this information was first observed in [14], while a polynomial algorithm for solving this problem in the SM setting is described in [9]. In [6], an algorithm for identifying all stable pairs for the SA with individual preferences is presented.

In the current work, we present an algorithm for solving the ALLSTABLEPAIRS problem in the MM setting under individual preferences. In the sequel, we show that this algorithm can also be used in the presence of responsive group preferences under (a) pairwise stability, or, (b) setwise stability when (in addition to responsiveness) the agents of one of the sets (W or F) have strongly substitutable preferences. To show (a), we prove that the *lattice* of stable matchings formed when agents have individual preferences coincides with the corresponding object when preferences are responsive. This settles a question posed in [2] regarding the efficiency of the two criteria (i.e., maxmin versus responsive) when pairwise stability is considered.

An alternative definition to the ALLSTABLEPAIRS problem can be given in the context of Constraint Programming (CP). Thus, if we consider the MM problem as a *global* constraint, the ALLSTABLEPAIRS problem asks for establishing *generalized arc consistency* to that constraint (see [11] for related definitions). Such a view has been adopted in [8] and [15] for the SM and SA case, respectively. However, the main theme in all these papers is the presentation and comparison of different encodings, none of which directly refers to the MM version of the problem. Further the issue of consistency is only partially resolved, since the algorithms presented just establish arc consistency without identifying all (non-)stable pairs.

2 Background

Under pairwise stability, if a matching M is not stable then there exists a worker w and a firm f such that $(w, f) \notin M$ but both w and f prefer each other to their current partners in M [10, 18]. Such a pair is said to *block* the matching M, or, equivalently, (w, f) forms a *blocking pair.* Similarly, a matching M can be blocked by an individual agent if this agent is matched to a member of the other set not appearing in its preference list (i.e., he prefers to remain single than to obtain a partner not in its preference list). In the case of setwise stability, a matching M can be blocked by a coalition bigger than a simple worker-firm pair [18].

In the case of pairwise stability under the maxmin criterion [2] or under responsive preferences [1], the MM problem has a non-empty solution set, namely $\mathfrak{M}$. Among the members of $\mathfrak{M}$, there exists a matching in which every worker is at least as better-off (worse-off) under it as under any other matching. Such a matching is called *workers' optimal (pessimal).* In fact, a *worker-oriented dominance relation* can be defined on the set of stable matchings.

Remark 3 *For $M, M' \in \mathfrak{M}$ and $M \neq M'$,*

(i) M dominates M' ($M \succ_W M'$) if $M \succeq_w M'$ for all $w \in W$ and

(ii) $M \succ_w M'$ implies $M' \succ_f M$ for all workers $w \in W$ and firms $f \in F$ such that $(w, f) \in (M \setminus M') \cup (M' \setminus M)$ (and vice versa).

Also, the *join (supremum)* $M \vee M'$ and the *meet (infimum)* $M \wedge M'$ are stable matchings, where $M \vee M'$ ($M \wedge M'$) assigns to each worker w the best (worst) of the two sets of firms $M(w)$ and $M'(w)$, and to each firm f the worst (best) of the workers $M(f)$ and $M'(f)$. Under this dominance relation, it can be shown that the set of stable matchings forms a *distributive lattice* (for definitions see [1, 2]). The *greatest* element of the lattice corresponds to the workers' optimal (firms' pessimal) matching and the *least* element corresponds to the worker's pessimal (firms' optimal) matching.

The situation is more complicated when setwise stability is considered. In the simpler SA setting, a matching is setwise stable if and only if it is pairwise stable. This is not true in the case of MM. In [18] it is shown that, in many-to-many matching models with responsive (thus substitutable) preferences, a pairwise stable matching need not be setwise stable. Also in [4] it is observed that the lattices that arise in many-to-many matching markets under setwise stability need no longer be distributive. On the positive side, in [5] it is proved that the set of stable matchings, under setwise stability, forms a non-empty lattice identical to the one created under the pairwise stability when members of the one (other) set have (strong) substitutable preferences. That particular model actually encompasses standard one-to-many theory, in which one of the sets represents colleges and the other students, since it is trivially true that colleges (students) have (strong) substitutable preferences. In the same work, it is shown (contrary to [4, Example 5.2]) that the lattice formed by the set of stable matchings is distributive in the case of the preferences being strong substitutable [5].

Finding a stable matching is intuitively related to the identification of stable pairs. In [10], an extended implementation of the Gale/Shapley algorithm [7] (called EGS) is described that in the process of findin a stable solution, identifies and eliminates some (but not all) non-stable pairs. The *worker-oriented* version of that method (hereafter called WEGS) in the MM setting identifies the workers' optimal solution under pairwise stability with individual preference lists in $O(m^2)$ steps [2]. Upon termination of WEGS, the workers' preference lists are reduced in a way that (a) the worker w is assigned either its best q_w stable partners, which are the first q_w firms in her (reduced) list, or a set of fewer than q_w firms, and (b) each firm is assigned its worst set of 'stable' workers. Note that it is not necessarily true that, in the workers' optimal (i.e. firms' pessimal) solution, a firm that fills all of its q_f places is assigned its q_f worst partners (of course, this is trivially true if a worker is underemployed). However, if M_0 is the workers' optimal matching and M' any other stable matching, then every firm prefers all the workers assigned to it in M' to all of those assigned to it in M_0 but not in M'. Also, in [1, 2] it is observed that (a) each worker is employed in the same number of firms in all stable matchings, (b) exactly the same workers are underemployed (or unemployed) in all stable solutions, and (c) any worker who is underemployed in one stable matching is matched with precisely the same firms in all stable matchings.

As stated previously, even if we successively apply WEGS, the resulting reduced lists may still allow for non-stable pairs to appear. It turns out that the ALLSTABLEPAIRS problem can be solved by exploiting the lattice structure of $\mathfrak{M}$, as discussed next.

3 Individual preferences

Assume that the agents have individual preferences. The implications are that (a) stability is defined in a pairwise context and (b) the stable matchings can be ordered (to form a distributive lattice) with respect to the maxmin criterion. Given a stable matching M, let $r_M(w)$ be the first firm $f \in P(w)$ such that $(w, f) \notin M$ and $w \succ_f \text{last}_M(f)$, i.e. firm f is not assigned to worker w but prefers it to its least preferred worker. Note that such a firm exists as long as M is not the firm-optimal matching. Further, $\text{next}_M(w)$ denotes $\text{last}_M(r_M(w))$. Since M is stable, w prefers all firms assigned to him in M to $r_M(w)$ if such an $r_M(w)$ exists (for example in firms' optimal matching $r_M(w)$ does not exist for some $w \in W$). Next, we explore the possibility of assigning the worker w to $f = r_M(w)$ not knowing yet which of his/hers current assignments to break to be employed by f. Next, firm f (which prefers w to $\text{next}_M(w)$) fires $\text{next}_M(w)$ and employes w. Worker $\text{next}_M(w)$ (who is now underemployed) proposes to $r_M(\text{next}_M(w))$ and so on, until a worker w' (who may or may not be w) comes up twice. Let $\rho = (w_1, r_M(w_n)), (w_2, r_M(w_1)), \ldots, (w_n, r_M(w_{n-1}))$ be an ordered list of pairs in a stable matching M such that $w_{i+1 \pmod n} = \text{next}_M(w_i)$, for all $i \in \{1, \ldots, n\}$. Then ρ is a *(meta-)rotation exposed* in M, and we say that w (or f) is *in rotation* ρ if there is a pair (w, f) in the ordered list defining ρ. If $\rho = (w_1, f_1), (w_2, f_2), \ldots, (w_n, f_n)$ is a rotation exposed in M, then M/ρ is called the *elimination* of rotation ρ from M and it denotes the matching M' derived from M if each worker w_i participating in ρ breaks her assignment to f_i and is employed by $f_{i+1 (\text{mod}\, n)} = r_M(w_i)$, while everyone else is employed by exactly the same firms as in M.

Rotations were introduced in [12] and used in [9] to develop an algorithm for solving the ALLSTABLEPAIRS problem in the SM setting. Rotations have also been used in the SA [6] and the MM context [3]. To employ rotations for solving the problem in the MM case, we present a series of statements, some of which extend known results and are therefore provided without a proof.

Lemma 4 *When a rotation ρ is eliminated from a stable matching M, the resulting matching M/ρ is stable and is dominated by M.*

Corollary 5 *Every worker-firm pair produced by eliminating a rotation is stable.*

The following result [9, Lemma 3] extends in the MM case.

Lemma 6 *Let ρ denote a rotation exposed in the stable matching M. Then, there is no stable matching 'between' $M, M' = M/\rho$ (i.e., M, M' correspond to adjacent points in the lattice).*

Lemma 7 *Let $\rho = (w_1, f_1), \ldots, (w_n, f_n)$ be a rotation exposed in the stable matching M and, for some $i \in \{1, \ldots, n\}$, let $f \in P(w_i)$ ($w \in P(f_i)$) such that* $\text{last}_M(w_i) \succ_{w_i} f \succ_{w_i} \text{last}_{M/\rho}(w_i)$ ($\text{last}_{M/\rho}(f_i) \succ_{f_i} w \succ_{f_i} \text{last}_M(f_i)$)*. Then (w_i, f) (respectively, (w, f_i)) is not a stable pair.*

Proof. We illustrate the result only with respect to the reference lists of workers (i.e. for pair (w_i, f)). Observe that, in w_i's list, f ranks lower than f_i and higher than $f_{i+1(\text{mod}\, n)}$, where $f_i = \text{last}_M(w_i)$ and $f_{i+1(\text{mod}\, n)} = r_M(w_i) = last_{M/\rho}(w_i)$. The procedure for exposing a rotation implies that $r_M(w_i)$ is the first firm ranking lower than $last_M(w_i)$ in w_i's list, which prefers w_i

to its least preferred worker in M. Since f ranks higher than $r_M(w_i)$, f prefers $last_M(f)$ to w_i (if not, it would itself be $r_M(w_i)$), i.e. $last_M(f) \succ_f w_i$.

Now suppose (w_i, f) is a stable pair in some stable matching M'. That implies that w_i is better-off in M, since it prefers f_i to f. However, so is f, since it prefers $last_M(f)$ to $last_{M'}(f)$, where $last_{M'}(f)$ may only be w_i or some worker ranking lower than w_i in f's list. It follows that $M \succ_{w_i} M'$ and $M \succ_f M'$, which contradicts Remark 3(ii). ■

Let hereafter M_0 (M_z) denote the workers' optimal (pessimal) stable matchings, respectively. The following theorem, generalizes [17, Theorem 4].

Theorem 8 *Any stable matching can be obtained from the workers' optimal by successively exposing and eliminating rotations.*

Proof. Let $M \neq M_0$. Since M is non-worker-optimal, there exists some worker w such that $M_0(w) \succ_w M(w)$, i.e. $last_{M_0}(w) \succ_w last_M(w)$. Thus, consider a rotation ρ exposed in M_0 that includes w. Within the procedure of exposing ρ, assume that worker w proposes to a choice worse than $last_M(w)$. That implies that w is rejected by $last_M(w)$ at some point during the procedure of exposing ρ. Equivalently, $last_M(w)$ is 'between' $last_{M_0}(w)$ and $r_{M_0}(w)$, i.e. $last_{M_0}(w) \succ_w last_M(w) \succ_w r_{M_0}(w)$. But then, Lemma 7 yields pair $(w, last_M(w))$ as non-stable, which contradicts the assumption that matching M is stable.

Therefore, exposing a rotation in M_0 can only produce a stable matching M', in which no worker has a worse choice than in M, while the procedure can be repeated if $M' \neq M$. Since, after exposing a rotation, at least two workers are employed by a worse choice, but not a choice worse than their choice in M, matching M is obtainable after eliminating a finite number of rotations. ■

Corollary 9 *Starting from the worker-optimal stable matching, the firm-optimal stable matching can be obtained by a number of successive exposures and eliminations of rotations.*

Theorem 10 *Let $M_0, M_1, \ldots, M_z$ be a sequence of stable matchings such that $M_i \succ_W M_{i+1}$ (M_i dominates M_{i+1}) for $i = 0, \ldots, z-1$, and there is no stable matching 'between' M_i and M_{i+1}. Then, every stable pair appears in at least one of the stable matchings of this sequence.*

Proof. For $w \in W$ and $f \in F$, let (w, f) be a stable pair appearing in no matching $M_i, i = 0, \ldots, z$. It follows that $f \neq last_{M_i}(w) = f_i$ for all $i = 0, \ldots, z$. Since $M_i \succ_W M_{i+1}$ for $i = 0, \ldots, z-1$, either $f_i = f_{i+1}$ or $f_i \succ_w f_{i+1}$. Thus, there exist two matchings in the sequence, namely M_i and M_{i+1} such that $f_i \succ_w f \succ_w f_{i+1}$.

Clearly, (w, f) appears in some other stable matching M, i.e. $M \neq M_i$ for $i = 0, \ldots, z$. Since the set of stable matchings forms a distributive lattice, we may construct matching $M' = M_i \wedge (M \vee M_{i+1})$. Observe that M' is a stable matching in which w is assigned to f, thus being different from both M_i and M_{i+1}, with $last_{M_i}(w) \succ_w last_{M'}(w) \succ_w last_{M_{i+1}}(w)$. Under the maxmin criterion, that implies $M_i \succ_W M' \succ_W M_{i+1}$, i.e. a contradiction to the hypothesis that there is no stable matching 'between' M_i and M_{i+1}. ■

Note that the underemployed workers (unsubscribed firms) participate in no rotations, since they are employed by precisely the same set of firms (workers) in all stable matchings. The following is an extension of [12, Lemma 4.7] (see [5] for a proof).

Lemma 11 *In any MM instance, no (w, f) pair can belong to two different rotations.*

Let us denote as FEGS the firm-oriented version of EGS that identifies the firms' optimal solution under pairwise stability. Algorithm 1 solves the ALLSTABLEPAIRS problem in the

Algorithm 1 Finding all stable pairs

Initially, the status of all pairs (w, f) is 'undefined';
Run WEGS and FEGS to produce reduced preference lists and M_0, M_z;
Use the reduced preference list to update the status of pairs (w, f);
$i \leftarrow 0$;
while $\exists(w, f)$ with undefined status **do**
  Expose ρ_i in M_i;
  $M_{i+1} \leftarrow M_i/\rho_i$;
  Set the status of pairs $(w, f) \in M_{i+1}$ to 'stable';
  for all $f \in \rho_i$ **do**
    for all w such that $\text{last}_{M_i/\rho_i}(f) \succ_f w \succ_f \text{last}_{M_i}(f)$ **do**
      Set the status of pair (w, f) to 'non-stable';
    end for
  end for
  for all $w \in \rho_i$ **do**
    for all f such that $\text{last}_{M_i/\rho_i}(w) \succ_w f \succ_w \text{last}_{M_i}(w)$ **do**
      Set the status of pair (w, f) to 'non-stable';
    end for
  end for
  Update preference lists;
end while
Output preference lists;

MM setting when agents have individual preferences. The algorithm assigns to each pair $(w, f) \in W \times F$ a status flag which can receive the values 'stable', 'non-stable', or 'undefined'. Initially, the status of all pairs is set to 'undefined'. The execution of WEGS and FEGS causes some of the pairs to obtain the status 'stable' (i.e. those appearing at M_0 and M_z) and some the status 'non-stable'. Starting from M_0, the algorithm exposes and eliminates rotations. Each time a new stable matching is reached, the pairs belonging to it receive the status 'stable' whereas the pairs indicated by Lemma 7 have their status set to 'non-stable' . The algorithm terminates when no pair remains with an 'undefined' status. Each time a pair (w, f) is found to be non-stable w, f are deleted from each other's list.

Theorem 12 *Algorithm 1 solves the ALLSTABLEPAIRS problem in $O(m^2)$ steps.*

Proof. Matching M_0 (M_z), together with the reduced preference lists is produced by WEGS (FEGS). As stated above, the algorithm starts exposing and eliminating rotations, thus creating a sequence of stable matchings $M_0, M_1, ..., M_z$ (Lemma 4). By Lemma 6, there is no stable matching 'between' M_i and M_{i+1} for all $i \in \{0, ..., z-1\}$. Since every stable pair is bound to appear in at least one of the matchings of this sequence (Theorem 10), Algorithm 1 reveals all stable pairs upon completion. (Note that the algorithm may terminate before deriving M_z as long as the status of all pairs has been defined.) The preference list of each agent, once Algorithm 1 terminates, contains only the mates that appear together with him in at least one stable matching. Thus, Algorithm 1 is correct.

Concerning the complexity of this algorithm, recall that WEGS and FEGS require $O(m^2)$ steps to create M_0 [2]. Exposing and eliminating a rotation can easily be implemented at constant time per pair by employing a stack (see [10, Section 4.2.4]). Since each pair appears in at most one rotation (Lemma 11) and the number of pairs is $O(m^2)$, Algorithm 1 requires $O(m^2)$

steps for exposing and eliminating all rotations, i.e. for computing matchings $M_1, ..., M_{z'}$, $z' \leq z$. ■

Since each pair should be examined at least once in order to determine its stability, Algorithm 1 is of the lowest possible complexity, i.e. it is asymptotically optimal.

4 Group preferences

In the setting of group preferences, the ALLSTABLEPAIRS problem amounts to computing all stable assignments of each agent, i.e. all sets of firms (workers) assigned to each worker (firm) in any stable matching. First, we show that this can be accomplished by Algorithm 1 under the maxmin criterion and for responsive group preferences.

Let L denote the distributive lattice corresponding to an MM instance. A sequence of stable matchings $P = M_0, M_1, \ldots, M_z$ is called a *maximal path of* L, if $M_{i+1} = M_i/\rho_i$ where ρ_i is a rotation exposed in matching $M_i, i = 0, ..., z-1$. Lemma 6 implies that there exists no stable matching 'between' M_i and M_{i+1}, i.e. the notion of a *maximal path* is analogous to that of a *maximal chain* of a general ring of sets [10]. Denote also as $M(w)$ the assignment to worker w in matching M, i.e. $M(w) = \{f : (w, f) \in M\}$.

Theorem 13 *Every rotation appears exactly once on every maximal path P of lattice L.*

Proof. It suffices to establish that every rotation is exposed at least once in P. For a rotation ρ not exposed while deriving P, Lemma 11 implies that a pair (w, f) appearing in ρ appears in no other rotation exposed while deriving P. But then, pair (w, f) is non-stable (Theorem 10), i.e. a contradiction to Corollary 5. ■

An implication of the previous theorem is that the all maximal paths have the same length $z + 1$, where z is the number of rotations exposed in all stable matchings.

Theorem 14 *A complete order of all stable assignments is obtained for worker w (firm f) by listing all stable assignments given to w (f) during the procedure of deriving a maximal path P.*

Proof. We show the result with respect to workers. Assume that assignment $M_{i+1}(w)$ is not included in worker w's list after exposing and eliminating all rotations. Let also $M_{i+1} = M_i/\rho_i$, pair (w, f) be appearing in rotation ρ_i and $f' = r_{M_i}(w)$. By Lemma 11, pair (w, f) appears in at most one rotation that is ρ_i, while Theorem 13 implies that ρ_i is exposed in some matching M_k appearing on P.

If $M_i = M_k$, $M_k/\rho_i = M_{i+1}$ thus $M_{i+1}(w)$ is actually included in worker w's list, i.e. a contradiction to the hypothesis. Otherwise ($M_i \neq M_k$), let $M_{k+1} = M_k/\rho_i$ thus yielding $last_{M_{i+1}}(w) = last_{M_{k+1}}(w) = f'$. For $M_{k+1}(w) = M_{i+1}(w)$, $M_{k+1}(w)$ is included in worker w's list since M_{k+1} belongs to P, i.e. a contradiction. For $M_{k+1}(w) \neq M_{i+1}(w)$, observe that $M_{k+1}(w)$ and $M_{i+1}(w)$ cannot be compared under the maxmin criterion since $last_{M_{k+1}}(w) = last_{M_{i+1}}(w)$, therefore one of them is not stable. It is not difficult to see that both cases yield a contradiction. ■

Evidently, Algorithm 1 can compute a maximal path if it is set to terminate only after obtaining the firms' optimal matching M_z. Therefore, this algorithm finds all stable assignments for all agents participating in any stable matching under the maxmin criterion, simply by retaining the workers (firms) assigned to each firm (worker) at each of the matchings $M_0, M_1, \ldots, M_z$. Moreover, all assignments of each worker (firm) are completely ranked in the (inverse) order they are identified. Hence the following.

Theorem 15 *Algorithm 1 computes all stable assignments of every agent for every instance of MM in $O(m^2)$ steps.*

Now, let us examine the case where group preferences are responsive to individual preferences and the stability criterion involves only pairs. The following theorem extends [18, Theorem 5.27].

Theorem 16 *Let M, M' be two stable matchings. If $M(w) \succ_w M'(w)$ for some worker w, then $f \succ_w f'$ for all $f \in M(w)$ and $f' \in M'(w) \backslash M(w)$.*

Theorem 16 together with Definition 1 lead to the following.

Corollary 17 *Let $P^{\#}(w)$ denote the preference relation of worker w over groups of firms which is responsive to his preferences $P(w)$ over individual firms. Then for every pair of stable matchings M and M', $M(w) \succ_w M'(w)$ under $P^{\#}(w)$ if and only if $M(w) \succ_w M'(w)$ under $P(w)$.*

Thus, Corollary 17 shows that the set of stable matchings is invariant to changes in the preference relations $P^{\#}(w)$ as long as these preferences remain responsive to the preferences $P(w)$ over individuals. A question posed in [2] is whether the maxmin criterion is significantly different in practice from responsive preferences, under pairwise stability. The following theorem answers this question in the negative. Recall that $\mathfrak{M}$ denotes the set of solutions of an MM instance.

Theorem 18 *The distributive lattice formed by $\mathfrak{M}$ under responsive preferences with pairwise stability is identical to the one formed by $\mathfrak{M}$ under the maxmin criterion.*

Proof. Consider an MM instance with preference lists $\mathbb{P}$ and recall that the matchings comprising $\mathfrak{M}$ form a distributive lattice L, under the maxmin criterion with pairwise stability [1]. Consider another MM instance, along with its associated set of solutions $\mathfrak{M}'$, having the exact same agents as $\mathfrak{M}$ but responsive preference relations $\mathbb{P}^{\#}$ that include the corresponding lists $\mathbb{P}$. We may safely assume that each agent's preference list contains the complete order of all stable assignments of that particular agent in $\mathfrak{M}'$ (computed by Algorithm 1), followed by the individual preference list $\mathbb{P}$. Then, observe that Corollary 17 implies a dominance relation for $\mathfrak{M}'$ identical to the one defined for $\mathfrak{M}$ (Remark 3), while the supremum and the infimum of any two $M, M' \in \mathfrak{M}'$ can also be defined as in the case of $\mathfrak{M}$. In other words, $\mathfrak{M}'$ is identical to $\mathfrak{M}$ and so are the corresponding lattices. ■

By Theorem 18, it becomes evident that pairwise stable matchings can be identified in the case of responsive group preferences without the complete knowledge of them, i.e. only the preferences over individuals are required. Note that, although the number of assignments included in a preference relation $P^{\#}$ can be exponential in m, Theorem 15 implies that the number of stable assignments is bounded by the number of rotations, which is trivially bounded by m^2. Therefore, the ALLSTABLEPAIRS problem can be solved using Algorithm 1 in the presence of responsive preference relations, under pairwise stability.

In the case of setwise stability, the question of finding all stable assignments becomes far more complex. However, a special case remains tractable through Algorithm 1, again because the set of stable matchings $\mathfrak{M}$ forms a distributive lattice. In [5], it is shown that if the workers' preferences are substitutable and the firms' preferences are strong substitutable (or vice-versa), then $\mathfrak{M}$ is the same under both setwise and pairwise stability, and forms a lattice. Since responsive preferences are also substitutable (Figure 1), it is easy to derive the following through Theorem 18.

Theorem 19 *The distributive lattice formed by $\mathfrak{M}$ under setwise stability, when both workers and firms have responsive preferences and at least one set of agents has also strong substitutable preferences, is identical to the one formed by $\mathfrak{M}$ under the maxmin criterion.*

It follows easily that, in the case of setwise stability implied by Theorem 19, Algorithm 1 computes all stable assignments in $O(m^2)$ steps.

5 Computational Results

This section discusses the application of Algorithm 1 to a set of randomly generated instances of MM. The code is written in VB.NET in the Microsoft Visual Studio .NET Professional 2003, while computational results are obtained on an Intel(R) Core(TM) 2 Duo CPU T7500 processor (2.20 GHz) with 2 GB of RAM under Windows Vista(TM).

The results are summarized in Table 1. Each row in this table corresponds to a different MM problem and presents the average results obtained on 10 randomly generated instances. Workers and firms have complete preference lists, i.e. each of the $|W| \cdot |F|$ pairs appears in two preference lists. Further, the quota of each worker (firm) is a random integer drawn uniformly from the interval $\{1, \ldots \max q_w\}$ ($\{1, \ldots \max q_f\}$). Those instance features are depicted in the first four columns of Table 1. The next two columns illustrate the time (in seconds) required by Algorithm 1 and the percentage of pairs that are deleted, respectively. The following two columns illustrate the percentage of time consumed and the percentage of pairs deleted, respectively, by the procedure of exposing and eliminating rotations (i.e. the remaining percentage of time and pairs deleted concern the WEGS and FEGS algorithms that comprise the first step of Algorithm 1). Thus, these two columns indicate the additional reduction achieved through Algorithm 1 (i.e. the percentage of non-stable pairs not identified by WEGS or FEGS) and the associated computational effort for finding all rotations. The penultimate column presents the number of rotations identified, while the last column provides the average number of stable assignments per worker, i.e. the average number of subsets of F that would not be deleted in any preference relation satisfying the criteria of Theorems 18 or 19.

Observe first that the reduction in the preference lists is substantial in all instances. This illustrates the importance of the proposed algorithm in terms of reducing the solution space. Thus, Algorithm 1, as a preprocessing step, could significantly enhance the performance of an algorithm searching for an 'optimal' solution or enumerating all solutions. Moreover, this finding provides an insight on the structure of MM, given that no theoretical bound on the number of stable pairs is known. Notice also that the reduction in the preference lists remains approximately the same for different values of $|F|$ and $|W|$ but decreases as quotas increase, since larger quotas imply more solutions and therefore fewer non-stable pairs.

Most of this reduction is achieved by using WEGS and FEGS, which also consume the larger percentage of total time. Still, the remainder of Algorithm 1 that utilizes the concept of rotations has a significant impact, with the percentage of non-stable pairs identified through exposing rotations (i.e. not found WEGS or FEGS) being 'proportional' to the percentage of the time required. For example, in the instance where $(|F|, |W|, \max q_f, \max q_w) = (100, 1000, 200, 20)$, 66% of a total of 10^5 pairs is deleted within less than a second; among those pairs, 90% is removed by using WEGS or FEGS in 76% of total time, while 10% is removed by the procedure of exposing rotations in 24% of total time. The 'time percentage over pair percentage' ratio (related to exposing rotations) actually increases in large instances (e.g. for $|W| = |F| = 2000$), although the number of rotations increases drastically, especially for larger quotas.

Algorithm 1 performs equally well in the presence of group preferences, since the number of stable assignments per worker (last column of Table 1) is always smaller than 20% of m.

One can easily construct responsive preference relations $\mathbb{P}^{\#}$, where the total number of subsets of firms contained in all workers' relations is $O\left(|W| \cdot \binom{|F|}{\max q_w}\right)$. It follows that Algorithm 1 can lead to an enormous reduction of preference relations when Theorems 18 or 19 become applicable.

Overall, the current work, apart from providing an efficient algorithm for the ALLSTABLE-PAIRS problem in the MM setting, has important computational implications under both individual and group preferences. Furthermore, Algorithm 1 is applicable for any type of preference relations that gives rise to a distributive lattice identical to the one formed under the maxmin criterion. Whether more such types exist remains to be further investigated.

Table 1: Computational results

Instance				Algorithm 1		Exposing Rotations			
|F|	|W|	$\max q_f$	$\max q_w$	Time	Pairs(%)	Time(%)	Pairs(%)	#Rot.	#Assign.
100	100	5	5	<1	90	14	6	90	7
100	100	20	20	<1	67	20	8	199	14
100	100	50	50	<1	40	11	3	155	11
100	500	25	5	<1	88	16	8	402	8
100	500	100	20	<1	67	22	9	825	14
100	500	250	50	<1	39	12	4	708	12
100	1000	50	5	<1	88	17	9	802	8
100	1000	200	20	<1	66	24	10	1643	15
100	1000	500	50	<1	39	12	9	1324	12
100	2000	100	5	1	88	17	9	1546	8
100	2000	400	20	3	67	23	9	3110	15
100	2000	1000	50	4	39	12	4	2566	12
500	500	25	25	1	87	27	15	1538	43
500	500	100	100	2	64	30	14	3011	82
500	500	250	250	2	36	17	7	2707	73
500	1000	50	25	4	86	30	18	2873	45
500	1000	200	100	5	63	33	17	5631	88
500	1000	500	250	7	35	19	8	4973	78
500	2000	100	25	13	86	32	19	5472	47
500	2000	400	100	19	63	33	17	10406	88
500	2000	1000	250	24	36	17	7	8581	73
1000	1000	50	50	11	86	34	22	5026	95
1000	1000	200	200	15	62	34	18	9509	179
1000	1000	500	500	22	35	19	8	8307	156
2000	2000	100	100	98	85	36	25	15038	197
2000	2000	400	400	138	61	37	21	29399	382
2000	2000	1000	1000	180	34	20	9	25325	327

References

[1] Alkan A.: On the properties of stable many-to-many matchings under responsive preferences, *Current trends in economics: theory and applications* **8**, edited by A. Alkan, C.D. Aliprantis, N.C. Yianellis, Studies in Economic Theory, Springer Verlag, (1999).

[2] Baiou M., Balinski M.: Many-to-Many Matching: stable polyandrous polygamy (or polygamous polyandry). *Discrete Applied Mathematics* **101**, 1-12 (2000).

[3] Bansal V., Agrawal A., Malhotra V.S.: Polynomial time algorithm for an optimal stable assignment with multiple partners". *Theoretical Computer Science* **379 (3)**, 317-328 (2007).

[4] Blair C.: The lattice structure of the set of stable matchings with multiple partners, *Mathematics of Operations Research* **13** (4), 619-628 (1988)

[5] Echenique F., Oviedo J.: A theory of stability in Many-to-Many matching markets, *Theoretical Economics* **1**, 233-273 (2006).

[6] Eirinakis P., Magos D., Mourtos I., Miliotis P.: Hyperarc Consistency for the Stable Admissions problem, In *Proceedings of ICTAI'07,* **1,** 239-242 (2007).

[7] Gale D., Shapley L.S.: College Admissions and the Stability of Marriage. *The American Mathematical Monthly* **69**, 9-15 (1962).

[8] Gent I.P., Irving R.W., Manlove D.F., Prosser P., Smith B.M.: A constraint programming approach to the stable marriage problem. In *Proceedings of CP'01*, *Lecture Notes in Computer Science* **239**, 225 – 239, Springer-Verlag (2001).

[9] Gusfield D.: Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing* **16**, 111-128 (1987).

[10] Gusfield D., Irving R.W.: The stable marriage problem: Structure and Algorithms. *MIT Press*, Cambridge, MA (1989).

[11] Hooker, J.N.: Logic Based Methods for Optimization, *Wiley*, NY (2000).

[12] Irving R., Leather P.: The complexity of counting stable marriages. *SIAM Journal on Computing* **15**, 655-667 (1986).

[13] Kelso A.S., Crawford V.: Job matching, coalition formation and gross substitutes, *Econometrica* **50** (6), 1483–1504 (1982).

[14] Knuth D.E. Marriage Stables. *Montreal University Press,* Montreal (1976).

[15] Manlove D.F., O'Malley G., Prosser P., Unsworth C.: A Constraint Programming approach to the Hospitals / Residents problem. In *Proceedings of the CPAIOR'07, Lecture Notes in Computer Science* **4510**, 28-43 (2007).

[16] Martínez R., Massó J., Neme A., Oviedo J.: An algorithm to compute the full set of many-to-many stable matchings. *Mathematical Social Sciences* **47**, 187-210 (2004)

[17] McVitie D.G., Wilson L.B.: The Stable Marriage Problem. *Communications of the ACM* **14**, 486-490 (1971).

[18] Roth A.E., Sotomayor M.A.O.: Two-sided Matching: a study in game theoretic modelling and analysis, *Economic Society Monographs* **18**, Cambridge University Press (1990).

[19] Sotomayor M.A.O.: Three remarks on the many-to-many stable matching problem, *Mathematical Social Sciences* **38**, 55-70 (1999).

A note on existence and uniqueness of vNM stable sets in marriage games *

Jun Wako †

June 10, 2008

Abstract

We prove that for any marriage game with strict preferences, there exists a unique von Neumann-Morgenstern stable set, which is also a unique subsolution of the game.

1. Introduction

We consider von Neumann-Morgenstern stable sets (vNM set, briefly) of two-sided one-to-one matching games with strict preferences, the so-called marriage games.

Gale and Shapley (1962) proved any marriage game has a nonempty core. The core is internally stable, since no core matchings dominate each other. However, it is not always externally stable, since there may exist non-core matchings that are not dominated by any core matchings. A vNM set is an internally and externally stable set of matchings. Ehlers (2007) proved that if a vNM set exists in a marriage game, it is a maximal distributive lattice including the core as a subset. In general, even if the core of a game is nonempty, a vNM set may not exist in the game. Thus, the existence of vNM sets in marriage games was unknown.

In this note, we show that there exists a unique vNM set for any marriage game. The proof is constructive:

(0) Initially, let C_0 be the core of a given marriage game G and set $n := 0$.

(1) Let UD^n be the set of matchings that are not dominated by any matchings in C_n. ($C_n \subseteq UD^n$ by Proposition 4.7, Definition 4.3.)

(2) If $C_n \subsetneq UD^n$, find the core C_{n+1} defined within UD^n. ($C_n \subsetneq C_{n+1}$ by Lemma 5.1.) Return to (1) with $n := n + 1$. If $C_n = UD^n$, then C_n gives the unique vNM set of G. (Theorems 5.1, 5.2.)

Moreover, the vNM set of a marriage game is a unique subsolution, which was defined by Roth (1976), of the game (Theorem 5.3).

We then reconsider stability of matchings in the vNM set. From the lattice property proved by Ehlers (2007), the vNM set of a marriage game has two polar matchings, each being optimal

*The author thanks for the very helpful comments from Lars Ehlers, Shigeo Muto, Alvin Roth, and an anonymous reviewer of ICALP 2008 workshop "Matching Under Preferences". Any errors are of course mine.

†Department of Economics, Gakushuin University, 1-5-1 Mejiro, Toshima-ku,Tokyo 171-8588, JAPAN. e-mail: jun.wako@gakushuin.ac.jp

for one side. They are immune to a sequence of myopic dominations, since a player in the advantageous side cannot end up being paired with more preferred partner. However, we argue that taking a more farsighted viewpoint, players in the disadvantageous side might breach matches unilaterally, and then the two sides could resume negotiations of which matching should be agreed on. To avoid such indeterminacy, all participants need to make a commitment to follow a matching prescribed by their agreed market clearing procedure.

2. Marriage games

Let $\mathcal{N}$ be the set of nonempty finite sets. For a given $S \in \mathcal{N}$, a *strict preference ordering* $\succ$ over the *choice set* S is a complete transitive antisymmetric binary relation over S. We mean by $h \succ j$ that h is preferable to j. Since $\succ$ is antisymmetric, let $h \succeq j$ indicate that $h \succ j$ or $h = j$ holds. The set of strict preference orderings over S is denoted by $\mathcal{P}(S)$.

A marriage game is played by two disjoint sets of players, a set $M \in \mathcal{N}$ of *men* and a set $W \in \mathcal{N}$ of *women*. Each men wants to be paired with a woman, and vice versa. Each player can also stay *unmatched* rather than being paired with an unacceptable partner. We formulate this game as follows. Let $N := M \cup W$. For each $i \in N$, let Y_i be the set of player i's potential partners, i.e., $Y_i = M$ if $i \in W$, or $Y_i = W$ if $i \in M$. Each player $i \in N$ has a strict preference ordering $\succ_i \in \mathcal{P}(Y_i \cup \{i\})$, where choosing him/herself i means "unmatched". Let $\succ$ denote the *preference profile* $(\succ_i)_{i \in N}$. The end result of the game is a *matching*, which is a bijection $\mu : N \to N$ such that for each $i \in N$, if $\mu(i) = j \neq i$ then $j \in Y_i$ and $\mu(j) = i$. We mean by $\mu(i) = i$ that i is unmatched, while $\mu(i) = j \neq i$ means that i and j belong to distinct sets and are matched.[1] The game thus formulated is referred to as *marriage game* $G = (M, W, \succ)$. We consider the following class $\mathbf{G}$ of marriage games: $\mathbf{G} = \{(M, W, \succ) \mid M \in \mathcal{N},\ W \in \mathcal{N},\ \succ_i \in \mathcal{P}(Y_i \cup \{i\})\ \forall i \in N\}$.

3. Core, vNM stable set, and subsolution

Let $G = (M, W, \succ)$ be a marriage game of class $\mathbf{G}$. The set of matchings in G is denoted by $\mathcal{M}$. Given $\mu, \nu \in \mathcal{M}$, we say "*μ dominates ν regarding $\succ$*" and write this as $\mu \,\mathrm{dom}[\succ]\, \nu$ if

(i) $\exists (i, j) \in M \times W$ with $\mu(i, j) = (j, i)$, $j \succ_i \nu(i)$, and $i \succ_j \nu(j)$; or

(ii) $\exists i \in N$ with $\mu(i) = i$ and $i \succ_i \nu(i)$.

When (i) is the case, we write it as "$\mu \,\mathrm{dom}[\succ]\, \nu$ via (i, j)." We also say "ν is *blocked* by (i, j)." The fact that μ does *not* dominate ν regarding $\succ$ is denoted by $\mu \,\neg\mathrm{dom}[\succ]\, \nu$. We omit symbol $[\succ]$ when no confusion may arise.

Let $IR[\succ] := \{\mu \in \mathcal{M} \mid \forall i \in N, \mu(i) \succeq_i i\}$, which is the set of *individually rational* matchings.[2] The *core* C of G is the set of matchings that are not dominated by any matching:

$$C := \{\mu \in \mathcal{M} \mid \nexists \nu \in \mathcal{M} : \nu \,\mathrm{dom}\, \mu\}.$$

[1] Since $\mu(i) = j \in Y_i \Leftrightarrow \mu(j) = i \in Y_j$, we also use notation $\mu(i, j) = (j, i)$.

[2] $IR[\succ]$ is simply denoted by IR when no confusion may arise.

The core of G can also be defined as the set of individually rational matchings that are not blocked by any $(i, j) \in M \times W$. We call a matching in the core a *core matching.*[3] For studying von Neumann-Morgenstern stable sets of marriage games, we give a more general definition of a core. For any nonempty subsets X and S of $\mathcal{M}$ and $\mu \in \mathcal{M}$, we define

$$D(\mu, X, \succ) := \{\nu \in X \mid \mu \,\mathrm{dom}[\succ]\, \nu\}, \quad D(S, X, \succ) := \cup_{\mu \in S} D(\mu, X, \succ), \text{ and}$$

$$UD(S, X, \succ) := X \setminus D(S, X, \succ).$$

We refer to $UD(X, X, \succ)$ as the *core in X regarding $\succ$*. The core C of G is the set $UD(\mathcal{M}, \mathcal{M}, \succ)$.

A *von Neumann-Morgenstern stable set in IR* (a vNM set, briefly) of G is defined to be any nonempty set $K \subseteq IR$ satisfying two stability conditions:

Internal stability: $\forall \mu, \nu \in K$, $\mu \neg\mathrm{dom}\, \nu$ and $\nu \neg\mathrm{dom}\, \mu$,

External stability: $\forall \nu \in IR \setminus K$, $\exists \mu \in K$: $\mu \,\mathrm{dom}\, \nu$.

The internal and external stability conditions can be represented as $K \subseteq UD(K, IR, \succ)$ and $K \supseteq UD(K, IR, \succ)$, respectively. A vNM set of G is thus a set $K \subseteq IR$ with

$$K = UD(K, IR, \succ),$$

which shows that a vNM set is a fixed point of set-to-set mapping $UD(\cdot, IR, \succ)$. Since multiple vNM sets may exist in G, let $\mathbf{vNM}(G)$ be the set of vNM sets of G.

A *subsolution in IR* (a subsolution, briefly) of G is defined to be any set $S \subseteq IR$ satisfying the **internal stability** above and

Self-protecting: $S = UD^2(S, IR, \succ) := UD(UD(S, IR, \succ), IR, \succ)$

The subsolution was defined by Roth (1976). It exists for any *abstract game* consisting of an outcome set X and a domination relation over X. Let $\mathbf{SB}(G)$ be the set of subsolutions of G.

Unlike the core, vNM sets and subsolutions, if they are defined in $\mathcal{M}$ by replacing each IR above with $\mathcal{M}$, can include not-individually-rational matchings. We thus need the whole preferences of all players to find a vNM set and a subsolution in $\mathcal{M}$. However, it is often the case with real-world matching environments that players only announce the individually rational part of their preferences.[4] Thus, we consider vNM sets and subsolutions in IR.

4. Basic properties

Let $G = (M, W, \succ)$ be any marriage game in $\mathbf{G}$. Symbols $\mathcal{M}$, IR, and C respectively denote the sets of matchings, individually rational matchings, and core matchings of G. For any $\mu \in \mathcal{M}$, define the set of unmatched players in μ to be the set $Un(\mu) := \{i \in N \mid \mu(i) = i\}$. Marriage game G has the following properties.

Proposition 4.1 (Gale and Shapley 1962). *$C = UD(IR, IR, \succ) \neq \emptyset$.*

Proposition 4.2 (Roth 1975,1976). *$\mathbf{SB}(G) \neq \emptyset$, and $C \subseteq S$ for any $S \in \mathbf{SB}(G)$.*

Proposition 4.3. *If $\mathbf{vNM}(G) \neq \emptyset$, then $C \subseteq K$ for any $K \in \mathbf{vNM}(G)$.*

---

[3] In the marriage game literature, a core matching is usually called a *stable* matching.

[4] In Japan, for example, we directly apply the Deferred Acceptance Algorithm by Gale and Shapley (1962) to the medical resident matching. There, the rank order lists submitted by the participants are treated as the individually rational part of their preference orderings.

Proposition 4.4. *For any $\mu, \nu \in \mathcal{M}$,*

(i) if $\mu \neg\text{dom}\, \nu$ and $\nu \neg\text{dom}\, \mu$, then $Un(\mu) = Un(\nu)$;

(ii) if $\mu, \nu \in C$, then $Un(\mu) = Un(\nu)$. $\quad (Un(C) := Un(\mu).)$

(iii) if $\mu \in C$ and $Un(\nu) \neq Un(C)$, then $\mu\, \text{dom}\, \nu$.

(iv) if $\mu \in C$ and $\mu \neg\text{dom}\, \nu$, then $Un(\nu) = Un(C)$;

Proposition 4.4.(i) means that if two matchings do not dominate each other, then the set of unmatched players are the same. Since any two core matchings do not dominate each other, we have property (ii): *The sets of unmatched players are the same throughout the core matchings.* This is a very well-known property of cores in marriage games. From this property, we define

$$Un(C) := Un(\mu) \text{ by picking any } \mu \in C.$$

We call $Un(C)$ the set of *core-unmatched* players. The set $N \setminus Un(C)$ gives the set of *core-matched* players. Property (iii) means that *each core matching dominates any matching whose set of unmatched players does not coincide with the set of core-unmatched players.* Property (iv) is the contraposition of (iii), which implies

$$Un(\mu) = Un(C) \text{ for each } \mu \in UD(C, IR, \succ).$$

Define $IR_0[\succ] := \{\mu \in IR \mid Un(\mu) = Un(C)\}$. we then have Proposition 4.5 below, since $C \subseteq UD(C, IR, \succ)$ by the definition of C. Since vNM sets are internally stable set including the core, Proposition 4.4.(ii) is extended to Proposition 4.6.

Proposition 4.5. $C \subseteq UD(C, IR, \succ) \subseteq IR_0[\succ]$.

Proposition 4.6 (Ehlers 2007)**.** *For any $K \in \mathbf{vNM}(G)$ and $\mu \in K$, $Un(\mu) = Un(C)$.*

Pick any $X \subseteq \mathcal{M}$ and $i \in N$, where X may be empty. Define $A(i, X)$ to be the set of i's *accessible partners in X*, i.e. $A(i, X) := \{j \in Y_i \mid \exists \mu \in X : \mu(i) = j\}$. For i's preference ordering $\succ_i$, we define a modified ordering as follows:

Definition 4.1. *For any given $X \subseteq \mathcal{M}$, player i's modified preference ordering $\succ[X]_i$ is the preference ordering in $\mathcal{P}(Y_i \cup \{i\})$ such that*

(i) for any $h, k \in A(i, X) \cup \{i\}$, $\quad h \succ[X]_i k \Leftrightarrow h \succ_i k$,

(ii) for any $h, k \in Y_i \setminus A(i, X)$, $\quad h \succ[X]_i k \Leftrightarrow h \succ_i k$,

(iii) for any $h \in A(i, X) \cup \{i\}$ and any $k \in Y_i \setminus A(i, X)$, $\quad h \succ[X]_i k$.

From condition (i), $\succ[X]_i$ keeps the same ordering as $\succ_i$ over $A(i, X) \cup \{i\}$. From condition (iii), $\succ[X]_i$ treats each of i's unaccessible partner in X as *unacceptable* ones.[5] We note that $j \in A(i, X) \Leftrightarrow i \in A(j, X)$. Using this modified preference ordering, we consider a marriage game that has a larger core than the original game.

Definition 4.2. *Let C be the core of G. The core extension of G is game $\hat{G} = (M, W, \hat{\succ}) \in \mathbf{G}$ with preference profile $\hat{\succ}$ defined by $\hat{\succ}_i := \succ [UD(C, IR, \succ)]_i$ for each $i \in N$.*

[5]The ordering among unaccessible partners defined by (ii) can be actually arbitrary in this note.

From Proposition 4.5, the set $IR[\widehat{\succ}]$ in $\hat{G}$ is given as below:

$$IR[\widehat{\succ}] = \left\{ \mu \in IR[\succ] \;\middle|\; \begin{array}{l} \forall i \in Un(C),\ \mu(i) = i; \\ \forall i \in N \setminus Un(C),\ \mu(i) \succeq_i i \text{ and } \mu(i) \in \{i\} \cup A(i, UD(C, IR[\succ], \succ)) \end{array} \right\}.$$

The core $\hat{C}$ of $\hat{G}$ is defined by $\hat{C} := UD(IR[\widehat{\succ}], IR[\widehat{\succ}], \widehat{\succ})$. The set $IR_0[\widehat{\succ}]$ is defined by $IR_0[\widehat{\succ}] := \{\mu \in IR[\widehat{\succ}] \mid Un(\mu) = Un(\hat{C})\}$. In the following, set $UD(C, IR[\succ], \succ) \setminus C$ plays an important role. Let $R := UD(C, IR[\succ], \succ) \setminus C$ and $\Delta C := UD(R, R, \succ)$.

Proposition 4.7. *Marriage game G and its core extension $\hat{G}$ have the following relations.*

(i) For any $\nu \in \mathcal{M}$ and any $\mu \in UD(C, IR[\succ], \succ)$, if $\nu \operatorname{dom}[\widehat{\succ}] \mu$, then there exists $\eta \in UD(C, IR[\succ], \succ)$ with $\eta \operatorname{dom}[\succ] \mu$.

(ii) $\hat{C}$ is equivalent to the core in $UD(C, IR[\succ], \succ)$ regarding $\succ$.

(iii) $C \subseteq \hat{C} \subseteq UD(\hat{C}, IR[\widehat{\succ}], \widehat{\succ}) \subseteq UD(C, IR[\succ], \succ)$. *(cf. Claim 7.2 for the proof)*

(iv) $C \cup \Delta C = \hat{C}$. *(cf. Claim 7.7 for the proof)*

(v) If $\mathbf{vNM}(\hat{G}) \neq \emptyset$, then $\hat{C} \subseteq K \subseteq UD(\hat{C}, IR[\widehat{\succ}], \widehat{\succ})$ for any $K \in \mathbf{vNM}(\hat{G})$.

(vi) $\mathbf{vNM}(\hat{G}) = \mathbf{vNM}(G)$.

We consider a sequence of marriage games iteratively defined by core-extension.

Definition 4.3. *(i) A sequence of marriage games starting with G is sequence $\{G_n\}$ with $G_0 = G$ and G_{n+1} being the core-extension $\hat{G}_n$ of G_n for each $n = 0, 1, \ldots$.*

(ii) Let $\succ^n$ and C_n denote the preference profile and core of G_n.

(iii) Let UD^n denote the set $UD(C^n, IR[\succ^n], \succ^n)$.

(iv) Let ΔC_n denote the set $UD(R^n, R^n, \succ^n)$, where $R^n := UD(C^n, IR[\succ^n], \succ^n) \setminus C^n$.

Applying Proposition 4.7 to each pair of games (G_0, G_1), (G_1, G_2),..., we have a non-decreasing sequence $\{C^n\}$ and a non-increasing sequence $\{UD^n\}$ such that

$$C_0 \subseteq \cdots \subseteq C_n \subseteq \cdots \subseteq UD^n \subseteq \cdots \subseteq UD^0 \subseteq IR[\succ^0].$$

Since $IR[\succ^0]$ is a finite set, *if $\{C^n\}$ is a strictly increasing sequence,* we have

$$C_0 \subset \cdots \subset C_k = UD(C^k, IR[\succ^k], \succ^k) \subseteq \cdots \subseteq UD^0.$$

at some finite number k. The set C^k is a vNM set of G^k, where $C_k = C_0 \cup \Delta C_0 \cup \cdots \cup \Delta C_{k-1}$ and $G^k = (M, W, \succ^k)$ with $\succ^k_i = \succ[UD(C^k, IR[\succ], \succ)]_i$ for each $i \in N$. By Proposition 4.7.(v) and (vi), C^k is the unique vNM set of the original game G.

5. Unique existence and equivalence of vNM set and subsolution

Let $G = (M, W, \succ)$ be any marriage game in $\mathbf{G}$ with the core C. For any matchings $\mu, \nu \in \mathcal{M}$, we define two mappings $\mu \wedge \nu$ and $\mu \vee \nu$ from N to N as follows:

For each $i \in M$, if $\mu(i) \succeq_i \nu(i)$, then $\mu \wedge \nu(i) := \nu(i)$ and $\mu \vee \nu(i) := \mu(i)$,
if $\nu(i) \succ_i \mu(i)$, then $\mu \wedge \nu(i) := \mu(i)$ and $\mu \vee \nu(i) := \nu(i)$.

For each $j \in W$, if $\mu(j) \succeq_j \nu(j)$, then $\mu \wedge \nu(j) := \mu(j)$ and $\mu \vee \nu(j) := \nu(j)$,
if $\nu(j) \succ_j \mu(j)$, then $\mu \wedge \nu(j) := \nu(j)$ and $\mu \vee \nu(j) := \mu(j)$.

Proposition 5.1. *For any core matchings $\mu, \nu \in C$,*

Conflict of interests (Knuth 1976): $[\mu(i) \succeq_i \nu(i)\ \forall i \in M] \Leftrightarrow [\nu(j) \succeq_j \mu(j)\ \forall j \in W]$,

Lattice property (Conway): $\{\mu \wedge \nu,\ \mu \vee \nu\} \subseteq C$.

Knuth (1976) gave an algorithm to find every core matching in a marriage game. Let X be any nonempty subset of $IR[\succ]$ in G. The core in X regarding $\succ$ is the set $UD(X, X, \succ)$. Suppose $\mu \in UD(X, X, \succ)$. If some matched players in μ can get *rematched* as described in the lemma below, then another core matching is obtained.[6]

Proposition 5.2. *Pick any $\mu \in UD(X, X, \succ)$, and suppose there exists an ordered set $S = \{(i_1, j_0), \ldots, (i_k, j_{k-1})\}$ of k pairs formed in μ such that*

(i) $|M| \geq k \geq 2$,

(ii) $j_{t-1} \succ_{i_t} j_t \succ_{i_t} i_t$, *and* $\nexists j \in A(i_t, X)$ *with* $j_{t-1} \succ_{i_t} j \succ_{i_t} j_t$ *for* $t = 1, \ldots, k$,

(iii) $i_t \succ_{j_t} i_{t+1}$ *for* $t = 0, 1, \ldots, k-1$,

where j_k and i_0 are treated as j_0 and i_k, respectively. Define matching ν by

$\nu(h) := \mu(h)$ *for each* $h \in N \setminus S$,

$\nu(i_t, j_t) := (j_t, i_t)$ *for each* $\{i_t, j_t\} \subset S$ *with* $t = 1, \ldots, k$.

If $\nu \in X$, then $\nu \in UD(X, X, \succ)$.

The condition (ii) means that j_t *is ranked lower next to j_{t-1} by i_t in X,* while each j_t prefers i_t to i_{t+1} from (iii). If they form pairs $(i_1, j_1), \ldots, (i_{k-1}, j_{k-1}), (i_k, j_0)$, we have a new core matching ν as far as ν is in X. This property proves Lemma 5.1 (see Section 7).

Lemma 5.1. *Let $\hat{G}$ be the core extension of G and $\hat{C}$ be the core of $\hat{G}$. Suppose $R = UD(C, IR[\succ], \succ) \setminus C \neq \emptyset$. Then we have $C \subsetneq C \cup \Delta C = \hat{C}$.*

Existence and uniqueness of a vNM set is easily derived from Propositions 4.7, Lemma 5.1, and the discussion associated with Definition 4.3.

Theorem 5.1. *For each marriage game $G \in \mathbf{G}$, a unique vNM set (in IR) exists.*

In general, the existence of an *individually rational* vNM set does not imply the existence of a vNM set in the set of all outcomes. In marriage games, the former implies the latter.

Theorem 5.2. *For each marriage game $G \in \mathbf{G}$, a unique vNM set in $\mathcal{M}$ exists.*

Proof. For any given $G = (M, W, \succ) \in \mathbf{G}$, let $G^\sharp$ be the game $(M, W, \succ^\sharp)$ in which each $i \in N$ has preference ordering $\succ_i^\sharp$ such that *(i) for any $h \in Y_i$, $h \succ_i^\sharp i$, and (ii) for any $h, k \in Y_i$, $h \succ_i^\sharp k \Leftrightarrow h \succ_i k$.* The set $\mathcal{M}$ of all matchings in G is then represented as $IR[\succ^\sharp]$ in $G^\sharp$. In addition, each $\succ_i^\sharp$ has the same ordering over Y_i as $\succ_i$. As Ehlers (2007) showed, we have that *K is a vNM set in $\mathcal{M}$ of G if and only if K is a vNM set in $IR[\succ^\sharp]$ of $G^\sharp$.* Since $G^\sharp \in \mathbf{G}$, it has a unique vNM set in $IR[\succ^\sharp]$ from Theorem 5.1. Thus, G has a unique vNM set in $\mathcal{M}$. □

We can also prove the equivalence of vNM sets and subsolutions. For any marriage game, its unique vNM set (in *IR* or $\mathcal{M}$) is also the unique subsolution (in *IR* or $\mathcal{M}$), respectively.

[6] Proposition 5.2 is the property discussed as the *elimination of a rotation* by Lemma 2.5.2 of Gusfield and Irving (1989), and also as the *cyclic matching* by Proposition 3.14 of Roth and Sotomayor (1990).

Theorem 5.3. *For each marriage game $G \in \mathbf{G}$, $\mathbf{SB}(G) = \{S\} = \mathbf{vNM}(G)$.*

Proof. In this proof, we simply write $UD(\cdot, IR, \succ)$ and $D(\cdot, IR, \succ)$ as $U(\cdot)$ and $D(\cdot)$. From Theorems 5.1, 5.2, and the discussion associated with Definition 4.3, game $G \in \mathbf{G}$ with the core C_0 has a unique vNM stable set K of the following form:

(i) $K = C_0$, or (ii) $K = C_0 \cup \Delta C_0 \cup \Delta C_1 \cup \cdots \cup \Delta C_{k-1}$ $(k \geq 1)$ such that

- ΔC_0 is the core in $R^0 := U(C_0) \setminus C_0$,
  ΔC_t is the core in $R^t := U(C_t) \setminus C_t$ for $t = 1, \ldots, k$,
  where $C_t := C_0 \cup \Delta C_0 \cup \cdots \cup \Delta C_{t-1}$, $R^k = \emptyset$, and $\Delta C_k = \emptyset$;
- $IR[\succ] = D(C_0) \cup C_0 \cup R^0$,
  $IR[\succ] = D(C_0) \cup D(\Delta C_0) \cup \cdots \cup D(\Delta C_{t-1}) \cup C_t \cup R^t$
  for $t = 1, \ldots, k$.

We will give a proof for Case (ii) above, from which the proof for Case (i) is obtained easily.

Since $\mathbf{SB}(G) \neq \emptyset$ by Proposition 4.2, let S be any subsolution of G. We then have $C_0 \subseteq S$. From the internal stability of S, we have $S \cap D(C_0) = \emptyset$. This implies $S \subseteq U(C_0)$. The fact that $C_0 \subseteq S$ also means $D(C_0) \subseteq D(S)$, and thus $U(S) \cap D(C_0) = \emptyset$.

Next assume that $\exists \mu \in \Delta C_0$ with $\mu \notin S$. Since ΔC_0 is the core in $R^0 = U(C_0) \setminus C_0$, there is no $\nu \in R^0$ with $\nu \operatorname{dom} \mu$. By the definition of $U(C_0)$, there is no $\nu \in C_0$ with $\nu \operatorname{dom} \mu$. Thus we have $\nu \operatorname{dom} \mu$ only if $\nu \in D(C_0)$. However, since $U(S) \cap D(C_0) = \emptyset$, we have $\mu \notin D(U(S))$, meaning $\mu \in U^2(S)$. However, this contradicts $\mu \notin S$, since $S = U^2(S)$. Thus $\Delta C_0 \subseteq S$.

Now that $\Delta C_0 \subseteq S$ and $C_0 \subseteq S$, the internal stability of S implies

$$C_0 \cup \Delta C_0 \subseteq S \subseteq U(C_0) \cap U(\Delta C_0) = U(C_0 \cup \Delta C_0). \tag{1}$$

$$U(S) \cap D(C_0 \cup \Delta C_0) = \emptyset. \tag{2}$$

Assume that $\exists \mu \in \Delta C_1$ with $\mu \notin S$. Since ΔC_1 is the core in $R^1 = U(C_1) \setminus C_1$ (recall $C_1 := C_0 \cup \Delta C_0$), there is no $\nu \in R^1$ with $\nu \operatorname{dom} \mu$. By the definition of $U(C_1)$, there is no $\nu \in C_1$ with $\nu \operatorname{dom} \mu$. Thus we have $\nu \operatorname{dom} \mu$ only if $\nu \in D(C_1)$. From (2), however, $\mu \in U^2(S)$. This contradicts $\mu \notin S$. Thus $\Delta C_1 \subseteq S$. Repeating the same argument, we have

$$C_0 \cup \Delta C_0 \cup \Delta C_1 \cup \cdots \cup \Delta C_{k-1} \subseteq S \subseteq U(C_0 \cup \Delta C_0 \cup \Delta C_1 \cup \cdots \cup \Delta C_{k-1}).$$

Since $K = C_0 \cup \Delta C_0 \cup \Delta C_1 \cup \cdots \cup \Delta C_{k-1}$ and $K = U(K)$, subsolution S is exactly the unique vNM stable set of G. Since S is any subsolution of G, its uniqueness is also proved. □

Remark. We have proved the existence of vNM sets in marriage games. However,the algorithm used in the proof is only guaranteed to end in finite steps. It is still an open question whether we can construct a polynomial-time algorithm to find a vNM set.

6. Reconsideration on stability of matchings in vNM stable sets

The internal and external stability conditions for a vNM set are not conditions that require each individual matching in the vNM set a certain kind of stability. However, if a particular matching in the vNM set is implemented by a centralized market clearing procedure, we should evaluate how robust the matching is by taking into account that more compounded domination behaviors may happen among players. We here consider two possible types of compounded domination: a sequential myopic domination and a farsighted breach of matches.

Let G be any marriage game in $\mathbf{G}$ and $\{G_n\}$ be a sequence of games starting with $G_0 = G$ (cf. Definition 4.3). Let C be the core of G. Suppose that the vNM set K of G was found at the kth game G_k as a fixed point C_k such that $C_k = UD(C_k, IR[\succ^k], \succ^k)$. As Ehlers (2007) proved, since this vNM set K can be regarded as the core C_k of G_k, it has the lattice property as shown in Proposition 5.1. Let us consider the case that $C \subsetneq K$ and the M-optimal matching μ^* in K is in the outside of C, i.e. $\mu^* := \vee_{\mu \in K} \mu \notin C$.

6.1. Immunity to a sequential myopic domination

Since $C_n = C_{n-1} \cup \Delta C_{n-1}$ for each n and $C_0 = C$ (cf. Definition 4.3), vNM set K is given as

$$K = C \cup \Delta C_0 \cup \Delta C_1 \cup \cdots \cup \Delta C_{k-1} \, (= C_k).$$

Since $\mu^* \in K \setminus C \subset IR[\succ]$, there exist $\nu_1 \in IR[\succ] \setminus K$ and $(i_1, j_1) \in M \times W$ such that

$$\nu_1 \text{ dom}[\succ] \, \mu^*, \;\; \nu_1(i_1) = j_1 \succ_{i_1} \mu^*(i_1), \;\text{ and }\; \nu_1(j_1) = i_1 \succ_{j_1} \mu^*(j_1).$$

From the external stability of K, $\mu_1 \text{ dom}[\succ] \, \nu_1$ for some $\mu_1 \in K$. If $\mu_1 \in C$ then μ_1 cannot be dominated by any matching. Let us suppose $\mu_1 \notin C$. In addition, assume $\mu_1 \in \Delta C_0$ to make explanation concise. Here, it should be recalled that

$C \cup \Delta C_0 (= C_1)$ is the core in $UD(C, IR[\succ], \succ)$ regarding $\succ$,

$D(C, IR[\succ], \succ) \cup UD(C, IR[\succ], \succ) = IR[\succ]$,

$D(C, IR[\succ], \succ) \cap UD(C, IR[\succ], \succ) = \emptyset$.

Since μ_1 is dominated by some matching, we have $\nu_2 \text{ dom}[\succ] \, \mu_1$ for some $\nu_2 \in D(C, IR[\succ], \succ)$. After all, the sequence of dominations ends up with a core matching $\mu_2 \in C$ with $\mu_2 \text{ dom}[\succ] \, \nu_2$. Let $\mu^+ := \vee_{\mu \in C} \mu$ be the M-optimal matching in C. Then we have

$$\nu_1(i_1) \succ_{i_1} \mu^*(i_1) \succeq_{i_1} \mu^+(i_1) \succeq_{i_1} \mu_2(i_1).$$

This show that player i_1 would not end up being paired with a better partner even if it might have been likely at the first domination. Thus, at least the M (or W)-optimal matching in a vNM set is immune to such a sequential myopic domination as above.

6.2. Indeterminacy from a breach of matches

If we take a more farsighted viewpoint, we see that *each* matching in a vNM set can be vulnerable to a unilateral breach of matches by players of the unsatisfied side. Those players may try to be matched with more suitable partners for them by becoming single temporarily.

Let μ be any matching in the vNM set K of G. Suppose a set S_w of matched women in μ breached their matches and become single. Let $S_m := \mu(S_w)$. By S_w's breach of matches, μ is changed to matching μ^- such that $\mu^-(i) = i$ for each $i \in S_w \cup S_m$; and $\mu^-(i) = \mu(i)$ for each $i \in N \setminus (S_w \cup S_m)$. Since K is obtained as the core C_k of $G_k = (M, W, \succ^k)$, we have $\mu^- \in IR[\succ^k]$ and $Un(\mu^-) \neq Un(C_k)$. From Proposition 4.4.(iii) and the definition of $\succ^k$,

$$\nu \text{ dom}[\succ] \, \mu^- \text{ for } \textit{each } \nu \in K.$$

Since Proposition 5.1.(Conflict of interests) also holds for the vNM set K, the fact that μ^- is dominated by each matching in K suggests that S_w's breaching matches could promote resumption of the whole negotiations on which matching the two sides should result in.

A breach of matches is most likely to occur when the M or W-optimal matching in K is implemented. In the case that the M-optimal matching $\mu^* \in K$ is implemented, if players in W breached matches, then it could make the W-players at least as well off as in μ^*. We should note however that a breach of matches can occur even when $\mu \in C$ $(\subseteq K)$ is implemented. To avoid such indeterminacy, all participants need to make a commitment to follow a matching prescribed by their agreed market clearing procedure.

7. Proof of Lemma 5.1

Lemma 5.1 plays a central role in deriving the main theorem 5.1. The other propositions are very well-known and/or can be proved easily.

Let $G = (M, W, \succ) \in \mathbf{G}$ be a marriage game with the core $C = UD(IR[\succ], IR[\succ], \succ)$. Let $\hat{G} = (M, W, \hat{\succ})$ be G's core extension (Definition 4.2) with the core $\hat{C} = UD(IR[\hat{\succ}], IR[\hat{\succ}], \hat{\succ})$. Let $R := UD(C, IR[\succ], \succ) \setminus C$ and $\Delta C := UD(R, R, \succ)$.

Claim 7.1. *If $R = \emptyset$, then $C = UD(C, IR[\succ], \succ)$, which is the unique vNM set of G.*

Proof. Since $UD(C, IR[\succ], \succ) \supseteq C \neq \emptyset$, if $R = \emptyset$ then $C = UD(C, IR[\succ], \succ)$, meaning $C \in \mathbf{vNM}(G)$. If $\exists K \in \mathbf{vNM}(G)$ with $K \neq C$, then $C \subsetneq K$ by Proposition 4.3. Thus $\exists \mu \in K \setminus C$. From the external stability of C, $\exists \nu \in C \subsetneq K$ with $\nu \operatorname{dom} \mu$. This contradicts the internal stability of K. Thus $\mathbf{vNM}(G) = \{C\}$. □

By Claim 7.1, we assume $R \neq \emptyset$ in the following. This assumption excludes the *inactive* marriage games having the core $C = \{\mu_0\}$ with $\mu_0(i) = i$ for each i. For those inactive games, $UD(C, IR[\succ], \succ) = \{\mu_0\}$ by Proposition 4.4.(iii). We have $R = \emptyset$, and Claim 7.1 holds.

Claim 7.2. $C \subseteq \hat{C} \subseteq UD(\hat{C}, IR[\hat{\succ}], \hat{\succ}) \subseteq UD(C, IR[\succ], \succ)$. (cf. Proposition 4.7.(iii))

Proof. Pick any $\mu \in C$. Since $\mu \in UD(C, IR[\succ], \succ)$, we have $\mu \in IR[\hat{\succ}]$. Assume $\mu \notin \hat{C}$, i.e. $\exists \nu \in IR[\hat{\succ}]$ with $\nu \operatorname{dom}[\hat{\succ}] \mu$. By the definition of $\hat{\succ}$, this means $\nu \operatorname{dom}[\succ] \mu$, which contradicts $\mu \in C$. Then $\mu \in \hat{C}$, and thus $C \subseteq \hat{C}$. By definition, $\hat{C} \subseteq UD(\hat{C}, IR[\hat{\succ}], \hat{\succ})$.

Pick any $\mu \in UD(\hat{C}, IR[\hat{\succ}], \hat{\succ})$. Since $\mu \in IR[\hat{\succ}] \subseteq IR[\succ]$, if $\mu \notin UD(C, IR[\succ], \succ)$, then $\exists \nu \in C$ with $\nu \operatorname{dom}[\succ] \mu$. Since $\nu \in UD(C, IR[\succ], \succ)$, $\nu \operatorname{dom}[\hat{\succ}] \mu$. This is impossible, since $\mu \in UD(\hat{C}, IR[\hat{\succ}], \hat{\succ})$ and $\nu \in C \subseteq \hat{C}$. Thus $\mu \in UD(C, IR[\succ], \succ)$. □

Claim 7.3. *If $A(i, R) \subseteq A(i, C)$ for each $i \in N$, then (i) $R = \Delta C$, and thus $C \subsetneq C \cup \Delta C$; and (ii) $C \cup \Delta C = \hat{C}$.*

Proof. (i) Suppose $\exists \{\mu, \nu\} \subseteq R$ with $\nu \operatorname{dom} \mu$. Since $\mu, \nu \in IR[\succ]$, $\exists (i, j) \in M \times W$ such that $\nu(i, j) = (j, i)$, $\nu(i) \succ_i \mu(i)$, and $\nu(j) \succ_j \mu(j)$. Since $A(i, R) \subseteq A(i, C)$ for each $i \in N$, $\exists \eta \in C$ with $\eta(i, j) = \nu(i, j) = (j, i)$ and $\eta \operatorname{dom} \mu$. This contradicts $\mu \in R \subset UD(C, IR[\succ], \succ)$. Thus $\nu \neg\operatorname{dom} \mu$ for any $\mu, \nu \in R$, i.e. $R = \Delta C$. Since $R \neq \emptyset$, we have $C \subsetneq C \cup \Delta C$. Claim 7.3.(ii) is proved under a more general setting in Claim 7.7. □

Lemma 5.1 will be proved by considering the case that $\exists i \in N$ with $A(i, R) \setminus A(i, C) \neq \emptyset$. For unmatched player $i \in Un(C)$, we have $A(i, R) \setminus A(i, C) = \emptyset$. Thus, letting $N_1 := N \setminus Un(C)$,[7]

[7] $N_1 \neq \emptyset$, since we have assumed $R \neq \emptyset$, which excludes the inactive cases with $Un(C) = N$,

we assume in the following that $\exists i \in N_1$ with $A(i,R) \setminus A(i,C) \neq \emptyset$.

Claim 7.4. *There exist $i \in N_1$, $h \in A(i,C)$, and $j \in A(i,R) \setminus A(i,C)$ such that $h \succ_i j$.*
Proof. Assume on the contrary that

$$\forall i \in N_1\ \forall h \in A(i,C)\ \forall j \in A(i,R) \setminus A(i,C),\ j \succ_i h. \tag{1}$$

Since $\exists i \in N_1$ with $A(i,R) \setminus A(i,C) \neq \emptyset$, pick any $i_0 \in N_1$ and $j_0 \in A(i_0,R) \setminus A(i_0,C)$. Then $j_0 \succ_{i_0} h$ for any $h \in A(i_0,C)$. We note $i_0 \in A(j_0,R) \setminus A(j_0,C)$. From (1), $i_0 \succ_{j_0} h$ for any $h \in A(j_0,C)$. Thus, $j_0 \succ_{i_0} \mu(i_0)$ and $i_0 \succ_{j_0} \mu(j_0)$ for any given $\mu \in C$. Let ν be any matching with $\nu(i_0,j_0) = (j_0,i_0)$. Then $\nu \operatorname{dom} \mu$ via (i_0,j_0). This contradiction deduces Claim 7.4. □

Let us write $UD(C, IR[\succ], \succ)$ as UD when no confusion arises. Note $UD = C \cup R$. From Claim 7.4, we may assume there are $i_1 \in M$, $j_0 \in A(i_1,C)$, and $j_1 \in A(i_1,R) \setminus A(i_1,C)$ with

$$j_0 \succ_{i_1} j_1, \tag{2}$$

$$\nexists j \in A(i_1, UD) \text{ with } j_0 \succ_{i_1} j \succ_{i_1} j_1. \tag{3}$$

Applying the lattice property of C (Proposition 5.1), define core matching $\mu^0 \in C$ by $\mu^0 := \wedge\{\mu \in C \mid \mu(i_1) \succeq_{i_1} j_0\}$.[8] We have $\mu^0(i_1,j_0) = (j_0,i_1)$, since $j_0 \in A(i_1,C)$.

Since $j_1 \in A(i_1,R) \setminus A(i_1,C)$, pick any $\nu^1 \in R$ with $\nu^1(i_1,j_1) = (j_1,i_1)$. Since μ^0 and ν^1 are bijections over finite set N ($= M \cup W$), there exists a set $S = \{i_1, \ldots, i_k, j_0, \ldots, j_{k-1}\}$ of $2k$ players (including i_1, j_0, and j_1) such that

$$|M| \geq k \geq 2,\ \{i_1,\ldots,i_k\} \subseteq M,\ \{j_0,\ldots,j_{k-1}\} \subseteq W, \tag{4}$$

$$\mu^0(i_t,j_{t-1}) = (j_{t-1},i_t) \text{ for } t = 1,\ldots,k, \tag{5}$$

$$\nu^1(i_t,j_t) = (j_t,i_t) \text{ for } t = 1,\ldots,k-1; \text{ and } \nu^1(i_k,j_0) = (j_0,i_k). \tag{6}$$

Claim 7.5. *For $t = 1,\ldots,k$, $\mu^0(i_t) \succ_{i_t} \nu^1(i_t)$. For $t = 0,\ldots,k-1$, $\nu^1(j_t) \succ_{j_t} \mu^0(j_t)$.*
Proof. First we have $\mu^0(i_1) = j_0 \succ_{i_1} \nu^1(i_1) = j_1$ [i.e., $\mu^0(i_t) \succ_{i_t} \nu^1(i_t)$ for $t = 1$].

If $\mu^0(j_0) = i_1 \succ_{j_0} \nu^0(j_0) = i_k$, then $\mu^0 \operatorname{dom} \nu^1$ via (i_1,j_0). This contradicts $\nu^1 \in R \subset UD$. Thus, $\nu^1(j_0) = i_k \succ_{j_0} \mu^0(j_0) = i_1$ [i.e., $\nu^1(j_t) \succ_{j_t} \mu^0(j_t)$ for $t = 0$].

If $\nu^1(i_k) = j_0 \succ_{i_k} \mu^0(i_k) = j_{k-1}$, then $\nu^1 \operatorname{dom} \mu^0$ via (i_k,j_0). This contradicts $\mu^0 \in C$. Thus, $\mu^0(i_k) = j_{k-1} \succ_{i_k} \nu^1(i_k) = j_0$ [i.e., $\mu^0(i_t) \succ_{i_t} \nu^1(i_t)$ for $t = k$].

If $\mu^0(j_{k-1}) = i_k \succ_{j_{k-1}} \nu^1(j_{k-1}) = i_{k-1}$, then $\mu^0 \operatorname{dom} \nu^1$ via (i_k,j_{k-1}). This contradicts $\nu^1 \in UD$. Thus, $\nu^1(j_{k-1}) = i_{k-1} \succ_{j_{k-1}} \mu^0(j_{k-1}) = i_k$ [i.e., $\nu^1(j_t) \succ_{j_t} \mu^0(j_t)$ for $t = k-1$]. Claim 7.5 is proved by repeating the same argument. □

From Claim 7.5 together with (4)and (5), we have $j_1 \in A(i_2,C)$, $j_2 \in A(i_2,R)$, and $j_1 \succ_{i_2} j_2$. However, j_2 may not be ranked lower next to j_1 by i_2 in UD. We then carry out the *Cycle search procedure* given below by setting counter $t := 1$.

Cycle search procedure:

(i) Replace the indices of i_1, j_0, i_2, and j_1 as follows: $i^*_t \leftarrow i_1$, $j^*_{t-1} \leftarrow j_0$, $i_1 \leftarrow i_2$, $j_0 \leftarrow j_1$.

(ii) (Under the replaced indices,) let $j_1 \in A(i_1, UD)$ be the element that is ranked lower next to j_0 by i_1 in UD.

[8] For any finite set of matchings $S = \{\mu_1,\ldots,\mu_t\}$, symbol $\wedge S$ denotes $\mu_1 \wedge \cdots \wedge \mu_t$.

(iii) Since $j_1 \in A(i_1, UD)$, pick any $\nu^{t+1} \in UD$ with $\nu^{t+1}(i_1, j_1) = (j_1, i_1)$.

(iv) If $j_1 \in \{j_0^*, \ldots, j_{t-1}^*\}$, then set $i_{t+1}^* \leftarrow i_1$, $j_t^* \leftarrow j_0$, and quit the procedure.

(v) Let $S = \{i_1, \ldots, i_k, j_0, \ldots, j_{k-1}\}$ be the set including i_1, j_0, and j_1 with properties (4)–(6) [where ν^1 is replaced with ν^{t+1}]. Return to (i) with $t := t + 1$.

Since player set N_1 is finite, the Cycle search procedure will end in finite steps. We will then obtain a set $S^* = \{i_{t_1}^*, \ldots, i_{t_q}^*, j_{t_0}^*, \ldots, j_{t_{q-1}}^*\}$ such that

$$|M| \geq q \geq 2,\ \{i_{t_1}^*, \ldots, i_{t_q}^*\} \subseteq M,\ \{j_{t_0}^*, \ldots, j_{t_{q-1}}^*\} \subseteq W, \tag{7}$$

$$\mu^0(i_{t_h}^*, j_{t_{h-1}}^*) = (j_{t_{h-1}}^*, i_{t_h}^*) \text{ for } h = 1, \ldots, q, \tag{8}$$

$$\nu^{t_h}(i_{t_h}^*, j_{t_h}^*) = (j_{t_h}^*, i_{t_h}^*) \text{ for } h = 1, \ldots, q-1; \text{ and } \nu^{t_q}(i_{t_q}^*, j_{t_0}^*) = (j_{t_0}^*, i_{t_q}^*). \tag{9}$$

Since each $\nu^{t_h} \in UD$ for $h = 1, ..., q$, Claim 7.5 holds for μ^0 and each ν^{t_h} and set S constructed at Step (v) of each round. Furthermore, by Step (ii), the elements of S^* satisfies

$$j_{t_h}^* \text{ is ranked lower next to } j_{t_{h-1}}^* \text{ by } i_{t_h}^* \text{ in } UD \text{ for } h = 1, \ldots, q-1, \tag{10}$$

$$j_{t_0}^* \text{ is ranked lower next to } j_{t_{q-1}}^* \text{ by } i_{t_q}^* \text{ in } UD \text{ (for } h = q), \tag{11}$$

$$i_{t_h}^* \succ_{j_{t_h}^*} i_{t_{h+1}}^* \text{ for } h = 1, \ldots, q-1; \quad \text{and} \quad i_{t_q}^* \succ_{j_{t_0}^*} i_{t_1}^*. \tag{12}$$

Define μ^* to be the matching such that

$\mu^*(i) := \mu^0(i)$ for each $i \in N \setminus S^*$,

$\mu^*(i_{t_h}^*, j_{t_h}^*) := \nu^{t_h}(i_{t_h}^*, j_{t_h}^*) = (j_{t_h}^*, i_{t_h}^*)$ for $\{i_{t_h}^*, j_{t_h}^*\} \subset S^*$ with $h = 1, \ldots, q-1$,

$\mu^*(i_{t_q}^*, j_{t_0}^*) := \nu^{t_k}(i_{t_k}^*, j_{t_0}^*) = (j_{t_0}^*, i_{t_q}^*)$ for $\{i_{t_q}^*, j_{t_0}^*\} \subset S^*$.

This matching turns out to be a core matching in $\hat{G}$.

Claim 7.6. *μ^* is a core matching in UD regarding $\succ$, and $\mu^* \in \hat{C} = UD(IR[\widehat{\succ}], IR[\widehat{\succ}], \widehat{\succ})$.*

Proof. First, we will show $\mu^* \in UD$ $(= UD(C, IR[\succ], \succ))$. The pairs formed by μ^* are those formed by $\mu^0 \in C$ and $\{\nu^{t_1}, \ldots, \nu^{t_q}\} \subset UD$. Thus $\mu^* \in IR[\succ]$. Assume on the contrary that $\exists \nu \in C$ with $\nu \,\text{dom}\, \mu^*$ via some pair $(i, j) \in M \times W$ with $\nu(i, j) = (j, i)$. If $\{i, j\} \subset N \setminus S^*$, it means that ν dominates $\mu^0 \in C$, which is a contradiction. If $i \in N \setminus S^*$ and $j = j_{t_h}^* \in S^* \cup W$, then $\nu(i) \succ_i \mu^*(i) = \mu^0(i)$ and $\nu(j_{t_h}^*) \succ_{j_{t_h}^*} \mu^*(j_{t_h}^*) = i_{t_h}^* \succ_{j_{t_h}^*} i_{t_{h+1}}^* = \mu^0(j_{t_h}^*)$. This also contradicts $\mu^0 \in C$. Next suppose $i = i_{t_r}^* \in S^* \cap M$ and $j = j_{t_h}^* \in S^* \cap W$. Since $\mu^*(i_{t_r}^*) = j_{t_r}^*$ is ranked lower next to $j_{t_{r-1}}^* = \mu^0(i_{t_r}^*)$ in UD, if $\nu(i_{t_r}^*) \succ_{i_{t_r}^*} \mu^*(i_{t_r}^*)$, then $\nu(i_{t_r}^*) \succ_{i_{t_r}^*} \mu^0(i_{t_r}^*) \succ_{i_{t_r}^*} \mu^*(i_{t_r}^*)$. As for $j_{t_h}^*$, we have $\nu(j_{t_h}^*) \succ_{j_{t_h}^*} \mu^*(j_{t_h}^*) = i_{t_h}^* \succ_{j_{t_h}^*} i_{t_{h+1}}^* = \mu^0(j_{t_h}^*)$. We can also deduce a contradiction when $i = i_{t_h}^* \in S^* \cap M$ and $j \in N \setminus S^*$. We have a contradiction for all possible cases. Thus, $\nexists \nu \in C$ with $\nu \,\text{dom}\, \mu^*$, i.e. $\mu^* \in UD$.

Since we have (10)–(12), it follows from Proposition 5.2 that μ^* is a core matching in UD regarding $\succ$. From the definition of $\widehat{\succ}$, we have $\mu^* \in \hat{C}$ (cf. Proposition 4.7.(ii)). □

Claim 7.7. *$C \cup \Delta C = \hat{C}$.* (cf. Proposition 4.7.(iv))

Proof. Assume that $\exists \mu \in \hat{C} \setminus (C \cup \Delta C)$. Since $\mu \in IR[\widehat{\succ}] \subseteq IR$, we have $\mu \in D(C, IR, \succ)$ or $\mu \in UD$ $(= UD(C, IR, \succ))$. If $\mu \in D(C, IR, \succ)$, then $\exists \nu \in C$ with $\nu \,\text{dom}[\succ]\, \mu$ via some (i, j). Since $C \subseteq UD$ and $\nu \in IR$, we have $\nu \in IR[\widehat{\succ}]$. This contradicts $\mu \in \hat{C}$. If $\mu \in UD$, the assumption that $\mu \notin C \cup \Delta C$ means $\mu \in R \setminus \Delta C$. Thus $\exists \nu \in R$ with $\nu \, \mathrm{dom}[\succ] \, \mu$ via some (i,j). Since $R \subset UD$ and $\nu \in IR$, we have $\nu \in IR[\widehat{\succ}]$. This contradicts $\mu \in \hat{C}$. A contradiction arises for each possible case. Thus $\hat{C} \subseteq C \cup \Delta C$.

Assume $\exists \mu \in (C \cup \Delta C) \setminus \hat{C}$. Since $\mu \in C \cup \Delta C$, we have $\mu \in IR[\widehat{\succ}]$. Thus, $\exists \nu \in IR[\widehat{\succ}]$ with $\nu \, \mathrm{dom}[\widehat{\succ}] \, \mu$ via some (i,j) with $\nu(i,j) = (j,i)$. This implies that $\exists \nu' \in UD$ with $\nu'(i,j) = \nu(i,j) = (j,i)$ and $\nu' \, \mathrm{dom}[\succ] \, \mu$ via (i,j). Note $C \subseteq UD(UD, IR, \succ)$. If $\mu \in C$, then $\nu' \, \neg\mathrm{dom}[\succ] \, \mu$ for any $\nu' \in UD$. This means $\mu \notin C$, and thus $\mu \in \Delta C$. Since $\Delta C \subset R \subset UD$, we have $\nu' \in R$. However, we cannot have $\nu' \, \mathrm{dom}[\succ] \, \mu$, since $\mu \in \Delta C = UD(R, R, \succ)$. Hence, $C \cup \Delta C \subseteq \hat{C}$. □

We finally prove that the core C of G is a strict subset of the core $\hat{C}$ of $\hat{G}$. This completes the proof of Lemma 5.1.

Claim 7.8. *$\mu^* \in \Delta C = UD(R, R, \succ)$, and thus $C \subsetneq C \cup \Delta C = \hat{C}$.*

Proof. Suppose $\{i_1^*, j_1^*\} \subset S^*$. This pair, reindexed by the Cycle search procedure, has the property that $j_1^* \in A(i_1^*, R) \setminus A(i_1^*, C)$. Thus, there is no core matching in C that contains pair (i_1^*, j_1^*). However, $\mu^* \in \hat{C}$ from Claim 7.6. Thus we have $\mu^* \in \Delta C$ from Claim 7.7.

If $\{i_1^*, j_1^*\} \not\subset S^*$, then $\{i_1^*, j_0^*\} \cap S^* = \emptyset$ from the definition of S^*. (Recall that i_1^* and j_0^* was indexed as i_1 and j_0 when they were first picked up.) From (7)–(12) and the definition of μ^*,

$$\mu^0(i_1^*) = j_0^* = \mu^*(i_1^*) \tag{13}$$

$$\mu^0(i) \succeq_i \mu^*(i) \text{ for each } i \in M, \tag{14}$$

$$\mu^0(i) \succ_i \mu^*(i) \text{ for each } i \in S^* \cap M. \tag{15}$$

If $\mu^* \in C$, then $\mu^* \in \{\mu \in C \mid \mu(i_1^*) \succeq_{i_1^*} j_0^*\}$ from (13). From the definition of μ^0, we then have $\mu^*(i) \succeq_i \mu^0(i)$ for *each* $i \in M$, which contradicts (15). However, $\mu^* \in \hat{C}$ from Claim 7.6. Thus $\mu^* \in \Delta C$ from Claim 7.7. The proof of Claim 7.8 is completed. □

References

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

L. Ehlers, Von Neumann-Morgenstern stable sets in matching problems, *Journal of Economic Theory,* vol.134 (2007) 537-547

D. Gusfield and R.W. Irving, *The Stable Marriage Problem: Structure and Algorithms,* MIT Press, 1989

D. Knuth, *Marriages Stables,* Les Presses de l'Université de Montreal, 1976

A. Roth, A Lattice Fixed-Point Theorem with Constraints, *Bulletin of The American Mathematical Society,* vol. 81 (1975) 136-138

A. Roth, Subsolutions and the Supercore of Cooperative Games, *Mathematics of Operations Research,* vol. 1 (1976) 43-49

A. Roth and M. Sotomayor,*Two-sided matching: A study in game-theoretic modeling and analysis,* Cambridge Univ. Press, 1990