MATCH-UP 2012:

the Second International Workshop on Matching Under Preferences

19-20 July 2012
Corvinus University of Budapest, Hungary
http://econ.core.hu/english/res/MATCH-UP_2012.html

co-located with SING8: The 8th Spain-Italy-Netherlands Meeting on Game Theory (http://sing8.iehas.hu/)

sponsored by
Faculty of Economics, Corvinus University of Budapest
Institute of Economics, Research Centre for Economic and Regional Studies,
Hungarian Academy of Sciences (under its Momemtum Programme, LD-004/2010).
School of Computing Science, University of Glasgow

MATCH-UP 2012:
the Second International Workshop on Matching Under Preferences

Preface

Celebrating the 50th anniversary of the seminal paper by Gale and Shapley, and following the success of the first MATCH-UP workshop in Reykjavík in 2008, we decided to organise another interdisciplinary workshop on stable matchings and related topics.

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 is to explore matching problems with preferences from the perspective of algorithms and complexity, discrete mathematics, combinatorial optimization, game theory, mechanism design and economics, and thus a key objective is to bring together the research communities of the related areas.

Unlike in 2008, this time we decided to call for two types of submissions. We required Format A papers to be original and at most 12-pages long for inclusion in these proceedings. Format B papers had no restriction on length or originality, with just the abstract of accepted papers being included in what follows.

Our call for papers generated much interest: we received 37 good quality submissions (17 Format A and 20 Format B), which were well-balanced in terms of representing the computing science and economics communities. Due to the time constraints and our strong intention to avoid parallel sessions, we accepted 26 submissions (10 Format A and 16 Format B). Following the withdrawal of one paper, 25 contributed papers will be presented at the workshop and appear in these proceedings.

We feel that these papers represent 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 programme committee (and additional reviewers), the invited speakers and the authors of all submitted papers for helping to make this workshop a success.

Péter Biró
Tamás Fleiner
David Manlove
Tamás Solymosi

Program Committee

- Péter Biró, chair, Institute of Economics, Hungarian Academy of Sciences
- Estelle Cantillon, Université Libre de Bruxelles
- Katarína Cechlárová, Univerzita Pavla Jozefa Safárika
- Paul David Dütting, EPFL, Lausanne
- Aytek Erdil, University of Cambridge
- Tamás Fleiner, Budapest University of Technology and Economics
- Guillaume Haeringer, Universitat Autonoma de Barcelona
- Elena Inarra, University of the Basque Country
- Zoltán Király, Eötvös Loránd University, Budapest
- Flip Klijn, Universitat Autonoma de Barcelona
- David Manlove, University of Glasgow
- Eric McDermid, 21CT
- Shuichi Miyazaki, Kyoto University
- Marina Nunez, Universitat de Barcelona
- Ildikó Schlotter, Budapest University of Technology and Economics
- Tamás Solymosi, Corvinus University of Budapest

Table of Contents

Invited speaker abstracts

Format A papers

Format B paper abstracts

Invited speaker abstracts

Two-Sided Matching with Partial Information

Nicole Immorlica
Department of Electrical Engineering and Computer Science,
Northwestern University,
Email: nickle@eecs.northwestern.edu

A critical problem with the traditional model of two-sided matching is that all agents are assumed to fully know their own preferences. As markets grow large, it quickly becomes impractical for participants to assess their precise preference rankings. We propose a novel model of two-sided matching in which agents are endowed with partially ordered preferences over candidates, but can refine these preferences through interviews. We further assume that an agents true preference ordering is some strict ordering consistent with this coarse ranking. To learn more about their preferences, agents must conduct interviews. We assume that the interviews reveal the pairwise rankings among all interviewed candidates. Our goal is to identify a centralized interview schedule that uncovers sufficient information to guarantee that the resulting matching is stable and optimal for a given side of the market, with respect to the underlying true and strict preferences of the participants. Clearly, such schedules exist; e.g., we could simply conduct all possible interviews. However, interviews are costly, so we aim to minimize their number. Our key contributions beyond the model itself are a formalization of what it means to minimize the number of interviews, a computationally efficient interview minimizing algorithm for a restricted setting, and an NP-completeness result that suggests identifying interview minimizing schedules are hard in general.

Joint work with Anne Condon, Kevin Leyton-Brown, and Baharak Rastegari.

Medical Matching in Scotland: Reflections on the Interplay of Theory and Practice

Rob Irving
School of Computing Science,
University of Glasgow,
Email: Rob.Irving@glasgow.ac.uk

Back in 1997, the Postgraduate Medical Institute in Glasgow proposed a Masters project involving the development of software to assist with the annual task of assigning graduating medical students to their training positions in hospitals. (In those days, departments throughout the University of Glasgow were invited to submit project proposals for students on the Masters course in Information Technology.) At that time, the assignment process was a ’free market’ graduates had to find their own positions by applying directly to hospitals. As had already been observed and documented in other contexts, this free market approach led to considerable chaos, and was profoundly unpopular with all of those involved, particularly the students.

My supervision of this Masters project initiated a period of collaboration with the medical authorities in Scotland that has continued right up to the present day. Of course, our use of matching algorithms to allocate medical students to hospital posts was not new the long history of the National Resident Matching Program (NRMP) in the US had been well-known and prominent in the literature for many years. However, each context typically has certain special features that distinguish it from other similar applications, and the requirements of the Scottish matching scheme, as they developed over the years, have thrown up a variety of interesting challenges, both theoretical and practical.

The key concept of the stability of a matching was recognised as crucial from the outset, and has remained so as the detailed requirements of the scheme have changed over time. The classical stable matching problem finding a stable matching of students to hospitals when strict preferences are expressed on both sides can be easily and optimally solved by the Gale-Shapley algorithm. However, the situation becomes potentially more interesting if, for example, students are to be assigned to pairs of hospitals, or if preferences are not strict, or if couples express joint preferences, or if hospital preferences are generated from a ’master’ list of student scores. Conditions of this kind have arisen in the Scottish scheme over the years, typically resulting in an NP-hard variant of the stable matching problem. The search for satisfactory solutions in these situations has led to some non-trivial extensions of the Gale-Shapley approach, and the need to carry out empirical evaluations of competing strategies. It is these variants of the classical stable matching problem, the relevant theoretical results that have been established, and the algorithms that have been developed to handle the problems in practice, that form the subject of this presentation.

Promoting School Competition Through School Choice: A Market Design Approach

Fuhito Kojima

The invited talk is based on a paper with the same title, a joint work with John William Hatfield and Yusuke Narita.

We study the effect of different school choice mechanisms on schools' incentives for quality improvement. To do so, we introduce the following criterion: A mechanism respects improvements of school quality if each school becomes weakly better off whenever that school becomes more preferred by students. We first show that no stable mechanism, or mechanism that is Pareto efficient for students (such as the Boston and top trading cycles mechanisms), respects improvements of school quality. Nevertheless, for large school districts, we demonstrate that any stable mechanism approximately respects improvements of school quality; by contrast, the Boston and top trading cycles mechanisms fail to do so. Thus a stable mechanism may provide better incentives for schools to improve themselves than the Boston and top trading cycles mechanisms.

Cadet-Branching at U.S. Army Programs

Tayfun Sönmez
Department of Economics,
Boston College,
Email: tayfun.sonmez@bc.edu

The invited talk is based on two papers. The first is a joint work with Tobias B. Switzer with title ”Matching with (Branch-of-Choice) Contracts at United States Military Academy”.

Prior to 2006, the United States Military Academy (USMA) matched cadets to military specialties (branches) using a single category ranking system to determine priority. Since 2006, priority for the last 25 percent of the slots at each branch has been given to cadets who sign a branch-of-choice contract committing to serve in the Army for three additional years. Of the three incentive plans implemented under the Officer Career Satisfaction Program (OCSP), this change in matching has been the most effective in combating historically low retention rates among junior army officers. Building on theoretical work of Hatfield and Milgrom (2005) and Hatfield and Kojima (2010), we show that the resulting new matching problem not only has practical importance but also it fills a gap in the market design literature. Even though the new branch priorities designed by the Department of the Army fail a substitutes condition, the cumulative offer algorithm of Hatfield-Milgrom gives a cadet-optimal stable outcome in this environment. The resulting mechanism restores a number of important properties to the current USMA mechanism including stability, strategy-proofness and fairness which not only increase cadet welfare consistent with OCSP goals but also provides the Army with very accurate estimates of the effect of a change in the parameters of the mechanism on number of man-year gains by the branch-of-choice incentive program. Our paper also shows that matching with contracts model have great potential to prescribe solutions to real-life resource allocation problems beyond domains that satisfy the substitutes condition.

The title of the second paper is ”Bidding for Army Career Specialties: Improving the ROTC Branching Mechanism”.

Motivated by low retention rates of USMA and ROTC graduates, the Army recently introduced incentives programs where cadets could bid three years of additional service obligation to obtain higher priority for their desired branches. The full potential of this incentives program is not utilized, due to ROTC’s deficient matching mechanism. We propose a design that eliminates these shortcomings and benefits the Army by mitigating several policy problems it has identified. In contrast to the ROTC mechanism, our design utilizes market principles more elaborately, and it is a hybrid between a market mechanism and a priority-based allocation mechanism.

Format A papers

Stability of Marriage and Vehicular Parking

Daniel Ayala, Ouri Wolfson, Bo Xu,
Bhaskar DasGupta, and Jie Lin

University of Illinois at Chicago

Abstract. The proliferation of mobile devices, location-based services and embedded wireless sensors has given rise to applications that seek to improve the efficiency of the transportation system. In particular, new applications are arising that help travelers find parking in urban settings. They convey the parking slot availability around users on their mobile devices. Nevertheless, while engaged in driving, travelers are better suited being guided to an ideal parking slot, than looking at a map and deciding which open slot to visit. Then the question of how an application should choose this ideal parking slot to guide the user towards it becomes relevant.

Vehicular parking can be viewed as vehicles (players) competing for parking slots (resources with different costs). Based on this competition, we present a game-theoretic framework to analyze parking situations. We introduce and analyze parking slot assignment games and present algorithms that choose parking slots ideally in competitive parking simulations. We also present algorithms for incomplete information contexts and show how these algorithms outperform greedy algorithms in most situations.

Keywords: Stable Marriage, Spatio-temporal resources, Vehicular Parking

1 Introduction

Finding parking can be a major hassle for drivers in some urban environments. The advent of wireless sensors that can be embedded on parking slots has enabled the development of applications that help mobile device users find available parking slots around their locations. A prime example of this type of application is SFPark [1]. It uses sensors embedded in the streets of the city of San Francisco, that can tell if a slot is available. When a user wants to find a parking slot in some area of the city, the application shows a map with marked locations of the open parking slots in the area.

While this type of application is useful for finding the open parking slots around you, it does raise some safety concerns for travelers. The drivers have to shift their focus from the road, to the mobile device they are using. Then they have to look at the map and make a choice about which parking slot to choose from all the available slots that are shown on the map. It would be better (safer) if the app just guided the driver to an exact location where they are most likely to find an open parking slot. Then the question arises, which algorithm should the mobile app use to choose such an *ideal* parking location?

Our main concern in this work is to answer the preceding question. Regardless of the safety concerns stated in the previous paragraph, the question still remains relevant. What is the optimal way of moving towards spatially located resources, to obtain a resource, when there is competition for the resource?

Parking can be viewed as a continuous query submitted by mobile devices to obtain information about spatial resources (parking slots). A mobile user wants to know which is the parking slot to visit in order to minimize various possible utilities like: distance traveled, walking distance to their destination, or monetary price of the parking slot. However, parking is also competitive in nature because after making a choice to visit a particular slot, the success in obtaining that slot will depend on if any other vehicles closer to that slot also made the same choice. This competition for resources (slots) lends itself for modeling this situation in a game-theoretic framework. We then present parking slot assignment games (PSAG) for studying competitive parking situations.

Two categories of PSAG will be considered in this work, complete and incomplete information PSAG. For the complete information PSAG, we relate the problem of finding the Nash equilibrium to the Stable Marriage problem [2]. We show the equivalency of Nash equilibria and Stable Marriage assignments for instances of PSAG.

For the incomplete information PSAG, the model that is most realistic and directly applicable to real-life application of parking slot choice, we present a gravitational approach for choosing parking. The Gravity-based Parking algorithm (GPA) is presented for this model. We also present an adaptation for GPA to road networks and show its merits through simulation.

2 General Setup and Notation

The general setup of the parking problem is as follows:

- There are two types of *objects* as follows.
  - A set of n vehicles $V = \{v_1, v_2, \ldots, v_n\}$.
  - A set of m open parking slots $S = \{s_1, s_2, \ldots, s_m\}$.
- $\mathsf{dist} : (V \cup S) \times S \to \mathbb{R}$ is a distance function. It denotes the distance between a vehicle and a slot, or the distance between two slots.
- $\mathsf{cost} : V \times S \to \mathbb{R}$ is a cost function. It denotes the *cost* of a slot $s_j \in S$ to a vehicle $v_i \in V$. This cost is a general cost. It could include the distance from the vehicle to the slot, $\mathsf{dist}(v_i, s_j)$, the walking distance from s_j to v_i's destination, and/or other utilities that v_i cares about when choosing a slot.
- Each vehicle is assumed to be moving independently of all other vehicles at a fixed velocity. Without loss of generality, we assume that the speeds of all vehicles are the same[1].

[1] Otherwise, we simply need to rescale the distances for each vehicle in our algorithmic strategies.

- A valid *assignment* of vehicles to slots is one where each vehicle is assigned to exactly one slot. It can be defined as a function $g : V \rightarrow S$, where $g(v)$ is the assigned slot for vehicle $v \in V$.[2]
- The *cost of an assignment* g for a player $v \in V$, $C_g(v)$, is defined as $\mathsf{cost}(v, g(v))$ if of all players assigned to slot $g(v)$, v is the closest to it; *i.e.*

$$v = \underset{v' \in V : g(v') = g(v)}{\operatorname{argmin}} \{\mathsf{dist}(v', g(v))\}. \tag{1}$$

  Here the argmin function returns the parameter that minimizes the given function. If some other vehicle assigned to $g(v)$ is closer to it than v, then v's cost based on g is $C_g(v) = \mathsf{cost}(v, g(v)) + \alpha$, where α is a large *penalty* (could be the sum of all costs) for not obtaining a parking slot.
- The *total cost of an assignment* g, C_g, is defined as:

$$C_g = \sum_{v \in V} C_g(v) \tag{2}$$

It should be noted that this type of model could be generalized to considering mobile agents (vehicles) that are looking to obtain one of a set of static resources (parking slots) on a map. Besides parking, another application that could conform to this model is one where taxicabs (mobile agents) are competing to obtain clients (static resources) that have a location on a map.

3 Parking Slot Assignment Games

One could define a model in which a centralized authority was in charge of assigning the vehicles to slots. This authority would be looking to minimize some system-wide objectives (optimizing social welfare). In the transportation literature this is usually called a system optimal assignment. In [3], we show how this system optimal assignment can be computed in polynomial time. Even though this centralized model shows good computational properties, it is difficult to justify in real life to distributed mobile users that make their own choices. This is because optimizing social welfare may imply that some travelers will incur a greater cost for the good of others.

We then model parking as a competitive game in which individual, selfish players are competing for the available slots. Any game has three essential components: a set of *players*, a set of possible *strategies* for the players and a payoff function (cost function) [4]. The payoff function determines what is the cost to each player based on a given *strategy profile.* If there are n players in the game then a *strategy profile* is an n-tuple in which the ith coordinate represents the strategy choice of the ith player. It basically represents the choices made by the n players.

[2] Based on this definition, there is a difference between where a vehicle is assigned and where a vehicle parks. If more than one vehicle is assigned to the same slot, then the closest one to it will park there. The others are left without parking. This will always happen when $n > m$.

In our case for the parking problem, we can define the *parking slot assignment game* (PSAG) as follows:

- The set of *players* in PSAG is V (the vehicles).
- The set of available *strategies* to each player is S (the slots).
- The *payoffs* (costs) for each player in this game can be defined by the C_g function introduced in section 2. Let $\mathcal{A} = (s_{v_1}, s_{v_2}, \ldots, s_{v_n})$ be the strategy profile chosen by the players, *i.e.* slot s_{v_i} is the chosen slot by vehicle v_i, $1 \leq i \leq n$. Let $g(v_i) = s_{v_i}$, then the cost for any player v_i will be $C_g(v_i)$.
- For this game, the penalty of not finding a parking slot, α, will be defined as a constant quantity larger than the sum of all the costs.

4 Nash Equilibrium for PSAG

In this section we introduce the Nash equilibrium for PSAG and establish its relationship with the Stable Marriage problem.

The Nash equilibrium [5] is the standard desired strategy that is used to model the individual choices of players in a game. It defines a situation in which no player can decrease its cost by changing strategy unilaterally. The standard definition of Nash equilibrium translates to the following definition for PSAG:

Definition 1 (Nash Equilibrium for PSAG). *Let $\mathcal{A} = (s_{v_1}, s_{v_2}, \ldots, s_{v_n})$ be a strategy profile for the PSAG. Let $\mathcal{A}_i^* = (s_{v_1}, s_{v_2}, \ldots, s_{v_{i-1}}, s_{v_i}^*, s_{v_{i+1}}, \ldots, s_{v_{n-1}}, s_{v_n})$, for $s_{v_i}^* \neq s_{v_i}$. Let g be the assignment function obtained from strategy profile $\mathcal{A}$ and g_i^* be the assignment function obtained from strategy profile $\mathcal{A}_i^*$. Then strategy profile $\mathcal{A}$ is a Nash equilibrium strategy for the players if $C_g(v_i) \leq C_{g_i^*}(v_i)$ for all i and any $s_{v_i}^* \neq s_{v_i}$.*

$\mathcal{A}_i^*$ is the strategy profile obtained by only player v_i changing strategy from s_{v_i} to any $s_{v_i}^* \neq s_{v_i}$ for any $1 \leq i \leq n$. If the condition in the definition holds then it means that no player can improve by him alone deviating from the Nash equilibrium strategy. For the remainder of the paper, *equilibrium* and *Nash equilibrium* will be used interchangeably.

4.1 Stability of Marriage in PSAG

A vehicle's preference in PSAG is to minimize its cost. Then a vehicle v's preference is to obtain the slot s that minimizes the function $\mathsf{cost}(v, s)$. Then, we say v prefers slot s over slot s' if $\mathsf{cost}(v, s) < \mathsf{cost}(v, s')$. Suppose that the slots had a similar preference order in which a slot s prefers a vehicle v over v' if v is closer to it than v', *i.e.* $\mathsf{dist}(v, s) < \mathsf{dist}(v', s)$.

Definition 2 (Unstable Marriage [2] in PSAG). *An assignment of vehicles to slots is called unstable if there are vehicles v_i and $v_{i'}$, assigned to slots s_j and $s_{j'}$ respectively, but $v_{i'}$ prefers s_j over $s_{j'}$ and s_j prefers $v_{i'}$ over v_i.*

In the following sections, we will show the relationship between the Nash equilibrium for PSAG and stable marriage assignments.

4.2 Computing the Psag Nash Equilibrium

Now we show that we can compute the Nash Equilibrium for Psag by computing stable marriages between the vehicles and the slots.

Theorem 1. *Suppose that the vehicles' preference order is determined by the* cost *function and the slots' preference order is determined by the* dist *function. Then an assignment g is a Nash equilibrium if and only if g is a stable marriage between the vehicles and slots.*

Proof. ($\rightarrow$) Let g be an assignment that is a Nash Equilibrium. Then for any $v \in V$, if v deviates strategy unilaterally from $g(v)$, v's cost will increase.

Suppose to the contrary that g is not a stable marriage between vehicles and slots. Then there exist $v, v' \in V$ and $s, s' \in S$ such that $g(v) = s$ and $g(v') = s'$ but v prefers s' over s and s' prefers v over v'. Then the following inequalities hold:

$$\begin{aligned} \mathsf{cost}(v, s') &< \mathsf{cost}(v, s) \\ \mathsf{dist}(v, s') &< \mathsf{dist}(v', s') \end{aligned}$$

But then if v deviates to strategy s' his cost will improve because v is closer to s' than v', and choosing s' has a lesser cost to him than his current choice s. This violates the Nash equilibrium assumption. Contradiction.

($\leftarrow$) Now let g be an assignment that is a stable marriage between vehicles and slots according to their preferences.

Suppose to the contrary that g is not a Nash equilibrium. Then there exists a vehicle that can deviate from the strategy given by g and improve its obtained cost. Let $v \in V$ be such a vehicle and let $g(v) = s$, where $s \in S$. Then v can choose a strategy $s' \neq s$ and improve its obtained cost. Suppose that slot s' was assigned to vehicle v', *i.e.* $g(v') = s'$.

There are two cases to consider.

Case 1: $C_g(v) = \mathsf{cost}(v, s)$
If $C_g(v) = \mathsf{cost}(v, s)$ then by definition v was the closest vehicle to s amongst those that chose s. Now suppose that v can improve its obtained cost by deviating to another strategy s'. If v improves its obtained cost it means that he will obtain his new chosen slot s', otherwise he would pay a penalty α that is larger than his previous cost. Then,

$$\mathsf{dist}(v, s') < \mathsf{dist}(v', s') \tag{3}$$

If v improves its obtained cost then it also means that his obtained cost on the new slot is better than the one he was paying with his previous slot. Then,

$$\mathsf{cost}(v, s') < \mathsf{cost}(v, s) \tag{4}$$

Condition (3) implies that slot s' preferred v over v'. Condition (4) implies that vehicle v preferred s' over s. These two conditions together are a violation of marriage stability. Therefore, g is not a stable marriage. Contradiction.

Case 2: $C_g(v) = \mathsf{cost}(v, s) + \alpha$
This is a case where $n > m$ and v chooses s but does not obtain it. We assume that for vehicles that will not obtain a slot, the stable marriage algorithm will simply assign the vehicle to its smallest cost slot. Suppose v can deviate to s' and improve its cost.

This means that it would definitely obtain the new slot and improve its cost that way (if the slot is not obtained then there's no way of improving). Then,

$$\mathsf{dist}(v, s') < \mathsf{dist}(v', s') \tag{5}$$

For this case, condition (5) is sufficient to show that g is not a stable marriage. v will not obtain any slot according to assignment g, and by (5), s' prefers v over v'. Then v should have been assigned to s' in the first place by the stable marriage algorithm since v really has no partner. Therefore, g is not a stable marriage. Contradiction.

Then by the contradictions obtained in both cases it follows that no vehicle could have improved by deviating strategies from those defined by the assignment g. Therefore, g is a Nash Equilibrium assignment.

By the equivalency obtained between the Nash equilibrium for Psag and stable assignments in Psag, one can compute an equilibrium by finding a stable assignment between the vehicles and slots. Then we found the equilibrium for this two-sided matching problem between agents (vehicles) and spatially located resources (parking slots) by assigning preference orders (based on distance to agent) to the items.

5 Gravitational Strategies for Incomplete Information Context

5.1 Incomplete Information Psag

We've shown how one can compute the Nash Equilibrium for Psag by computing a stable marriage assignment between the vehicles and the slots. But this equilibrium is applicable only in a complete information setting. This is one where the vehicles are aware of what their payoffs will be based on their decisions and the decisions of others. For Psag, this means that each vehicle is aware of the locations of all the other vehicles and are aware of their cost functions.

This complete information model is hard to justify in practice because of privacy and security concerns. Not all vehicles will be willing to share the location information at all times. Furthermore, tracking the locations of vehicles at all times, and sharing the locations of all of them with all the users of a system so that they can have up-to-the-second location data on all other potential parking competitors seems infeasible.

Then we wish to analyze Psag in an incomplete information context. In this context, the players have no knowledge about the locations of the other players. Since they do not have complete access to the distance function, dist, then they have no way of knowing the payoff function for this game; *i.e.* given a strategy profile, none of the players have a way of knowing what its payoff will be.

In the incomplete information Psag, players make some prior probabilistic assumptions about the locations of the other vehicles in the game and the analysis is performed based on the expectations given by the prior distributions. One can compute the expected costs based on the distribution that is used to denote the location of a vehicle. Then a player will be looking to minimize its expected cost. In this context, the analysis will compute the Nash equilibrium strategies for the players but considering expected costs. This equilibrium is analogous to the Nash equilibrium for Psag (Definition 1) but instead of using cost given by the cost functions (C_g), it uses expected cost.

For this work, each player will assume that other players are distributed uniformly across the map. Unfortunately, computing the equilibrium for this incomplete information context is very complicated in general, even for simple cases in the number line [3]. Then heuristics are needed to compute ideal strategies for players in this more realistic model.

5.2 Gravity for Parking

The heuristic we want to introduce is one that pushes vehicles towards areas where they are most likely to find a parking slot. Since all other vehicles are assumed to be distributed uniformly across space, this will increase the probability of finding a parking slot upon arrival to the area with a larger amount of available slots. Also, we want the algorithm to take into account the vehicle's location and its proximity to the surrounding slots. In [3], we proposed the Gravity-based Parking Algorithm (GPA), which encompasses these desired properties by using vector addition of force vectors.

In the GPA, slots are said to have a gravitational pull on the vehicles. At any point in time, each slot has a gravitational force on the vehicle that will depend on the distance from the vehicle (magnitude) and location of the slot (direction). So then for each slot, a force vector is generated around the vehicle. Then, all of these vectors are added and the vehicle moves in the direction of the resultant vector (total gravitational force) for a specified time step. Then the process is repeated at the beginning of each time interval.

The classical formula for gravitational force is $F = \frac{Gm_1m_2}{d^2}$ where G is the gravitational constant, m_1 and m_2 are the masses of the respective objects and d is the distance between the objects. But for our purposes we can assume that the masses of the objects are constant. We want to compute the vector that represents total gravitational force generated by all the available slots to a vehicle and use the direction of that vector to move the vehicle in that direction. Then we consider a more simplified formula for gravitational force, since all the masses are constant, represented by:

$$F(v, s) = 1/\mathsf{dist}(v, s)^2 \tag{6}$$

$F(v,s)$ is the gravitational force generated by slot s towards vehicle v.

To consider general costs, this formula can be generalized to:

$$F(v,s) = 1/\mathsf{cost}(v,s)^2 \tag{7}$$

With formula (7), one will compute gravitational pull by considering the general cost as the distance between the vehicle and the slot.

5.3 Gravity-based Parking Algorithm (GPA)

Let z denote the velocity of each vehicle (in units/s), which is constant for all vehicles. Each time step for the algorithm will be 1 second. Each vehicle v will perform the following steps in order to move one time-step at a time towards a parking slot:

- Let S' be the set of currently available slots (updated at every time step). Then for each $s \in S'$ generate vector of magnitude $F(v,s)$ that starts at v's location in direction of s.
- Add the computed force vectors and the result will be the total gravitational force generated by all the available slots on v.
- Move z units (velocity) in the direction given by the total force vector. If the closest slot to v is at a distance less than z then move straight to the closest slot.

These steps define the proposed heuristic for vehicles to use in the incomplete information Psag. The intuition behind the algorithm is that a vehicle is better served moving towards areas of higher density of parking slots when the force to closer slots (determined by distance to them) is not strong enough.

Figure 1 shows what a gravitational force field generated by five sample slots would look like. The arrows represent the direction at which a vehicle will move when it is located at the start point of the arrow and the small dots represent the slots. This diagram gives us an idea of how vehicles move across the map when using GPA and it shows that they will eventually converge to a slot. The GPA was evaluated and performed well in simulations against a greedy approach [3].

6 GPA on a Road Network

On a real-world road network, vehicles are constrained to move only on roads. In this setting we will still use a gravitational approach. It will also be based on the gravity equation defined by equation (6), but now the distance between a vehicle and slot is computed by using the travel distance across the network.

A vehicle can only make a routing choice upon arrival to an intersection, whereas before (in free-space) a vehicle could change direction at any point in time. Therefore, the GPA algorithm will only be used at each intersection by each vehicle. The road network is modeled as a graph $G = (N, E)$ where the vertices (N) represent intersections and the edges (E) are the road segments that connect the intersections.

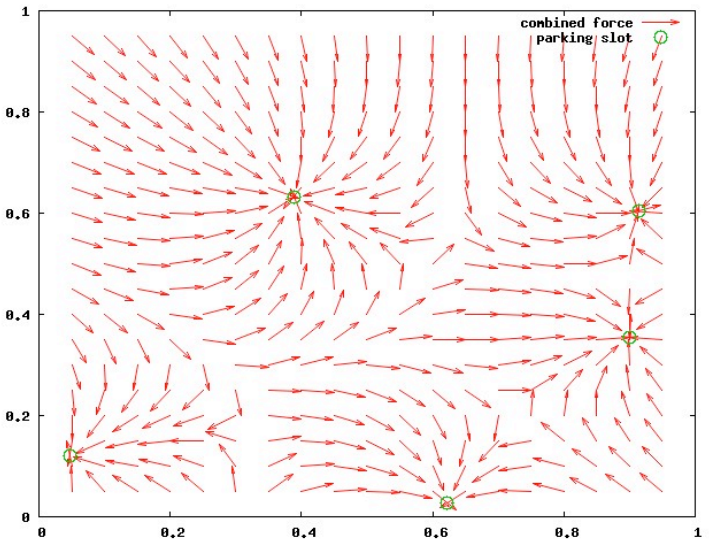


Fig. 1. Force field generated by 5 slots

Instead of adding up all the gravity vectors for all slots (as in Euclidean space), the vehicle will aggregate the gravity information for all slots into special direction vectors (one for each possible direction out of the intersection). Suppose that the intersection where vehicle v is located has k outgoing edges $e_1, e_2, ..., e_k \in E$. Then there will be k direction vectors $g_1, g_2, ..., g_k$ where each vector will have a direction according to its respective embedded edge. The magnitudes of these vectors will start at 0.

Then for each slot s, the shortest path is computed from v to s and the gravity force g is computed using equation (6). Let e_i be the first edge to be taken according to the computed shortest path. Then g_i is updated to be $g_i = g_i + g$.

After repeating this procedure for each slot, the vehicle will use the computed direction vectors $g_1, g_2, ..., g_k$ to make its route choice. From this point, we will introduce two variants of the GPA algorithm that will be evaluated as candidate algorithms for using a gravity-based approach for parking on embedded road networks. The two variants will only differ in how the eventual edge to be taken is computed based on the direction vectors.

6.1 Deterministic Angular GPA (DA-GPA)

In the Deterministic Angular GPA (DA-GPA) the direction vectors $g_1, g_2, ...g_k$ will be added to produce a resultant vector r. This resultant vector will be located between two of the directions to choose from, say $e_i \in E$ and $e_j \in E$. Let θ_i be the angle distance between r and e_i and θ_j be the angle distance between r and

e_j. Then, if $\theta_i < \theta_j$, v will choose e_i as the next edge to travel, otherwise it will choose e_j as the next edge to travel through.

6.2 Randomized Magnitude GPA (RM-GPA)

In the Randomized Magnitude GPA (RM-GPA) the direction vectors $g_1, g_2, ..., g_k$ will be used as part of a probabilistic scheme. Let $T = |g_1| + |g_2| + ... + |g_k|$, *i.e.* the addition of the magnitudes of the k direction vectors. Then let $p_i = |g_i|/T$ for $1 \leq i \leq k$. Then each edge $e_i \in E$ which is an outgoing edge of v's current intersection will be chosen with probability p_i.

6.3 Deterministic Magnitude GPA (DM-GPA)

In the Deterministic Magnitude GPA (DM-GPA) the direction vectors $g_1, g_2, ..., g_k$ will be used to choose the next direction to move towards. The direction with the vector with the largest magnitude will be chosen.

The efficiency of these three methods will be evaluated through simulation.

7 Simulation and Results

In this section we will evaluate DA-GPA, RM-GPA, and DM-GPA against the greedy parking algorithm. The greedy algorithm simply moves each vehicle towards its current closest slot.

7.1 Simulation Environment

The simulation tests the three GPA variants with varying number of values of n and m for the embedded road network in Euclidean space. The simulation is run on a one mile by one mile map where roads are generated that either run from east to west or north to south. Locations for slots on these roads are pre-generated as well.

The map is then partitioned into 16 equal-sized square regions. A random permutation of the regions is generated (uniform distribution) and is used as the ranking of the popularity of each region for available slots. To choose each of the m open slots, first a random number is generated to determine which region to choose a slot from. The Zipf distribution with its skew parameter and the regional popularity previously generated are used to generate this random number. Then a random slot (uniform) is chosen from the region denoted by the Zipf number. The n vehicles' initial positions are generated using the uniform distribution on the grid.

After generating the vehicles and slots, the algorithms are tested. The GPA algorithms were tested against the greedy parking algorithm, which simply moves each vehicle towards its current closest slot. For the GPA algorithms, the vehicles will move as described in the procedures on section 6.

When a vehicle reaches an open parking slot, the time it took for it to find that slot is saved. Then a new slot is chosen randomly (uniform) on a randomly chosen region (Zipf). Also a new vehicle is generated at a random location on the grid (uniform). The simulation run stops when a given time horizon of 3,600 seconds is surpassed.

The parameters for the simulation are:

- n - the number of vehicles.
- m - the number of slots.
- k - the regional skew of the Zipf distribution.

The values that were tested for each parameter are detailed in table 1. For each configuration of the parameters, 20 different simulation runs were generated and tested.

Parameter	**Symbol**	**Range**
Vehicles	n	{40,80}
Slots	m	{20,30,40}
Zipf Skew	k	{0, 1, 2, 3}

Table 1. Parameters tested on Simulation

7.2 Simulation Results

Figure 2 shows the improvement of the DM-GPA algorithm over the greedy parking algorithm. In the best case, the highest improvement that was attained was one of 40% ($n = 40, m = 40, skew = 1$). We can see that the lowest improvement is seen when the skew is 0 (uniformly distributed). Higher improvements are seen in highly skewed situations (skew of 1 or above), although as the regional skew increases past 1, the performance decreases. The RM-GPA and DA-GPA also showed positive improvements over the greedy algorithm but were not better in performance than the DM-GPA. The results for RM-GPA and DA-GPA are thus omitted for space considerations.

8 Conclusions

In this paper our main goal was to analyze vehicular parking. We presented two models that can be used to study the parking problem in a game-theoretic framework.

For the complete information model, in which vehicles are aware of the location and cost information of other players, we presented an algorithm for computing the Nash equilibrium for parking slot assignment games (PSAG). We established the relationship between the parking problem and the stable marriage problem. We also showed that the Nash equilibrium was actually equivalent to a stable marriage between vehicles and slots.

For the incomplete information model, vehicles are not aware of the locations of the other mobile users that are also looking for parking. For this model we presented the Gravity-based Parking Algorithm (GPA). For the adapted GPA to road networks we presented the DA-GPA, RM-GPA and DM-GPA. The merits of the GPA's were tested using simulations.

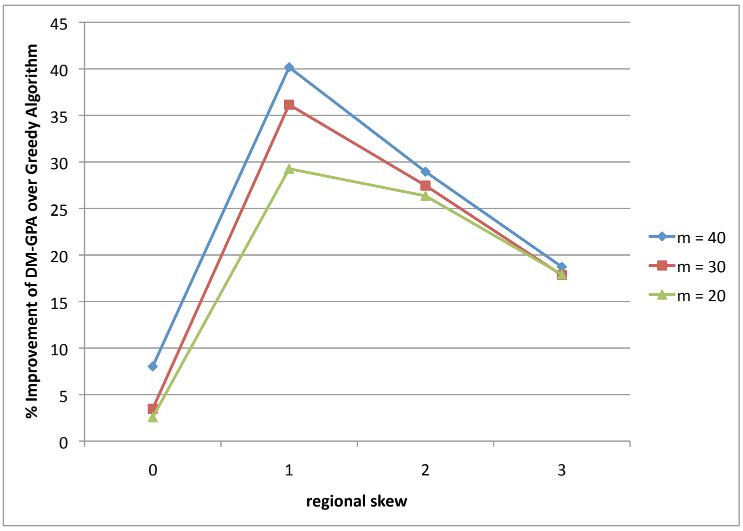


Fig. 2. % improvement of DM-GPA over greedy algorithm ($n = 40$, varying m)

References

1. http://sfpark.org/
2. Gale, D., Shapley, L.: College admissions and the stability of marriage. The American Mathematical Monthly **69**(1) (1962) 9–15
3. Ayala, D., Wolfson, O., Xu, B., Dasgupta, B., Lin, J.: Parking slot assignment games. In: Proc. of the 19th Intl. Conf. on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2011), Chicago, IL (November 2011)
4. Rasmusen, E.: Games and Information. 4th edn. Blackwell Publishing (2006)
5. Nash, J.: Equilibrium points in n-person games. Proceedings of the National Academy of Sciences **36**(1) (1950) 48–49

Testing Substitutability of Weak Preferences

Haris Aziz*, Markus Brill, Paul Harrenstein

Institut für Informatik, Technische Universität München, 85748 Garching bei München, Germany

Abstract

In many-to-many matching models, substitutable preferences constitute the largest domain for which stable matchings are guaranteed to exist. Recently, Hatfield et al. [4] have proposed an efficient algorithm to test substitutability of strict preferences. In this note we show how the algorithm by Hatfield et al. can be adapted in such a way that it can test substitutability of weak preferences as well. When restricted to the domain of strict preferences, our algorithm is faster than Hatfield et al.'s original algorithm by a linear factor.

Keywords: Substitutability, Many-to-Many Matchings, Computational Complexity, and Preference Elicitation.
JEL: C62, C63, and C78

1. Introduction

In matching problems, the aim is to match agents in a stable manner to objects or to other agents while considering the preferences of the agents involved. Matching theory has significant applications in assigning residents to hospitals, students to schools, etc. and has received tremendous attention in mathematical economics, computer science, and operations research [see, e.g., 3, 9].

In various matching models individual preferences are supposed to be *responsive*, i.e., for any two sets that differ only in one object, the agent prefers the set containing the more preferred object [9, page 128f.]. For example in the case in which a hospital can hire multiple doctors, the hospitals are commonly assumed to submit preferences that render the choice between a pair of doctors independent of other available outcomes [4]. An alternative is to allow hospitals to submit substitutable preferences, which allows for considerably more flexibility in expressing preferences over groups of doctors. An agent's preferences are *substitutable* if whenever its most preferred set of objects from a set of objects

*Corresponding author
Email addresses: aziz@in.tum.de (Haris Aziz), brill@in.tum.de (Markus Brill), harrenst@in.tum.de (Paul Harrenstein)

S contains an object w, then so will its most preferred set of objects from any subset of S that still includes w [9, page 173f.].

Substitutable preferences were introduced by Roth [10] and constitute the largest domain in which stable matchings are guaranteed to exist. In many matching models, substitutability is in fact a necessary and sufficient condition for the existence of stable allocations [see 4, footnote 4].[1] The significance of substitutability leads to the natural algorithmic problem of testing whether a given preference relation is substitutable or not. Recently, Hatfield et al. [4] have presented a polynomial-time algorithm for this problem, which they point out in their conclusion *"could be distributed to market participants for use in the preparation of their preference relations for submission."*

As for most results in the literature concerning substitutability, the original definition of substitutability and the algorithm of Hatfield et al. assume that individuals can only express *strict* preferences, i.e., preferences without indifferences. In many settings, however, allowing indifferences is not only a natural relaxation but also a practical necessity. Allowing for indifferences, however, may significantly affect the properties and structure of stable matchings. For example, stable matchings may obtain different cardinalities [7] and for marriage markets man-optimal or woman-optimal stable matchings are no longer guaranteed to exist [9]. Moreover, weak preferences may also be a source of complexity for many computational problems concerning stable matchings. For instance, checking whether a stable roommate matching exists is polynomial-time solvable for strict preferences [6], but becomes NP-complete when indifferences are allowed [8].

For the more general domain which allows individuals to express indifferences (*weak* preferences), Sotomayor [12] extended the concept substitutability and provided an appropriate definition. Moreover, she showed that a stable matching in many-to-many matching markets with weak and substitutable preferences is still guaranteed to exist.

In this note, we examine the notion of substitutability for the general case of weak preferences from a computational point of view. We formulate conditions that characterize substitutable preferences. Using these conditions, we find that testing substitutability of weak preferences can be performed in polynomial time. When restricted to the domain of strict preferences, our algorithm is faster than the algorithm of Hatfield et al. [4] by a linear factor.

2. Preliminaries

Let U be a finite set of alternatives. A *weak preference relation* is a transitive and complete relation R on 2^U. R is said to be *strict* if it is also anti-symmetric. Let P and I denote the strict and symmetric parts of R, respectively. A set $X \subseteq U$ is called *acceptable* if $X \; R \; \emptyset$. By an *equivalence class of* R we understand

[1] These settings include many-to-one matchings, many-to-many matchings, and many-to-many matchings with contracts.

a family $\{Y \in U : X\ I\ Y\}$ for some subset X in U. Each preference relation R induces a choice function C that returns, for each $X \subseteq U$, the set of all R-maximal subsets of X, i.e.,

$$C(X) = \{Y \subseteq X : Y\ R\ Z \text{ for all } Z \subseteq X\}.$$

Observe that, defined thus, $C(X)$ invariably contains a single set if R is strict but may contain more than one set if R allows for indifferences. Even if $C(X)$ may contain the empty set, $C(X)$ itself is never empty.

Example 1. *Let $U = \{a, b, c, d\}$ and let the preference relation R restricted to the acceptable set be defined as follows.*

$$\{a, b, d\}\ I\ \{b, c, d\}\ P\ \{a, b\}\ I\ \{b, c\}\ I\ \{a, c\}\ P\ \emptyset.$$

Then, $C(U) = \{\{a, b, d\}, \{b, c, d\}\}$ and $C(\{a\}) = \{\emptyset\}$.

Given a preference relation R, let s denote the maximal size of an indifference class consisting of acceptable sets. Observe that for each $X \subseteq U$ the size of $C(X)$ is bounded by s and that a preference relation is strict if and only if $s = 1$. Furthermore, let u denote the size of U and ℓ the number of acceptable sets.

A very general and expressive way of representing R is via a preference list L that contains all acceptable sets in descending order of preferability and using brackets to group sets in the same equivalence class.[2] For a preference relation R represented in list form and for $X \subseteq U$ it can be checked in time $O(\ell|X|)$ whether a given alternative is in $C(X)$.

3. Substitutability and weak preferences

In the restricted setting of strict preferences the choice function invariably chooses (a family consisting of) a single set. For weak preferences the choice function may select a family of any number of sets and the definition of substitutability for strict preferences has to be adapted accordingly. Sotomayor [12] observed that substitutability for strict preferences can be characterized in two equivalent ways.[3] Both of these definitions can naturally and conservatively be extended to the domain of weak preferences. In this more general setting, however, the conditions these generalizations give rise to (S1 and S2 in Definition 1 below) are no longer equivalent. Sotomayor has argued that both conditions capture an essential aspect of substitutability and suggested that for weak preferences substitutability be defined as their conjunction.

[2] This list representation is reminiscent to the *representation by individually rational lists of coalitions* used in the context of hedonic coalition formation games [1].

[3] Substitutability for strict preferences is identical to Sen's condition α used in choice theory [11] but the meanings of the choice sets are different [see 5, footnote 4]. Also from the perspective of choice theory, Brandt and Harrenstein [2] considered choice functions that are rationalized by relations over sets of alternatives, which is formally similar to the setting considered here.

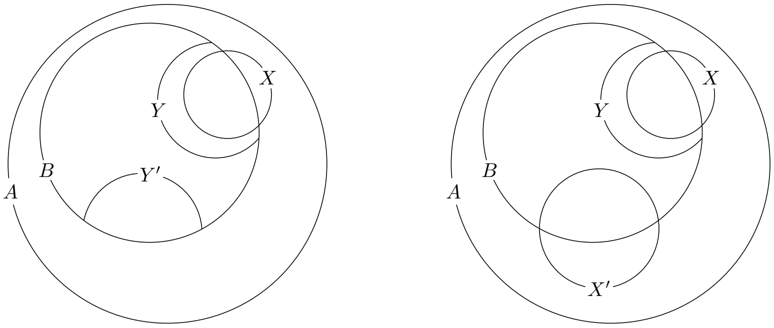


Figure 1: Conditions S1 and S2. For the diagram on the left, we assume that $C(A) = \{X\}$ and $C(B) = \{Y, Y'\}$. Then, S1 is satisfied but S2 is not. In the diagram on right, we assume that $C(A) = \{X, X'\}$ and $C(B) = \{Y\}$. Then, S2 is satisfied but S1 is not.

Definition 1. *A preference relation R is* substitutable *if and only if the following two conditions hold:*

(*S1*) *for all non-empty $A, B \subseteq U$ with $B \subseteq A$ we have that for all $X \in C(A)$ there is some $Y \in C(B)$ such that $X \cap B \subseteq Y$, and*

(*S2*) *for all non-empty $A, B \subseteq U$ with $B \subseteq A$ we have that for all $Y \in C(B)$ there is some $X \in C(A)$ such that $X \cap B \subseteq Y$.*

Example 2. *Consider the preference relation R from Example 1. It can be verified that R satisfies S1 and violates S2. For the latter, take $A = U$ and $B = \{a, b, c\}$. Then,*

$$C(B) = \{\{a, b\}, \{b, c\}, \{a, c\}\}.$$

Now $Y = \{a, c\}$ is in $C(B)$, but there exists no $X \in C(A)$ such that $X \cap B \subseteq Y$. Hence, R is not substitutable.

The following lemma captures the intuitive idea that a most-preferred subset will remain most-preferred and that no subsets will become most-preferred that were not previously so when some other subsets are withdrawn from consideration without other subsets being added [cf. 4, Lemma 1].

Lemma 1. *For all $A, B \subseteq U$ with $B \subseteq A$,*

$$C(A) \cap 2^B \neq \emptyset \textit{ implies } C(B) = C(A) \cap 2^B.$$

Proof. Assume $C(A) \cap 2^B \neq \emptyset$. Then, $X \in C(A) \cap 2^B$ for some $X \subseteq U$. First consider an arbitrary $Y \in C(B)$. Then $Y \; R \; X$. Hence, $Y \in C(A) \cap 2^B$ as well. Now consider an arbitrary $Y \notin C(B)$. If $Y \notin 2^B$, immediately $Y \notin C(A) \cap 2^B$. If $Y \in 2^B$, we have $X \; P \; Y$ and therefore $Y \notin C(A)$. Also then $Y \notin C(A) \cap 2^B$. □

4. Testing substitutability

We now outline a way to test substitutability of weak preferences. The idea utilizes an insight of Hatfield et al. [4] that instead of checking all violations of substitutability, one may restrict one's attention to violations of a specific type. Formally, by an *S1-violation for R* we understand a pair $(A, B) \in 2^U \times 2^U$ such that $B \subseteq A$ and for some $X \in C(A)$ it is the case that $X \cap B \nsubseteq Z$ for all $Z \in C(B)$. Obviously, a preference relation R satisfies S1 if and only if there are no S1-violations for R.

Lemma 2. *Let R be a preference relation. If there exists an S1-violation for R, then there exist acceptable sets $X, Y \subseteq U$ and $x \in X$ such that $(X \cup Y, Y \cup \{x\})$ is also an S1-violation for R.*

Proof. Assume that (A, B) is an S1-violation for R. Then there is some $X \in C(A)$ such that $X \cap B \nsubseteq Z$ for all $Z \in C(B)$. As $C(B) \neq \emptyset$, there is some $Y \in C(B)$ such that $X \cap Y$ is maximal with respect to set inclusion, i.e., $X \cap Y \subsetneq X \cap Z$ for no $Z \in C(B)$. Obviously, X and Y are acceptable. By our assumption, $X \cap B \nsubseteq Y$ and we may therefore assume the existence of some $x \in (X \cap B) \setminus Y$ (see Figure 2). We prove that $(X \cup Y, Y \cup \{x\})$ is an S1-violation for R, in particular we show that

(*i*) $Y \cup \{x\} \subseteq X \cup Y$,

(*ii*) $X \in C(X \cup Y)$, and

(*iii*) $X \cap (Y \cup \{x\}) \nsubseteq Z$ for all $Z \in C(Y \cup \{x\})$.

As $x \in X$, it is obvious that (*i*) holds. As for (*ii*), observe that $X \in C(A) \cap 2^{X \cup Y}$. Lemma 1 implies $C(X \cup Y) = C(A) \cap 2^{X \cup Y}$ and thus $X \in C(X \cup Y)$.

Finally, consider an arbitrary $Z \in C(Y \cup \{x\})$. Observe that $Y \in C(B) \cap 2^{Y \cup \{x\}}$. By another application of Lemma 1, we get $C(Y \cup \{x\}) = C(B) \cap 2^{Y \cup \{x\}}$ and, therefore, $Z \in C(B)$. Moreover, by choice of Y, it is not the case that $X \cap Y \subsetneq X \cap Z$, i.e., either $X \cap Y \nsubseteq X \cap Z$ or $X \cap Y = X \cap Z$. If the former, then there is some $z \in X \cap Y$ with $z \notin X \cap Z$. If the latter, then $x \notin Z$. In either case, there is some $z \in X \cap (Y \cup \{x\})$ such that $z \notin Z$. Hence, $X \cap (Y \cup \{x\}) \nsubseteq Z$, which proves (*iii*). □

S2-violations are defined similar to S1-violations. A pair $(A, B) \in 2^U \times 2^U$ is an *S2-violation for R* if $B \subseteq A$ and for some $Y \in C(B)$ it is the case that $Z \cap B \nsubseteq Y$ for all $Z \in C(A)$. Clearly, R satisfies S2 if and only if there are no S2-violations for R. Moreover, R is substitutable if and only if there are neither S1-violations nor S2-violations for R.

Lemma 3. *Let R be a preference relation. If there exists an S2-violation for R, then there exist acceptable sets $X, Y \subseteq U$ and $x \in X$ such that $(X \cup Y, Y \cup \{x\})$ is also an S2-violation for R.*

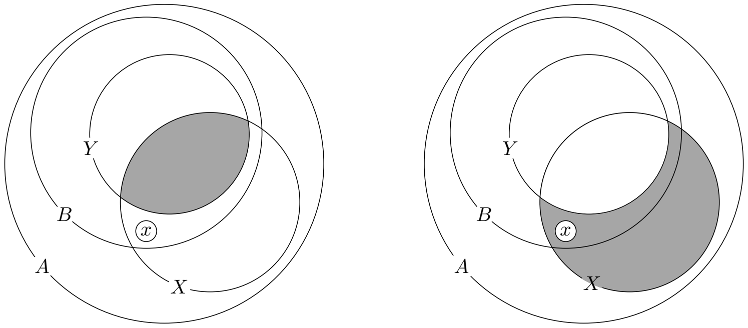


Figure 2: The left diagram illustrates Lemma 2, the right one Lemma 3. In either case, Y is chosen so as to maximize and minimize the areas in gray, respectively.

Proof. Assume that (A, B) is an S2-violation for R. Then there is some $Y \in C(B)$ such that $Z \cap B \nsubseteq Y$ for all $Z \in C(A)$. As $C(A) \neq \emptyset$, there is some $X \in C(A)$ such that $X \setminus Y$ is minimal with respect to set-inclusion, i.e., $Z \setminus Y \subsetneq X \setminus Y$ for no $Z \in C(A)$. Obviously, X and Y are acceptable. By our assumption, $X \cap B \nsubseteq Y$ and we may assume the existence of some $x \in (X \cap B) \setminus Y$ (see Figure 2). We prove that $(X \cup Y, Y \cup \{x\})$ is also an S2-violation for R, in particular we show that

(*i*) $Y \cup \{x\} \subseteq X \cup Y$,

(*ii*) $Y \in C(Y \cup \{x\})$, and

(*iii*) $Z \cap (Y \cup \{x\}) \nsubseteq Y$ for all $Z \in C(X \cup Y)$.

As $x \in X$, (*i*) obviously holds. As for (*ii*), observe that $Y \in C(B) \cap 2^{Y \cup \{x\}}$. Lemma 1 implies that $C(Y \cup \{x\}) = C(B) \cap 2^{Y \cup \{x\}}$ and thus $Y \in C(Y \cup \{x\})$.

Finally, consider an arbitrary $Z \in C(X \cup Y)$. Observe that $X \in C(A) \cap 2^{X \cup Y}$. Another application of Lemma 1 yields $C(X \cup Y) = C(A) \cap 2^{X \cup Y}$ and, therefore, $Z \in C(A) \cap 2^{X \cup Y}$. Moreover, by choice of X, it is not the case that $Z \setminus Y \subsetneq X \setminus Y$. Observe that $Z \setminus Y \subseteq X \setminus Y$ and it follows that $Z \setminus Y = X \setminus Y$. Hence, $x \in Z$ and, since $x \notin Y$, we obtain $Z \cap (Y \cup \{x\}) \nsubseteq Y$, which proves (*iii*). □

We can exploit Lemmas 2 and 3 to obtain a polynomial-time algorithm to check the substitutability of a preference relation. The algorithm works as follows. Instead of checking all potential violations of S1 and S2, due to Lemmas 2 and 3 we can restrict our attention to S1- and S2-violations of the form $(X \cup Y, Y \cup \{x\})$, where $X, Y \subseteq U$ are acceptable and $x \in U$. The number of these potential violations is polynomial in the number of acceptable subsets in R and a polynomial-time algorithm is obtained by exhaustively checking each of them. We note that the algorithm is not different from that of Hatfield et al. [4] in that it exhaustively checks certain violations.

Theorem 1. *It can be checked in time $O(\ell^2 u^2(\ell + s^2))$ whether a given preference relation in list representation is substitutable.*

Proof. To test substitutability, we need to check whether both S1 and S2 hold for a preference relation R represented by list L. This is equivalent to verifying that neither S1-violations nor an S2-violations exist for R.

Let us first consider the case of S1. To check S1, we know from Lemma 2, that we can restrict our attention to violations of the form $(X \cup Y, Y \cup \{x\})$ for some $X, Y \in L$ and $x \in X$. Therefore, the maximum number of pairs we need to check is upper-bounded by $\binom{\ell}{2}u$.

Verifying an S1-violation of type $(A, B) = (X \cup Y, Y \cup \{x\})$ requires us to perform the following three steps. First, compute $C(A)$. This takes time $O(\ell u)$. Then, compute $C(B)$, which also takes time $O(\ell u)$. Finally, test the main condition:

$$\text{for all } X \in C(A) \text{ there is some } Y \in C(B) \text{ such that } X \cap B \subseteq Y.$$

This can be performed in time $O(s^2 u)$. In total, verifying a violation of type $(A, B) = (X \cup Y, Y \cup \{x\})$ takes time

$$O(\ell u) + O(\ell u) + O(s^2 u) = O(\ell u + s^2 u).$$

The time needed to check whether an S1-violation exists is then equal to the maximum number of pairs we need to check multiplied by the time required to verify one S1-violation, which equals

$$O\big(\tbinom{\ell}{2}u\big) \times O(\ell u + s^2 u) = O(\ell^2 u(\ell u + s^2 u)).$$

The same analysis holds for checking whether an S2-violation exists. Therefore there exists an algorithm which runs in time $O(2\ell^2 u(\ell u + s^2 u)) = O(\ell^2 u(\ell u + s^2 u)) = O(\ell^2 u^2(\ell + s^2))$ and tests the substitutability of a preference relation. □

By letting $s = 1$, we get the following result for strict preferences as a corollary.

Corollary 1. *It can be checked in time $O(\ell^3 u^2)$ whether a given* strict *preference relation is substitutable.*

On the domain of strict preferences, the (worst case) asymptotic running time of the algorithm turns out to be slightly faster than the algorithm of Hatfield et al. [4]: $O(\ell^3 u^2)$ as compared to $O(\ell^3 u^3)$. The reason for this is that Hatfield et al. considered violations the form $(x, y, (X \cup Y) \setminus \{x, y\})$, involving two alternatives and two acceptable sets, whereas in this paper we found that one can restrict attention to S1- and S2-violations of the form $(X \cup Y, Y \cup \{x\})$, which involve one alternative less. In practice one might have some expectation that s would be a polynomial function of ℓ. In that case, we could even obtain a polynomial bound entirely in terms of ℓ and u.

Acknowledgements

This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/9-1 and BR 2312/10-1.

References

[1] Ballester, C., 2004. NP-completeness in hedonic games. Games and Economic Behavior 49 (1), 1–30.

[2] Brandt, F., Harrenstein, P., 2011. Set-rationalizable choice and self-stability. Journal of Economic Theory 146 (4), 1721–1731.

[3] Gusfield, D., Irving, R. W., 1989. The stable marriage problem: structure and algorithms. MIT Press, Cambridge, MA, USA.

[4] Hatfield, J. W., Immorlica, N., Kominers, S. D., 2011. Testing substitutability. Games and Economic Behavior.

[5] Hatfield, J. W., Milgrom, P. R., 2005. Matching with contracts. American Economic Review 95 (4), 913–935.

[6] Irving, R. W., 1985. An efficient algorithm for the "stable roommates" problem. Journal of Algorithms 6 (4), 577–595.

[7] Manlove, D., Irving, R. W., Iwama, K., Miyazaki, S., Morita, Y., 2002. Hard variants of stable marriage. Theoretical Computer Science 276 (1–2), 261–279.

[8] Ronn, E., 1990. NP-complete stable matching problems. Journal of Algorithms 11, 285–304.

[9] Roth, A., Sotomayor, M. A. O., 1990. Two-Sided Matching: A Study in Game Theoretic Modelling and Analysis. Cambridge University Press.

[10] Roth, A. E., 1984. Stability and polarization of interests in job matching. Econometrica 52, 47–57.

[11] Sen, A. K., 1970. Collective Choice and Social Welfare. North-Holland.

[12] Sotomayor, M. A. O., 1999. Three remarks on the many-to-many stable matching problem. Mathematical Social Sciences 38 (1), 55–70.

On rank-profiles of stable matchings [*]

E. Boros[†] L. Fedzhora[‡] V. Gurvich [†] S. Jaslar [§]

May 19, 2012

Abstract

We study the quality of stable matchings from the individuals' viewpoint. To each matching we associate its *rank-profile* describing the individuals' satisfaction with the matching. We provide a complete and computationally efficient characterization of the rank-profiles that can arise from men-optimal, women-optimal, and arbitrary stable matchings. We also study *uniquely stable* rank-profiles, that is, for which there exists a stable matching problem that has only one stable matching and this matching has this particular rank-profile. We give some necessary and some sufficient conditions for unique stability and show that characterizations of men-optimal and women-optimal rank-profiles is reduced to the characterization of uniquely stable rank-profiles. Our characterizations imply that the set of all stable rank-profiles is monotone, unlike the sets of men-optimal, women-optimal, and uniquely stable rank-profiles. We also show that both stable and uniquely stable matchings may be highly disadvantageous for all participating individuals, simultaneously. Namely, we show that there are stable and even uniquely stable rank-profiles in which no individual gets a better partner than his/her middle choice. Furthermore, this result is sharp, since a stable matching in which all individuals get a partner ranked below their middle choice cannot be stable. Finally, we demonstrate an "instability of stable matchings" from a quality point of view.

Key Words: stable matching, preference list, rank-profile, Gale-Shapley algorithm and theorem.

1 Introduction

In this paper we consider the *stable matching problem* in which members of two groups of individuals (*men* and *women*) have strict rankings of the individuals of the other group. A *matching* is called *stable* if no men-women couple prefer one another more than their respective partners in the matching.

The celebrated result by [10] shows that a stable matching always exists, regardless of the individuals' rankings, and can be found algorithmically efficiently. Furthermore, there are unique group-optimal stable matchings (men-optimal and women-optimal), in which all members of the group simultaneously are paired with their "best possible" partners, in the sense that no individual gets a more preferred partner in any other stable matching. These properties, the equilibrium nature of stable matchings, the coincidence of individual and group optimality, and the algorithmic tractability make stable matching based models very attractive in social sciences and economics.

[*] The first author was supported in part by the National Science Foundation (IIS-0803444 and CMMI-0856663).

[†] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; ({boros,gurvich}@rutcor.rutgers.edu).

[‡] Virginia Department of Transportation, 1401 E Broad Street, Richmond, VA 23219; (liliya.fedzhora@vdot.virginia.gov)

[§] FBI, Engineering Research Facility Building 27958A Quantico, VA 22135; (steven.jaslar@ic.fbi.gov)

In fact, a very large number of recent publications utilize stable matchings for various applications; see, for example, [1, 2, 3, 4, 5, 6, 7, 9, 13, 14, 20, 18, 17, 16, 19].

Naturally, the quality and fairness of stable matchings based models received a lot of attention. A natural parameter to consider in such investigations is the rank of the matched partner in the individual's preference list. For instance, [12] considered the so called *egalitarian* stable matchings in which the sum of the ranks of the partners of all individuals in a stable matching is minimized, and showed that such an egalitarian stable matching can also be obtained in polynomial time. [15] solved the so called *minimum regret* stable matching problem, in which the maximum rank of any partner (or in other words, the unhappiness of the least satisfied individual) is minimized.

Most of the studies in the literature focus on optimization and algorithmic issues, arising in the course of finding a specific, extremal (optimal, according to some objective) stable matching *for a given preference list.* It is however equally interesting to understand how "bad" or "good" stable matching solutions could be *over all possible preference lists* (in a worst case sense, even if we optimize for each particular instance). More precisely, we are interested in the individuals' satisfaction, in the worst scenario, that is, we consider the maximum happiness of the happiest individual in a stable matching, when the maximization takes place over all stable matchings of a given instance, and ask how unhappy such a "most happy" individual can be? For instance, is it possible that in a men-optimal stable matching no man is coupled with his most preferred woman partner? We show that unfortunately this is quite possible, and that even more so, there are infinitely many instances when even the happiest individual in the community is pretty-pretty unhappy.

More generally, we associate a so called rank-profile to each matching, describing the level of satisfaction of all individuals with their partners within the given matching, and provide a computationally efficient characterization of the rank-profiles that can arise from the men-optimal, women-optimal, and arbitrary stable matchings. We also show that characterizing the men-optimal and women-optimal stable rank-profiles is equivalent, in some sense, with characterizing rank-profiles that arise from the stable matchings instances that have a unique stable matching. We obtain several interesting properties of such rank-profiles.

Finally, we demonstrate "an instability of the stable matchings" by exhibiting instances for which the addition of just one more couple can "spoil the life of a happy community".

In this short extended abstract we could not include all of the above mentioned results and their proofs, and refer the reader for the full version to [8].

2 Main Definitions and Results

We consider stable matching problems involving n men and n women, where n is a given positive integer. Let $\mathbf{N} = \{1, \ldots, n\}$ denote the set of indices, while $\mathbf{M} = \{m_1, \ldots, m_n\}$ and $\mathbf{W} = \{w_1, \ldots, w_n\}$ denote the sets of men and women, respectively. A stable matching problem is described by the set of individual preferences, that is, by a set of $2n$ mappings:

$$r_w : \mathbf{M} \to \mathbf{N} \text{ for } w \in \mathbf{W} \quad \text{ and } \quad r_m : \mathbf{W} \to \mathbf{N} \text{ for } m \in \mathbf{M}. \tag{1}$$

For example, we say that man $m \in \mathbf{M}$ is of *rank* $j \in \mathbf{N}$ for woman $w \in \mathbf{W}$ if $r_w(m) = j$ and that woman $w \in \mathbf{W}$ *prefers* man $m \in \mathbf{M}$ to man $m' \in \mathbf{M}$ if $r_w(m) < r_w(m')$. We consider the most traditional model, in which the preference lists contain no ties, that is, all $2n$ mappings r_w, $w \in \mathbf{W}$ and r_m, $m \in \mathbf{M}$ are bijections. In what follows, a stable matching problem is given by a *preference list* $\sigma = \{r_w^\sigma \mid w \in \mathbf{W}\} \cup \{r_m^\sigma \mid m \in \mathbf{M}\}$ consisting of $2n$ bijections, as in (1), and we shall denote by Σ_n the set of all possible preference lists of men $\mathbf{M}$ and women $\mathbf{W}$. When it is unambiguous, which preference list is meant, we will omit the upper index σ and refer simply by r_w and r_m to the corresponding preferences.

A set $\pi \subseteq \mathbf{M} \times \mathbf{W}$ is called a *matching (or pairing)*, if for each man $m \in \mathbf{M}$ there is a unique woman $w \in \mathbf{W}$ such that $(m, w) \in \mu$ and for each woman $w \in \mathbf{W}$ there is a unique man $m \in \mathbf{M}$ such that $(m, w) \in \pi$). If $(m, w) \in \pi$, we say that w is m's *partner*, and m is w's *partner* in the matching π.

Given a preference list $\sigma \in \Sigma_n$, and a matching $\pi \subseteq \mathbf{M} \times \mathbf{W}$, we say that a pair $(m, w) \in \mathbf{M} \times \mathbf{W}$ is a *breaking couple* for π, if m and w mutually prefer each other to their current partners in π, that is, if $r^\sigma_w(m) < r^\sigma_w(m')$ and $r^\sigma_m(w) < r^\sigma_m(w')$ for $m' \in \mathbf{M}$ and $w' \in \mathbf{W}$ for which $(m, w') \in \pi$ and $(m', w) \in \pi$.

A matching π is called *stable*, if there is no breaking couple for it. The celebrated result of [10] claims that every preference list has a stable matching. (In fact, it may have many, and those form a lattice; see Section 3 for more details). Let us denote by $\Pi(\sigma)$ the set of all stable matchings of a preference list $\sigma \in \Sigma_n$.

Given a preference list $\sigma \in \Sigma_n$ and a matching $\pi \subseteq \mathbf{M} \times \mathbf{W}$, let us associate to them two vectors $k(\pi, \sigma) \in \mathbf{N}^{\mathbf{M}}$ and $\ell(\pi, \sigma) \in \mathbf{N}^{\mathbf{W}}$ defined for $m \in \mathbf{M}$ and $w \in \mathbf{W}$ by

$$\begin{aligned} k_m(\pi, \sigma) &= j \quad \text{if } r_m(w) = j \text{ for the woman } w \in \mathbf{W} \text{ for which } (m, w) \in \pi, \\ \ell_w(\pi, \sigma) &= j \quad \text{if } r_w(m) = j \text{ for the man } m \in \mathbf{M} \text{ for which } (m, w) \in \pi. \end{aligned}$$

Let us call the pair of vectors $(k(\pi, \sigma), \ell(\pi, \sigma))$ the *rank-profile* of the matching π with respect to the preference list σ. Let us note that these two vectors themselves carry all necessary information about the level of satisfaction of the individuals in $\mathbf{M} \cup \mathbf{W}$ with respect to the matching π, even though these vectors alone do not carry enough information to determine π or σ. To simplify notation, we refer to $k(\pi, \sigma)$ and $\ell(\pi, \sigma)$ as $k(\pi)$ and $\ell(\pi)$ whenever σ is clearly defined by the context, as $k(\sigma)$ and $\ell(\sigma)$ whenever π is clearly defined by the context, and simply by (k, ℓ) whenever both π and σ are clearly defined.

Furthermore, whenever π is not explicitly defined, we assume that $\pi = \{(m_1, w_1), (m_2, w_2), ..., (m_n, w_n)\}$, since for any matching we can relabel the women to obtain π in this form. Moreover, we will write simply k_i and ℓ_j instead of k_{m_i} and ℓ_{w_j}, and consequently, we consider rank-profiles as pairs of vectors $k, \ell \in \mathbf{N}^n$.

Let us denote by $\mathcal{R}_n$ the set of all rank-profiles, that is, the set of all pairs of vectors (k, ℓ), where $k \in \mathbf{N}^{\mathbf{M}}$ and $\ell \in \mathbf{N}^{\mathbf{W}}$. Let us note that not all rank-profiles correspond to stable matchings. For instance, $k = (2, 2)$ and $\ell = (2, 2)$, for $n = 2$, cannot, that is, no instance $\sigma \in \Sigma_2$ has a stable matching $\pi \in \Pi(\sigma)$ such that $k = k(\pi, \sigma)$ and $\ell = \ell(\pi, \sigma)$. This is simply because no matter how we choose a matching $\pi \in \mathbf{M} \times \mathbf{W}$ and a preference list $\sigma \in \Sigma_2$, any pair $(m_i, w_j) \in (\mathbf{M} \times \mathbf{W}) \setminus \pi$ must be a breaking couple for π in σ.

In this paper we will characterize the rank-profiles of stable matchings. For a given preference list $\sigma \in \Sigma_n$ let us denote by $\mathcal{S}(\sigma)$ the set of rank-profiles of all stable matchings of σ,

$$\mathcal{S}(\sigma) = \{ (k(\pi, \sigma), \ell(\pi, \sigma)) \mid \pi \in \Pi(\sigma) \},$$

and let $\mathcal{S}_n$ denote the set of all *stable rank-profiles*, that is, $\mathcal{S}_n = \bigcup_{\sigma \in \Sigma_n} \mathcal{S}(\sigma)$.

Our first result characterizes stable rank-profiles.

Theorem 1 $(k, \ell) \in \mathcal{S}_n$ *if and only if*

$$\sum_{i \in I} k_i + \sum_{j \in J} \ell_j \leq n|I| + n|J| + |I \cap J| - |I||J| \tag{2}$$

holds for all subsets $I, J \subseteq \mathbf{N}$. Furthermore, the membership $(k, \ell) \in \mathcal{S}_n$ can be tested in $O(n^5)$ time, and if $(k, \ell) \in \mathcal{S}_n$, then we can construct in the same time an instance $\sigma \in \Sigma_n$ for which $k = k(\sigma)$ and $\ell = \ell(\sigma)$.

Consider first a few small examples. For instance, for $n = 3$ the rank-profile $((1,1,2),(3,3,1))$ is not stable, since inequality (2) fails for the subsets $M = \{3\}, W = \{1,2\}$, because $2 + (3+3) = 8 > 7 = 3(1+2) - (1 \times 2 - 0)$. Furthermore, $((1,2,2),(2,3,3)) \notin \mathcal{S}_3$, because for the subsets $M = \{2,3\}$, and $W = \{1,2,3\}$ we get $(2+2) + (2+3+3) = 12 > 11 = 3(2+3) - (2 \times 3 - 2)$. Similarly, $((1,2,3),(2,1,3)) \notin \mathcal{S}_3$, since $(2+3)+(2+3) = 10 > 9 = 3(2+2) - (2 \times 2 - 1)$ for $M = \{2,3\}$, and $W = \{1,3\}$.

Let us consider now several corollaries of Theorem 1. Two special cases of Theorem 1, corresponding to $|I| = |J| = 1$ and $|I| = |J| = n$, can be reformulated as follows.

Corollary 1 *If $(k, \ell) \in \mathcal{S}_n$ then $(k_i, \ell_j) \neq (n, n)$ whenever $i \neq j$.*

Proof. Apply (2) with $I = \{i\}$ and $J = \{j\}$. Then the left hand side of (2) is $2n$, while the right hand side is only $2n - 1$. □

For example, rank-profiles $((1,2),(2,1))$ for $n = 2$ and $((1,1,3),(1,3,1))$ for $n = 3$ are not stable according to the above Corollary 1.

Corollary 2 *If $(k, \ell) \in \mathcal{S}_n$ then $\sum_{i \in \mathbf{N}} k_i + \sum_{j \in \mathbf{N}} \ell_j \leq n(n+1)$.*

Proof. Apply (2) with $I = \mathbf{N}$ and $J = \mathbf{N}$. □

For example, the following rank-profiles are not stable:

$((1,2),(2,2))$, $((1,2,3),\ (2,2,3))$, $((2,2,2),\ (2,2,3))$.

We shall derive two more consequences of Theorem 1. Let us write $f \leq g$ for two vectors f and g, if those are labeled by the same domain, and the inequality holds componentwise.

Corollary 3 *Set $\mathcal{S}_n$ is anti-monotone with respect to the relation $\leq$ applied between rank-profiles. In other words, if $(k, \ell) \in \mathcal{S}_n$ and $(k', \ell') \in \mathcal{R}_n$ such that $k' \leq k$ and $\ell' \leq \ell$ hold, then $(k', \ell') \in \mathcal{S}_n$.*

Proof. Conditions (2) are clearly monotone in k and ℓ. □

Corollary 4 *If n is an odd integer and $k_i = \ell_j = \frac{n+1}{2}$ for all $i, j \in \mathbf{N}$ then $(k, \ell) \in \mathcal{S}_n$.*

Proof. We need to derive from Theorem 1 that for arbitrary subsets $I, J \subseteq \mathbf{N}$ we have

$$\frac{n+1}{2}(|I| + |J|) \leq n|I| + n|J| + |I \cap J| - |I||J|.$$

Since $|I \cap J| - |I||J| \leq |I||J| - (|I| + |J|) + n$, it is enough to show that

$$\frac{n+1}{2}(|I| + |J|) \leq n(|I| + |J|) - |I||J| + |I| + |J| - n,$$

which follows readily by $|I| \leq n$ and $|J| \leq n$. □

Corollary 4 implies, somewhat surprisingly, that there are instances $\sigma \in \Sigma_n$ and stable matchings $\pi \in \Pi(\sigma)$ for which all individuals are pretty unhappy. Of course, for some other stable matchings of the same instance, some individuals may be much happier. Let us return to the problem of measuring individuals' satisfaction. As one of the simplest measures of a given preference list $\sigma \in \Sigma_n$ and a stable matching $\pi \in \Pi(\sigma)$, let us introduce

$$h(\pi, \sigma) = \min\left\{\min_i k_i(\pi, \sigma),\ \ \min_j \ell_j(\pi, \sigma)\right\} \quad \text{and} \quad h(\sigma) = \min_{\pi \in \Pi(\sigma)} h(\pi, \sigma), \tag{3}$$

that is, the "happiness" of the "happiest" individual in his/her "luckiest" stable matching of σ. Clearly, if $h(\sigma) = 1$, then there is a "very happy" individual, who gets his/her best choice in some of the stable matchings of σ.

It was observed by Conway (see, for example, [15]) that for every preference list $\sigma \in \Sigma_n$ the set $\Pi(\sigma)$ forms a lattice, that is, if $\alpha, \beta \in \Pi(\sigma)$ are two arbitrary stable matchings, then there exists $\gamma, \delta \in \Pi(\sigma)$ such that

$$k_i(\gamma) = \max\{k_i(\alpha), k_i(\beta)\} \text{ and } \ell_j(\gamma) = \min\{\ell_j(\alpha), \ell_j(\beta)\};$$

$$k_i(\delta) = \min\{k_i(\alpha), k_i(\beta) \quad \text{and} \quad \ell_j(\delta) = \max\{\ell_j(\alpha), \ell_j(\beta)\}$$

for all $i, j \in \mathbf{N}$. In particular, for every preference list $\sigma \in \Sigma_n$ there exists unique *men-optimal* and *women-optimal* matchings $\pi^M, \pi^W \in \Pi(\sigma)$, for which we have

$$k(\pi^M) \geq k(\pi) \geq k(\pi^W) \quad \text{and} \quad \ell(\pi^W) \geq \ell(\pi) \geq \ell(\pi^M) \tag{4}$$

for all stable matchings $\pi \in \Pi(\sigma)$. In fact the algorithm by [10] can be executed in two natural ways, the so called men-oriented and women-oriented way, and these produce the unique matchings π^M and π^W. Thus, both of these extremal stable matchings can be obtained in $O(n^2)$ time, implying that $h(\sigma)$ can in fact be computed for every instance σ in $O(n^2)$ time, because by (3) we have

$$h(\sigma) = \min\left\{\min_i k_i(\pi^M, \sigma),\ \ \min_j \ell_j(\pi^W, \sigma)\right\}. \tag{5}$$

Of course, $h(\sigma)$ is a very weak measure of happiness of the community, since there might be many very unhappy individuals at the same time, who can get only their very low ranked choices, no matter how we choose a stable matching for σ. However, even this simple measure can be very informative; for example, if for some preference list $h(\sigma)$ has a high value, then all individuals will be unhappy to some extent no matter how we choose a stable matching for σ. Let us note that by Corollary 4 we have some preference lists $\sigma \in \Sigma_n$ for which $h(\pi, \sigma)$ is very high for some stable matching $\pi \in \Pi(\sigma)$. For the same preference list, however, we may have another stable matching $\pi' \in \Pi(\sigma)$ for which $h(\pi', \sigma)$ is much lower. It is an interesting question, how high $h(\sigma)$ can be. To investigate this, let us introduce

$$h(n) = \max_{\sigma \in \Sigma_n} h(\sigma). \tag{6}$$

In studying $h(n)$, special instances which have a unique stable matching will play an important role. Let us denote by Σ_n^* the set of all those preference lists $\sigma \in \Sigma_n$ for which $|\Pi(\sigma)| = 1$, and let

$$h^*(n) = \max_{\sigma \in \Sigma_n^*} h(\sigma). \tag{7}$$

Clearly, for $\sigma \in \Sigma_n^*$ we have $\pi^M = \pi^W = \pi$ by (3); hence, the computation of $h(\sigma)$ in (5) can be simplified for such instances. Furthermore, since $\Sigma_n^* \subseteq \Sigma_n$, we get by (6) and (7) that $h^*(n) \leq h(n)$ for all $n \in \mathbb{Z}_+$.

Our next result claims that the example in Corollary 4 is essentially tight, even for instances with a unique stable matching:

Theorem 2 *For every positive integer n we have*

$$\left\lceil \frac{n}{2} \right\rceil \geq h(n) \geq h^*(n) = \left\lfloor \frac{n}{2} \right\rfloor . \tag{8}$$

Interesting open problems are (i) to describe the instances for which $h(\sigma)$ is high (for example, $h(\sigma) > cn$ for some constant c) and (ii) to compute the value of $h(\sigma)$ for typical (say, random) instances.

Besides stable rank-profiles, it would also be very interesting to understand which rank-profiles can arise from men-optimal and/or from women-optimal stable matchings. As we have recalled earlier, for every preference list $\sigma \in \Sigma_n$ there exists a unique men-optimal stable matching $\pi^M = \pi^M(\sigma)$, and a unique women-optimal stable matching $\pi^W = \pi^W(\sigma)$. Let us denote by $(k^M(\sigma), \ell^M(\sigma))$ and $(k^W(\sigma), \ell^W(\sigma))$ respectively, the corresponding rank-profiles, and introduce

$$\mathcal{M}_n = \left\{ (k^M(\sigma), \ell^M(\sigma)) \;\middle|\; \sigma \in \Sigma_n \right\} \text{ and } \mathcal{W}_n = \left\{ (k^W(\sigma), \ell^W(\sigma)) \;\middle|\; \sigma \in \Sigma_n \right\}.$$

We say that a rank-profile (k, ℓ) is *men-optimal* (*women-optimal*) if $(k, \ell) = (k^M(\sigma), \ell^M(\sigma))$ (respectively, $(k, \ell) = (k^W(\sigma), \ell^W(\sigma))$) for some preference list $\sigma \in \Sigma_n$. As we have seen above, preference lists which have a unique stable matching play a helpful role in providing lower bounds for $h(n)$, and in the study of stable rank-profiles, in general. Let us finally define $\mathcal{U}_n = \bigcup_{\sigma \in \Sigma_n^*} \mathcal{S}(\sigma)$ and let us call $(k, \ell) \in \mathcal{U}_n$ a *uniquely stable* (US) rank-profile. It follows from the above definitions that

$$\mathcal{U}_n \subseteq \mathcal{W}_n \cap \mathcal{M}_n \subseteq \left\{ \begin{array}{c} \mathcal{W}_n \\ \mathcal{M}_n \end{array} \right\} \subseteq \mathcal{S}_n \subseteq \mathcal{R}_n.$$

We shall show that in fact most containment relations above are strict. Despite this, the characterization of men-optimal rank-profiles $\mathcal{M}_n$ and women-optimal rank-profiles $\mathcal{W}_n$ can be reduced to the characterization of uniquely stable rank-profiles $\mathcal{U}_n$.

3 Further notations and properties

In this section we recall several well-known properties of stable matchings from the literature, which will be instrumental in our proofs.

First of all, it will be convenient to represent an input preference list in one matrix, as in Figure 1. For a pair $m \in \mathbf{M}$ and $w \in \mathbf{W}$ the cell (m, w) contains in the upper left corner the rank $r_m(w)$

	x	y	z
a	1 2	**3** **3**	2 2
b	**1** **3**	2 1	3 1
c	3 1	1 2	**2** **3**

Figure 1: An example $\sigma \in \Sigma_3$ involving three men $\mathbf{M} = \{a, b, c\}$ and three women $\mathbf{W} = \{x, y, z\}$.

of woman w in the preference list of man m, while the lower right corner contains $r_w(m)$, the rank of m in the preference list of w. For example, for man $a \in \mathbf{M}$ and woman $x \in \mathbf{W}$ in Figure 1 we have $r_a(x) = 1$, i.e., x is a's first choice, while $r_x(a) = 2$, that is, a is only the second choice of

x. The bold entries in Figure 1 correspond to the matching $\pi = \{(a,y),(b,x),(c,z)\}$, which is not stable, since the pair (a,z) is a breaking couple for π.

Let us recall from [10] that in the "men-oriented" algorithm the following two stages are repeated, until a stable matching is found: each man without a current partner makes an offer to the first woman in their preference list who did not yet reject him. Each woman rejects all offers, except one man's who is ranked the highest in her preference list among those who made an offer to her. When no man is rejected, all men must have a partner, and the algorithm stops. In the "women-oriented" variant, the roles of men and women are interchanged. Despite a high degree of freedom (in what order men make offers, etc.) it is known that these procedures always converge to the same stable matching. More precisely, it was proven in [10] that:

Fact 1 *For every instance $\sigma \in \Sigma_n$ the men-oriented and the women-oriented algorithms always produce, in at most $O(n^2)$ steps respectively, the unique men-optimal $\pi^M = \pi^M(\sigma)$ and unique women-optimal $\pi^W = \pi^W(\sigma)$ stable matchings, for which $k(\pi^M) \le k(\pi)$ and $\ell(\pi^W) \le \ell(\pi)$ hold for all stable matchings $\pi \in \Pi(\sigma)$.* □

For instance, for the example of Figure 1 in the men-oriented version, first both a and b make offers to x, and c makes an offer to y. The offer of b is rejected, and hence he makes a second offer to y, who is his second choice. Then y has two offers, one from c in the first step, and one from b, and since she prefers b to c, she rejects c, who then makes his second offer to z. At this moment all women have exactly one offer, so nobody is rejected and the algorithm stops, outputting $\pi^M = \{(a,x),(b,y),(c,z)\}$ as the men-optimal stable matching. Analogously, the women-oriented procedure produces $\pi^W = \{(a,z),(b,y),(c,x)\}$. In fact, in this example we have only two stable matchings, $\Pi(\sigma) = \{\pi^M, \pi^W\}$. We can see that $k(\pi^M) = (1,2,2)$ and $\ell(\pi^W) = (1,1,2)$, while we have $k(\pi^W) = (2,2,3)$ and $\ell(\pi^M) = (2,1,3)$, and indeed, for these we have $k(\pi^M) \le k(\pi^W)$ and $\ell(\pi^W) \le \ell(\pi^M)$ (where we listed these vectors keeping the natural orders, $\{a,b,c\}$ for the men, and $\{x,y,z\}$ for the women). An immediate corollary of Fact 1 is the following:

Fact 2 *If $\pi^M(\sigma) = \pi^W(\sigma)$ for a preference list $\sigma \in \Sigma_n$ then $|\Pi(\sigma)| = 1$, that is, $\sigma \in \Sigma_n^*$.* □

Let us call a pair $(m_i, w_j) \in \mathbf{M} \times \mathbf{W}$ a *men rejection* pair if in the course of the men-oriented algorithm w rejects the offer of m at one point. Analogously, we can define *women rejection* pairs. The following properties are well-known and easy to derive.

Fact 3 *Given an instance $\sigma \in \Sigma_n$, the men rejection pairs and the women rejection pairs are always the same in any variant of the men-oriented and women-oriented algorithms. Denoting these sets respectively by $R^M = R^M(\sigma)$ and $R^W = R^W(\sigma)$, we obtain that $R^M \cap R^W = \emptyset$, because*

- *for all $(m_i, w_j) \in R^M$ we have $r_{m_i}(w_j) < k_i(\pi^M)$ and $r_{w_j}(m_i) > \ell_j(\pi^M)$;*
- *for all $(m_p, w_q) \in R^W$ we have $r_{m_p}(w_q) > k_p(\pi^W)$ and $r_{w_q}(m_p) < \ell_q(\pi^W)$.*

For instance, for the example of Figure 1 we have $R^M = \{(b,x),(c,y)\}$ and $R^W = \{(b,z)\}$.

The following very useful observation can be found for instance in [11].

Fact 4 *The men-oriented algorithm terminates when the last woman receives her first offer. Analogously, the women-oriented algorithm terminates when the last man receives his first offer.*

4 Men-optimal, women-optimal, and uniquely stable rank-profiles

4.1 Main concepts and relations between them

In section 4 we consider men-optimal (MO), women-optimal (WO), and uniquely stable (US) rank-profiles. Given n, we denote the corresponding sets of rank-profiles by $\mathcal{M}_n, \mathcal{W}_n$ and $\mathcal{U}_n$, respectively. Let us recall that we denote the set of all rank-profiles by $\mathcal{R}_n$ and the set all stable ones by $\mathcal{S}_n$.

By definition, each US rank profile is simultaneously MO and WO and each MO or WO rank-profile is stable, that is, $\mathcal{U}_n \subseteq \mathcal{M}_n \cap \mathcal{W}_n, \quad \mathcal{M}_n \cup \mathcal{W}_n \subseteq \mathcal{S}_n$. The second inclusion is strict already for $n = 3$. For example, rank-profile $((2,2,2),(2,2,2)) \in \mathcal{R}_3$ is stable, yet not uniquely stable (SYNUS), and it is neither men- nor women-optimal, see section **??**.

We conjecture that the first containment is in fact an equality.

Conjecture 1 *If a rank-profile is both men- and women-optimal then it is US, that is,* $\mathcal{U}_n = \mathcal{M}_n \cap \mathcal{W}_n$.

This statement is not obvious, since a rank-profile can be MO for one preference list and WO for another one. However, computations for $n = 2, 3$ and 4 confirm this conjecture.

By Theorem 1, the set of all stable rank-profiles $\mathcal{S}_n$ is monotone, that is, $(k, \ell) \in S_n$ and $(k', \ell') \leq (k, \ell)$ imply that $(k', \ell') \in S_n$. Yet, it is not so with $\mathcal{M}_n, \mathcal{W}_n$, and $\mathcal{U}_n$ already for $n = 3$.

For example, we show that $((1,1,3),(2,2,1))$ is SYNUS; it is men-optimal but not women-optimal, while $((1,1,3),(2,2,3))$ is uniquely stable. Hence, $\mathcal{U}_3$ is not monotone, for example, $((1,1,3),(2,2,1)) \leq ((1,1,3),(2,2,3))$. Further we show that $((1,2,3),(2,2,2))$ is SYNUS; it is WO but not MO. Hence, $\mathcal{W}_3$ is not monotone, since $((1,1,3),(2,2,1)) \leq ((1,2,3),(2,2,2))$. By symmetry, $\mathcal{M}_3$ is not monotone either.

In general, it seems difficult to characterize $\mathcal{M}_n, \mathcal{W}_n$ or $\mathcal{U}_n$. Here we provide only some sufficient conditions for unique stability, and also show (Theorem 3) that characterization of $\mathcal{M}_n$ (as well as $\mathcal{W}_n$) is reduced to characterization of $\mathcal{U}_n$. It is an open question whether a membership in $\mathcal{M}_n, \mathcal{W}_n$ or $\mathcal{U}_n$ can be tested in polynomial time.

Necessary conditions for the unique stability can be found in Appendix B of [8].

4.2 Sufficient conditions

Proposition 1 *If* $(k, \ell) \in \mathcal{U}_n$, *then* $(k', \ell') \in \mathcal{U}_{n+1}$ *for all* $k' = (k, k_{n+1})$ *and* $\ell' = (\ell, \ell_{n+1})$, *where* k_{n+1} *and* ℓ_{n+1} *are arbitrary integers between* 1 *and* $n+1$.

Proof. Assume that $(k, \ell) \in \mathcal{U}_n$, and consider a corresponding preference list $\sigma \in \Sigma^*_n$. To describe a preference list $\sigma' \in \Sigma^*_{n+1}$ which corresponds to the rank-profile $(k', \ell') \in \mathcal{U}_{n+1}$ we define the preferences of the individuals as follows:

$$r'_{m_i}(w_j) = r_{m_i}(w_j) \text{ for } i,j \in \mathbf{N};\ r'_{w_j}(m_i) = r_{w_j}(m_i) \text{ for } i,j \in \mathbf{N};\ r'_{m_i}(w_{n+1}) = n+1$$
$$\text{for } i \in \mathbf{N};\ r'_{w_j}(m_{n+1}) = n+1 \text{ for } j \in \mathbf{N};\ r'_{m_{n+1}}(w_{n+1}) = k_{n+1},\ r'_{w_{n+1}}(m_{n+1}) = \ell_{n+1}.$$

All the undefined preferences (of man m_{n+1} and woman w_{n+1}) can be filled in arbitrarily. Then, due to the properties of σ' constructed above, in the men-oriented procedure only man m_{n+1} makes an offer to woman w_{n+1}, and in the women-oriented procedure only woman w_{n+1} makes an offer to man m_{n+1}. Thus, in both cases (m_{n+1}, w_{n+1}) form a couple in the final stable matching, and since the preferences of the first $n - n$ individuals did not change, these stable matchings coincide, by our assumption about σ. Hence, we have $\sigma' \in \Sigma^*_{n+1}$ and has rank-profile (k', ℓ'). □

For example, $((1),(1)) \in \mathcal{U}_1$; hence $((1,2),(1,2)) \in \mathcal{U}_2$; this in its turn, implies that

$((1,2,3),(1,2,3)) \in \mathcal{U}_3, \dots, ((1,2,\dots,n),(1,2,\dots,n)) \in \mathcal{U}_n$.

There is an alternative proof. Clearly, the unique matching, $((1,1),\dots,(n,n))$, is stable with respect to the *unanimous* instance, when all men and women have the same preference list $(1,\dots,n)$. Hence, the corresponding rank-profile $(k^u, \ell^u) = ((1,2,\dots,n),(1,2,\dots,n))$ is uniquely stable. In fact, Proposition 1 implies a stronger claim.

Proposition 2 *A rank-profile* $(k,\ell) \in \mathcal{U}_n$ *whenever* $(k,\ell) \le (k^u, \ell^u)$.

Proof. Let us generate uniquely stable rank-profiles recursively beginning with $(k,l) = ((1),(1)) \in \mathcal{U}_1$ and applying all extensions of Proposition 1. Clearly, (k,ℓ) will be obtained in such a way if and only if $(k,\ell) \le (k^u, \ell^u)$. □

Of course, not every uniquely stable rank-profiles can be obtained in this way. For example, rank-profiles $((1,1,3),(2,2,3)), ((1,1,3),(2,2,2)), ((1,2,2),(2,2,2)) \in \mathcal{U}_3$ but they are not majorized by $((1,2,3),(1,2,3))$.

Proposition 3 *If* $(k,\ell) \in \mathcal{U}_n$ *then* (k_i, ℓ_j), (k_i, k_j), *and* (ℓ_i, ℓ_j) *are not equal* (n,n) *whenever* $i \ne j$.

Proof. The fact that $(k_i, \ell_j) \ne (n,n)$ for $i \ne j$ follows by Corollary 1. The other two claims are symmetric, and it is enough to show e.g. that $(\ell_i, \ell_j) \ne (n,n)$ whenever $i \ne j$. For this, assume indirectly that $\ell_i = \ell_j = n$ for some $i \ne j$, and consider the preference list $\sigma \in \Sigma_n^*$ for which (k,ℓ) is the rank-profile of the unique stable matching. In the women-oriented algorithm for σ we have both women w_i and w_j making offers to all men, implying that all men receives at least two offers. This contradicts the fact that the algorithm terminates when the last man receives his first offer. This contradiction proves the claim. □

An important relation between unique stability and men-optimality is given by the next claim.

Theorem 3 *If a rank-profile* $(k,\ell) \in \mathcal{S}_{n-1}$ *is men-optimal then* $(k', \ell') \in \mathcal{U}_n$ *whenever* $\ell' = (\ell, n)$, $k' = (k, k_n)$, *and* $1 \le k_n \le n$. *Moreover, for* $k_n = 1$ *the inverse is also true, namely, if* $((k,1),(\ell,n)) \in \mathcal{U}_n$ *then* $(k,\ell) \in \mathcal{S}_{n-1}$ *and it is men-optimal.*

Obviously, the symmetric claim for WO rank-profiles holds, too. By this Theorem, the membership test for $\mathcal{M}_{n-1}$ (or $\mathcal{W}_{n-1}$) is reduced to such a test for $\mathcal{U}_n$. There are several other corollaries.

Corollary 5 *Rank-profile* $((k,k_n),(\ell,n)) \in \mathcal{R}_n$ *is uniquely stable whenever* $k_i = 1$ *and* $l_i < n$ *for all* $i = 1,\dots,n-1$.

Proof. $(k,\ell) \in \mathcal{R}_{n-1}$ is men-optimal, since $k = (1,\dots,1)$. By Theorem 3, $((k,k_n),(\ell,n)) \in \mathcal{U}_n$. □

Let us consider examples. Rank-profile $((2,2),(1,1))$ is WO but not MO. Hence, $((2,2,1),(1,1,3))$ is not US. However, by corollary 5, $((2,2,3),(1,1,3)) \in \mathcal{U}_3$ (and $((2,2,2),(1,1,3)) \in \mathcal{U}_3$, too). This shows that the assumption $k_n = 1$ in the second part of Theorem 3 is essential and also that $\mathcal{U}_3$ is non monotone. Furthermore, $((1,1,3),(2,2,1))$ is men-optimal (though not US), hence $((1,1,1,3),(2,2,4,1))$ is US. In contrast, $((2,2,2),(2,2,2))$ is not men-optimal (see subsection **??**) and hence $((2,2,2,1),(2,2,2,4))$ is not uniquely stable (though it satisfies all necessary conditions of unique stability which will be given in subsection **??**). Let us also remark that $((2,2,2,2),(2,2,2,4))$, $((2,2,2,3),(2,2,2,4))$, $((2,2,2,4),(2,2,2,4)) \in \mathcal{U}_4$.

In contrast to Corollary 5, not any rank-profile majorized by $(k,\ell) = ((1,...,1,k_n),(n-1,...,n-1,n))$ is uniquely stable. For example, $((1,1,3),(2,2,1))$ is SYNUS, it is MO but not WO (see section **??**), though $((1,1,3),(2,2,2))$ and $((1,1,3),(2,2,3))$ are US. In fact all components of (k,ℓ) can be reduced without loss of the unique stability, except $\ell_n = n$ which cannot be reduced to 1.

Somewhat surprisingly, Theorem 3 implies the following "anti-monotone" property.

Corollary 6 *If $((k,1),(\ell,n)) \in \mathcal{U}_n$ then $((k,k_n),(\ell,n)) \in \mathcal{U}_n$, where $1 \le k_n \le n$ and $(k,\ell) \in \mathcal{R}_{n-1}$.*

Proof. By Theorem 3, if $((k,1),(\ell,n)) \in \mathcal{U}_n$ then $(k,\ell))$ is men-optimal and if $(k,\ell))$ is men-optimal then $((k,k_n),(\ell,n)) \in \mathcal{U}_n$ for any k_n between 1 and n. □

This anti-monotone property can be strengthened as follows.

Proposition 4 *If $((k,k_n),(\ell,n)) \in \mathcal{U}_n$ then $((k,k_n+1),(\ell,n)) \in \mathcal{U}_n$, where $1 \le k_n < n$ and $(k,\ell) \in \mathcal{R}_{n-1}$.*

Proof. Since we assume $((k,k_n),(\ell,n)) \in \mathcal{U}_n$, there exists a preference list $\sigma \in \Sigma_n^*$ for which both the men- and women-oriented algorithms yield the same rank-profile $((k,k_n),(\ell,n))$.

Consider then the women-oriented algorithm. In this procedure woman w_n makes n offers, and is rejected by all men m_j, $j < n$. Therefore, we must have

$$r_{m_j}(w_n) > k_j \quad \text{for all} \quad j = 1,...,n-1. \tag{9}$$

Furthermore, man m_n must be the one who receives his first offer last, and consequently, he does not receive any other offer in this algorithm. This implies that we must also have

$$r_{w_j}(m_n) > \ell_j \quad \text{for all} \quad j = 1,...,n-1. \tag{10}$$

Let us now choose index j such that $r_{m_n}(w_j) = k_n + 1$, and create a new preference list σ' from σ by interchanging preferences $r_{m_n}(w_j) = k_n + 1$ and $r_{m_n}(w_n) = k_n$, and leaving all other preference values unchanged.

We claim that $\sigma' \in \Sigma_n^*$, completing the proof. To see this claim, let us run first the women-oriented algorithm for σ'. Since the first $n-1$ rows of the preference table did not change, and since the inequalities (9), this algorithm will run exactly the same way as for σ.

Let us compare next the runs of the men-oriented algorithm on σ and σ'. Notice that on σ' man m_n makes now one new offer to woman w_j, which he did not make in the run on σ. This new offer may result in the rejection of some other men's offers, which were not rejected in the run on σ. This is however not the case, since if another men's offer, say m_i's, to woman w_j is rejected by this new offer of man m_n, then we must have $r_{w_j}(m_i) > r_{w_j}(m_n) > \ell_j$ by (10), implying that $i \ne j$, and hence man m_i's offer must have been rejected in the run on σ, as well (otherwise that run could not have terminated with rank-profile $((k,k_n),(\ell,n))$). Thus, both algorithms terminate the same way on σ' as on σ, proving our claim. □

It seems that this claim can be strengthen further as follows.

Conjecture 2 *If $((k,k_n),(\ell,n)) \in \mathcal{U}_n$ then $((k,k'_n),(\ell,n)) \in \mathcal{U}_n$, where $(k,\ell) \in \mathcal{R}_{n-1}$, $1 \le k_n \le n$, and $2 \le k'_n \le n$.*

This claim was verified by computer for $n \le 4$. For $k_n = 1$ and, more generally, for $k_n \le k'_n$ it follows from Theorem 3. As we already know, it does not generalize the case $1 = k'_n < k_n$.

Lemma 1 *Let $n \ge 3$. If $((1,\dots,1),(\ell_1,...,\ell_{n-1})) \in \mathcal{U}_{n-1}$ such that $\ell_j < n-1$ for all $j = 1,...,n-1$, and $\ell_n < n$ is a positive integer, then we have $((1,...,1),(\ell_1+1,\ell_2+1,...,\ell_{n-1}+1,\ell_n)) \in \mathcal{U}_n$.*

Proposition 5 *If $k = (1, \ldots, 1)$ then a rank-profile $(k, \ell) \in \mathcal{U}_n$ if and only if $\ell_j = n$ for at most one $j \in \mathbf{N}$.*

Proof. By Corollary 6, at most one of the ℓ_j values can be equal to n. If $\ell_j = n$ for some j then Theorem 3 implies the statement. If $\ell_j < n$ for all $j = 1, .., n$ then Lemma 1 implies the claim. □

According to this proposition, the following rank-profiles are SYNUS:
$((1,1),(2,2))$, $((1,1,1),(1,3,3))$, $((1,1,1,1),(1,1,4,4))$, . . . , $((1,\ldots,1,1,1),(1,\ldots,1,n,n))$,
while the following (together with all rank-profiles which are majorized by them) are US:
$((1,1),(1,2))$, $((1,1,1),(2,2,3))$ $((1,1,1,1),$ $(3,3,3,4)),\ldots,((1,\ldots,1,1),(n-1,\ldots,n-1,n))$.

References

[1] A. Abdulkadiroglu, P.A. Pathak, and A.E. Roth. The new york city high school match. *American Economic Review*, May 2005. forthcoming.

[2] A. Abdulkadiroglu, P.A. Pathak, A.E. Roth, and T. Sonmez. The boston public school match. *American Economic Review*, May 2005. forthcoming.

[3] A. Abdulkadiroglu and T. Sonmez. House allocation with existing tenants. *Journal of Economic Theory*, 88:233–260, 1999.

[4] J. Alcalde and S. Barbera. Top dominance and the possibility of strategy-proof stable solutions to the marriage problem. *Economic Theory*, 4:417–435, 1994.

[5] J. Alcalde, D. Perez-Castrillo, and A. Romero-Medina. Hiring procedures to implement stable allocations. *Journal of Economic Theory*, 82:469–480, 1998.

[6] J. Alcalde and A. Romero-Medina. Simple mechanisms to implement the core of college admissions problems. *Games and Economic Behavior*, 31:294–302, 2000.

[7] Y. Blum, A.E. Roth, and U.G. Rothblum. Vacancy chains and equilibration in senior-level labor markets. *Journal of Economic Theory*, 76(2):362–411, 1997.

[8] E. Boros, L. Fedzhora, V. Gurvich, and S. Jaslar. On rank-profiles of stable matchings. RUTCOR Research Report 16, Rutgers University, 2012.

[9] B. Dutta and J. Masso. Stability of matchings when individuals have preferences over colleagues. *Journal of Economic Theory*, 75(2):464–475, 1997.

[10] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.

[11] D. Gusfield and R.W. Irving. *The Stable Marriage Problem.* MIT Press, Cambridge, Mass., 1989.

[12] R.W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the 'optimal' stable marriage. *Journal of the ACM*, 34:532–543, 1987.

[13] U. Kamecke. Wage formation in a centralized matching market. *International Economic Review*, 39(1):33–53, 1998.

[14] F. Klaus and F. Klijn. Stable matchings and preferences of couples. *Journal of Economic Theory*, 121(1):75–106, 2005.

[15] D.E. Knuth. *Mariages Stables.* Les Presses de l'Universite de Montreal, Montreal, 1976.

[16] H. Li and W. Suen. Self-fulfilling early-contracting rush. *International Economic Review*, 45(1):301–324, 2004.

[17] E. Miyagawa. House allocation with transfers. *Journal of Economic Theory*, 100:329–355, 2001.

[18] A. Romero-Medina. Implementation of stable solutions in a restricted matching market. *Review of Economic Design*, 3(2):137–147, 1998.

[19] A.E. Roth and E. Peranson. The redesign of the matching market for american physicians: Some engineering aspects of economic design. *American Economic Review*, 89(4):748–780, 1999.

[20] T. Sonmez. Can pre-arranged matches be avoided in two-sided matching markets? *Journal of Economic Theory*, 86:148–56, 1999.

An exploration into why some matchings are more likely than others

James W. Boudreau*[†]
Dept. of Economics & Finance
University of Texas-Pan American

May, 2012

Abstract

The purpose of this paper is to determine what makes decentralized matching markets more or less likely to be inclined toward efficient outcomes when efficiency is measured in terms of agents' satisfaction with their partners (a matching's choice-count). The analysis is based on one-to-one, two-sided matching markets that proceed to stability by way of a decentralized, randomized process of matching and rematching. While each one of a market's stable outcomes occurs with positive probability as a result of the process, some stable matchings are more likely to occur than others. Importantly, the most efficient stable matchings are not always the most likely to occur. Based on data generated by simulation methods, probit analysis shows that the number of stable matchings a market has, as well as the degree of correlation in a market's preferences, are significant predictors of a market's tendency toward efficiency. Both relationships appear to be non-linear.

JEL Classification Codes: C78, D63
Keywords: Matching; randomized tâtonnement; ordinal preferences.

*Mailing address: 1201 W. University Dr., Edinburg, TX, 78539-2999. Email: boudreaujw@utpa.edu.

[†] The author wishes to thank two extremely helpful referees for not only helping to improve the exposition of this paper, but also for extremely helpful suggestions that will facilitate future work on this topic.

1 Introduction

Many economic transactions consist of simple, short-lived and informal relationships, with the typical buyer-seller relationship being the classic example. Yet some of the most important economic relationships are much more long-term and formal. The employee-employer relationship, for example, or a marriage-type partnership between two individuals. For those types of relationships the process by which agents seek out and form their partnerships – the matching process – is especially crucial. Each individual wants to secure the best partner possible, but in doing so they must face the fact that others are competing for the same partners, and potential mates may not always reciprocate their feelings.

While some matching markets are now centralized in order to streamline the costly process of searching for and impressing potential partners[1], the majority are not. Most labor and marriage markets remain decentralized, and can therefore be imagined as a sequence of interactions in which agents periodically have the opportunity to break their current partnerships to form new ones. That is the type of matching process this paper is concerned with.

The popular equilibrium concept for matching markets is *stability.* A *stable* matching is an assignment of partnerships such that no matched agent would rather be single and no two agents not matched with one another would prefer each other over their current partners. Stability is intuitively appealing since it means that no agent can unilaterally improve their situation. It thus represents an ultimate resting point for a system of self-interested agents looking for their best match. But will a completely unguided matching market necessarily attain such an outcome? The answer to that question was answered in the affirmative by Roth and Vande Vate (1990), who outline a fully decentralized process (described in detail in the next section) by which one-to-one two-sided matching markets converge to stability with probability one.[2]

Importantly, however, many matching markets possess more than one stable matching, and a corollary of Roth and Vande Vate's (1990) main result is that all stable matchings can be arrived at with some positive probability if all agents are initially unmatched. Such a possibility result is certainly interesting, but it leads to more questions. Since different stable matchings can have very different welfare implications, both for individual agents and for the market as a whole, it seems important to consider whether or not some matchings will be more likely to occur than others when a matching market proceeds in a decentralized fashion. Further, if some matchings

[1] Examples include the market matching new doctors to residencies (Roth and Peranson, 1990) and markets matching city children to public schools (Abdulkadíroǧlu, Pathak and Roth, 2005; Abdulkadíroǧlu, Pathak, Roth, and Sönmez, 2005). In these instances of centralized matching, all participants submit ordinal lists of preferences or priorities for partners and an algorithm is used to assign partnerships.

[2] Their work has since been extended (with minor restrictions on preferences) by Klaus and Klijn (2007) and Kojima and Unver (2008) for the cases of many-to-one and many-to-many matching markets respectively.

are more likely than others, which types of matchings are more or less likely?

Previous work (Boudreau, 2011) has shown that, indeed, some stable matchings are more likely to occur than others for certain matching markets. That same work also shows that the most efficient stable matching in terms of agents' combined satisfaction with their partners is not always the most likely stable matching to occur as the result of Roth and Vande Vate's decentralized process. This latter result is an important, though somewhat unfortunate, result. While Roth and Vande Vate's (1990) work shows that decentralized matching markets can always attain a form of equilibrium, Boudreau (2011) demonstrates that the equilibrium most likely to be reached may be less desirable than other equilibria that could have been reached.

This paper builds on that work, further exploring the functioning of decentralized matching markets. Given that the most efficient stable matching is not always the most likely outcome for a decentralized market, this paper attempts to identify factors that contribute to that phenomenon. That is, it attempts to figure out which types of matching markets will see their most efficient outcome also be the most likely result of decentralized matching, and which will not.

Accomplishing this task theoretically proves to be difficult due to the complexity of categorizing markets based on their ordinal preference lists. Instead, as in some previous studies of matching markets (Boudreau, 2008; Boudreau and Knoblauch, 2010), Monte Carlo methods are used. By generating a wide variety of sample markets with randomly endowed preferences and simulating the decentralized matching process for each market thousands of times, it is possible to obtain an estimate of the probability distribution of each market's stable outcomes. Along with the features of each sample market (such as characteristics of agents' ordinal preference lists), these estimates provide a large data set that can be analyzed empirically.

Based on data from thousands of simulated matching markets, probit regression analysis supports two major conclusions. First, a market's potential level of efficiency, as measured by its most efficient stable outcome, alone is not a significant predictor of a market's tendency toward efficiency. Second, there are two measurable factors that do have significant, non-linear impacts on the likelihood that a market's most likely outcome will be (in)efficient: the number of stable matchings that the market possesses and the level of correlation among the preferences of each side of the market.

A market obviously must have more than one stable matching in order to have any question of which stable outcome will emerge. Accordingly, a larger number of stable matchings makes a market more likely to tend toward a less-efficient outcome, though perhaps only to a point. The effect appears to be quadratic, suggesting that at some point a larger number of stable matchings allows for more opportunity for a market to tend toward efficiency. Preference correlation has a similar effect, although the relationship is flipped. As preferences on either or side of the market become more correlated – as men all agree on the order in which they rate women's attractiveness, for example – the likelihood of an efficient outcome being the most likely result of decentralized matching increases. This makes sense since perfectly correlated preferences eliminate any debate over efficiency. Preference correlation too displays evidence of a quadratic effect, however, suggesting that high-but-not-complete levels of correlation do make less-efficient outcomes more likely.

2 Model

The type of matching markets considered here are classic "marriage" matching markets, which were first formally introduced by Gale and Shapley (1962). A marriage market consists of a set M of men and a set W of women, each of whom wants to match with just one agent from the set they do not belong to (hence the term one-to-one two-sided matching). Each man $i = 1, \ldots, n$ possesses a complete, strict, and transitive preference ordering over the set of all women and the option of remaining single dictated by a ranking function, r_{m_i}. A ranking of $r_{m_i}(w_j) = \ell$ indicates that woman j is man i's ℓth choice, so man i prefers woman j to woman k if $r_{m_i}(w_j) < r_{m_i}(w_k)$. Each woman $j = 1, \ldots, n$, similarly has a complete, strict, and transitive preference ordering over men dictated by r_{w_j}.

The collections of ordinal preference rankings for the two sets of agents are specified as R_m for men and R_w for women. For simplicity it is assumed that that $|M| = |W| = n$, and that all agents rank the option of remaining single as last, so $r_{m_i}(m_i) = n + 1$. A marriage matching market can then be defined in terms of the number of agents on each side of the market and their corresponding preferences, (n, R_m, R_w).

A *matching*, μ, is a one-to-one function from $M \cup W$ to itself such that, for all $m \in M$ and $w \in W$, $\mu(m) = w$ if and only if $\mu(w) = m$. If $\mu(m)$ is not contained in W then $\mu(m) = m$, indicating that man m is unmatched or matched with himself, with a symmetric statement holding for unmatched women. For any given matching μ if there exists a man and woman that are not matched with each other but who prefer one another to their current partners at μ, that couple is called a *blocking pair*. A *stable* matching is one that does not possess any blocking pairs.[3]

Gale and Shapley (1962) proved that there always exists at least one stable matching for any marriage matching market. In many cases, however, there may be more than one stable matching, in which event the set of stable matchings can be ranked in a variety of ways. One way, which will be used here, is to rank stable matchings according to their *choice-count*, which is the sum of the rankings that agents give their partners under a given matching μ: $\sum_{a \in M \cup W} r_a(\mu(a))$. Since stable matchings with lower choice-counts mean a higher level of aggregate satisfaction for the members of the market, this is one notion of efficiency used for matching markets (McVitie

[3] If all agents do not rank the option of being single as last, stability also requires that no individual would rather remain single than be with their assigned partner.

and Wilson, 1971; Irving, Leather and Gusfield, 1987). The most efficient matching(s) in this sense would therefore be the matching(s) with the lowest choice-count.[4]

In terms of how a decentralized market can arrive at a stable outcome, Roth and Vande Vate (1990) proved that the following process converges to a stable matching in a finite number of steps with probability one, starting from any initial matching. As in Boudreau (2011) the process is referred to as *randomized tâtonnement* because the French term for "groping" embodies the idea of unguided market adjustment.[5]

Randomized Tâtonnement Process:

Step 1. Find all possible blocking pairs existing in the current match, μ_t.

Step 2. Randomly select one of the blocking pairs, $\{m_i, w_j\}$, and satisfy them by matching them together so that $\mu_{t+1}(m_i) = w_j$. If m_i and w_j were not single in μ_t, this leaves their former partners, $\mu_t(m_i)$ and $\mu_t(w_j)$ single in μ_{t+1}. Note that in the current model, this pair is necessarily then a blocking pair for the next round.

⋮

Step k. Repeat steps 1. and 2. until no more blocking pairs exist.

The idea of the randomized tâtonnement process is simple and intuitive. Starting from an initial matching of partners, agents periodically encounter potential new partners at random. If two agents encountering one another both prefer the other over their current partner (or over being single if they are currently unmatched), each agent leaves their current partner and the two form a new relationship together, leaving their former companions alone for the time being. Repeating this process over and over, eventually there will be no two agents willing to leave their current partners for one another. Furthermore, Roth and Vande Vate (1990) also show that if the initial matching is the empty matching (all agents unmatched), every stable matching has some positive probability of being arrived at as long as each blocking pair has some probability of being selected at step 2 above.

An important assumption for this study is that blocking pairs are selected with uniform probability at step 2. The reason for that specification is first and foremost to keep the process as decentralized and unguided as possible. Any bias in favor of certain pairs being selected would obviously skew the likelihood of certain stable outcomes. The goal here, on the other hand, is to study pure tâttonement without any such direction. While some blocking pairs of agents might

[4] Of course, there are many other ways to rank matchings. See Boudreau and Knoblauch (2011) for a collection of such measures that have been used in the literature. Accordingly, the intention to extend the approach of this paper to account for other notions of welfare in decentralized matching markets will be emphasized in the concluding section. For now it is worth noting that the choice-count of a matching is a measurement closely related to the welfare concept of envy used by Romero-Medina (2001) and Klaus (2009).

[5] The same process is also referred to as "random better response dynamics" in Ackermann et al. (2008) and "unperturbed blocking dynamics" in Klaus, Klijn and Walzl (2010).

be more likely than others to encounter one another in real-world matching markets, which pairs are more or less likely to do so is a question beyond the scope of this study. Thus, allowing each existing blocking pair an equal chance of being selected at step 2 is also an assumption of simplicity and maximum entropy. Similar reasoning supports the choice of allowing blocking pairs to be satisfied myopically rather than endowing agents with more complicated far-sighted expectations.

3 Analysis and Results

A theoretical analysis of matching markets that proceed by randomized tâtonnement is difficult due to the presence of cycles along the path to stability. Since the process is completely unguided it is possible, indeed even likely, that an unstable matching will be upset by a sequence of blocking pairs that leads directly back to that initial matching (see Boudreau, 2008, for examples). Due to the possibility of cycling, and even more so due to the fact that cycles may repeat themselves any number of times (though not indefinitely), the number of possible paths to stability is intractably large even for fairly small markets with few individuals. This in turn makes an analytical calculation of the probability of each stable outcome for even small markets quite complex.[6]

Alternatively, the randomized tâtonnement process can simply be simulated for a given market (n, R_m, R_w). By repeating the process – beginning from the empty matching and selecting from the pool of blocking pairs with uniform probability at each step until arriving at one of the market's stable outcomes – and keeping track of which outcome is selected each time, it is possible to obtain an estimate of the probability distribution of that market's outcomes. To be sure the estimate is reliable the number of trials can simply be increased to monitor for any significant change.

Boudreau (2011) used that simulation approach to investigate whether or not the most efficient stable matching was always the most likely outcome to occur. The intuition behind why that might be the case is simple. If two agents are paired up but rank each other poorly, many potential blocking pairs can exist since each has many partners they would prefer; if paired agents rank each other highly there will be relatively fewer blocking pairs since there are fewer preferred partners. Thus, if a matching market evolves based on the random satisfaction of blocking pairs

[6] After the submission of this paper an extremely helpful referee pointed out that there are in fact methods to calculate explicit probabilities for the outcomes of the randomized tâtonnement process, as well as tools for checking such calculations, both of which are detailed in Biro and Norman (2011). Fortunately for this particular study, subsequent comparisons have confirmed that the estimates arrived at by way of the simulation procedures outlined below are accurate. Nevertheless, the efficiency Biro and Norman's (2011) tools will be extremely helpful for future work on this topic, especially in allowing for the study of larger markets, and as such the helpful pointer is gratefully acknowledged.

it seems reasonable to suspect that stable matchings with lower choice-counts would be more likely to result than those with higher choice-counts. Simulating an example market (Boudreau, 2011, example 1), however, proves that not to be the case.

The purpose of this paper is therefore to begin the process of identifying which factors make a particular market more or less likely to have its most efficient matching also be the most likely outcome of the randomized tâtonnement process. To accomplish that goal, the approach is simple: generate matching markets of a fixed size with different preference rankings, simulate the randomized tâtonnement process to determine whether or not the most efficient matching is the most likely outcome, and then compare the characteristics of the various markets and their outcomes.

The artificial matching markets that make up the evidence for this study were of size $n = 6$, with preferences generated uniform-randomly. For each market, every agent was assigned n independently drawn uniform-random numbers between 0 and 1 that served as their scores for each of their n possible partners. Agents' preferences were then ranked according to those randomly drawn scores, with the lowest score being ranked first, the second-lowest score ranked second, and so on. With preferences generated, the set of stable matchings was then determined by the McVitie-Wilson (1971) algorithm. With the stable set determined, the randomized tâtonnement process was then simulated 10,000 times (always beginning from the empty matching) to form an estimate of the probability distribution of stable outcomes. A total of 20,000 such markets were generated.

Of the 20,000 artificial markets, 12,020 of them had more than one stable outcome. Of those, 6,070, featured a less-efficient stable outcome – one that was not best in terms of choice-count – as the most likely to occur. This type of inefficiency is therefore prevalent: out of a fairly large random sample, the most efficient stable matching was not the most likely decentralized outcome for almost a third of all markets, and roughly half of those with more than one stable outcome.

The dependent variable of interest for this study, then, is whether or not the market attained its most efficient stable outcome. Arbitrarily, a value of 1 is assigned to any market whose most likely outcome is not the most efficient, 0 to a market whose most likely outcome is the most efficient. Probit analysis can then be used to identify features which make this phenomenon more or less likely. A positive sign for a marginal effect indicates that less-efficient outcomes are more likely, while a negative sign indicates that efficient outcomes are more likely.

As per the logic above, the first considerations for explanatory variables involve the choice-counts of matchings. The first two columns of table 1 present probit estimates of the impact the choice-count of a matching market's most likely matching ($likelyCC$)– the one to occur as the result of randomized tâtonnement – and the choice-count of a market's most efficient stable matching $minCC$. Unsurprisingly, the higher the choice-count of a market's most likely matching, the more likely it is that the outcome is less-efficient. More interesting, however, is that the choice-count of a market's most efficient stable matching does not have a significant impact at all. This strengthens the results from Boudreau (2011), suggesting that the utilitarian efficiency of the choice-count measure alone is no predictor of which outcome is most likely for a decentralized matching market.

Table 1: Marginal Effects from Probit Estimation

$likelyCC$	0.0739^{***} (0.0017)					
$minCC$		-0.0013 (0.0016)				
$\#stable$			0.3574^{***} (0.0253)		0.3510^{***} (0.0253)	
$\#stable^2$			-0.0340^{***} (0.0037)		-0.0334^{***} (0.0037)	
ρ				-4.4418^{***} (0.7567)	-2.3021^{***} (0.7584)	
ρ^2				32.5134^{***} (9.7168)	18.2647^{*} (9.488)	
Φ						-0.0722 (0.0537)
LR	2269.10	0.64	618.52	43.11	629.00	1.81

Notes: n=12,020 for all estimates. Standard errors appear in parentheses.
***,**, and * indicate statistical significance at the 1%, 5%, and 10% levels, respectively.

Another variable that seems like it should have significant a impact on a decentralized market's tendency toward efficiency is the number of stable outcomes the market possesses ($\#stable$). With more possibilities, it could mean that there is a greater chance the market could drift to less-efficient stable matchings. The third and fifth columns of table 1 show that this intuition is at least somewhat correct. The effect appears to be quadratic, however, perhaps suggesting that with enough stable outcomes available there will be more and more that share the same, most efficient, choice-count.

In addition to properties of the stable matchings themselves, another variable that should have an impact is the degree of correlation in the market's preferences. Since matching market preferences are almost exclusively treated as ordinal, it is difficult to characterize them. One way, however, is to measure the degree of shared agreement between agents on the same side of the market. That is, the extent to which women (men) agree on which man (woman) is most preferred, which is second-most preferred, and so on. Celik and Knoblauch (2007) developed a general measure for this trait, which is described in detail in the appendix of this paper. Since the degrees of correlation among men's and women's preferences have symmetric effects on the market, the product of the two measures is used to represent the degree of correlation in the market as a whole (ρ).

As displayed in the fourth and fifth columns of table 1, correlation in the market's preferences seems to increase the likelihood of an efficient outcome. This makes sense since very "un"correlated preferences can allow for very inefficient stable matchings. When all women have a different first choice among men, for example, all women getting their first choice is stable even if each woman's first choice lists that woman as their last choice. Perfectly correlated preferences, on the other hand, limit any debate over efficiency, since their will only be one stable outcome and no matter what someone will get their first choice, someone will get their second choice, and so on. Unexpected is that, like the number of stable matchings, correlation also seems to have a quadratic effect, indicating that there may be some levels of high correlation that do allow for less-efficient outcomes to be most likely.

In thinking about the effects of correlation, however, it is worthwhile to acknowledge that the degree of preference correlation on either side of a matching market does have a relationship with the number of stable matchings that will exist for that market. Thus, the two effects suggested by columns 3-5 of table 1 certainly may be related. But one final result suggests that the relationship between preference correlation and the number of stable matchings, on its own, does not tell the whole story.

An additional way to characterize a matching market's preferences is by their degree of intercorrelation–that is, that degree to which agents across the two sides of the market agree in terms of their rankings of one another. Consider, for example, a market in which each man's highest-ranked woman also ranks him highly, versus a market in which each man's highest-ranked woman ranks him poorly. Boudreau and Knoblauch (2010) define a measure for that trait (Φ), described in detail in the appendix, and show that it is extremely influential in determining the size of a market's set of stable matchings. It is also closely related to the amount of correlation in a market's preferences. This can be seen by noting that if all men rank the same woman as highest, she can not rank them all highly in return.

The last column of table 1 therefore examines the impact of preference intercorrelation on the likelihood of an (in)efficient outcome being most likely. Quite interestingly, especially in light of the significant effects of both preference correlation and the number of stable matchings, intercorrelation does not itself have a significant effect. Indeed, although the estimates are not reported due to space limitations, the lack of significance persists even when accounting for the possibility of quadratic effects or when combining intercorrelation with various permutations of the other variables in consideration. This lack of an effect is unexpected, but nevertheless underscores the separate effects of the number of stable matchings and preference correlation.

4 Conclusion

When a matching market proceeds toward stability by randomized tâtonnement, there are surely many factors that determine whether or not the most likely outcome will be the most efficient stable matching. This study was not an exhaustive search for all such factors. Rather, it simply sought to identify some significant determinants in order to guide future research.

Since most matching markets, particularly labor and dating markets, remain decentralized, it is important to understand what features determine their patterns. In particular, it is important to understand which types of stable matchings different markets are most likely to drift toward on their own. The results reported here, which are based on simulation experiments, confirm that the choice-counts of stable matchings by themselves do not determine which outcome is most likely. Certain characteristics of the market's ordinal preferences, however, such as their degree of correlation or the number of stable matchings they permit, do seem to influence whether or not the matching with the lowest choice-count is the most likely to occur.

Future work will extend the ideas of this paper to further pursue the question of which matching outcomes are most likely in decentralized settings. As noted earlier in the paper, the notion of choice-count is only one of many possible ways to rank matchings. Other criteria may also be critical in identifying which matchings are most likely. Recent evidence from laboratory experiments (Echenique and Yariv, 2011; Pais, Pinter, and Veszteg, 2012), for example, suggests that median matchings–those matchings which balance the interests of the two sides of the market–may indeed be more likely to occur than others. Boudreau (2011) shows that, like minimum choice-count matchings, median matchings are not always the most likely outcomes of the randomized tâtonnement procedure. Nevertheless, work that considers the role of balance between the interests of the two sides of the market, as well as other ordinal properties of matchings, in determining the likelihood of decentralized matching outcomes is already under way. Alternatively, cardinal aspects of matching markets such as the intensity of preferences are very likely to also play a role their decentralized progression toward stability, and thus also merit future consideration.

Ideally, in the future a more full categorization of decentralized matching market outcomes will be possible; perhaps even a complete theoretical characterization. This paper is a step in that direction of better understanding.

References

[1] Abdulkadíroğlu, A., P. Pathak & Roth, A.E. (2005). The New York City High School Match. *American Economic Review*, *95*(2), 364–367.

[2] Abdulkadíroğlu, A., P. Pathak, Roth, A.E. & T. Sönmez (2005). The Boston Public Schools Match. *American Economic Review*, *95*(2), 368–371.

[3] Ackermann, H., P.W. Goldberg, Mirrokni, V., H. Roglin & Vocking, B. (2008). Uncoordinated Two-Sided Matching Markets. *Proc. of the 9th ACM Conference on Electronic Commerce*, 256–263.

[4] Biro, P. & G. Norman (2011). Analysis of Stochastic Matching Markets. *IEHAS Discussion Paper*, MT-DP–2011/32.

[5] Boudreau, J.W. (2008). Preference Structure and Random Paths to Stability in Matching Markets. *Economics Bulletin*, *3* (67), 1–12.

[6] Boudreau, J.W. (2011). A Note on the Efficiency and Fairness of Decentralized Matching. *Operations Research Letters*, *39* (4), 231–233.

[7] Boudreau, J.W. & V. Knoblauch (2010). Marriage Matching and Intercorrelation of Preferences. *Journal of Public Economic Theory, 12* (3), 587–602.

[8] Boudreau, J.W. & V. Knoblauch (2011). What Price Stability? Social Welfare in Matching Markets. *Working Paper*, Universtiy of Connecticut.

[9] Celik, O. B. & V. Knoblauch (2007). Marriage Matching with Correlated Preferences. *Working Paper*, University of Connecticut.

[10] Echenique, F. & L. Yariv (2011). An Experimental Study of Decentralized Matching. *Working Paper*, Caltech.

[11] Gale, D. & L.S. Shapley (1962). College Admissions and the Stability of Marriage. *American Mathematical Monthly*, *69*, 9–15.

[12] Irving, R.W., P. Leather & Gusfield, D. (1987). An Efficient Algorithm for the Optimal Stable Marriage. *Journal of the ACM*, *34*, 532–544.

[13] Klaus, B. (2009). Fair Marriages: An Impossibility. *Economics Letters, 34*, 1–11.

[14] Klaus, B. & F. Klijn (2007). Paths to Stability for Matching Markets with Couples. *Games and Economic Behavior, 58*, 154–171.

[15] Klaus, B., F. Klijn & Walzl, M. (2010). Stochastic Stability for Roommate Markets. *Journal of Economic Theory, 145* (6), 2218–2240.

[16] Kojima, F. & M. U. Unver (2008). Random Paths to Pairwise Stability in Many-to-Many Matching Problems: a Study on Market Equilibration. *International Journal of Game Theory, 36*, 473–488.

[17] McVitie, D.G., & L.B. Wilson (1971). The Stable Marriage Problem. *Communications of the ACM*, *14* (7), 9–15.

[18] Pais, J., A. Pinter & Veszteg, R. (2012). Decentralized Matching Markets: A Laboratory Experiment. *Working Paper*, ISEG, Technical University of Lisbon.

[19] Romero-Medina, A. (2001). Sex-Equal Stable Matchings. *Theory and Decision*, *50*, 197–212.

[20] Roth, A.E. & J.T. Vande Vate (1990). Random Paths to Stability in Two-Sided Matching. *Econometrica, 58* 1475–1480.

Appendix: Measures of Correlation and Intercorrelation

The measure of preference correlation used in this paper comes directly from Celik and Knoblauch (2007). As in their paper, $Avei = \frac{1}{n}\sum_{j=1}^{n} r_{w_j}(m_i)$ is the average ranking of m_i by all n women, so the total measure of correlation in women's preferences is

$$\rho_w = \frac{\sum_{i=1}^{n}(Avei)^k - n(\frac{n+1}{2})^k}{\sum_{i=1}^{n} i^k - n(\frac{n+1}{2})^k},$$

where $k \geq 2$. All experiments here use $k = 9$, as recommended by Celik and Knoblauch (2007). The measure of correlation in men's preferences, ρ_m, is defined symmetrically. The combined measure used for analysis here is simply the product of the correlation of the two sides, $\rho = \rho_m \times \rho_w$.

The measure of preference intercorrelation used here comes directly from Boudreau and Knoblauch (2010). First, for each man i, square the difference between the rank m_i gives w_j and the rank w_j gives m_i and add over all women:

$$\phi_{m_i} = \sum_{j=1}^{n}(r_{m_i}(w_j) - r_{w_j}(m_i))^2.$$

Then sum across the men and divide by n to get

$$\phi_{ave} = \frac{\sum_{i=1}^{n}\phi_{m_i}}{n}.$$

Finally, the score is normalized by using the maximum possible ϕ_{ave} score that is obtained when no two men agree on the rank of any woman and each man is ranked last by his first-ranked woman, second last by his second ranked woman, third last by his third ranked woman, etc.

$$\Phi = \frac{\sum_{k=1}^{n}(n+1-2k)^2 - \phi_{ave}}{\sum_{k=1}^{n}(n+1-2k)^2}$$

Perfect positive intercorrelation therefore yields $\Phi = 1$ and perfect negative intercorrelation yields $\Phi = 0$.

Maximum Locally Stable Matchings

Christine T. Cheng* Eric McDermid†

Abstract

Motivated by the observation that most companies are more likely to consider job applicants suggested by their employees than those who apply on their own, Arcaute and Vassilvitskii modeled a job market that integrates social networks into stable matchings in an interesting way. We call their model HR+SN because an instance of their model is an ordered pair (I, G) where I is a typical instance of the Hospital/Residents problem (HR) and G is a graph that describes the social network (SN) of the residents in I. A matching μ of hospitals and residents has a *local blocking pair* (h, r) if (h, r) is a blocking pair of μ, and there is a resident r' so that r' is simultaneously an employee of h and a neighbor of r in G. Such a pair is likely to compromise the matching because the participants have access to each other through r': r can give her resume to r' who can then forward it to h. A *locally stable matching* is a matching with no local blocking pairs.

This paper continues the study of locally stable matchings, focusing on those with maximum cardinality. We refer to them as *maximum locally stable matchings*. First, we present families of instances where finding a maximum locally stable matchings is computationally easy. For one family of instances, every stable matching is a maximum locally stable matching. This family includes the case when G is a complete graph. For the other family of instances, every maximum cardinality matching is a maximum locally stable matching. This family includes the case when G is an empty graph. Next, we provide a bound on how good a stable matching approximates a maximum locally stable matching based on the size of a maximum matching of $\bar{G}$, the complement of G. An implication of this bound is that when G is almost a complete graph, a stable matching is almost a maximum locally stable matching. We then consider the case when G is almost an empty graph and show that finding a maximum locally stable matchings is still easy. Nonetheless, finding a maximum locally stable matching is in general computationally hard. In particular, we prove that finding a locally stable matching of a certain size is NP-complete and that approximating the size of a maximum locally stable matching within $21/19 - \delta$ is NP-hard.

1 Introduction

Motivated by the observation that most companies are more likely to consider job applicants suggested by their employees than those who apply on their own, Arcaute and Vassilvitskii [2] modeled a job market that integrates social networks into stable matchings. Formally, an instance of their model consists of a set of firms F, a set of workers W, and a social network graph G of the workers. Each member of $F \cup W$ has a preference list that ranks members of the opposite group that it or she finds acceptable in some linear order. Each firm f has a capacity q_f, the maximum number of

*Supported by NSF award CCF-0830678. Department of Computer Science, University of Wisconsin–Milwaukee. Email: ccheng@uwm.edu

†21st Century Technologies. Email: emcdermid@21CT.com

workers it can employ. A firm-worker pair is *acceptable* if they appear in each other's preference lists. A (many-to-one) *matching* μ of F and W is a set of acceptable firm-worker pairs where each firm is part of at most q_f pairs and each worker is part of at most one pair. It has a *blocking pair* (f, w) if (i) (f, w) is an acceptable pair, (ii)f has an opening or f prefers w to its worst employee under μ, and (iii) w is unemployed or w prefers f to her employer under μ. This blocking pair is *local* if additionally f and w have access to each other – i.e., f has an employee that is also a neighbor of w in G. In two-sided matching theory, a common goal is to find *stable matchings*, which are matchings with no blocking pairs. In this model, the matchings of interest are *locally stable matchings*, which are matchings with no local blocking pairs.

We emphasize that a local stable matching can contain blocking pairs, but these blocking pairs will unlikely to compromise the matching. This may seem odd – if f still has an opening or prefers w over its worst employee, why can't f just make a job offer to w? If w is unemployed or prefers f over her current employer, why can't she just apply to f? In the job market context, we can think of the preference lists of f and w as being constructed in an online fashion. In particular, w can be included in f's list only *after* f has seen w's resume. But for f to consider w, some employee of f who is also a friend of w must forward w's resume to f. Similarly, f can be included in w's list only *after* w has gained some reliable information about f. For this to happen, w must have a friend who works at f. This friend of w that is also an employee of f is a *point of contact* between f and w. The main assumption in this model is that a blocking pair of a matching cannot affect the matching if the firm-worker pair has no points of contact.

The first part of Arcaute and Vassilvitskii's paper [2] explores the combinatorial differences between stable matchings and locally stable matchings. Among others, they show that there are instances whose set of locally stable matchings do not form a distributive lattice under the standard ordering relation used for stable matchings. There are also instances whose locally stable matchings vastly outnumber its stable matchings. The second part of their paper examines the evolution of the job market. They consider a decentralized version of Gale and Shapley's algorithm and show that for a specific case the algorithm converges to a locally stable matching under weak stochastic conditions. They then go on to analyze the goodness of the resulting locally stable matching. The recent work of Hoefer [9] expands on the latter line of inquiry significantly.

Our Contribution. In this paper, we continue the study of locally stable matchings, focusing on those with maximum cardinality. We call them the *maximum locally stable matchings*. In our opinion, not only are the locally stable matchings interesting in itself but they are also an intriguing alternative to stable matchings. In some applications, requiring a matching to be stable can be too strong a requirement. It can also unnecessarily limit the size of the matching. This has led researchers to suggest other kinds of matchings that still take participants' preferences into consideration. They include popular matchings [1] and its many variants (e.g. [16], [14], [12], etc.), rank maximal matchings [10], and "almost stable" maximum matchings – which are maximum matchings with few blocking pairs [3]. In the job market context, locally stable matchings may not only be larger than stable matchings, they may be just as robust since participants will unlikely leave their assignments. Here are our main contributions:

- First, we present families of instances where finding a maximum locally stable matchings is computationally easy. For one family of instances, every stable matching of the instance is a maximum locally stable matching. This family includes the case when G, the social network of the workers, is a complete graph. For the other family of instances, every maximum matching of the firms and workers is a maximum locally stable matching. This family includes the case when G is an empty graph.

- Next, we show that when $\bar{G}$, the complement of G, has a maximum matching of size r, the size of a maximum locally stable matching of the instance is at most r more than the size of a stable matching of the instance. Thus, when G is almost a complete graph, a stable matching of the instance is a good approximation to its maximum locally stable matching. On the other hand, we show that when G has a constant number of edges – i.e., G is almost an empty graph – finding a maximum locally stable matching can still be done in polynomial time.

- Finally, in spite of the results above, we show that finding a maximum locally stable matching is computationally hard in general. In particular, we prove that finding a locally stable matching of a certain size is NP-complete and that approximating the size of a maximum locally stable matching within $21/19 - \delta$ is NP-hard.

The rest of the paper is organized as follows: In Section 2, we state facts and preliminary results. We present the first two results in Section 3, and the last result in Section 4. We conclude in Section 5.

2 Preliminaries

In the stable matchings literature, the problem of finding a stable matching in the Arcaute-Vassilvitskii model sans the social network is often referred to as the Hospital/Residents problem (HR). The firms correspond to the hospitals while the workers correspond to the residents. In their seminal paper on stable matchings [5], Gale and Shapley presented an algorithm that finds a stable matching for every HR instance I. It can be implemented in $O(|I|)$ time where $|I|$ is the size of I. In general, I can have many stable matchings. Nonetheless, Gale and Sotomayor [6] showed that every stable matching of I has the same size and matches exactly the same set of residents. Throughout this paper, we shall assume that every HR instance we can consider has the property that a resident r is in a hospital h's preference list if and only if h is also in r's preference list.

An example. In the following instance, let the hospitals be h_1, h_2, h_3 whose capacities are $2, 2, 4$ respectively. Let the set of residents be $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8$. Here are their preference lists:

h_1:	r_1	r_2	r_5	r_6	r_1:	h_1	h_3	r_5:	h_1
h_2:	r_3	r_4	r_7	r_8	r_2:	h_1	h_3	r_6:	h_1
h_3:	r_1	r_2	r_3	r_4	r_3:	h_2	h_3	r_7:	h_2
					r_4:	h_2	h_3	r_8:	h_2

It is not difficult to see that $\mu = \{(h_1, r_1), (h_1, r_2), (h_2, r_3), (h_2, r_4)\}$ is a stable matching of the instance. It is, however, smaller than $\sigma = \{(h_1, r_5), (h_1, r_6), (h_2, r_7), (h_2, r_8), (h_3, r_1), (h_3, r_2),$ (h_3, r_3), $(h_3, r_4)\}$, which is a maximum matching of the instance.

For HR instance I, let $B[I]$ denote the bipartite graph where the hospitals are the vertices on one side, and the residents on the other side. A pair (h, r) is an edge if and only if they form an acceptable pair. Thus, every matching of I is a subgraph of $B[I]$. Finding a maximum matching of I can be done by solving a maximum flow problem with $B[I]$ as the "base graph": Create a source

s and a directed edge from s to every hospital h, and set its capacity to q_h. Direct all edges $\{h, r\}$ in $B[I]$ from h to r and set its capacity to 1. Finally, create a sink t and a directed edge from every resident r to t, and set its capacity to 1. There is a one-to-one correspondence between the maximum flows of this network and the maximum matchings of $B[I]$.

Let us suppose that all hospitals in I have capacity 1, and we wish to compare two of its matchings μ and σ. In this case, it is useful to consider their symmetric difference $\mu \bigoplus \sigma$. In $B[I]$, it is made up of what are called *alternating paths and cycles* – i.e., paths and cycles whose edges alternately belong to μ and σ. In cycles and even-length alternating paths, the number of μ-edges and the number of σ-edges are the same; in odd-length alternating paths, the numbers differ by 1. Additionally, when $|\sigma| > |\mu|$, there is at least one odd-length alternating path with one more σ-edge than μ-edge. We shall call such a path a *σ-alternating path.*

When not all hospitals in I have capacity 1, we can transform I to another instance where this is the case. Here is a standard trick [7]. Denote by $I^{(1)}$ the instance obtained from I by doing the following: for each hospital h_i of I with capacity q_{h_i}, replace h_i by q_{h_i} clones of h_i: $h_{i,1}, h_{i,2}, \ldots, h_{i,q_{h_i}}$. Let their capacities be 1, and let their preference lists be exactly the same as that of h_i. Then for each resident r_j that has h_i in her preference list, replace h_i with the linear order $(h_{i,1}, h_{i,2}, \ldots, h_{i,q_{h_i}})$. By transforming I to $I^{(1)}$, the many-to-one matchings of I can now be viewed as one-to-one matchings of $I^{(1)}$. Let μ be a matching of I. Create the corresponding matching $\mu^{(1)}$ of $I^{(1)}$ as follows: when h_i is matched to residents $r_{j_1}, r_{j_2}, \ldots, r_{j_k}$ in μ and these residents are arranged according to its preference, let $h_{i,1}, h_{i,2}, \ldots, h_{i,k}$ be matched to $r_{j_1}, r_{j_2}, \ldots, r_{j_k}$ respectively in $\mu^{(1)}$. Notice that μ and $\mu^{(1)}$ have the same size. Moreover, it is easy to verify that this mapping is a bijection from the set of stable matchings of I to the set of stable matchings of $I^{(1)}$. Now, suppose we want to compare μ with another matching σ of I. The task becomes equivalent to comparing $\mu^{(1)}$ and $\sigma^{(1)}$ in $I^{(1)}$, and the symmetric difference technique described in the previous paragraph can now be applied.

Proposition 1. *In the HR instance I, let μ be a stable matching and σ be a maximum matching of I. Then $|\mu| \leq |\sigma| \leq 2|\mu|$.*

Proof. By definition, $|\mu| \leq |\sigma|$. Now, construct $I^{(1)}$ and the matchings $\mu^{(1)}$ and $\sigma^{(1)}$ from I, μ and σ respectively. Since $|\mu| = |\mu^{(1)}|$ and $|\sigma| = |\sigma^{(1)}|$, $|\mu^{(1)}| \leq |\sigma^{(1)}|$. This means that in $\mu^{(1)} \bigoplus \sigma^{(1)}$ there is a $\sigma^{(1)}$-alternating path. But there cannot be a $\sigma^{(1)}$-alternating path which simply consists of one edge (h, r) from $\sigma^{(1)}$ because this means that h and r are acceptable to each other in $I^{(1)}$ but are unmatched in $\mu^{(1)}$ – i.e., $\mu^{(1)}$ is not a stable matching of $I^{(1)}$ since (h, r) is a blocking pair. This contradicts the fact that $\mu^{(1)}$ was constructed from μ, a stable matching of I. Thus, in every $\sigma^{(1)}$-alternating path, the ratio of edges belonging to $\sigma^{(1)}$ and to those belonging to $\mu^{(1)}$ is at most 2:1. Hence, $|\sigma^{(1)}| \leq 2|\mu^{(1)}|$. It follows that $|\sigma| \leq 2|\mu|$. □

The previous example shows that the bound in Proposition 1 is tight.

2.1 HR+SN and max-HR+SN

Following the above terminology, we shall call the problem of finding a locally stable matching and a maximum locally stable matching in the Arcaute-Vassilvitskii's model HR+SN and max-HR+SN respectively, where SN stands for *social network.* We will, however, revert back to the original context and use firms in place of hospitals and workers in place of residents. An instance of HR+SN is an ordered pair (I, G) where I is an HR instance and G is a social network of the workers.

Example continued. In the previous example, suppose G consists of two cliques, one containing r_1, r_2, r_3, r_4 and another containing r_5, r_6, r_7, r_8. Then σ is a maximum locally stable matching of (I, G). It has several blocking pairs – $(h_1, r_1), (h_1, r_2), (h_2, r_3), (h_2, r_4)$ – but none of the pairs have a point of contact.

Proposition 2. *If μ is a locally stable matching in the HR+SN instance (I, G), then $\mu^{(1)}$ is a locally stable matching in the HR+SN instance $(I^{(1)}, G)$.*

Proof. Assume $\mu^{(1)}$ is not a locally stable matching of $(I^{(1)}, G)$ so it has a local blocking pair (f_{ik}, w_j). Thus, f_{ik} and w_j has a point of contact, say w', who is also the only employee of f_{ik} in $\mu^{(1)}$. If w_j is unmatched or is employed by a firm that is not a clone of f_i in $\mu^{(1)}$, then (f_i, w_j) is a local blocking pair of μ with w' as a point of contact – a contradiction. If w_j is employed by a clone of f_i in $\mu^{(1)}$, say $f_{ik'}$, then the fact that w_j prefers f_{ik} over $f_{ik'}$ means that $k < k'$. On the other hand, f_{ik} prefers w_j over its only employee w' means that f_i prefers w_j over w'. But this contradicts the way $\mu^{(1)}$ is constructed because f_i should prefer the worker matched to f_{ik} over the worker matched to $f_{ik'}$. Hence, we have shown that all cases lead to a contradiction. Therefore, $\mu^{(1)}$ is a locally stable matching of $(I^{(1)}, G)$. □

We note though that the converse of Proposition 2 is not always true. That is, a matching may be locally stable in $(I^{(1)}, G)$ but its corresponding matching in (I, G) is not locally stable. In the next proposition, we provide a bound similar to Proposition 1.

Proposition 3. *In the HR+SN instance (I, G), let μ be a stable matching and $\hat{\mu}$ be a maximum locally stable matching. Then $|\mu| \leq |\hat{\mu}| \leq 2|\mu|$.*

Proof. Let σ be a maximum matching of I. By definition, $|\mu| \leq |\hat{\mu}| \leq |\sigma|$. According to Proposition 1, $|\sigma| \leq 2|\mu|$. Hence, $|\hat{\mu}| \leq 2|\mu|$. □

When we appended our running example with the social network consisting of a clique containing r_1, r_2, r_3, r_4 and another clique containing r_5, r_6, r_7, r_8, σ is a maximum locally stable matching. Its size is twice that of μ. This shows that the bound of Proposition 3 is tight. The next proposition describes the interaction between the preference lists in I and the edges in G.

Proposition 4. *Let (I, G) be an HR instance. Suppose two workers w_1 and w_2 do not have a firm in common in their preference list or, equivalently, there is no firm that has w_1 and w_2 in its preference list. Let $e = \{w_1, w_2\}$. Then $(I, G - e)$ and $(I, G + e)$ have the same set of locally stable matchings as (I, G).*

Proof. Without loss of generality, assume e is an edge of G. It is easy to verify that when G' is a subgraph of G, every locally stable matching of (I, G) is also a locally stable matching of (I, G'). Thus, to prove the proposition, we simply have to show that every locally stable matching of $(I, G - e)$ is also a locally stable matching of (I, G).

Suppose μ is a locally stable matching of $(I, G - e)$ but has a local blocking pair (f, w) in (I, G). Let f and w's point of contact be w'. Hence, both w and w' have f in their preference lists; that is, $\{w, w'\} \neq \{w_1, w_2\}$. Thus, the edge $\{w, w'\}$ is in $G - e$ so that (f, w) is also a local blocking pair of μ in $(I, G - e)$, a contradiction. It follows that μ has no blocking pairs in (I, G). □

In Section 4, we shall consider max-HR+SN. Given an HR+SN instance (I, G), let A be an algorithm that outputs a locally stable matching of (I, G), which we denote as $A(I, G)$. Then A is an *$f(N)$-approximation algorithm of max-HR+SN* if for all instances (I, G) of size N, $|\mu^*(I, G)|/|A(I, G)| \leq f(N)$ where $\mu^*(I, G)$ is a maximum locally stable matching of (I, G). Thus, according to Proposition 3, the Gale-Shapley algorithm is a 2-approximation algorithm of max-HR+SN. The problem max-HR+SN is *NP-hard to approximate within $f(N)$* if the existence of an efficient $f(N)$-approximation algorithm implies P=NP.

3 The easy cases

In this section, we present a family of HR+SN instances where finding a maximum locally stable matching is easy.

Theorem 1. *Let (I, G) be an HR+SN instance. Suppose that whenever two workers have a firm in common in their preference lists, the two workers also share an edge in G. Then every stable matching of I is a maximum locally stable matching of (I, G). Consequently, when G is the complete graph, every stable matching of I is a maximum locally stable matching of (I, G).*

Proof. For now, assume that all firms in I have capacity 1. Suppose (I, G) has a locally stable matching σ that is larger than the stable matchings of I. Let μ be one of these stable matchings. Then $\sigma \bigoplus \mu$ has a σ-alternating path of the form $f_1, w_1, f_2, w_2, \ldots, f_k, w_k$ such that $(f_i, w_i) \in \sigma$ for $i = 1, \ldots, k$ and $(f_{i+1}, w_i) \in \mu$ for $i = 1, \ldots, k-1$. Since f_1 is unmatched in μ, w_1 must prefer f_2 over f_1; otherwise, (f_1, w_1) is a blocking pair of μ. Now w_1 and w_2 both have f_2 in their preference lists so they share an edge in G. It must be the case then that f_2 prefers w_2 over w_1; otherwise, (f_2, w_1) is a local blocking pair of σ. Continuing in this fashion, we have that for $i = 1, \ldots, k-1$, w_i prefers f_{i+1} over f_i because μ is a stable matching of I while f_{i+1} prefers w_{i+1} over w_i because w_i and w_{i+1} are adjacent in G and σ is a locally stable matching of (I, G). Consequently, f_k must prefer w_k over w_{k-1}. But w_k is unmatched in μ so this implies that (f_k, w_k) is a blocking pair of μ – a contradiction. Hence, σ cannot exist, and μ is a maximum locally stable matching of (I, G).

So suppose some firms in I have capacity greater than 1. Construct the HR+SN instance $(I^{(1)}, G)$ from (I, G). Notice that the property "whenever two workers have a firm in common in their preference lists, the two workers also share an edge in G" is preserved in $(I^{(1)}, G)$. Let σ be a locally stable matching of (I, G), and let μ be a stable matching of I. Consider their corresponding matchings $\sigma^{(1)}$ and $\mu^{(1)}$. From Proposition 2, $\sigma^{(1)}$ is also a locally stable matching of $(I^{(1)}, G)$. We also know that $\mu^{(1)}$ is a stable matching of $I^{(1)}$. If $|\sigma| > |\mu|$, $|\sigma^{(1)}| > |\mu^{(1)}|$. But all firms in $I^{(1)}$ have capacity 1, and according to the previous paragraph $\mu^{(1)}$ is a maximum locally stable matching of $(I^{(1)}, G)$. Hence, $|\sigma| \leq |\mu|$, and μ is a maximum locally stable matching of (I, G). □

The next theorem provides a bound that is different from the one presented in Proposition 3. It shows that when G is almost a complete graph, a stable matching of I and a maximum locally stable matching of (I, G) will almost have the same size.

Theorem 2. *Let (I, G) be an HR+SN instance. Suppose that the largest matching in $\bar{G}$, the complement of G, is r. Let $\hat{\mu}$ be a maximum locally stable matching of (I, G) and μ be a stable matching of I. Then $|\hat{\mu}| \leq |\mu| + r$.*

Proof. Once again, let us begin the proof by assuming that all firms in I have capacity 1. First, notice that $|\hat{\mu}| - |\mu|$ is bounded above by the number of $\hat{\mu}$-alternating paths in $\hat{\mu} \bigoplus \mu$. Furthermore, the $\hat{\mu}$-alternating paths are vertex-disjoint. Now, in the proof of Theorem 1, we argued that when $f_1, w_1, f_2, w_2, \ldots, f_k, w_k$ forms a $\hat{\mu}$-alternating path and $w_1, w_2, \ldots, w_k$ is a path in G, a contradiction arises. Thus, at least one of the edges w_1w_2, $w_2w_3, \ldots, w_{k-1}w_k$ must be missing from G and therefore present in $\bar{G}$. If $\hat{\mu} \bigoplus \mu$ has x $\hat{\mu}$-alternating paths, $\bar{G}$ has at least x pairwise vertex-disjoint edges. Since $x \leq r$, $|\hat{\mu}| - |\mu| \leq r$.

Now suppose some firms in I have capacity greater than 1. Again, construct the HR+SN instance $(I^{(1)}, G)$ from (I, G). Since $\hat{\mu}$ is a maximum locally stable matching of (I, G), $\hat{\mu}^{(1)}$ is a locally stable matching of $(I^{(1)}, G)$, but it may not be the largest such matching. Let τ be a maximum locally stable matching of $(I^{(1)}, G)$. Thus, $|\hat{\mu}| - |\mu| = |\hat{\mu}^{(1)}| - |\mu^{(1)}| \leq |\tau| - |\mu^{(1)}|$. From the previous paragraph, the latter is bounded by r. Hence, $|\hat{\mu}| - |\mu| \leq r$. □

We now consider the opposite case of Theorem 1.

Theorem 3. *Let (I, G) be an HR+SN instance. Suppose that whenever two workers have a firm in common in their preference lists, the two workers do not share an edge in G. Then the matchings of I are exactly the locally stable matchings of (I, G). Hence, every maximum matching of I is a maximum locally stable matching of (I, G). Consequently, when G is the empty graph, the matchings of I are exactly the locally stable matchings of (I, G), and every maximum matching of I is a maximum locally stable matching of (I, G).*

Proof. Let μ be an arbitrary matching of I. Suppose μ is not stable and contains a blocking pair (f, w). In order for (f, w) to be a local blocking pair, f and w must have a point of contact w'; i.e., both w and w' have f in their preference lists, and both are neighbors in G. But by our assumption on G, this cannot be the case. Hence, all blocking pairs of μ are not local so μ is a locally stable matching. Thus, every matching of I is a locally stable matching of (I, G). Since every locally stable matching of (I, G) is also a matching of I, it follows that the matchings of I are exactly the locally stable matchings of (I, G). The rest of the theorem follows. □

In this next theorem, we consider the case when G is almost an empty graph.

Theorem 4. *Suppose that in the HR+SN instance (I, G), G has a constant number of edges. Then a maximum locally stable matching of (I, G) can be found in time polynomial in $|I|$.*

Proof. Let $W_1 \subseteq W$ be the smallest set of workers whose induced subgraph G_1 in G contains all the r edges of G. Let $W_2 = W - W_1$, and let G_2 be the subgraph induced by W_2, which in this case is an empty graph. Thus, $G = G_1 \cup G_2$. Furthermore, every matching μ of I can be expressed as $\mu_1 \cup \mu_2$ where each μ_i is a matching involving the workers in W_i, for $i = 1, 2$. Let I_1 be the HR instance derived from I by restricting the set of workers to W_1. With some abuse in notation, let $I - \mu_1$ be the HR instance obtained from I by removing the workers matched in μ_1 and decreasing the capacities of the firms according to the number of matches they received in μ_1. Thus, μ_1 is a matching of I_1 and μ_2 is a matching of $I - \mu_1$. Conversely, if μ_1 is a matching of I_1 and μ_2 is a matching of $I - \mu_1$, putting them together as $\mu = \mu_1 \cup \mu_2$ results in a matching of I.

It is also straightforward to verify that when μ is a locally stable matching of (I, G), μ_1 and μ_2 are locally stable matchings of (I_1, G_1) and $(I - \mu_1, G_2)$ respectively. Let us now argue the converse. Suppose μ_1 and μ_2 are locally stable matchings of (I_1, G_1) and $(I - \mu_1, G_2)$ respectively but μ has a local blocking pair (f, w) whose point of contact is w_1. Thus, in μ, f has an opening or prefers w over its worst employee w_2, and that w is either unmatched or prefers f over her current employer. Furthermore, since G_2 is an empty graph, w and w_1 must both be in W_1 and neighbors in G_1. If f has an opening in μ or w_2 is in W_2, then f has an opening after the matching μ_1 so (f, w) is a local blocking pair of μ_1. On the other hand, if w_2 is in W_1, then f prefers w to its worst employee w_2 in μ_1 so that (f, w) is again a local blocking pair of μ_1. All cases lead to a contradiction. Thus, μ must be a locally stable matching of (I, G).

To find a maximum locally stable matching of (I, G), we do what is essentially a brute force method. We consider all possible matchings of I_1. For each such matching μ_1, we check to see if μ_1 is a locally stable matching of (I_1, G_1). If it is, we construct $I_1 - \mu_1$ and then find a maximum matching μ_2 of the instance. According to Theorem 3, μ_2 is a maximum locally stable matching of $(I_1 - \mu_1, G_2)$ since G_2 is an empty graph. If $\mu = \mu_1 \cup \mu_2$ is currently the largest locally stable matching of (I, G) we have seen, we store μ; otherwise, we move on to the next matching of I_1.

There are at most $2r$ workers in W_1. Each one is either unmatched or employed by one of the $|F|$ firms. Thus, the number of possible matchings of I_1 is $O((|F| + 1)^{2r})$. Verifying if a matching μ_1 of I_1 is locally stable in (I_1, G_1), constructing $I_1 - \mu_1$, and finding a maximum matching of the instance can all be done in $O(poly(|F|, |W|))$ time. Hence, finding a maximum locally stable matching of (I, G) takes $O((|F| + 1)^{2r} poly(|F|, |W|))$ time, which is polynomial in $|F|$ and $|W|$ when r is a constant. □

4 Hardness results

An SMI (Stable Marriage with Incomplete Lists) instance is just like an HR instance only all the firms have capacity 1. Hence, all of its stable matchings have the same size. An SMTI (Stable Marriage with Ties and Incomplete Lists) instance is an SMI instance except that the participants' preference lists are allowed to contain ties. For this problem, a pair (f, w) is a blocking pair of matching μ if (i) (f, w) is an acceptable pair, (ii)f has an opening or f *strictly* prefers w to its only employee under μ, and (iii) w is unemployed or w *strictly* prefers f to her employer under μ. Once again, a matching is (weakly) *stable* if it has no blocking pairs. Unlike SMI instances, the stable matchings of an SMTI instance can have different sizes. In this section, we will show that certain kinds of SMTI instances can be encoded as HR+SN instances. This will allow us to translate hardness results known for max-SMTI, the problem of computing a maximum (cardinality) stable matching of an SMTI instance, to max-HR+SN.

Let I be an SMTI instance. Suppose the ties in the preference lists of I are broken arbitrarily to create the SMI instance I'. Clearly, every stable matching of I' is also a stable matching of I. The converse, however, is not true. Let us say that the ties in I are *consistent* if for every pair of participants q and q', whenever q and q' appear in the preference lists of p and p', q and q' are in a tie in the preference list of p if and only if they are also in a tie in the preference list of p'. In the next theorem, we show that when only the firms' preference lists contain ties and these ties are consistent, then the stable matchings of I can be retrieved from I' provided we consider the locally stable matchings of (I', G) instead where G is constructed appropriately.

Theorem 5. *Let I be an SMTI instance where only the firms' preference lists contain ties, and the ties are consistent. Let I' be the SMI instance obtained by breaking the ties in the preference lists of I arbitrarily. Let G be a graph such that whenever two workers w and w' appear together in some firm's preference list in I, w and w' are adjacent if and only if they are not in a tie. Then the following is true:*

(i) Every stable matching of I is also a locally stable matching of (I', G).

(ii) Every locally stable matching μ' of (I', G) can be transformed into a stable matching μ of I such that $|\mu| \geq |\mu'|$ in time polynomial in the size of I. Consequently, every maximum locally stable

matching of (I', G) can be transformed into a maximum stable matching of I of the same size in time polynomial in $|I|$.

Proof. Let μ be a stable matching of I. Suppose μ is not a locally stable matching of (I', G), and (f, w) is one of its local blocking pairs with w' as a point of contact. Since f has capacity 1, w' is the only employee of f and f prefers w over w'. In order for (f, w) to not be a blocking pair of μ in I, w and w' must be in a tie in f's preference list in I. But this cannot be the case – w and w' are adjacent in G and ties are consistent in I. Hence, μ cannot have a local blocking pair in (I', G) and must therefore be a locally stable matching of the instance.

For (ii), suppose μ' is a locally stable matching of (I', G). Let (f, w) be a blocking pair of μ' in I. Without loss of generality, assume that w is the worker that f prefers the most among those that form a blocking pair with f. First, we note that f cannot be matched in μ'. Otherwise, if it is matched to some worker w_1 then f must strictly prefer w over w_1 so that w and w_1 are adjacent in G. In (I', G), (f, w) has w_1 as a point of contact, implying that μ' cannot be a locally stable matching because (f, w) is a local blocking pair. Since this is a contradiction, f has to be unmatched in μ'. Next, if w is unmatched in μ', let $\mu'' = \mu' \cup \{(f, w)\}$; otherwise, let $\mu'' = \mu' - \{(\mu'(w), w)\} \cup \{(f, w)\}$. If μ'' has a local blocking pair, it will involve either f or w. By our choice of w, no worker will form a local blocking pair with f. If (f_1, w) is a local blocking pair of μ'', then (f_1, w) must be a local blocking pair of μ' too – a contradiction. Hence, μ'' is still a locally stable matching of (I', G).

What we have shown is that as long as a locally stable matching of (I', G) has a blocking pair with respect to I, the matching can be modified so that (i) its size stays the same or is larger by 1, and (ii) the modified matching is still a locally stable matching of (I', G) where one worker's employer improved while everyone else's stayed the same. If we keep applying this modification, at some point there will be no more worker whose employer can be improved. The locally stable matching of (I', G) under consideration is now a stable matching of I.

Checking whether μ' is a stable matching of I, finding a blocking pair of μ' if it is not, and finding a worker that f prefers the most among those that form a blocking pair with f can be done in time polynomial in the size of I. Since the number of modifications cannot be more than $|F| \times |W|$, it follows that starting at a locally stable matching μ' of (I', G), we can arrive at a stable matching μ of I such that $|\mu| \geq |\mu'|$ in time polynomial in the size of I.

Finally, we note from (i) that a maximum locally stable matching of (I', G) is at least as large as a maximum stable matching of I. Hence, if a maximum locally stable matching of (I', G) is *also* a stable matching of I, it must be a maximum stable matching of I. Our argument in the previous paragraphs show that the last part of (ii) is true. □

In the statement of Theorem 5, we simply described what edges should be in G: whenever two workers w and w' appear together in some firm's preference list in I, w and w' are adjacent if and only if they are not in a tie. That is, we are ambivalent about edges formed by workers that do not appear together in a firm's preference list in I since according to Proposition 4, the absence or presence of these edges in G have no effect on the set of locally stable matchings of (I, G).

Consistent ties arise naturally when firms and/or workers derive their preference lists from a *master list* [11]. A master list of workers L_W is an ordering of all the workers which may or may not contain ties. Each firm's preference list contains all the workers acceptable to it and ranked in accordance with the master list. Thus, when w and w' is part of the preference list of a firm f, they are in a tie in f's list if and only if they are in a tie in L_W. A master list of firms L_F is defined similarly, and each worker's preference list is obtained in the same way. Let SMTI-2ML denote the

SMTI problem where both groups of participants derived their preference lists from a master list. The following result is known about SMTI -2ML.

Fact 1. *(Irving et al. [11]) Suppose that in the SMTI-2ML instance I, there are n firms and n workers. Determining if I has a stable matching of size n is NP-complete even if the ties occur in one master list only. The result holds even when (i) there is only one tie in that master list or (ii) all the ties are of length* 2.

We now translate this result to HR+SN.

Theorem 6. *Suppose that in the HR+SN instance (I, G), there are n firms and n workers, and each firm has capacity* 1*. Determining if I has a locally stable matching of size n is NP-complete. The result holds even when $G \cong K_n - K_{n'}$ where $n' < n$ or $G \cong K_n - F$ where F is a matching in K_n.*

Proof. Let I' be an SMTI-2ML instance with n firms and n workers, and the ties occur in one master list only. Without loss of generality, assume that it is the firms' master list of workers that contains the ties. Create the HR+SN instance (I'', G) according to Theorem 5. If there is an efficient algorithm for determining if (I'', G) has a locally stable matching of size n, which is clearly a *maximum* locally stable matching of (I'', G), then there is also an efficient algorithm for determining if I' has a stable matching of size n. But Fact 1 states that the latter is an NP-complete problem. It follows that determining if I'' has a locally stable matching of size n is also NP-complete. The form of G is based on Proposition 4, Fact 1 and Theorem 5. □

Next, we argue that max-HR+SN is NP-hard to approximate within some δ_0 by appealing to the details of the following result of Halldórsson et al. [8] with regards to approximating max-SMTI.

Fact 2. *(Halldórsson et al. [8]) It is NP-hard to approximate max-SMTI within a factor of* $21/19 - \delta$ *for any constant* $\delta > 0$.

Theorem 7. *It is NP-hard to approximate max-HR+SN within a factor of* $21/19 - \delta$ *for any constant* $\delta > 0$.

Proof. To prove Fact 2, Halldórsson et al. [8] relied on a result of Dinur and Safra [4] about approximating a minimum vertex cover of a graph. Given a graph $H = (V, E)$, we now describe how they constructed the SMTI instance I_H. Note that we shall use firms in place of men and workers in place of women. For each vertex v_i of H, create three firms v_i^A, v_i^B, v_i^C, and three workers v_i^a, v_i^a, v_i^c. Thus, there are a total of $3|V|$ firms and $3|V|$ workers. Suppose v_i is adjacent to d vertices $v_{i_1}, v_{i_2}, \ldots, v_{i_d}$. Here are the preference lists of the firms and workers corresponding to v_i:

$$\begin{array}{llllll@{\qquad\qquad}llllll}
v_i^A: & v_i^a & & & & & v_i^a: & v_i^B & v_{i_1}^C & \cdots & v_{i_d}^C & v_i^A \\
v_i^B: & (v_i^a & v_i^b) & & & & v_i^b: & v_i^B & v_i^C & & & \\
v_i^C: & v_i^b & v_{i_1}^a & \cdots & v_{i_d}^a & v_i^c & v_i^c: & v_i^C & & & &
\end{array}$$

Clearly, I_H can be constructed from H in time polynomial in the size of H. Also, notice that the ties in I_H are consistent since v_i^a and v_i^b appear together in v_i^B's preference list only. Using the reduction in Theorem 5, let (I'_H, G) be the HR+SN instance that corresponds to I_H. Let $s^+(I_H)$ and $s^+(I'_H, G)$ denote the sizes of a maximum stable matching in I_H and a maximum locally stable matching in (I'_H, G) respectively. According to Theorem 5, $s^+(I_H) = s^+(I'_H, G)$. Let $\beta^-(H)$ denote the size of a minimum vertex cover of H. They showed that $s^+(I_H) = 3|V| - \beta^-(H)$. Since

$\beta^-(H)$ is NP-hard to approximate, it follows that $s^+(I_H)$ is also NP-hard to approximate. Since $s^+(I_H) = s^+(I'_H, G)$, and HR+SN instance (I'_H, G) can be constructed from I_H in polynomial time, we also have that $s^+(I'_H, G)$ is NP-hard to approximate. We refer readers to [8] for the derivation of the factor $21/19 - \delta$. □

5 Final Remarks

Theorem 7 provides a lower bound while Proposition 3 provides an upper bound to the factor of the best approximation algorithm for max-HR+SN. Can this gap be narrowed? We suspect that the answer is yes since the source of our hardness results, max-SMTI, has a number of 3/2-approximation algorithms [15, 13, 17]. An important strategy in these algorithms is to come up with a stable matching μ so that if μ^* is a maximum stable matching of the instance, there is no length-3 μ^*-alternating path in $\mu^* \bigoplus \mu$. In other words, in all μ^*-alternating path in $\mu^* \bigoplus \mu$, the ratio of the number of μ^*-edges to the number of μ-edges is at most 3 : 2. We end with the theorem below which shows that for some HR+SN instances, choosing μ to be equal to a stable matching of the instance yields such a result.

Theorem 8. *In the HR+SN instance (I, G), let W' be the set of workers that get matched in every stable matching of I. Suppose that in graph G, whenever $w_1 \in W'$ and $w_2 \in W - W'$ have a firm in common in their preference lists, w_1 and w_2 are adjacent in G. Let μ be a stable matching of I and $\hat{\mu}$ be a maximum locally stable matching of (I, G). Then $|\hat{\mu}| \le \frac{3}{2}|\mu|$.*

Proof. Recall the proof of Theorem 1. For now, assume all firms in I have capacity 1. Consider $\hat{\mu} \bigoplus \mu$. Suppose it has a length-3 $\hat{\mu}$-alternating path: f_1, w_1, f_2, w_2. This means that f_1 is unmatched in μ so w_1 must prefer f_2 over f_1. Similarly, w_2 is unmatched in μ so f_2 must prefer w_1 over w_2. Thus, (f_2, w_1) is a blocking pair of $\hat{\mu}$. Furthermore, $w_1 \in W_1$ and $w_2 \in W - W_1$ are adjacent in G so w_2 is a point of contact between f_2 and w_1 and (f_2, w_1) is a *local* blocking pair of $\hat{\mu}$ – a contradiction. It follows that $\hat{\mu} \bigoplus \mu$ has no length-3 $\hat{\mu}$-alternating paths.

Now, suppose some firms in I have capacity greater than 1. Once more, construct $(I^{(1)}, G)$ from (I, G). Let τ be a maximum locally stable matching of $(I^{(1)}, G)$. From the previous paragraph, $|\tau| \le \frac{3}{2}|\mu^{(1)}|$. Thus, $|\hat{\mu}|/|\mu| = |\hat{\mu}^{(1)}|/|\mu^{(1)}| \le |\tau|/|\mu^{(1)}| \le 3/2$. □

References

[1] D. Abraham, R. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37:1030–1045, 2007.

[2] E. Arcaute and S. Vassilvitskii. Social networks and stable matchings in the job market. In *Proceeding of WINE '09: the 5th Workshop on Internet and Network Economics*, volume 5929 of *Lecture Notes in Computer Science*, pages 220–231. Springer, 2009.

[3] P. Biró, D. F. Manlove, and S. Mittal. Size versus stability in the marriage problem. *Theoretical Computer Science*, 411:1828–1841, March 2010.

[4] I. Dinur and S. Safra. The importance of being biased. In *Proceedings of STOC '02: the 34th Symposium on the Theory of Computing*, pages 33–42. ACM, 2002.

[5] D. Gale and L. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.

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

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

[8] M. Halldórsson, K. Iwama, S. Miyazaki, and H. Yanagisawa. Improved approximation of the stable marriage problem. *ACM Transactions on Algorithms*, 3(3), 2007.

[9] M. Hoefer. Local matching dynamics in social networks. In *Proceedings of ICALP '11: 39th International Colloquium on Automata, Languages and Programming*, volume 6756 of *Lecture Notes in Computer Science*, pages 113–124. Springer, 2011.

[10] R. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. Rank-maximal matchings. *ACM Transactions on Algorithms*, 2(4):602–610, 2006.

[11] R. Irving, D. Manlove, and S. Scott. The stable marriage problem with master preference lists. *Discrete Applied Mathematics*, 156:2959–2977, 2008.

[12] T. Kavitha and M. Nasre. Popular matchings with variable job capacities. In *ISAAC '09: Proceedings of the 20th International Symposium on Algorithms and Computation*, pages 423–433, Berlin, Heidelberg, 2009. Springer-Verlag.

[13] Z. Király. Better and simpler approximation algorithms for the stable marriage problem. *Algorithmica*, 60(1):3–20, 2011.

[14] R. McCutchen. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In *Proceedings of LATIN 2008: the 8th Latin-American Theoretical Informatics symposium*, volume 4957 of *Lecture Notes in Computer Science*, pages 593–604. Springer, 2008.

[15] E. McDermid. A 3/2-approximation algorithm for general stable marriage. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP 2009)*, Lecture Notes in Computer Science, pages 689–700, 2009.

[16] J. Mestre. Weighted popular matchings. In *Proceedings of ICALP '06: the 33rd International Colloquium on Automata, Languages and Programming*, volume 4051 of *Lecture Notes in Computer Science*, pages 715–726. Springer, 2006.

[17] K. Paluch. Faster and simpler approximation of stable matchings. Available at `http://arxiv.org/abs/0911.5660`, 2011.

Stable Flows over Time

Ágnes Cseh* Jannik Matuschke* Martin Skutella*

March 19, 2012

Abstract

In this paper, the notion of stability is extended to network flows over time. As a useful device in our proofs, we present an elegant preflow-push variant of the Gale-Shapley algorithm that operates directly on the given network and computes stable flows in pseudo-polynomial time both in the static flow and the flow over time case. We show periodical properties of stable flows over time on networks with infinite time horizon. Finally, we discuss the influence of storage at vertices, with different results depending on the priority of the corresponding holdover edges.

1 Introduction

In the *stable marriage problem* every vertex of an undirected bipartite graph G represents either a woman or a man. Their sympathy for each person of the opposite gender is expressed by their *preference lists*: the more beloved person has the higher rank. A marriage scheme is a matching M on G. We say that such a scheme is *stable*, if there is no pair of participants willing to leave their partners in order to marry each other. More formally: an edge *uv blocks* M, if u and v both are unpaired or prefer each other to their partners in M. A matching M is *stable*, if there is no blocking edge in G. Gale and Shapley [7] were the first to state that a stable matching always exists. Their well-known deferred-acceptance algorithm finds a stable marriage in strongly polynomial time.

One of the most advanced extensions of the stable marriage problem is the *stable allocation problem*. It was introduced by Baïou and Balinski [1] in 2002. Here we talk about jobs and machines instead of men and women. Edges have capacity and participants have quota on their matched edges. This quota stands for the time a job needs to get done, and the time a machine is able to work in total. Edges are used for assigning jobs to machines such that none of the machines spends more time on a job than the edge capacity allows them.

The goal is to find a feasible set of contracts such that no machine-job pair exists where both could improve their states by breaking the scheme. Edge e is *blocking*, if it is unsaturated and neither end vertex of e could fill up its quota with at least as good edges as e. An allocation x is *stable*, if none of the edges of G is blocking. Baïou and Balinski [1] give two algorithms to solve the problem: while their augmenting path algorithm runs in strongly polynomial time, the refined Gale-Shapley algorithm is more efficient in simple cases, e. g., on instances where all jobs get one of their best choices it only needs sublinear time. Dean et al. [2, 3] succeeded to speed up the first method relying on sophisticated data structures such as dynamic trees. They also extended the second one to the case of irrational data.

Stable allocations have been further generalized to *stable flows* by Ostrovsky [9] and Fleiner [4]. Fleiner also gave a constructive proof for the existence of a stable flow in every network by solving a stable allocation problem in a modified network. In an instance of stable flow, special

*TU Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany, e-mail: `{cseh,matuschke,skutella}@math.tu-berlin.de`

vertices are designated as terminals, and each non-terminal vertex has a preference list of its incoming and outgoing edges that specifies from which edges it prefers to receive flow and which edges it prefers to send flow along. Stable flows are well-suited for modeling real-world market situations, as they capture the different trading preferences of vendors and customers.

We extend this model to the setting of *flows over time.* Flows over time were introduced by Ford and Fulkerson [5, 6]. In addition to the stable flow instance we have transit times on the edges and a time horizon specifying the end of the process. We prove existence of stable flows over time and, moreover, show that, for the case of an infinite (or sufficiently large) time horizon, there is a stable flow over time which converges to a static stable flow. By introducing time to the stable flow setting, one can achieve a considerably more realistic and more interesting description of real market situations. With flows over time and transit times on the edges we can model transportation problems or illustrate distances amongst the vendors.

Structure of the paper In Section 2, we introduce the stable flow problem and present an elegant version of the Gale-Shapley algorithm that operates directly on the given network. We also give a simple proof of its correctness. In Section 3, we present the stable flow over time problem and show existence of stable flows even in the case of networks with infinite time horizon. We conclude the section by analyzing how different variations of storage at vertices can influence the stability of flows in a network.

2 Stable flows

Although most bipartite matching problems can be easily interpreted as network flow problems, stability was defined for flows only in 2008 by Ostrovsky [9]. He proved the existence and some basic properties of stable flows. Two years later, Fleiner [4] came up with a generalized setting and further results. In this paper, we use this general setting and build upon Fleiner's remarkable achievements.

2.1 Basic notions

We consider a network (D, c), where D is a directed graph whose vertices $V(D)$ are partitioned into a set of terminals $S \subseteq V(D)$ and non-terminal vertices $V(D) \setminus S$. Moreover, there is a capacity function $c : E(D) \to \mathbb{R}_{>0}$ on the edges. The digraph D might contain multiple edges as well as loops. This concession forces us to modify slightly the structure of preference lists: each vertex $v \in V(D)$ sets up a strictly ordered list of the neighboring edges instead of the neighboring vertices. For convenience, we consider the orderings on incoming and outgoing edges as two separate lists. The set of these lists are denoted by O. Vertex v prefers uv to wv, if uv has a lower number on v's preference list than vw. In this case we say that uv *dominates* wv at v and denote it by $uv <_v wv$. The same notation is used for outgoing edges.

Definition 2.1 (flow)**.** *A* flow *f in network (D, c) is a function $f : E(D) \to \mathbb{R}_{\geq 0}$ such that the following properties hold:*

1. *$f(uv) \leq c(uv)$ for all edges $uv \in E(D)$;*
2. *$\sum\limits_{uv \in E(D)} f(uv) = \sum\limits_{vw \in E(D)} f(vw)$ for all vertices $v \in V(D) \setminus S$.*

We would like to emphasize that we do not distinguish sources and sinks in S, their role is the same: they are the vertices in D that do not have to obey the Kirchhoff law.

Definition 2.2 (stable flow)**.** *A* blocking walk *of flow f is a directed walk $W = (v_1, e_1, ..., e_{k-1}, v_k)$ such that all of the following properties hold:*

1. *each edge e_i, $i = 1, ..., k-1$, is unsaturated;*
2. *$v_1 \in S$ or there is an edge $e' = v_1u$ such that $f(e') > 0$ and $e_1 <_{v_1} e'$;*
3. *$v_k \in S$ or there is an edge $e'' = wv_k$ such that $f(e'') > 0$ and $e_{k-1} <_{v_k} e''$.*

A network flow is stable*, if there is no blocking walk in the graph.*

The network can be seen as a market situation where the vertices are the traders and the edges connecting them are possible deals. All participants rank their partners for arbitrary reasons: e. g., quality, price or location. These rankings are the preference lists of our instance. Notice that edges do not necessarily correspond to deals involving the same product and there may be cycles on the graph, even of length two. An unsaturated walk is a possible deal between vendors v_1 and v_k. If they are suppliers or consumers (terminals) or can improve their situation using the unsaturated walk, then they will agree to send some flow along it and break the existing scheme.

It was shown by Fleiner [4] that each instance $\mathcal{I}$ of the stable flow problem can be converted into an equivalent instance $\mathcal{I}'$ of the stable allocation problem such that every stable flow corresponds to a stable allocation and vice versa. Since there always exists a stable allocation [1], the existence of stable flows directly follows from the equivalence of stability on $\mathcal{I}$ and $\mathcal{I}'$. The construction shows that stable flows can be found in polynomial time.

Theorem 2.3 (Fleiner, 2010 [4])**.** *There is a stable flow on every instance (D, c, O).*

Theorem 2.4 (Fleiner, 2010 [4])**.** *For a fixed instance, the value of every stable flow is the same. Moreover, each edge incident to a terminal vertex has the same value in every stable flow.*

The stable flow problem can be seen as a generalization of the stable allocation problem. We introduce two terminal vertices to G and connect s with all vertices representing jobs, and t with all vertices representing a machine. The new edges get the quota of their non-terminal end vertex as capacity. Now we orient all edges from s to the jobs, from the jobs to the machines and from the machines to t in order to get a directed network. On this network all stable flows induce a stable allocation on the original graph and vice versa.

2.2 Algorithms to find stable flows

The stable allocation problem can be solved in polynomial time [1, 3]. As mentioned in the introduction, the augmenting path algorithm is not always the most efficient way to find stable allocations: the Gale-Shapley algorithm terminates faster in some cases. It can be run on $\mathcal{I}'$ in order to give a stable allocation, which yields a stable flow on $\mathcal{I}$. Note that this holds for irrational data as well. The fact that this method can be directly applied to instance $\mathcal{I}$ is briefly mentioned by Fleiner [4]. In the following we will show how to interpret the direct application of the Gale-Shapley algorithm on the network as a preflow-push-type algorithm and prove its correctness. We will provide two variants, a basic preflow-push variant that is easy to understand and one that resembles the alternating proposal/refusal scheme of the original Gale-Shapley algorithm.

Proposal and refusal pointers For each vertex $v \in V(D)$, the algorithm maintains two pointers $p[v]$ and $r[v]$. The first pointer, $p[v]$, iterates through v's list of outgoing edges from the highest to the lowest priority edge. It points to that edge which v is currently willing to offer more flow along. Likewise, $r[v]$ iterates through v's list of incoming edges from the lowest to the highest priority edge. It points to that edge which v is going to refuse next, if necessary. For technical reasons we introduce one more element to each preference list: after passing through all neighbors $p[v]$ reaches a state encoded by $p[v] = 0$. This means that v cannot submit any more offers. Likewise $r[v] = 0$ initially, as v has no intention to refuse flow in the beginning.

Initialization The algorithm starts by saturating all edges leaving the terminal set, i. e., $f(sv) = c(sv)$ for all $sv \in E(D)$ with $s \in S$. We define the *excess* of a vertex v w.r.t. f by $\text{ex}(v, f) := \sum_{uv \in \delta^-(v)} f(uv) - \sum_{vw \in \delta^+(v)} f(vw)$, where $\delta^-(v)$ denotes the set of incoming edges, while $\delta^+(v)$ stands for the outgoing ones. Note that f initially is not a feasible flow, as $\text{ex}(v, f) > 0$ for some non-terminal vertices $v \in V(D) \setminus S$—we will call such vertices *active*.

Preflow-push variant The algorithm iteratively selects an active vertex $v \in V(D)$ and pushes as much flow as possible along $p[v]$, advancing the proposal pointer whenever the edge is saturated or an already refused edge is encountered. If $p[v]$ reaches the 0-state before all excessive flow has been pushed out of the vertex, it continues by decreasing the flow on the incoming edge $r[v]$, advancing the refusal pointer whenever the flow on the edge reaches 0. After a push operation, the excess of the vertex is 0 and another active vertex is selected. The algorithm terminates once there is no active vertex left. For a pseudo-code listing of this preflow-push approach see Algorithm 1.

Simultaneous variant An alternative variant that resembles the Gale-Shapley algorithm more closely can be obtained by performing alternating rounds of proposal and refusal steps, respectively, on all active vertices simultaneously (cf. Algorithm 2). This variant of the algorithm will prove useful when analyzing stable flows in a time-expanded network later in this paper.

Algorithm 1 Preflow-Push Algorithm

```
Initialize $p, r$. Saturate all edges leaving $S$.
while $\exists v \in V(D) \setminus S :\ \text{ex}(v, f) > 0$ do
    while $\text{ex}(v, f) > 0$ do
        if $p[v] \neq 0$ then
            PROPOSE($p[v]$)
        else
            REFUSE($r[v]$)
        end if
    end while
end while
```

Algorithm 2 Simultaneous Push Algorithm

```
Initialize $p, r$. Saturate all edges leaving $S$.
while $\exists v \in V(D) \setminus S :\ \text{ex}(v, f) > 0$ do
    for all $v \in V(D) :\ p[v] \neq 0$ do
        PROPOSE($p[v]$)
    end for
    for all $v \in V(D) :\ p[v] = 0$ do
        REFUSE($r[v]$)
    end for
end while
```

```
procedure PROPOSE($e = (v, w)$)
    if ($r[w] >_w e$ or $w \in S$) and $f(e) < c(e)$ then
        $f(e) := \min(f(e) + \text{ex}(v, f),\ c(e))$
    else
        ADVANCE(p[v])
    end if
end procedure
```

```
procedure REFUSE($e = (v, w)$)
    if $r[w] \neq 0$ and $f(e) > 0$ then
        $f(e) := \max(f(e) - \text{ex}(w, f),\ 0)$
    else
        ADVANCE(r[w])
    end if
end procedure
```

A special execution of Algorithm 1 gives the McVitie-Wilson algorithm [8] for the stable marriage problem. We can determine the choice of active vertices while running Algorithm 1 on the flow instance defined by the stable matching instance. At initialization the source sends one unit of flow to each vertex symbolizing a man. The active vertex chosen arbitrarily in the first step is one of them. He proposes along his best edge, the asked lady accepts the offer and sends the flow further to the sink. Now the second man will be chosen arbitrarily, he proposes along his best edge. If any lady gets more than one offers, her vertex enters the active set and she needs to be chosen next. Afterwards, the refused man must play the role of the selected vertex, and so on. This way we run the deferred-acceptance algorithm by taking the men one by one to the instance, always setting up a current stable matching.

Theorem 2.5. *If c is integral, both algorithms return an integral stable flow in at most $O\left(\sum_{e \in E} c(e)\right)$ iterations.*

The proof is split into three parts:

Claim 1. *Throughout the course of the algorithms, f is integral.*

Proof. We prove this by induction. The claim is true after initialization as the capacities of all edges are integral. Thus, before a call of REFUSE or PROPOSE, the excess of the corresponding vertex is integral as well. This implies that the flow value of the corresponding edge is changed by an integral amount. □

Claim 2. *Both algorithms terminate after $O\left(\sum_{e \in E} c(e)\right)$ steps.*

Proof. In each call of PROPOSE, the flow value of the corresponding edge is increased by an integral amount (by Claim 1), or the pointer p is advanced. Likewise, in each call of REFUSE, the flow value of the corresponding edge is decreased by an integral amount, or the pointer r is advanced. Once REFUSE is called for some edge uv, the flow value cannot be increased by a PROPOSE call anymore. Thus, there can be only at most $O\left(\sum_{e \in E} c(e)\right)$ calls of PROPOSE and REFUSE. □

Claim 3. *The algorithms return a stable flow.*

Proof. After termination, f is a feasible flow, as the excess of every non-terminal vertex is 0. Now suppose there is a blocking walk in the network. There are two reasons for leaving unsaturated edges in the network: either the edge was refused or there was no proposal along it using all its capacity, it stayed at least partly unexamined. We will study which case can come up at which position in the blocking walk.

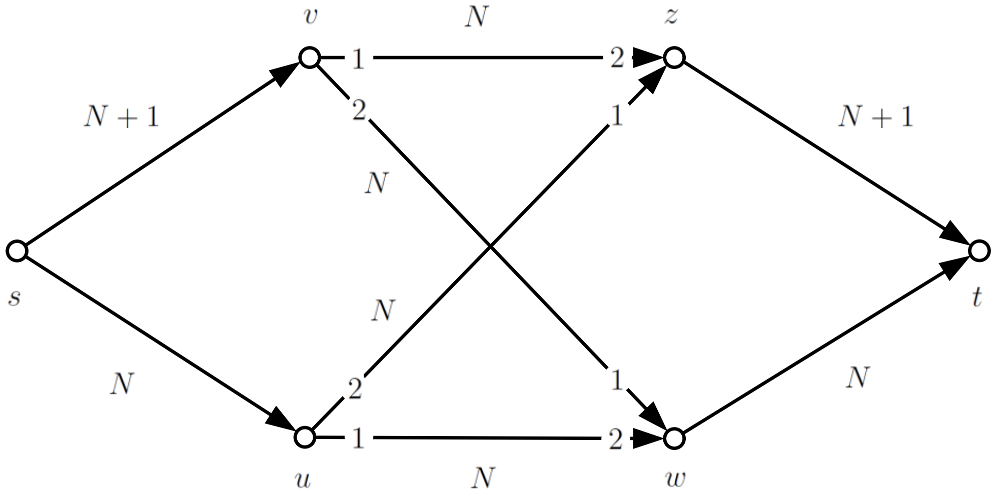
If an unsaturated walk starts at a terminal vertex, then the first edge of it has been refused, since $s \in S$ must try to fill all its adjacent edges with maximum capacity. If the walk ends at a terminal vertex, there was no full proposal along that edge in the algorithm, since terminal vertices do not refuse any flow. If the blocking walk starts at a non-terminal vertex, then there must be a dominating edge starting at the same vertex and having nonzero value. This proves that the unsaturated edge has been refused, because vertices submit offers along edges in their order in the preference list of the start vertex. A similar argument can applied to the end vertex of the blocking walk: there must be a dominating edge ending at the same vertex and having nonzero value. The unsaturated edge can not be a refused one, since we always refuse the worst edges.

The argument above shows that the blocking walk must start with a refused edge and end with a not fully proposed one. This means that along the walk there has to be at least one refused edge uv and an at least partly unexamined one vw. This implies that vertex v refused flow although it has not filled up its outgoing quota, contradiction. □

Corollary 2.6. *If c is rational, both algorithms return a rational stable flow.*

Proof. Any instance with rational capacity vector can be transformed to an instance with integral capacity vector by multiplying all capacities with their smallest common denominator. □

A simple network setting can illustrate how large capacities may cause a long running time. Note that this example is the flow extension of the allocation problem described by Baïou and Balinski [1].



In the example above $S = \{s,t\}$ and N is an arbitrary large number. After saturating sv and su the simultaneous push algorithm saturates uw and vz and proposes along vw with one unit. This offer will be accepted by w, forcing it to refuse one flow unit along uw. This way u has to submit an offer to z, that needs to accept it and reject a flow unit from v. An alternating cycle of new offers and refusals will be made along v, w, u, z as long as there is any flow to refuse along uw and vz. This means in total N augmentations along the cycle.

3 Flows over time

3.1 Basic notions

We are given a network (D, c, τ) consisting of a directed graph D, some terminal vertices $S \subseteq V(D)$ and a capacity function $c : E(D) \to \mathbb{R}_{>0}$ on the edges. The last element is the *transit time* function: $\tau : E(D) \to \mathbb{Z}_{\geq 0}$. Besides these an instance contains a *time horizon* $T \in \mathbb{Z}_{>0}$ as well. Loops and multiple edges are allowed in D.

Definition 3.1 (flow over time)**.** *Functions $f_e : \{0, 1, ..., T-1\} \to \mathbb{R}_{\geq 0}$ for each edge $e \in E(D)$ form a* flow over time *or* dynamic flow *with time horizon T, if they fulfill all of the following requirements:*

1. *$f_e(\theta) = 0$ for $\theta \geq T - \tau(e)$*
   This ensures that flow can be sent from u along an edge uv, only if there is enough time left for it to reach v.

2. *$f_e(\theta) \leq c(e)$ for all $e \in E(D)$ and $\theta \in \{0, 1, ..., T-1\}$*
   Capacity constraints hold all the time.

3. *$\sum\limits_{e \in \delta^-(v)} \sum\limits_{\xi \leq \theta - \tau(e)} f_e(\xi) = \sum\limits_{e \in \delta^+(v)} \sum\limits_{\xi \leq \theta} f_e(\xi)$ for all $v \in V(D) \setminus S$ and $\theta \in \{0, 1, ..., T-1\}$*
   Flow conservation is fulfilled at every point in time.

Any dynamic flow problem can be converted into an ordinary flow problem with the help of the *time-expanded network*. We create T copies of $V(D)$, with v_i denoting the $(i+1)$th copy of vertex i. For every edge $uv \in E(D)$ and every $i \in \{0, \ldots, T-1-\tau(uv)\}$, we connect u_i with $v_{i+\tau(uv)}$. The vertices of D^T with a fixed index i form a *timeslot*. Though this construction reduces the flow over time concept to the static setting without transit times, the price of this simplification is a considerably larger time-expanded network D^T, having a size linear in T and thus exponential in the input size. This does not always cause a problem: in the maximum dynamic flow problem there is always a temporally repeated maximum flow that can be found in polynomial time.

We extend the flow over time instance to stable flows over time, by introducing preference lists. They have the same behavior as in the stable flow problem and they do not change in time. The notion of stability can be extended to this instance the following way:

Definition 3.2 (stable flow over time)**.** *A directed walk $W = (v_1, e_1, ..., e_{k-1}, v_k)$ is a* blocking walk *of flow over time f, if there is a certain point in time $0 \leq \theta \leq T-1$ such that all the following properties hold:*

1. *each edge e_i is unsaturated at time $\theta + \sum\limits_{j=1}^{i-1} \tau(e_j)$*

2. *$v_1 \in S$ or there is an edge $e' = v_1 u$ such that $f_{e'}(\theta) > 0$ and $e_1 <_{v_1} e'$*

3. *$v_k \in S$ or there is an edge $e'' = wv_k$ such that $f_{e''}\left(\theta + \sum\limits_{j=1}^{k-1} \tau(e_j) - \tau(e'')\right) > 0$ and $e_{k-1} <_{v_k} e''$*

A blocking walk W can be interpreted in a similar way as in the static case: if the two participants symbolized by the end vertices agree that sending some flow along W would improve their situation, then the scheme will be broken by them.

Note that a flow over time is stable, if and only if the corresponding flow in the time-expanded network is stable in the classical sense.

Theorem 3.3. *For every (D, c, τ, O) and time horizon $T \in \mathbb{Z}_{>0}$ there is a stable flow over time.*

Proof. A stable static flow exists in the time-expanded network. □

Corollary 3.4. *In every stable flow over time with $T \in \mathbb{Z}_{>0}$ the terminal vertices send and receive the same amount of flow in a fixed timeslot on all edges incident to them.*

Proof. This follows from Theorem 2.4. □

Note that these two statements hold even if the preference lists may change in time.

The following lemma gives important structural insights on the stable flow computed by the simultaneous push algorithm (Algorithm 2), which will prove useful in the following sections.

Lemma 3.5. *Let $v \in V(D) \setminus S$ and $i < j$. At any step of the Algorithm 2, the following statements hold:*

1. $p[v_i] \leq_v p[v_j]$

2. If $p[v_i] =_v p[v_j] \neq 0$, *then* $f(p[v_i]) \leq f(p[v_j])$.

3. $r[v_i] \geq_v r[v_j]$

4. If $r[v_i] =_v r[v_j] \neq 0$, *then* $f(r[v_i]) \geq f(r[v_j])$.

Proof. We prove the lemma by induction on the algorithm. Clearly, all statements are true after initialization. We show that they also stay true after the end of each proposal loop. The statement for refusal loops can be shown analogously.

First observe that if $p[v_i] <_v p[v_j]$ before the loop, then (1) or (2) cannot be invalidated. Thus, let $p[v_i] =_v v_i w_{i+\tau(e)}$ and $p[v_j] =_v v_j w_{j+\tau(e)}$ for some $e = vw \in E(D)$.

(1) cannot be invalidated by a proposal loop: $p[v_i]$ will only be advanced if either $f(p[v_i]) = c(p[v_i])$ or $r[w_k] \leq_w p[v_i]$ at the beginning of the loop. In the former case, $f(p[v_j]) \geq f(p[v_i]) = c(p[v_j])$, in the latter case $r[w_l] \leq_w r[w_k] \leq_w p[v_j]$ by (3), and thus in either case $p[v_j]$ will be advanced as well.

In order to see that (2) cannot be invalidated either, observe that the inflow of v_j is at least the inflow of v_i: If $e' = uv \in E(D)$, then edge $u_{i-\tau(e')}v_i$ exists in D^T only if edge $u_{j-\tau(e')}v_j$ exists as well, and by (1) and (2), $f(u_{i-\tau(e')}v_i) \leq f(u_{j-\tau(e')}v_j)$ in this case. On the other hand, the the outflow of v_j on edges other than $p[v_j]$ is at most the same as on edges of v_i: Let $e' = (v, u)$ with $e' \neq p[v_j]$. If $e' >_v p[v_j]$, then $f(v_j u_{j+\tau(e')}) = 0$. If $e' <_v p[v_j] =_v p[v_i]$, then either $f(v_i u_{i+\tau(e')}) = c(e) \geq f(v_j u_{j+\tau(e')})$ or $r[u_{i+\tau(e)}] \leq_u e'$, which implies $f(v_j u_{j+\tau(e')}) \leq f(v_i u_{i+\tau(e')})$ by (3) and (4). Thus, $f(p[v_j])$ is set to a value at least as large as $f(p[v_i])$.

(3) and (4) stay true during a proposal loop as no r-pointer and no flow value of any edge $r[u]$ for any vertex $u \in V(D^T)$ is modified. □

3.2 Infinite time

In this section we will prove the existence of a stable flow even if the time horizon is infinite. This flow can be constructed by applying the simultaneous push algorithm on D^∞ (under the assumption that it can apply the PROPOSE and REFUSE steps on all vertices simultaneously). Even more, after a certain point in time, the stable flow is identical to a temporal repetition of the stable flow computed by the same algorithm in the static network D.

We define D_i^∞ to be the subgraph induced by the vertices v_j with $j > i$. We will run Algorithm 2 in parallel on D and D^∞, assuming we can execute the PROPOSE and REFUSE steps simultaneously on all vertices of D^∞. Let f be the flow values in D and f' be the flow values in D^∞ that occur throughout the run of the algorithm, and let p, r and p', r' be the corresponding pointers, respectively.

Our main theorem in this section states that there is a point in time, from which on all computations in the infinite time expanded network correspond one-to-one to those in the static network.

Theorem 3.6. *There is a time* $0 \leq i < \infty$ *such that throughout the course of the algorithm, for every* $j \geq i$, $f'(u_j v_k) = f(uv)$ *for all edges* $uv \in E(D)$ *and* $p'[v_j] =_v p[v]$ *and* $r'[v_j] =_v r[v]$ *for all* $v \in V(D)$.

Proof. Let $\tau_{\max} := \max_{e \in E(D)} \tau(e)$. We will first prove by induction that after K iterations of the algorithm the statement of the theorem is true for $i_K := K \cdot \tau_{\max}$. Clearly, this is true after initialization, as all edges leaving the terminal set are saturated and all pointers are at their initial state.

Now assume that after iteration K, the state of all flow variables and pointers in $D_{i_K}^\infty$ is identical to that in D. Now let $v \in V(D)$ be any vertex that is subject to a PROPOSE operation in D. Note that for $j > i_{K+1}$, vertex v_j is only incident to edges in $D_{i_K}^\infty$, as $i_{K+1} = i_K + \tau_{\max}$.

Therefore, $\mathrm{ex}(v_j, f') = \mathrm{ex}(v, f)$. As $p'[v_j] =_v p[v]$ and $r'[w_{j+\tau(e)}] =_w r[w]$, the PROPOSE call has the same effect on $p'[v_j]$ and $f'(p[v_j])$ as it has on $p[v]$ and $f(p[v])$. An analogous statement holds for calls of REFUSE.

We now have shown that up to iteration K, the statement of the theorem holds for i_K. Now let K_0 be the number of iterations performed by Algorithm 2 before f is a stable flow in D. After iteration K_0, no flow values or pointers in D are changed. Let $i := i_{K_0+1}$. We will show that after iteration K_0, also no flow values or pointers in D_i^∞ are changed, which concludes the proof of the theorem.

After iteration K_0, all vertices in D_i^∞ are inactive. Thus, their state can only change if the flow value on an edge $v_k w_l$ for $k < i$ and $l \geq i$ is increased. This can only happen, if $f'(v_k w_l) < c(e)$, as well as $r'[w_l] >_w v_k w_l$, and $p[v_k] =_v vw$. Note that after iteration K_0, $f'(v_{k+\tau_{\max}} w_{l+\tau_{\max}}) = f'(v_k w_l)$ and $r'[w_{l+\tau_{\max}}] =_w r'[w_l] >_w vw$ by choice of $i = i_{K_0+1}$. Now, by Lemma 3.5, $p[v_{k+\tau_{\max}}] \leq_v p[v_k]$, and as $v_{k+\tau_{\max}} w_{l+\tau_{\max}}$ is neither refused nor saturated, $p[v_{k+\tau_{\max}}] =_v vw =_v p[v_k]$. Again by Lemma 3.5, $f'(v_k w_l) \leq f'(v_{k+\tau_{\max}} w_{l+\tau_{\max}})$ throughout the remaining course of the algorithm, i. e., $f'(v_k w_l)$ cannot be increased before $v_{k+\tau_{\max}}$ becomes active. Thus, $f(v_k w_l)$ cannot be increased. □

Corollary 3.7. *Algorithm 2 constructs a stable flow f' in D^T such that $f'(v_i w_{i+\tau(vw)}) = f(vw)$ for all $vw \in E(D)$ and all $j \geq (K_0 + 1) \cdot \tau_{\max}$, where K_0 is the number of iterations performed by the same algorithm to compute the stable flow f in D.*

3.3 Storing at vertices

The third point in the definition of a flow over time requires flow conservation in every timeslot. A different model allows storing at vertices: a vendor may delay the shipment of goods at his convenience, as long as the flow arrives the terminal vertices within the time horizon. More formally, we generalize the definition of the *excess* of vertex v at time θ as the amount of stored goods at vertex v at time θ:

$$\mathrm{ex}_f(v, \theta) := \sum_{e \in \delta^-(v)} \sum_{\xi \leq \theta - \tau(e)} f_e(\xi) - \sum_{e \in \delta^+(v)} \sum_{\xi \leq \theta} f_e(\xi)$$

While *strict flow conservation* requires $\mathrm{ex}_f(v, \theta) = 0$ for all non-terminal vertices and times in $\{0, ..., T-1\}$, *weak flow conservation* allows $\mathrm{ex}_f(v, \theta) \geq 0$ except for time $T-1$, where $\mathrm{ex}_f(v, T-1) = 0$ must hold for all $v \in V(D) \setminus S$.

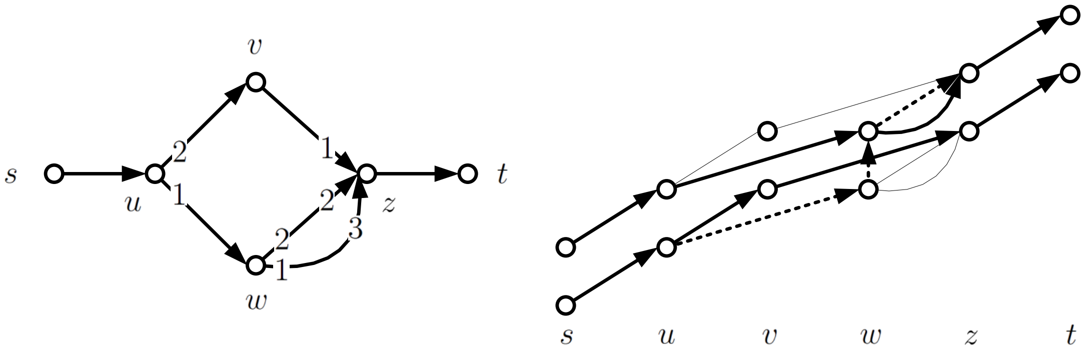
The time-expanded network can be adapted for weak flow conservation by introducing so-called *holdover edges* $v_i v_{i+1}$ of infinite capacity for each $v \in V(D)$ and $i \in \{0, ..., T-2\}$. Setting the ranks of these holdover edges in the preference lists allows the investigation of interesting variations. One intuitive view comes from the fact that storing goods is expensive and people are impatient, thus letting the edges $v_i v_{i+1}$ be the last on v_i's preference list and the first on v_{i+1}'s preference list. In the following we will investigate the four main cases with weak flow conservation, given by varying first and last preferences on the end vertices of holdover edges. In each case we study how the sets of stable flows obeying the different flow conservation rules are related to each other.

We adapt the definition of blocking walk for the case of intermediate storing. Apart from a starting time θ we need the exact time θ_i when the walk in D leaves a vertex v_i. A walk in D together with a sequence $\{\theta, \theta_1, \theta_2, ..., \theta_k\}$ uniquely determines a walk on D^T.

Example 3.8. *Introducing waiting at vertices stable flows may lose their stability, independent from the rank of holdover edges.*

Proof. Consider the following example:

All edges in the static network have unit capacity and unit transit time, s and t are the non-terminal vertices. The unobvious preferences are shown on the edges. The time-expanded



network with $T = 5$ stands on the right side, showing only the vertices and edges that may be used in a feasible flow. On the static network above there are three stable flows, each of them uses a different incoming edge of z. If waiting is not allowed, then the dynamic flow shown by the thick edges on the second network is stable, since all unsaturated paths are dominated by it at one end. Adding holdover edges to the time-expanded network breaks the scheme: the walk $\{u_1, w_2, w_3, z_4\}$ (denoted by dashed lines) blocks the flow over time. Note that this is independent of the rank of the holdover edges, since we only need dominance at the ends of the blocking walk. □

Theorem 3.9. *If T is sufficiently large and holdover edges always stand on the first place on preference lists, then there is no stable flow with value 0 on all holdover edges.*

Proof. Suppose there is a stable flow on the time-expanded network that does not use any of the holdover edges. Consider all the copies of an arbitrary non-terminal vertex v: If there are two of them that have a positive value on all incoming (and outgoing) edges in total, then holdover edges between them form a walk that blocks the flow. This way a stable flow over time passes through an arbitrary non-terminal vertex maximal once. Apart from the terminal-terminal edges that do not affect stability—a flow with this property has positive value on at most $2|V(D) \setminus S|$ edges in the time-expanded network.

These maximum $2|V(D) \setminus S|$ edges have to ensure that there is no unsaturated walk between terminal vertices. If the length (w.r.t. transit times) of the shortest walk is denoted by $\tau(W_{min})$, then $T - \tau(W_{min}) + 1$ is a lower bound for the number of disjoint walks between terminal vertices. In order to saturate at least one edge along these walks, the inequality $2|V(D) \setminus S| \geq T - \tau(W_{min}) + 1$ must hold, otherwise at least one copy of the shortest walk is unsaturated, hence it blocks the flow. □

Theorem 3.10. *If $T \in \mathbb{Z}_{>0}$ and the holdover edges stand on the last place on preference lists, then the stable flow produced by the simultaneous push algorithm has value 0 on all holdover edges.*

Proof. By contradiction assume, there is a holdover edge with positive flow value at some vertex v. W.l.o.g., assume i to be minimal with $f(v_i v_{i+1}) > 0$. Since the time horizon is finite, there also is a latest point in time j such that $f(v_j v_{j+1}) > 0$.

Since $f(v_i v_{i+1}) > 0$, the pointer $p[v_i]$ must already have passed all outgoing non-holdover edges of v_i, offering as much flow as the corresponding vertices are willing to accept. Therefore, by Lemma 3.5, the flow on each of these edges must be at least the flow on the corresponding edges of v_{j+1} (if they exist). Thus, the total outflow of v_i exceeds the total outflow of v_{j+1} by at least $f(v_i v_{i+1})$. On the other hand, $r[v_{j+1}]$ points either to 0 or the incoming holdover edge, and thus v_{j+1} does not refuse any flow on the regular incoming edges. Again by Lemma 3.5, the flow on those edges is at least the flow on the corresponding edges of v_i. Thus, the total inflow of v_{j+1} exceeds the total outflow of v_i by at least $f(v_j v_{j+1})$. Putting this together yields inflow(v_{j+1}) − outflow$(v_{j+1}) \geq$ inflow$(v_i) + f(v_j v_{j+1})$ − outflow$(v_i) + f(v_i v_{i+1}) > 0$, contradicting flow conservation.

□

In the other two cases, when holdover edges are the best for one of the end vertices and the worst for the other one, storing can be used or avoided depending on the network. There are even networks where all stable flows have positive value on some holdover edges.

Conclusion and open questions

We introduced stable flows over time, extending the concept of stability to network flows over time. As initial results, we proved the existence of stable flows both for finite and infinite time horizon. In both cases, a stable flow can be computed in pseudo-polynomial time by applying a preflow-push algorithm operating directly on the flow network. We also showed that the possibility of storage at non-terminal vertices has an effect on the set of stable flows, depending on the preference given to holdover edges in the time expanded network.

Although this paper provides first structural insights, many questions brought up by the definition of stable flows over time remain open, most prominently the complexity of the stable flow problem: Is there a polynomial time algorithm for finding a stable flow? As of today, it is not even clear whether a stable flow over time can be encoded in polynomial space. First research in this direction indicates that at least the concept of (generalized) temporally repeated flows cannot be applied directly.

Also, further extension of stability can be studied on the new setting, e. g., special edges or ties on preference lists, as can be extensions of the flow over time model, e. g., connections to earliest arrival flows. Finally, we conjecture a stronger form of Theorem 3.10: If holdover edges have lowest priority at both start and end vertex, no stable flow uses storage at non-terminal vertices.

Acknowledgements This work was supported by the Berlin Mathematical School and by the DFG Research Center Matheon in Berlin.

References

[1] Mourad Baïou and Michel Balinski. The stable allocation (or ordinal transportation) problem. *Math. Oper. Res.*, 27(3):485–503, 2002.

[2] Brian C. Dean, Michel X. Goemans, and Nicole Immorlica. Finite termination of "augmenting path" algorithms in the presence of irrational problem data. pages 268–279, 2006.

[3] Brian C. Dean and Siddharth Munshi. Faster algorithms for stable allocation problems. *Algorithmica*, 58(1):59–81, 2010.

[4] Tamás Fleiner. On stable matchings and flows. In *Proceedings of the 36th International Workshop on Graph-Theoretic Concepts in Computer Science*, WG'10, pages 51–62, Berlin, Heidelberg, 2010. Springer-Verlag.

[5] Lester R. Ford and Delbert R. Fulkerson. Constructing Maximal Dynamic Flows from Static Flows. *Operations Research*, 6(3):419–433, 1958.

[6] Lester R. Ford and Delbert R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[7] David Gale and Lloyd Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 1:9–14, 1962.

[8] David G. McVitie and Leslie B. Wilson. The stable marriage problem. *Commun. ACM*, 14(7):486–490, July 1971.

[9] Michael Ostrovsky. Stability in supply chain networks. *American Economic Review*, 98(3):897–923, June 2008.

Matching with partially ordered contracts

Rashid Farooq[*] Tamás Fleiner[†] Akihisa Tamura[‡]

July 20, 2012

Abstract

In this paper, we study a many-to-many matching model with contracts. We extend the economic model of Hatfield and Milgrom by allowing a partial order on the possible bilateral contracts of the agents in a two-sided market economy. To prove that a generalized stable allocation exists, we use generalized form of properties like path-independence and substitutability. The key to our results is the well-known lattice theoretical fixed point theorem of Tarski. The constructive proof of this fixed point theorem for finite sets turns out to be the appropriate generalization of the Gale-Shapley algorithm also in our general setting.

1 Introduction

A significant step in the research of generalized two-sided matching markets is the observation that there is a close connection between stability and Tarski's well-known fixed point theorem [9]. The first such step was probably done by Adachi [1] who described stable marriages as fixed points of a monotone function. Fleiner [5, 6] proved that a fairly general class of stability problems defined with the help of choice functions (including many-to-many versions and matroid-generalizations and much more) fits into the Tarski-based framework. Choice functions in this framework must have the so called comonotone property that is closely related to the well-known substitutability condition. Fleiner pointed out that the well-known theorem by Blair [4] on the lattice structure of generalized stable matchings follows more or less directly from Tarski's fixed point theorem. He also pointed out that the proposal algorithm of Gale and Shapley [7] can be regarded as an iteration method of a monotone function for finding a maximal or a minimal fixed point. A key ingredient in Fleiner's approach is that he considered the set of edges of the underlying bipartite graph as the domain of the key monotone mapping. Fleiner defined the so-called increasing property of a choice function meaning that if we extend the choice set then the number of choices picked cannot decrease. He proved that if choice functions are increasing (beyond comonotone and path-independent) then fixed points of the corresponding monotone function form a sublattice. Hence, the lattice operations on the stable solutions can be calculated directly by the corresponding choice

[*] Centre for Advanced Mathematics and Physics, National University of Sciences and Technology, H-12, Islamabad, Pakistan. E-mail: farooq@camp.nust.edu.pk

[†] Department of Computer Science and Information Technology, Budapest University of Technology and Economics, H-1117 Magyar Tudósok körútja 2, Hungary. Research was supported by the OTKA K 69027 research project and the MTA-ELTE Egerváry Research Group. E-mail: fleiner@cs.bme.hu

[‡] Department of Mathematics, Keio University, Yokohama 223-8522, Japan. This work was supported by Grants-in-Aid for Scientific Research (B) from JSPS. E-mail: aki-tamura@math.keio.ac.jp

functions. An easy consequence of this is that a natural generalization of the rural hospitals theorem holds for increasing, comonotone and path-independent choice functions. (The "rural hospital theorem" says that if a certain hospital h cannot fill up its quota in some stable matching then hospital h gets the same set of residents in any stable matching.) Based on the Tarski-framework and other well-known results, Fleiner also gave a linear description of several stable matching related polyhedra.

Independently and after Fleiner's work, Alkan [2] and Alkan and Gale [3] proved that if choice functions are cardinal monotone, or more generally size monotone then stable matchings form a lattice and an extension of the rural hospital theorem also holds. These works do not lean on Tarski's theorem and hence the basis of the proofs is a natural generalization of the proposal algorithm of Gale and Shapley.

A major breakthrough in popularizing the Tarski framework was done by Hatfield and Milgrom in [8]. They rediscovered several results of Fleiner and formulated them in a terminology that is much closer to Economists than the former Mathematical approach. In particular, they called the set of edges of the underlying bipartite graph "contracts" and defined substitutable mappings on them. They formulated a stability concept equivalent to Fleiner's in [5, 6] and proved that if contracts are substitutes (that is, if the choice functions are comonotone) for the hospitals and doctors have a linear preference order then in this two-sided one-to-many market stable allocations are basically the fixed points of a monotone function. They also pointed out that the Gale-Shapley algorithm is a monotone function iteration. Another important achievement of [8] is the formulation of the "law of aggregate demand" that corresponds to Fleiner's increasing property and Alkan's cardinal monotonicity. A main result is that if this condition also holds for the hospitals' choice functions then the rural hospital theorem can be generalized and that honest behaviour is a dominant strategy for the doctors if the doctor optimal stable assignment is realized after some bargaining process.

We propose a model that is a genuine generalization of the models by Fleiner [5, 6] and Hatfield and Milgrom [8]. We give an example of natural and practical preferences which are substitutable in our framework but not in the existing models. This shows that our model is not just a mathematical generalization but also a practically interesting one. Our main results are contained in Section 4 where we prove the existence of a stable solution and we extend Blair's result on the lattice structure of stable assignments [4] to our model. We point out that the proposal algorithm of Gale and Shapley can be regarded as an iteration of a monotone mapping. We show another related result in this section by demonstrating that a generalization of the proposal algorithm of Gale and Shapley can be used to calculate the lattice operations on stable solutions.

2 Preliminaries

In this section, we recall some concepts related to partially ordered sets (posets) that are essential in our framework.

A *partially ordered set* (or *poset*) P on a ground set X is a pair $P = (X, \leq)$ where $\leq$ is a reflexive, antisymmetric and transitive binary relation on X. (That is, for any $x, y, z \in X$ we have $x \leq x$ and ($x \leq y \leq x \Rightarrow x = y$) and ($x \leq y \leq z \Rightarrow x \leq z$.) Elements x and y of poset $P = (X, \leq)$ are *comparable* if $x \leq y$ or $y \leq x$, otherwise x and y are *incomparable.* If $P = (X, \leq)$ is a poset then a *lower ideal* is a set X' of X such that if $y \leq x \in X'$ then $y \in X'$ holds. Poset $P = (X, \leq)$

is called *trivial* if no two different elements of X are comparable. We shall often abuse notation by identifying a poset with its ground set, so for example a mapping $f : P \to P$ means simply a mapping $f : X \to X$ if we want to emphasize the underlying partial order. Or a subset P' of poset $P = (X, \leq)$ means a poset $P' = (X', \leq|_{X'})$ for some subset X' of X where $\leq|_{X'}$ means the restriction of binary relation $\leq$ on X'.

A subset A of X is an *antichain* of P if no two elements of A are comparable in P, that is, if $a \not\leq a'$ and $a' \not\leq a$ for different elements a, a' of A. Let $\mathcal{L}(P)$ and $\mathcal{A}(P)$ denote the set of lower ideals and antichains of P, respectively. Note that if P is trivial then $\mathcal{L}(P) = \mathcal{A}(P) = 2^X$, which consists of all subsets of X. For finite posets P, there is a natural bijection between $\mathcal{L}(P)$ and $\mathcal{A}(P)$. If $L \in \mathcal{L}(P)$ is lower ideal then clearly $\text{Max}(L) = \{x \in L : x \leq x' \in L \Rightarrow x = x'\}$ is an antichain, so $\text{Max} : \mathcal{L}(P) \to \mathcal{A}(P)$. Moreover, if $A \in \mathcal{A}(P)$ is an antichain then $\text{Li}(A) := \{x \in X : \exists a \in A \text{ such that } x \leq a\}$ is a lower ideal, hence $\text{Li} : \mathcal{A}(P) \to \mathcal{L}(P)$. It is easy to check that for any finite poset $\text{Li}(\text{Max}(L)) = L$ and $\text{Max}(\text{Li}(A)) = A$ hold for any lower ideal L of $\mathcal{L}(P)$ and any antichain A of $\mathcal{A}(P)$, that is, Max and Li are inverses of one another and both of them define a bijection between $\mathcal{L}(P)$ and $\mathcal{A}(P)$.

Recall that a poset $\mathcal{L} = (X, \leq)$ is a *lattice* if any two elements x and y of $\mathcal{L}$ have a least upper bound (denoted by $x \vee y$) and a greatest lower bound (denoted by $x \wedge y$), that is, if $(y \leq z \geq x \Rightarrow z \geq x \vee y)$ and $(y \geq t \leq x \Rightarrow t \leq x \wedge y)$ hold. A lattice is called *complete* if any (possibly infinite) subset Y of X has a least upper bound $\bigvee Y$ and a greatest lower bound $\bigwedge Y$. An example of a complete lattice is $(2^X, \subseteq)$. Clearly, the lattice operations in $(2^X, \subseteq)$ are $\cup$ and $\cap$. If $\mathcal{L}$ is a lattice then subset $\mathcal{L}'$ of $\mathcal{L}$ is a *sublattice* if $\mathcal{L}'$ is closed with respect to lattice operations $\vee$ and $\wedge$. If $\mathcal{L}'$ is closed even with respect to the infinite lattice operations $\bigvee$ and $\bigwedge$ then $\mathcal{L}'$ is a *complete sublattice* of $\mathcal{L}$. We shall also need a less restrictive definition of a substructure. A subset $\mathcal{L}'$ of $\mathcal{L}$ is a *(complete) lattice subset* of $\mathcal{L}$ if $\mathcal{L}'$ is a (complete) lattice for the restriction of $\leq$. It is clear from the definition that any (complete) sublattice is a (complete) lattice subset but the converse is not true.

Assume that $P = (X, \leq)$ is a poset on X. Clearly, $\mathcal{L}(P) \subseteq 2^X$, and we have equality if and only if P is trivial. For nontrivial posets P the following is true.

Observation 2.1. *$\mathcal{L}(P)$ is a complete sublattice of $(2^X, \subseteq)$, but not all complete sublattices of $(2^X, \subseteq)$ are of this form.*

We shall lean on Tarski's fixed point theorem, an important result on complete lattices. A mapping $f : X \to X$ on poset $P = (X, \leq)$ is *monotone* if $x \leq y$ implies $f(x) \leq f(y)$.

Theorem 2.2 (Tarski [9])**.** *If $\mathcal{L} = (X, \leq)$ is a complete lattice and $f : X \to X$ is monotone then the set of fixed points $F := \{x \in X : f(x) = x\}$ forms a nonempty complete lattice subset of $\mathcal{L}$.*

For a finite lattice $\mathcal{L}$, there is a straightforward proof of the existence of a fixed point in Theorem 2.2. Namely, if 0 denotes the smallest element of $\mathcal{L}$ then by monotonicity we have that $0 \leq f(0) \leq f(f(0)) \leq f(f(f(0))) \leq \ldots$ and by finiteness, there has to be an iterated image of 0 that is mapped to itself. It is easy to see that the fixed point of f constructed this way is a lower bound in $\mathcal{L}$ to any other fixed point of f. Similarly, if we start iterating f from 1 (that denotes the maximal element of $\mathcal{L}$) then we get a decreasing sequence $1 \geq f(1) \geq f(f(1)) \geq \ldots$ that eventually arrives to the maximal fixed point of f.

3 The economic model

In this section, we give a mathematical description of our model that is a genuine extension of that of Hatfield and Milgrom described in [8].

Let D and H be two disjoint sets of agents. We regard D as the set of doctors and H as the set of hospitals. By a contract x, we always mean an agreement between doctor $D(x) \in D$ and hospital $H(x) \in H$. Let X denote the finite set of all possible contracts in the model. For any subset X' of X, doctor d of D and hospital h of H, $X'(d) = \{x \in X' : D(x) = d\}$ and $X'(h) = \{x \in X' : H(x) = h\}$ denote all the contracts that involve doctor d and hospital h, respectively.

The main difference between our model and that of Hatfield and Milgrom in [8] is that in our model we allow certain implications between contracts. An example is that if x is a contract that assigns doctor $D(x)$ to hospital $H(x)$ for some i days a week then it is always possible to choose contract x' between $D(x)$ and $H(x)$ that describes the same job as x does except that the total weakly workload is j days for $j < i$. Or, instead of contract x doctor $D(x)$ and hospital $H(x)$ may agree on signing a contract x' for a job that needs a lower qualification than x needs. In these examples, the possibility of contract x implies the possibility of contract x' and we denote this fact by $x' \preceq x$. We assume that $P = (X, \preceq)$ is a poset on the set X of possible of contracts[1]. It is easy to check that if there is no implication between contracts whatsoever (that is, if any two contracts are incomparable in poset P, i.e. if P is trivial) then our model reduces to that of Hatfield and Milgrom.

Just like in the Hatfield-Milgrom model, hospitals and doctors have certain preferences on the contracts they participate in. This is described by choice functions as follows. Assume that $X' \subseteq X$ is a lower ideal of P. Then $C_d(X')$ denotes those contracts of $X'(d)$ that doctor d would pick from $X'(d)$ if she is allowed to choose freely. Note that though in the Hatfield-Milgrom model, choice function C_d always selects at most one contract (hence it is a so-called one-to-many matching market), we do not assume this property. For any hospital h, we have a similar choice function C_h that selects the favourite contracts of hospital h from $X'(h)$. We assume that C_d and C_h always select an antichain of P. (That is, if d can work t or t' hours ($t \leq t'$) for h according to contracts x and x' ($x \preceq x'$) then d never wants to sign both contracts x and x' and the same is true for h.) As each agent in our two-sided market has a choice function, we can define two joint choice functions: one for the doctors and one for the hospitals. Formally,

$$C_D(X') = \bigcup \{C_d(X') : d \in D\} \qquad \text{and} \qquad C_H(X') = \bigcup \{C_h(X') : h \in H\}$$

denote the doctors' and hospitals' choice function, respectively. Clearly, each choice function C we defined so far is mapping lower ideals of P into antichains of P such that $C(L) \subseteq L$ holds for any lower ideal L of P. For such a choice function $C : \mathcal{L}(P) \to \mathcal{A}(P)$, we define another choice function $C^* : \mathcal{L}(P) \to \mathcal{L}(P)$ by $C^*(L) := \mathrm{Li}(C(L))$. As there is a bijection between antichains and lower ideals of P, not only C determines C^*, but we can calculate C from C^* by $C(L) = \mathrm{Max}(C^*(L))$. Obviously, if P is trivial then $C = C^*$. We can also talk about choice functions in a more general sense. If $\mathcal{F}$ is a subset of 2^X then a choice function on $\mathcal{F}$ is a mapping $C : \mathcal{F} \to \mathcal{F}$ such that

[1] Later we shall see that all our results are true in the more general setting where we do not assume any acyclicity about implications between contracts. One can define "lower ideals" on the transitive closure of the implication digraph and these "lower ideals" form a complete sublattice of $(2^X, \subseteq)$.

$C(F) \subseteq F$ holds for any element F of $\mathcal{F}$. Note that choice functions C^*_D and C^*_H are choice functions in this latter sense, as well.

There are two important properties of choice functions that we shall assume in our model. Choice function $C : \mathcal{F} \to 2^X$ on subset $\mathcal{F}$ of 2^X is *path-independent* if

$$C(F) \subseteq F' \subseteq F \Rightarrow C(F) = C(F') \tag{3.1}$$

holds for any two members F and F' of $\mathcal{F}$. Note that in the Hatfield-Milgrom model, choice functions are defined by a strict linear order on the subsets of X such that $C(Y)$ is that subset of Y that comes first in this linear order. (We shall see an example of such a choice function in Example 3.7.) Clearly, such choice functions are path-independent by definition. Note that the "traditional" definition of path-independence is different from ours. Actually, (3.1) is weaker than that.

Observation 3.1. *Let $\mathcal{L}$ be a lattice. If for choice function $C : \mathcal{L} \to \mathcal{L}$ identity $C(A \cup B) = C(C(A) \cup C(B))$ holds for any members $A, B \in \mathcal{L}$ then* (3.1) *is also true for C.*

In Lemma 3.5 we shall see that assuming substitutability (that we define a bit later) of C then "traditional" path-independence is equivalent to (3.1). The following statement is easy to check.

Observation 3.2. *If P is a poset on X then choice function $C : \mathcal{L}(P) \to \mathcal{A}(P)$ is path-independent if and only if choice function $C^* : \mathcal{L}(P) \to \mathcal{L}(P)$ is path-independent.*

From now on, $\mathcal{L}$ denotes a complete sublattice of $(2^X, \subseteq)$. To get some intuition, the reader might simply think that $\mathcal{L} = \mathcal{L}(P)$, but our results that we claim for general complete sublattices are more general than the ones with this restriction. We do think that general complete sublattices still capture some interesting Economics models that do not fit in the poset-framework.

If $C : \mathcal{L} \to \mathcal{L}$ is a choice function then we can compare certain members of $\mathcal{L}$ with the help of C in the following way. We say that member L is *C-better* than member L' (denoted by $L' \preceq_C L$) if $C(L \cup L') = L$. We can extend this notion for antichains if choice function $C : \mathcal{L}(P) \to \mathcal{A}(P)$ maps lower ideals to antichains. This way, antichain A of P is *C-better* than $A' \in \mathcal{A}(P)$ (denoted by $A' \preceq_C A$) if $C(\mathrm{Li}(A \cup A')) = A$. Note that the same notation for lower ideals and antichains does not cause ambiguity as the range of C determines which one we talk about. Note further that $\preceq_C$ is not necessarily a partial order.

The second important property of a choice function is substitutability (or comonotonicity, as called by Fleiner in [6]) that we define here in a somewhat unusual way. A mapping $U : \mathcal{L} \to \mathcal{L}$ is called *antitone* if $U(L') \subseteq U(L)$ holds whenever $L \subseteq L'$ holds for elements L and L' of $\mathcal{L}$. Choice function $C^* : \mathcal{L} \to \mathcal{L}$ is *substitutable* if there exists an antitone mapping $U : \mathcal{L} \to \mathcal{L}$ such that $C^*(L) = L \cap U(L)$ holds for each member L of $\mathcal{L}$. A choice function $C : \mathcal{L}(P) \to \mathcal{A}(P)$ that selects an antichain is called *substitutable* if C^* is substitutable. Recall that $C^*(\cdot) = \mathrm{Li}(C(\cdot))$, and $C^* = C$ if underlying order of the poset is trivial. A choice function C in a traditional two-sided market model selects $C(Y)$ from a set Y of alternatives such that $C(Y)$ is the set of all those choices that are undominated by set Y of alternatives. The substitutability property captures the fact that a broader set of alternatives leaves less undominated choices. Or, equivalently, if the choice set is growing then the set of dominated (hence rejected) alternatives is also growing. This phenomenon is used in the definition of substitutes by Hatfield and Milgrom: elements of X are *substitutes* for choice function $C : 2^X \to 2^X$ if the set of rejected elements is a monotone mapping, that is, $R(Y) := Y \setminus C(Y) \subseteq Y' \setminus C(Y') = R(Y')$ whenever $Y \subseteq Y' \subseteq X$.

Observation 3.3. *If elements of X are substitutes for choice function $C : 2^X \to 2^X$ in the sense of Hatfield and Milgrom then C is substitutable.*

Proof. As rejection function defined by $R(Y) := Y \setminus C(Y)$ is monotone, its complement $U(Y)$ defined by $U(Y) := X \setminus R(Y)$ is antitone. As partial order of P is trivial, $U(Y)$ is a lower ideal. Observe that $Y \cap U(Y) = Y \cap (X \setminus R(Y)) = Y \setminus R(Y) = Y \setminus (Y \setminus C(Y)) = C(Y)$, hence C is indeed substitutable. □

Example 3.4. *Assume that hospital h has a linear preference order on $X(h)$ and $C_h(X')$ for $X' \subseteq X$ is the q_h best elements of $X'(h)$. (Here, there is no partial order on X, or if we insist on having one then it is trivial.) It is easy to check that C_h is path-independent and contracts in $X(h)$ are substitutes. To see that C_h is substitutable, we define $U : 2^X \to 2^X$ by $U(X')$ denoting the set of those contracts x of $X(h)$ such that $X'(h)$ contains at most $q_h - 1$ contracts that are better than x according to the preference order of h. Clearly, if $X' \subseteq X''$ then $U(X') \supseteq U(X'')$, so U is antitone. It is also clear by the definition of U that $C_h(X') = X' \cap U(X')$, that is, C_h is indeed substitutable.*

It is well-known that our definition of path-independence is equivalent to the "traditional" one for substitutable choice functions.

Lemma 3.5. *If choice function $C^* : \mathcal{L} \to \mathcal{L}$ is substitutable and path-independent then identity $C^*(A \cup B) = C^*(C^*(A) \cup C^*(B))$ holds for any members A, B of $\mathcal{L}$.*

The following theorem points out an interesting property of substitutable choice functions.

Theorem 3.6. *If choice function $C : \mathcal{L}(P) \to \mathcal{A}(P)$ is path-independent and substitutable for some poset P on a finite ground set X then $\preceq_C$ is a partial order on $\{C(L) : L \in \mathcal{L}(P)\}$, that is, on those antichains of P that are in the range of C.*

Proof. Assume that $A = C(L)$ for some $L \in \mathcal{L}(P)$. This means that $A \subseteq \mathrm{Li}(A) \subseteq L$ hence $C(\mathrm{Li}(A)) = A$ by path-independence of C. Obviously, $C(\mathrm{Li}(A \cup A)) = C(\mathrm{Li}(A)) = A$, hence $A \preceq_C A$, that is, $\preceq_C$ is reflexive. Now assume that $C(\mathrm{Li}(A')) = A'$ also holds. Clearly, if $A' \preceq_C A$ and $A \preceq_C A'$ then $A = C(\mathrm{Li}(A \cup A')) = A'$, hence $A = A'$, so $\preceq_C$ is antisymmetric. Note that we did not use the substitutable property of C so far.

To prove transitivity, assume that $C(\mathrm{Li}(A'')) = A''$ and $A \preceq_C A' \preceq_C A''$ hold. Define $L := \mathrm{Li}(A), L' := \mathrm{Li}(A')$ and $L'' := \mathrm{Li}(A'')$. From the assumption we have that $C(L \cup L') = A'$ and $C(L' \cup L'') = A''$, or, as P is finite this is equivalent to saying that $C^*(L \cup L') = L'$ and $C^*(L' \cup L'') = L''$. Since C^* is substitutable, there exists an antitone mapping U with $C^*(L) = L \cap U(L)$ for all $L \in \mathcal{L}(P)$. From the definition and the antitone property of U we get that

$$\begin{aligned} C^*(L \cup L' \cup L'') &= (L \cup L' \cup L'') \cap U(L \cup L' \cup L'') \\ &= ((L \cup L') \cap U(L \cup L' \cup L'')) \cup ((L' \cup L'') \cap U(L \cup L' \cup L'')) \\ &\subseteq ((L \cup L') \cap U(L \cup L')) \cup ((L' \cup L'') \cap U(L' \cup L'')) \\ &= C^*(L \cup L') \cup C^*(L' \cup L'') = L' \cup L''. \end{aligned}$$

This means that $C^*(L \cup L' \cup L'') \subseteq L' \cup L'' \subseteq L \cup L' \cup L''$, hence by path-independence of C^*, we have $C^*(L \cup L' \cup L'') = C^*(L' \cup L'') = L''$. So $C^*(L \cup L' \cup L'') = L'' \subseteq L \cup L'' \subseteq L \cup L' \cup L''$ and again, path-independence implies $C^*(L \cup L'') = L''$. This follows that $C(L \cup L'') = A''$, or, in other words $A \preceq_C A''$. We conclude that $\preceq_C$ is transitive, so it is indeed a partial order. □

The following example shows that our poset-based model is more general than that of Hatfield and Milgrom in [8].

Example 3.7. *Assume that we have one hospital h, two doctors d, d' and six contracts $X = \{x_3, x_4, x_5, x'_3, x'_4, x'_5\}$. Contracts x_3, x_4, x_5 and x'_3, x'_4 and x'_5 represent a $3, 4$ and 5 days job for d and d', respectively. Assume that h has the following preference order on feasible contract sets (starting from the best):*

$$\{x_4, x'_4\}, \{x_5, x'_3\}, \{x_3, x'_5\}, \{x_4, x'_3\}, \{x_3, x'_4\},$$
$$\{x_3, x'_3\}, \{x_5\}, \{x'_5\}, \{x_4\}, \{x'_4\}, \{x_3\}, \{x'_3\}.$$

So $C_h(Y)$ is that subset of Y which is the first in the above order. In particular, we have that $C_h(x_5, x_4, x_3, x'_5, x'_3) = \{x_5, x'_3\}$, so $x_4 \in R_h(x_5, x_4, x_3, x'_5, x'_3)$. Hence, if contracts were substitutes for C_h then R_h is monotone thus $x_4 \in R_h(x_5, x_4, x_3, x'_5, x'_4, x'_3) = R_h(X)$. This means that $x_4 \notin C_h(X)$ contradicting $C_h(X) = \{x_4, x'_4\}$.

The preference of h is natural and practical in the sense that h wants to employ doctors as many days as possible up to 8 days on primary criterion, as equally as possible on secondary criterion, and thirdly h prefers d to d'. Unfortunately, the framework of Hatfield and Milgrom excludes it because it is not substitutable in the framework. The reason is that several subsets of X, e.g., $\{x_5, x_4, x_3, x'_5, x'_3\}$, are inappropriate (why does not d' work 4 days even if he can work 3 or 5 days?) However, the above C_h easily fits in our framework if we define poset P by $x_3 \preceq x_4 \preceq x_5$ and $x'_3 \preceq x'_4 \preceq x'_5$. Clearly, this C_h is path-independent by definition. To see that C_h is also substitutable in the above framework, for $L \in \mathcal{L}(P)$, define $U(L) := \{x_3, x_4, x'_3, x'_4\} \cup u(L) \cup u'(L)$ where $u(L) = \emptyset$ if $x_4 \in L$ and $u(L) = \{x'_3, x'_4, x'_5\}$ if $x_4 \notin L$, and similarly $u'(L) = \emptyset$ if $x'_4 \in L$ and $u'(L) = \{x_3, x_4, x_5\}$ if $x'_4 \notin L$. As both u and u' are antitone, U is also such. Hence choice function C^* defined by $C^*(L) = L \cap U(L)$ is substitutable and one can easily check that $C^* = C^*_h$ on lower ideals of P. Hence, our model is indeed a genuine generalization of Hatfield and Milgrom's.

Note that for a substitutable choice function $C^* : \mathcal{L} \to \mathcal{L}$, there might be several antitone functions $U : \mathcal{L} \to \mathcal{L}$ such that $C^*(L) = L \cap U(L)$ holds for any member L of $\mathcal{L}$. The next statement shows that there is a canonical one among these antitone functions and this is in fact the minimal of those. (Actually, there is a maximal such U as well, but we do not need this fact.) For a choice function $C^* : \mathcal{L} \to \mathcal{L}$ define $U^* : \mathcal{L} \to \mathcal{L}$ by

$$U^*(L) := \bigcup\{Y \in \mathcal{L} : Y \subseteq C^*(L \cup Y)\} = \bigcup\{Y \in \mathcal{L} : Y \subseteq U(L \cup Y)\}. \tag{3.2}$$

Note that the second equality in (3.2) holds by the definition of substitutability, and this means that the right hand side defines the same U^* no matter which U (that defines C^*) we use.

Observation 3.8. *If choice function $C^* : \mathcal{L} \to \mathcal{L}$ is substitutable then U^* in* (3.2) *is antitone and for any member L of $\mathcal{L}$ we have $C^*(L) = L \cap U^*(L)$.*

Proof. Assume that C^* is substitutable, and $U : \mathcal{L} \to \mathcal{L}$ is an antitone function such that $C^*(L) = L \cap U(L)$ holds for any member L of $\mathcal{L}$. Define

$$U'(L) := \{Y \in \mathcal{L} : Y \subseteq C^*(L \cup Y)\} = \{Y \in \mathcal{L} : Y \subseteq U(L \cup Y)\},$$

that is, $U^*(L) = \bigcup U'(L)$. Observe that if L and L' are members of $\mathcal{L}$ with $L \subseteq L'$ and $Y \in U'(L')$ then $Y \subseteq U(L' \cup Y) \subseteq U(L \cup Y)$ by the antitone property of U. This means that $U'(L') \subseteq U'(L)$, hence $U^*(L') = \bigcup U'(L') \subseteq \bigcup U'(L) = U^*(L)$, so U^* is indeed antitone.

For the second part, observe that $C^*(L) \in U'(L)$ by definition, hence $C^*(L) \subseteq U^*(L)$ and $C^*(L) \subseteq L \cap U^*(L)$. Moreover, if $Y \in U'(L)$ then $Y \cap L \subseteq Y \subseteq U(L \cup Y) \subseteq U(L)$, hence $Y \cap L \subseteq L \cap U(L) = C^*(L)$ holds for any $Y \in U'(L)$. This follows that

$$L \cap U^*(L) = L \cap (\bigcup U'(L)) = \bigcup \{L \cap Y : Y \in U'(L)\} \subseteq C^*(L),$$

so $L \cap U^*(L) = C^*(L)$ as we claimed. □

There is another useful fact about the antitone function U^* that defines a path-independent substitutable choice function.

Observation 3.9. *If choice function $C^* : \mathcal{L} \to \mathcal{L}$ is path-independent and substitutable then $U^*(L) = U^*(C^*(L))$ holds for any member L of $\mathcal{L}$.*

Proof. It follows from the antitone property of U^* and $C^*(L) \subseteq L$ that $U^*(L) \subseteq U^*(C^*(L))$. To show the opposite inclusion, assume that $Y \in U'(C^*(L))$, that is, $Y \subseteq C^*(C^*(L) \cup Y)$ holds. We show that

$$Y \subseteq C^*(L \cup Y), \tag{3.3}$$

that is, $Y \in U'(L)$, hence $U'(C^*(L)) \subseteq U'(L)$ and $U^*(C^*(L)) = \bigcup U'(C^*(L)) \subseteq \bigcup U'(L) = U^*(L)$.

To prove (3.3), observe that

$$C^*(L \cup Y) = (L \cup Y) \cap U^*(L \cup Y) \subseteq (L \cup Y) \cap U^*(L) \subseteq (L \cap U^*(L)) \cup Y = C^*(L) \cup Y,$$

hence $C^*(L \cup Y) \subseteq C^*(L) \cup Y \subseteq L \cup Y$. Path-independence of C^* gives $C^*(L \cup Y) = C^*(C^*(L) \cup Y)$ and our assumption $Y \subseteq C^*(C^*(L) \cup Y)$ proves (3.3) that concludes the proof. □

At this point, we can generalize the notion of stability to our framework. Let D and H be the sets of doctors and hospitals, respectively, and let X denote the set of possible contracts between doctors and hospitals. Assume that we are given a (complete) sublattice $\mathcal{L}$ of $(2^X, \subseteq)$ (for example as the set of lower ideals of a poset P on X), and let $C^*_D = (C_D)^*$ and $C^*_H = (C_H)^*$ denote the joint choice functions of the doctors and of the hospitals, respectively. For members L_1 and L_2 of $\mathcal{L}$, pair (L_1, L_2) is called a *stable pair* if

$$U^*_D(L_1) = L_2 \qquad \text{and} \qquad U^*_H(L_2) = L_1 \tag{3.4}$$

hold. If $\mathcal{L} = \mathcal{L}(P)$ for some poset P on X then antichain A of P is called *stable* if

$$U^*_D(\mathrm{Li}(A)) \cap U^*_H(\mathrm{Li}(A)) = \mathrm{Li}(A). \tag{3.5}$$

Later we shall see that stable pairs are closely related to stable antichains. These latter represent the solution concept of two-sided market situations in our model. What does it mean that an antichain is stable? The first requirement is that if both doctors and hospitals select freely from those contracts that antichain A represents or implies then doctors select $C_D(\mathrm{Li}(A)) = \mathrm{Max}(\mathrm{Li}(A) \cap U^*_D(\mathrm{Li}(A)) = \mathrm{Max}(\mathrm{Li}(A)) = A$, as $\mathrm{Li}(A) \subseteq U^*_D(\mathrm{Li}(A))$. Similarly, it follows that $C_H(\mathrm{Li}(A)) = A$, so hospitals also pick the same antichain A of contracts. Moreover, if there are some further choices available that are represented by antichain Y and both the doctors and the hospitals are happy to pick those (formally, if $Y \subseteq C_D(\mathrm{Li}(A) \cup \mathrm{Li}(Y))$ and $Y \subseteq C_H(\mathrm{Li}(A) \cup \mathrm{Li}(Y))$) then

$$\begin{aligned}
\mathrm{Li}(Y) &\subseteq C^*_D(\mathrm{Li}(A) \cup \mathrm{Li}(Y)) \cap C^*_H(\mathrm{Li}(A) \cup \mathrm{Li}(Y)) \\
&\subseteq U^*_D(\mathrm{Li}(A) \cup \mathrm{Li}(Y)) \cap U^*_H(\mathrm{Li}(A) \cup \mathrm{Li}(Y)) \\
&\subseteq U^*_D(\mathrm{Li}(A)) \cap U^*_H(\mathrm{Li}(A)) = \mathrm{Li}(A).
\end{aligned}$$

So $Y \subseteq C_D(\mathrm{Li}(A) \cup \mathrm{Li}(Y)) = C_D(\mathrm{Li}(A)) = A$. This means that we cannot add further choices to A such that both the doctors and the hospitals will select them.

In the Hatfield-Milgrom model, $A \subseteq X$ is a *stable allocation* if $C_D(A) = C_H(A) = A$ and there exists no hospital h and set of contracts $X'' \neq C_h(A)$ with $X'' = C_h(A \cup X'') \subseteq C_D(A \cup X'')$. Assume that A is a feasible allocation, that is, $C_D(A) = C_H(A) = A$ and $A' \not\subseteq A$ is a blocking set: $A' \subseteq C_D(A \cup A')$ and $A' \subseteq C_H(A \cup A')$. This means that there is a hospital h that picks a different assignment from A and from $A \cup A'$. Let $X'' = C_h(A \cup A')$ denote the choice of this hospital h. Since $X'' = C_h(A \cup A') \subseteq A \cup X'' \subseteq A \cup A'$, we have $X'' = C_h(A \cup X'')$. Because of $X'' \subseteq C_D(A \cup A')$ and $A = C_D(A)$, each doctor in $\cup_{x \in X''} D(x)$ has the same choice as in $A \cup X''$, that is, $X'' \subseteq C_D(A \cup X'')$. So X'' blocks A in the Hatfield-Milgrom sense. This proves that a stable antichain of contracts in our framework with a trivial underlying partial order is a stable allocation in the Hatfield-Milgrom framework. It is not difficult to see that the other direction is also true: any stable allocation in the Hatfield-Milgrom framework is a stable antichain.

Example 3.10. *(Example 3.7 continued) Assume that both doctors d and d' want to work as many days as possible, and that C_d and $C_{d'}$ are defined according to the preference. Obviously, C_d and $C_{d'}$ are substitutable and path-independent. Then $U_H^*(\mathrm{Li}(A)) = U_H^*(L) = L_1$ and $U_D^*(\mathrm{Li}(A)) = U_D^*(L) = L_2$ hold for*

$$A = \{x_4, x_4'\}, \quad L = \mathrm{Li}(A) = \{x_3, x_4, x_3', x_4'\},$$
$$L_1 = \{x_3, x_4, x_3', x_4'\}, \quad L_2 = \{x_3, x_4, x_5, x_3', x_4', x_5'\}.$$

As $U_D^*(L_1) = L_2$, $U_H^*(L_2) = L_1$ and $L_1 \cap L_2 = L$, it follows that (L_1, L_2) is a stable pair and A is a stable antichain.

4 Main result

In this section, we prove our main results. Let X be a ground set and define partial order $\sqsubseteq$ on pairs of subsets of X by $(A, B) \sqsubseteq (A', B')$ if $A \subseteq A'$ and $B \supseteq B'$ holds. It is clear that for any sublattice $\mathcal{L}$ of $(2^X, \subseteq)$, $\sqsubseteq$ defines a lattice on $\mathcal{L} \times \mathcal{L}$ with lattice operations $(A, B) \sqcap (A', B') = (A \cap A', B \cup B')$ and $(A, B) \sqcup (A', B') = (A \cup A', B \cap B')$. The following theorem generalizes some results by Hatfield and Milgrom in [8].

Theorem 4.1. *Let X be a set of possible contracts between set D of doctors and H of hospitals and let $\mathcal{L}$ be a complete sublattice of $(2^X, \subseteq)$. Assume that joint choice functions C_D^* of doctors and C_H^* of hospitals are substitutable. Then stable pairs form a nonempty complete lattice subset of $(\mathcal{L} \times \mathcal{L}, \sqsubseteq)$. In particular, there does exist a stable pair and there is a greatest and a lowest such pair.*

Moreover, if $\mathcal{L} = \mathcal{L}(P)$ is the lattice of lower ideals of some poset $P = (X, \preceq)$ and both joint choice functions C_D and C_H are substitutable and path-independent then $\preceq_{C_D}$ and $\preceq_{C_H}$ are opposite partial orders on stable antichains and both of them define a lattice.

Note that the 2nd part of Theorem 4.1 generalizes the following well-known result of Blair on the lattice structure of many-to-many stable matchings [4].

Corollary 4.2 (Blair [4])**.** *If both doctors' and hospitals' choice functions are substitutable and path-independent and moreover no two different contract is possible between the same doctor and* *hospital then $\preceq_{C_D}$ and $\preceq_{C_H}$ are opposite partial orders on stable assignments and both of them define a lattice.*

Proof of Theorem 4.1: Define mapping $f : \mathcal{L} \times \mathcal{L} \to \mathcal{L} \times \mathcal{L}$ by $f(L_1, L_2) := (U^*_H(L_2), U^*_D(L_1))$ where U^*_D and U^*_H are the functions defined according to (3.2) from C^*_D and C^*_H. By definition, a pair (L_1, L_2) is stable if and only if $f(L_1, L_2) = (L_1, L_2)$. Assume that $(L_1, L_2) \sqsubseteq (L'_1, L'_2)$, i.e. $L_1 \subseteq L'_1$ and $L_2 \supseteq L'_2$. Functions U^*_D and U^*_H are antitone by Observation 3.8, hence $U^*_H(L_2) \subseteq U^*_H(L'_2)$ and $U^*_D(L_1) \supseteq U^*_D(L'_1)$, that is, $f(L_1, L_2) \sqsubseteq f(L'_1, L'_2)$ holds. This means that f is monotone on complete lattice $(\mathcal{L} \times \mathcal{L}, \sqsubseteq)$, hence its fixed points form a nonempty complete lattice subset of $(\mathcal{L} \times \mathcal{L}, \sqsubseteq)$ by Theorem 2.2 of Tarski. This proves the first part of Theorem 4.1.

To show the second part of Theorem 4.1 in the special case where $\mathcal{L} = \mathcal{L}(P)$ and choice functions are path-independent, we prove that there is a natural bijection between stable pairs and stable antichains in such a way that the partial order on stable antichains induced by the natural bijection and partial order $\sqsubseteq$ coincides with both $\preceq_{C_D}$ and $\succeq_{C_H}$ (the opposite of $\preceq_{C_H}$). As soon as we do so, the second part of Theorem 4.1 immediately follows from the first one.

So assume that $\mathcal{L} = \mathcal{L}(P)$ and choice functions of doctors' and hospitals' are substitutable and path-independent. Let (L_1, L_2) be a stable pair of lower ideals of P. Observe that

$$C^*_D(L_1) = L_1 \cap U^*_D(L_1) = L_1 \cap L_2 = U^*_H(L_2) \cap L_2 = C^*_H(L_2)$$

so

$$A(L_1, L_2) := C_D(L_1) = C_H(L_2) = \mathrm{Max}(L) \qquad (4.6)$$

is an antichain, where $L := L_1 \cap L_2$. From (4.6) and Observation 3.9 it follows that $U^*_D(\mathrm{Li}(A(L_1, L_2))) = U^*_D(C^*_D(L_1)) = U^*_D(L_1) = L_2$ and similarly, $U^*_H(\mathrm{Li}(A(L_1, L_2))) = U^*_H(C^*_H(L_2)) = U^*_H(L_2) = L_1$, hence $U^*_D(\mathrm{Li}(A(L_1, L_2))) \cap U^*_H(\mathrm{Li}(A(L_1, L_2))) = L_1 \cap L_2 = \mathrm{Li}(A(L_1, L_2))$, so $A(L_1, L_2)$ is indeed stable.

Now assume that A is a stable antichain and define $L := \mathrm{Li}(A)$, $L_1 := U^*_H(L)$ and $L_2 := U^*_D(L)$. We show that (L_1, L_2) is a stable pair such that $A = A(L_1, L_2)$. By stability of antichain A, we have that $L = U^*_D(L) \cap U^*_H(L)$. This means that $L \subseteq U^*_D(L)$, hence $L = L \cap U^*_D(L) = C^*_D(L)$ and $C^*_D(L_1) = L_1 \cap U^*_D(L_1) \subseteq L_1 \cap U^*_D(L) = L_1 \cap L_2 = L$. This means that $C^*_D(L_1) \subseteq L \subseteq L_1$, and path-independence of C^*_D implies that $C^*_D(L_1) = C^*_D(L) = L$. Observation 3.9 yields that $U^*_D(L_1) = U^*_D(L) = L_2$. A similar argument shows that $U^*_H(L_2) = L_1$. We got that (L_1, L_2) is indeed a stable pair, and moreover $A(L_1, L_2) = \mathrm{Max}(L_1 \cap L_2) = \mathrm{Max}(U^*_H(L) \cap U^*_D(L)) = \mathrm{Max}(L) = A$.

To prove the existence of a natural bijection between stable pairs and antichains, we only have to show that the stable pair we construct from $A(L_1, L_2)$ according to the above paragraph is (L_1, L_2) for any stable pair (L_1, L_2). Actually, this stable pair is (L'_1, L'_2) for $L'_1 = U^*_H(\mathrm{Li}(A(L_1, L_2)))$ and $L'_2 = U^*_D(\mathrm{Li}(A(L_1, L_2)))$. We have seen that $C^*_D(L_1) = \mathrm{Li}(A(L_1, L_2)) = C^*_H(L_2)$, so Observation 3.9 implies that $L_1 = U^*_H(L_2) = U^*_H(C^*_H(L_2)) = U^*_H(\mathrm{Li}(A(L_1, L_2))) = L'_1$ and $L_2 = U^*_D(L_1) = U^*_D(C^*_D(L_1)) = U^*_D(\mathrm{Li}(A(L_1, L_2))) = L'_2$. This shows that there is indeed a natural bijection between stable antichains and stable pairs.

To finish the proof by justifying the generalization of Theorem 4.2 by Blair, we show that $\sqsubseteq$ and the natural bijection induces a partial order that coincides with $\preceq_{C_D}$ and $\succeq_{C_H}$ on stable antichains. So assume now that (L_1, L_2) and (L'_1, L'_2) are stable pairs that correspond to stable antichains A and A', respectively. This means that $L_1 = U^*_H(\mathrm{Li}(A)), L_2 = U^*_D(\mathrm{Li}(A))$ and $L'_1 =$

$U_H^*(\mathrm{Li}(A')), L_2' = U_D^*(\mathrm{Li}(A'))$ on one hand and $\mathrm{Li}(A) = L_1 \cap L_2 = C_D^*(L_1) = C_H^*(L_2), \mathrm{Li}(A') = L_1' \cap L_2' = C_D^*(L_1') = C_H^*(L_2')$ on the other hand.

Assume first that $(L_1, L_2) \sqsubseteq (L_1', L_2')$, i.e. $L_1 \subseteq L_1'$ and $L_2 \supseteq L_2'$. Consequently, (using Lemma 3.5)

$$C_D^*(\mathrm{Li}(A) \cup \mathrm{Li}(A')) = C_D^*(C_D^*(L_1) \cup C_D^*(L_1')) = C_D^*(L_1 \cup L_1') = C_D^*(L_1') = \mathrm{Li}(A')$$

and similarly

$$C_H^*(\mathrm{Li}(A) \cup \mathrm{Li}(A')) = C_H^*(C_H^*(L_2) \cup C_H^*(L_2')) = C_H^*(L_2 \cup L_2') = C_H^*(L_2) = \mathrm{Li}(A).$$

In other words, $C_D(A \cup A') = A'$ and $C_H(A \cup A') = A$, hence $A \preceq_{C_D} A'$ and $A \succeq_{C_H} A'$.

Now suppose that $A \preceq_{C_D} A'$, that is, $A' = C_D(A \cup A')$. This follows that $\mathrm{Li}(A') = C_D^*(\mathrm{Li}(A) \cup \mathrm{Li}(A'))$. Observation 3.9 and the antitone property of U^* yields that

$$L_2' = U_D^*(\mathrm{Li}(A')) = U_D^*(\mathrm{Li}(A) \cup \mathrm{Li}(A')) \subseteq U_D^*(\mathrm{Li}(A)) = L_2. \tag{4.7}$$

As (L_1, L_2) and (L_1', L_2') are stable pairs, the antitone property of U_H^* implies that

$$L_1 = U_H^*(L_2) \subseteq U_H^*(L_2') = L_1'. \tag{4.8}$$

From (4.7) and (4.8), $(L_1, L_2) \sqsubseteq (L_1', L_2')$ follows. A similar argument justifies that $(L_1', L_2') \sqsubseteq (L_1, L_2)$ holds whenever $A \preceq_{C_H} A'$ and this concludes our proof. □

Theorem 4.1 points out a close connection between the notion of stability and fixed points of a monotone function that always exist by Theorem 2.2 of Tarski. We have already indicated that one can construct the maximal and minimal fixed points by iterating the monotone function starting from the maximum or from the minimum element of the underlying lattice, respectively. Probably, it was Fleiner in [5] who first pointed out that the well-known proposal algorithm of Gale and Shapley that finds a man-optimal stable marriage scheme can be regarded as an iteration of a certain monotone mapping. Later, the same observation was made by Hatfield and Milgrom for a special case of Fleiner's model. Actually, the same connection also holds in our present settings that generalize both Fleiner's and the Hatfield-Milgrom's framework. The generalized Gale-Shapley algorithm for finding a stable antichain works as follows.

Let us denote by 0 and 1 the minimal and maximal elements of $\mathcal{L}$, respectively. It is straightforward to check that mapping $f : \mathcal{L} \times \mathcal{L} \to \mathcal{L} \times \mathcal{L}$ on lattice $(\mathcal{L} \times \mathcal{L}, \sqsubseteq)$ defined by $f(L_1, L_2) := (U_H^*(L_2), U_D^*(L_1))$ is monotone by the antitone property of U_D^* and U_H^*. Clearly, $(L_1, L_2) = (U_H^*(L_2), U_D^*(L_1))$ is a fixed point if and only if (L_1, L_2) is a stable pair, and in case of $\mathcal{L} = \mathcal{L}(P)$, it is equivalent to $\mathrm{Max}(L_1 \cap L_2)$ is a stable antichain. So to find the maximal (doctor-optimal) stable antichain, we only have to start to iterate f from the maximal element of $\mathcal{L} \times \mathcal{L}$ to get a $\sqsubseteq$-decreasing sequence $(1, 0) \sqsupseteq f(1, 0) \sqsupseteq f(f(1, 0)) \sqsupseteq \ldots$. Hence after at most 2ℓ iterations (where ℓ denotes the hight (the length of the longest chain) of poset $\mathcal{L}$) we arrive to a fixed point and find the doctor-optimal stable antichain A_D. If we start the iteration from the bottom of the lattice $\mathcal{L} \times \mathcal{L}$ then the $\sqsubseteq$-minimal fixed point at the "end" of increasing sequence $(0, 1) \sqsubseteq f(0, 1) \sqsubseteq f(f(0, 1)) \sqsubseteq \ldots$ represents the hospital-optimal stable antichain A_H. According to Theorem 4.1 for any stable antichain A we have $A_H \preceq_{C_D} A \preceq_{C_D} A_D$ and $A_D \preceq_{C_H} A \preceq_{C_H} A_H$, so for example $C_D(\mathrm{Li}(A \cup A_D)) = A_D$ and $C_H(\mathrm{Li}(A \cup A_H)) = A_H$. This means that if doctors are offered all the choices that the contracts in some stable antichain represent or imply then from this choice set doctors pick contracts of A_D, and a similar property is true for the hospitals with respect to A_H.

It seems that no one observed so far that the monotone function iteration is more powerful than the Gale-Shapley algorithm itself that (in its original form) finds the man-optimal and (with an exchanges of roles) the woman-optimal stable matchings. The iteration method can be used to calculate the lattice operations on the fixed points of the monotone function. Consequently, we can construct the $\preceq_{C_D}$-least and $\preceq_{C_D}$-greatest stable antichains that are the least upper and greatest lower bounds of any given nonempty set of stable antichains. This works as follows: take stable antichains $A_1, A_2, \ldots, A_k$ that correspond to stable pairs $(L_1, K_1), (L_2, K_2), \ldots, (L_k, K_k)$ and define $L := \bigcup_{i=1}^k L_i$ and $K := \bigcap_{i=1}^k K_i$. By the antitone property of U_H^* and U_D^* we get

$$U_H^*(K) = U_H^*(\bigcap_{i=1}^k K_i) \supset \bigcup_{i=1}^k U_H^*(K_i) = \bigcup_{i=1}^k L_i = L \text{ and}$$
$$U_D^*(L) = U_D^*(\bigcup_{i=1}^k L_i) \subset \bigcap_{i=1}^k U_D^*(L_i) = \bigcap_{i=1}^k K_i = K,$$

so $(L, K) \sqsubseteq (U_H^*(K), U_D^*(L)) = f(L, K)$. Now monotonicity of f gives that $(L, K) \sqsubseteq f(L, K) \sqsubseteq f(f(L, K)) \sqsubseteq f(f(f(L, K)))\ldots$ and at most 2ℓ iterations of f this $\sqsubseteq$-increasing sequence arrives to the $\sqsubseteq$-least stable pair that is $\sqsubseteq$-greater than (L, K). This fixed point clearly corresponds to the stable antichain that is the least upper bound of stable antichains $A_1, A_2, \ldots, A_k$. A similar argument shows that for $L' := \bigcap_{i=1}^k L_i$ and $K' := \bigcup_{i=1}^k K_i$ we have $(L', K') \sqsupseteq f(L', K') \sqsupseteq f(f(L', K')) \sqsupseteq f(f(f(L', K'))) \sqsupseteq \ldots$ and the "end" of this decreasing sequence corresponds to the meet of stable antichains $A_1, A_2, \ldots, A_k$.

References

[1] Adachi, H.: On a characterization of stable matchings. Econom. Lett. **68**, 43–49 (2000)

[2] Alkan, A.: A class of multipartner matching markets with a strong lattice structure. Econom. Theory **19**, 737–746 (2002)

[3] Alkan, A., Gale, D.: Stable schedule matching under revealed preference. J. Econom. Theory **112**, 289–306 (2003)

[4] Blair, C.: The lattice structure of the set of stable matchings with multiple partners. Math. Oper. Res. **13**, 619–628 (1988)

[5] Fleiner, T.: Stable and crossing structures. Ph.D. thesis, Tecnische Universiteit Eindhoven (2000)

[6] Fleiner, T.: A fixed point approach to stable matching and some applications. Math. Oper. Res. **38**, 103–126 (2003)

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

[8] Hatfield, J.W., Milgrom, P.R.: Matching with contracts. Amer. Econom. Rev. **95**, 913–935 (2005)

[9] Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific J. Math. **5**, 285–309 (1955)

Weight-Maximal Matchings[*]

Chien-Chung Huang
villars@informatik.hu-berlin.de
Humboldt-Universität zu Berlin
Berlin, Germany

Telikepalli Kavitha
kavitha@tcs.tifr.res.in
Tata Institute of Fundamental Research
Mumbai, India

Abstract

We introduce the following optimization problem, called the WEIGHT-MAXIMAL MATCHING problem. The input is a bipartite graph $G = (A \cup B, E)$ along with a family of weight functions $\langle w_1, \ldots, w_r \rangle$ where $w_i : E \to \{0, 1, \ldots, W\}$, for $1 \le i \le r$. Each weight function captures a particular attribute (say, size, reward, compatibility, etc) and we have the following order among these functions: w_1 measures the most important attribute, the next most important attribute is measured by w_2, and so on, and at the end comes w_r. Our problem is to compute a matching M in G that captures the most important attribute optimally, subject to this constraint, M captures the second most important attribute optimally, and so on, and finally, subject to the first $r-1$ constraints, M captures the r-th attribute optimally.

This problem is an abstraction of several known variants of matching problems with preferences: fair matching, rank-maximal matching, and maximum cardinality rank-maximal matching. We present an iterative combinatorial algorithm whose running time is at least as good as or better than the running times of algorithms to compute fair/rank-maximal/maximum cardinality rank-maximal matchings. In addition, we show how to solve a generalized version of the weighted vertex cover problem in bipartite graphs using a single-source shortest paths computation – this could be of independent interest.

1 Introduction

Let $G = (A \cup B, E)$ be a bipartite graph, where each edge has several attributes associated with it. For instance, these attributes might include quantities such as the weight of this edge, reward associated with this edge, the compatibility of the endpoints of the edge, and so on. These attributes are captured via edge weight functions $w_1, \ldots, w_r$, where $w_i : E \to \{0, 1, \ldots, W\}$, for $1 \le i \le r$. Thus $w_i(e)$ is a measure of how well edge e fares with respect to the i-th attribute. We assume that W (the largest edge weight) is a constant. These attributes are ordered in terms of importance as follows: attribute 1 (measured by w_1) is the most important, next comes attribute 2 (measured by w_2), and so on, and at the very end comes attribute r (measured by w_r).

A matching is a collection of edges, no two of which share an endpoint. The goal is to compute a matching M in G such that M is optimal with respect to these r attributes as follows: *M is a maximum weight matching with respect to attribute 1, subject to this constraint, M is a maximum weight matching wrt attribute 2, and so on, and finally subject to the first $r-1$ constraints, M is a maximum weight matching wrt attribute r.* We call such a matching a WEIGHT-MAXIMAL MATCHING. The following definition will be useful.

Definition 1.1 *For any matching M in G, let* signature(M) *be the r-tuple $(w_1(M), w_2(M), \ldots, w_r(M))$, where for $1 \le i \le r$, $w_i(M)$ is the weight of matching M under weight function w_i, i.e., $w_i(M) = \sum_{e \in M} w_i(e)$.*

[*] This work was supported by IMPECS (the Indo-German Max Planck Center for Computer Science).

The matching whose signature is lexicographically the best is the weight-maximal matching. So if M is such a matching, then we have $\mathsf{signature}(M) \succeq \mathsf{signature}(M')$, for all matchings M', where $\succeq$ is the lexicographic order on signatures. The weight-maximal problem is an abstraction of several matching problems whose objectives can be reformulated in this setting. Thus, if we can solve the abstract version of these problems, then we effectively propose a unified scheme for solving many related problems. We now give concrete examples of weight-maximal matchings.

Matching with Preferences.

Let $G = (A \cup B, E)$ be a bipartite graph where each vertex in $A \cup B$ ranks its neighbors in an order of preference, possibly involving ties. The set A can be viewed as a set of applicants and the set B can be viewed as a set of jobs, where each applicant a has a ranking (with ties allowed) over the jobs that a is interested in and similarly, each job b has a ranking (with ties allowed) over the applicants that b is interested in. Thus, each edge $e = (a, b) \in E$ is associated with two values, one is the ranking of a for b and the other is the ranking of b for a. This is the same as an instance of the stable marriage problem with incomplete lists and ties [6, 7].

The focus in the stable marriage problem is to find matchings that are *stable*. However, there are many applications where stability is not a proper objective. Rather we seek matchings that satisfy several optimality criteria. For instance, in assigning summer internships to interns, there is a global authority who seeks to maximize the size of the resulting matching, subject to this constraint, then may seek to maximize the "value" of the matching in terms of matching vertices to highly-valued neighbors, then may seek to maximize the number of women matched, and so on. Another example is in assigning counselors to students, where the global authority again seeks a maximum cardinality matching that minimizes the number of "incompatible" pairs. We summarize below three such optimality criteria that have been studied so far. Let r be the largest or worst rank used by any vertex.

Definition 1.2 *A matching M in G is said to be*

- rank-maximal*: if M matches the maximum number of vertices to rank 1 neighbors, and subject to that, M matches the maximum number of vertices to rank 2 neighbors, and so on.*
- maximum cardinality rank-maximal*: if M is a maximum cardinality matching, and subject to that, M is rank-maximal.*
- fair*: if M leaves the minimum number of vertices unmatched*[1]*, and subject to that, M matches the minimum number of vertices to rank r neighbors, and subject to that, M matches the minimum number of vertices to rank $(r-1)$ neighbors, and so on.*

Rank-maximal/fair/maximum cardinality rank-maximal matchings have several applications: for instance, when assigning counselors to students, a fair matching fits the bill of the desired matching since the matching is of maximum cardinality, and subject to this, as few people as possible are matched to their rank r neighbors, subject to this, as few people as possible are matched to their rank $(r-1)$ neighbors, and so on. Similarly, when assigning summer internships to interns, a maximum cardinality rank-maximal matching fits the bill of a desired matching since here we seek a maximum cardinality matching that matches as many vertices as possible to their topmost choice, subject to this, matches as many vertices as possible to their second best choice, and so on. Also, here one may seek a

[1] Note this is the same as saying M has to be of maximum cardinality.

maximum cardinality rank-maximal matching that satisfies additional properties, for instance, one that maximizes the number of women being matched.

These three matching problems with preferences have been studied when vertices on only one side of the bipartite graph have preferences [12, 13, 14, 16, 17]. In fact, Irving [12] originally referred to rank-maximal matchings with one-sided preferences as *weight-maximal* matchings. It is rather surprising that the above problems in the context of 2-sided preferences have never been addressed so far (though the technique of Mehlhorn and Michail [16] for 1-sided preferences can be generalized to the 2-sided preferences setting—see the discussion below).

It is easy to see that these three matching problems are special cases of weight-maximal matchings. Let the total number of ranks used be r. Note that rank 1 is the top choice while rank r is the worst choice.

- *Rank-Maximal Matching*: create r weight functions w_1 to w_r. For any edge $e = (a, b)$, for $1 \leq i \leq r$, define $w_i(e)$ as follows: it is 2 if both a and b rank each other as rank i neighbors, it is 1 if exactly one of $\{a, b\}$ ranks the other as a rank i neighbor, otherwise it is 0.

- *Maximum Cardinality Rank-Maximal Matching*: create weight functions w_1 up to w_{r+1}. The function w_1 is for the "cardinality" of the matching: thus $w_1(e) = 1$ for all $e \in E$. For $2 \leq i \leq r + 1$, define $w_i(e)$ as follows: it is 2 if both a and b rank each other as rank $(i - 1)$ neighbors, it is 1 if exactly one of $\{a, b\}$ ranks the other as a rank $(i - 1)$ neighbor, otherwise it is 0.

- *Fair Matching*: create weight functions w_1 up to w_r. As in the preceding case, the function w_1 is for the cardinality of the matching: thus $w_1(e) = 1$ for all $e \in E$. For any $2 \leq j \leq r$, define $w_j(e)$ as follows: it is 2 if both a and b rank each other as rank $\leq r - j + 1$ neighbors, it is 1 if exactly one of $\{a, b\}$ ranks the other as a rank $\leq r - j + 1$ neighbor, otherwise it is 0.

Our Results and Technique.

Though the weight-maximal matching problem specifies a family of weight functions $\langle w_1, \ldots, w_r \rangle$, in order to compute a weight-maximal matching, we can effectively reduce all these weights into a *single* weight function as follows. We assign a weight of $\sum_{j=1}^{r} w_j(e) n^{2(r-j)}$ to the edge e. It is not difficult to see that the maximum weight matching in the reduced problem is also the weight-maximal matching in the original problem and vice versa.

However the above brute-force approach can be expensive even if we use the fastest maximum-weight bipartite matching algorithms [1, 2, 4, 5]. The running time will be $O(rmn)$ or $O(r^2\sqrt{n}m \log n)$ where $n = |A \cup B|$ and $m = |E|$. (These complexities follow from the customary assumption that each arithmetic operation has constant cost on numbers of magnitude $O(n)$.)

Mehlhorn and Michail [16] showed that with the same reduction, by a more sophisticated implementation of the Gabow-Tarjan scaling algorithm [5], a weight-maximal matching can be found in $O(r\sqrt{n}m \log n)$ time.[2] Their algorithm maintains "reduced" edge costs and it is known that if the reduced cost of any edge becomes $8n$ or more, then such an edge may as well be deleted. Thus the reduced costs are always $O(n)$ and the Gabow-Tarjan scaling algorithm takes $O(r\sqrt{n}m \log n)$ time here. Rather than using scaling, here we present an iterative combinatorial algorithm for the weight-maximal matching problem. We show the following result.

[2] The Mehlhorn-Michail algorithm was originally designed for matching problems with one-sided preferences. The complexity of $O(r\sqrt{n}m \log n)$ is for fair/rank-maximal/maximum cardinality rank-maximal matchings. But their algorithm can be generalized to solve our weight-maximal matching problem with the same time complexity.

Theorem 1.3 *A weight-maximal matching M in $G = (A \cup B, E)$ can be computed in $O(r^* \sqrt{n}m \log n)$ time, or in $\tilde{O}(r^* n^\omega)$ time with high probability, where $r^* \leq r$ is the largest index such that $w_{r^*}(M)$ is strictly positive, $n = |A \cup B|$, $m = |E|$, and $\omega \approx 2.37$ is the exponent of matrix multiplication.*

Thus our running times improve the running time of the Mehlhorn-Michail algorithm, in particular when the graph is relatively dense, i.e., when $m = \Omega(n^{1.88})$. We note that as the Mehlhorn-Michail algorithm is based on the Gabow-Tarjan scaling algorithm, it seems unlikely that one can replace the term $\sqrt{n}m$ with n^ω in its running time—we are not aware of such scaling algorithms.

Note that for matching problems with preferences, the above complexity implies that we can find a rank-maximal or maximum cardinality rank-maximal matching in $O(r^* \sqrt{n}m \log n)$ time, or in $\tilde{O}(r^* n^\omega)$ with high probability, where r^* is the worst rank used in the solution. For fair matchings, we achieve the same running time with some preprocessing. Note that for one-sided preferences, a rank-maximal matching can be computed in $O(\min(r^* \sqrt{n}m, mn))$ time [13].

Theorem 1.4 *A rank-maximal/maximum cardinality rank-maximal/fair matching in $G = (A \cup B, E)$ with 2-sided preference lists, can be computed in $O(r^* \sqrt{n}m \log n)$ time, or in $\tilde{O}(r^* n^\omega)$ time with high probability, where $r^* \leq r$ is the worst rank used in the solution, $n = |A \cup B|$, and $m = |E|$.*

We also show the following structural result.

Theorem 1.5 *When our algorithm terminates, we show a subgraph G' of G and a subset $V' \subseteq A \cup B$ such that M is a weight-maximal matching in G if and only if M is a matching in G' that matches all vertices in V'.*

Our algorithm is based on linear programming duality and is combinatorial in nature, i.e., we do not rely on any scaling used in the Gabow-Tarjan algorithm. Our algorithm is iterative and in each iteration, we solve a variant of the maximum weight matching problem and its dual. The dual problem that we solve in the i-th iteration is the following.

Generalized minimum weighted vertex cover problem. Let $G_i = (A \cup B, E_i)$ be a bipartite graph with edge weights given by $w_i : E \to \{0, 1, \ldots, W\}$. Let $K_{i-1} \subseteq A \cup B$. Find a cover $\{y^i_u\}_{u \in A \cup B}$ so that $\sum_{u \in A \cup B} y^i_u$ is minimized subject to the following conditions: (1) for each $e = (a, b) \in E_i$, $y^i_a + y^i_b \geq w_i(e)$, and (2) $y^i_u \geq 0$ if $u \notin K_{i-1}$.

When $K_{i-1} = \emptyset$, then the above problem reduces to the original weighted vertex cover problem. We show that the generalized minimum weighted vertex cover problem can be solved using a single-source shortest paths subroutine in directed graphs, by a non-trivial extension of a technique of Iri [11].

Organization of the paper. We discuss preliminaries in Section 2. Section 3 has the algorithm for finding a weight-maximal matching. Section 4 has an efficient algorithm for solving the generalized minimum weighted vertex cover problem.

2 Preliminaries

Let OPT denote a weight-maximal matching. So $\mathsf{signature}(\mathsf{OPT}) \succeq \mathsf{signature}(M')$ for all matchings M' in G, where $\succeq$ is the lexicographic order on signatures. For any matching M and $1 \leq j \leq r$, let $\mathsf{signature}_j(M)$ denote the j-tuple obtained by truncating $\mathsf{signature}(M)$ to its first j coordinates.

Definition 2.1 *A matching M is j-optimal if $\mathsf{signature}_j(M) = \mathsf{signature}_j(\mathsf{OPT})$.*

Our algorithm is iterative and it computes a $(j-1)$-optimal matching M_j in the j-th iteration. Our algorithm runs for $r^* + 1$ iterations, where $r^* \le r$ is the largest index such that the r^*-th coordinate of $\mathsf{signature}(\mathsf{OPT})$ is positive. Thus if $w_r(\mathsf{OPT}) > 0$, then $r^* = r$, otherwise $r^* < r$.

For every $1 \le j \le r^* + 1$, our algorithm maintains the $(j-1)$-optimality of M_j as follows. Let G_j be the graph in the j-th iteration. We maintain a *critical* set $K_{j-1} \subseteq A \cup B$ at the end of the $(j-1)$-st iteration and M_j will be a maximum w_j-weight matching in G_j under the constraint that all vertices of K_{j-1} have to be matched in M_j. We will show that this ensures the $(j-1)$-optimality of M_j.

The problem of computing M_j can be expressed as a linear program (rather than an integer program) as the constraint matrix is totally unimodular and hence the corresponding polytope is integral. The problem of computing M_j will be referred to as the primal program of the j-th iteration. This linear program and its dual are given below. (Let $\delta(v)$ be the set of edges incident on vertex v.)

$$\begin{aligned} &\max \sum_{e \in E} w_j(e) x_e^j \\ &\sum_{e \in \delta(v)} x_e^j \le 1 \qquad \forall v \in A \cup B \\ &\sum_{e \in \delta(v)} x_e^j = 1 \qquad \forall v \in K_{j-1} \\ &x_e^j \ge 0 \qquad \forall e \text{ in } G_j. \end{aligned}$$

$$\begin{aligned} &\min \sum_{v \in V} y_v^j \\ &y_a^j + y_b^j \ge w_j(e) \qquad \forall e = (a,b) \text{ in } G_j \\ &y_v^j \ge 0 \qquad \forall v \in (A \cup B) \setminus K_{j-1}. \end{aligned}$$

Proposition 2.2 *M_j and y^j are the optimal solutions to the primal and dual programs respectively, iff the following hold:*

1. *if u is unmatched in M_j (thus u has to be outside K_{j-1}), then $\mathsf{y}_u^j = 0$;*
2. *if $e = (u,v) \in M_j$, then $\mathsf{y}_u^j + \mathsf{y}_v^j = w_j(e)$;*

Proposition 2.2 follows from the complementary slackness conditions in the linear programming duality theorem. This suggests the following strategy once the primal and dual optimal solutions M_j and y^j are found in the j-th iteration.

- to prune "wrong" edges: if $e = (u,v)$ and $y_u^j + y_v^j > w_j(e)$, then no optimal solution of the j-th iteration primal program can contain e. So we prune such edges from G_j and let G_{j+1} denote the resulting graph. The graph G_{j+1} will be used for the $(j+1)$-st iteration.

- to grow the critical set K_{j-1}: if $\mathsf{y}_u^j > 0$ and $u \notin K_{j-1}$, then u *has to* be matched in every optimal solution of the primal program of the j-th iteration. Hence u should be added to the critical set. Adding such vertices u to K_{j-1} yields the critical set K_j for the $(j+1)$-st iteration.

3 Our main algorithm

We first present our algorithm that runs for r iterations, where r is the total number of weight functions. In case there is an r^* such that $w_j(\mathsf{OPT}) = 0$ for all $r^* + 1 \le j \le r$, we then show how to terminate our algorithm in $r^* + 1$ iterations.

1. *Initialization.* Let $G_1 = G$ and $K_0 = \emptyset$.

2. For $j = 1$ to r do
   (a) Let M_j be a maximum w_j-weight matching in G_j subject to the constraint that all vertices in K_{j-1} are matched (thus M_j is primal optimal).
   (b) Find an optimal solution $\{\mathsf{y}_u^j\}_{u \in A \cup B}$ to the dual program.
   (c) Every edge (a, b) such that $\mathsf{y}_a^j + \mathsf{y}_b^j > w_j(e)$ is pruned from G_j. Call the pruned graph G_{j+1}.
   (d) Add all vertices with positive dual values to the critical set. That is, $K_j = K_{j-1} \cup \{u\}_{\mathsf{y}_u^j > 0}$.
3. Return the matching M_r.

We now prove the correctness of our algorithm in Lemma 3.2. First we need to show that there is always a feasible solution for the primal program of the j-th iteration and its dual, so that our algorithm is never "stuck" in Steps 2(a) and 2(b).

Lemma 3.1 *The primal program of the j-th iteration and its dual are feasible, for $1 \leq j \leq r$.*

Proof. By linear programming duality, if the primal program of the j-th iteration is bounded from above and admits a feasible solution, then there is also a feasible solution for its dual. It is obvious that the primal program is bounded from above since it is upper bounded by $\sum_{e \in E} w_j(e) \leq Wm$. Therefore, to prove this lemma, we just need to show the feasibility of the primal program.

The base case is $j = 1$. Since $K_{j-1} = \emptyset$, any matching in $G_1 = G$ is a feasible solution for the first primal program. For $j > 1$, we need to show that the primal program of the j-th iteration is feasible. By induction hypothesis, assume that the primal program of the $(j-1)$-st iteration is feasible. Let M_{j-1} denote its optimal solution. Since M_{j-1} is a feasible point of the primal program of the $(j-1)$-st iteration, M_{j-1} uses only edges in G_{j-1} and matches all vertices in K_{j-2}. Since M_{j-1} is, in fact, an *optimal* solution to the primal program of the $(j-1)$-st iteration, we will show that M_{j-1} has to be a feasible point of the primal program of the j-th iteration by arguing that M_{j-1} does not use any of the edges pruned from G_{j-1} and all vertices in K_{j-1} are matched in M_{j-1}.

In step 2(c) of the $(j-1)$-st iteration, we remove only those edges $e = (a, b)$ such that $\mathsf{y}_a^{j-1} + \mathsf{y}_b^{j-1} > w_{j-1}(e)$ from G_{j-1} to form G_j. By the optimality of M_{j-1}, we know from Proposition 2.2.2 that M_{j-1} has no *slack* edges, thus all edges in M_{j-1} are retained in G_j.

We also know from Proposition 2.2.1 that if $y_u^{j-1} > 0$, then u must be matched in M_{j-1}. Therefore, all vertices in $K_{j-1} \backslash K_{j-2}$ are matched in M_{j-1}. Moreover, as M_{j-1} is also a feasible solution in the $(j-1)$-st primal program, all vertices in K_{j-2} are matched in M_{j-1}. This completes the proof of Lemma 3.1.

□

Lemma 3.2 *For every $0 \leq j \leq r$, the following hold:*

1. *any matching M in G_{j+1} that matches all vertices in K_j is j-optimal;*
2. *conversely, a j-optimal matching in G is a matching in G_{j+1} that matches all vertices in K_j.*

Proof. We proceed by induction. The base case is $j = 0$. As $K_0 = \emptyset$, $G_1 = G$, and all matchings are, by definition, 0-optimal, the lemma holds vacuously.

For the induction step $j > 1$, suppose that the lemma holds up to $j - 1$. As $K_j \supseteq K_{j-1}$ and G_{j+1} is a subgraph of G_j, M is a matching in G_j that matches all vertices of K_{j-1}. Thus by induction hypothesis, M is $(j-1)$-optimal. For each edge $e = (a, b) \in M$ to be present in G_{j+1}, e must be a tight edge in the j-th iteration, i.e., $\mathsf{y}_a^j + \mathsf{y}_b^j = w_j(e)$. Furthermore, as $K_j \supseteq \{u\}_{\mathsf{y}_u^j > 0}$, we have

$$w_j(M) = \sum_{e=(a,b)\in M} w_j(e) = \sum_{e=(a,b)\in M} \mathsf{y}_a^j + \mathsf{y}_b^j \geq \sum_{u\in A\cup B} \mathsf{y}_u^j,$$

where the final inequality holds because all vertices v with positive y_v^j are matched in M. By linear programming duality, M must be optimal in the primal program of the j-th iteration. So the j-th primal program has optimal solution of value $w_j(M)$.

Recall that by definition, OPT is also $(j-1)$-optimal. By (2) of the induction hypothesis, OPT is a matching in G_j and OPT matches all vertices in K_{j-1}. So OPT is a feasible solution of the primal program in the j-th iteration. Thus $w_j(\mathsf{OPT}) \leq w_j(M)$. However, it cannot happen that $w_j(\mathsf{OPT}) < w_j(M)$, otherwise, $\mathsf{signature}(M) \succ \mathsf{signature}(\mathsf{OPT})$, since both OPT and M_j have the same first $j-1$ coordinates in their signatures. So we conclude that $w_j(\mathsf{OPT}) = w_j(M)$, and this implies that M is j-optimal as well. This proves (1).

In order to show (2), let M' be a j-optimal matching in G. Note that since M' is j-optimal, it is also $(j-1)$-optimal and by (2) of the induction hypothesis, it is a matching in G_j that matches all vertices in K_{j-1}. So M' is a feasible solution to the primal program of the j-th iteration. As $\mathsf{signature}(M')$ has $w_j(\mathsf{OPT})$ in its j-th coordinate, M' must be an optimal solution to this primal program; otherwise there is a $(j-1)$-optimal matching with a value larger than $w_j(\mathsf{OPT})$ in the j-th coordinate of its signature, contradicting the optimality of OPT. What remains to be shown is that all edges of M' are present in G_{j+1} and all vertices in $K_j \setminus K_{j-1}$ are matched in M'. The former fact follows from Proposition 2.2.2: the optimal solution M' uses only tight edges. For the latter fact, note that by Proposition 2.2.1, all vertices $u \notin K_{j-1}$ with $\mathsf{y}_u^j > 0$ have to be matched by the optimal solution M'. This completes the proof of (2). □

Combining Lemma 3.1 and Lemma 3.2.1, we conclude that M_r is an r-optimal matching. This implies that M_r is a weight-maximal matching, since $\mathsf{signature}(M) = \mathsf{signature}(\mathsf{OPT})$. Lemma 3.2.2 yields Theorem 1.5 by setting G' to be G_{r+1} and V' to be K_r.

We now discuss how to modify our algorithm so that it ends in r^*+1 iterations, rather than in r iterations, in case there is an r^* such that $w_j(\mathsf{OPT}) = 0$ for all $r^*+1 \leq j \leq r$. Insert the following step right before Step 2(a).

> Step 2(0): Define the graph G'_j as follows: the edge set is exactly the same as in G_j; however the edge weight function w'_j is as follows. *For each edge e in G'_j: set $w'_j(e) = 1$ if $\sum_{i=j}^{r} w_i(e) > 0$, else set $w'_j(e) = 0$.*
>
> – Now find a maximum weight matching M'_j in G'_j so that all the vertices in K_{j-1} are matched. If the weight of M'_j is 0, then return M_{j-1} as the final solution.

Lemma 3.3 *The following claims hold:*

(1) If the matching M'_j has weight 0, then M_{j-1} is a weight-maximal matching.

(2) If r^* is the largest index such that $\mathsf{signature}(\mathsf{OPT})$ *is positive in this coordinate, then in Step 2(0) of the j-th iteration where $j = r^*+1$, the matching M'_j has weight 0.*

Proof. We first show (1). Suppose $w'_j(M'_j) = 0$. Since all edge weights are non-negative, we claim that any matching M in G_j that matches all vertices of K_{j-1} must have $w_j(M) = w_{j+1}(M) = \cdots = w_r(M) = 0$, otherwise, $w'_j(M) > 0$, a contradiction to the assumption that the maximum weight matching M'_j in G'_j satisfies $w'_j(M'_j) = 0$.

As a result, both M_{j-1} and OPT have 0 in the last $r-j+1$ coordinates in their signatures, since M_{j-1} (by its optimality in G_{j-1}) and OPT (by Lemma 3.2.2) are matchings in G_j that match all vertices of K_{j-1}. By Lemma 3.2.1, M_{j-1} is $(j-1)$-optimal. Hence $\mathsf{signature}(M_{j-1}) = \mathsf{signature}(\mathsf{OPT})$, thus M_{j-1} is weight-maximal and (1) is proved.

We prove (2) by contradiction. Suppose that $w'_j(M'_j) > 0$, where $j = r^*+1$. Since M'_j is a matching in G_j that matches all vertices in K_{j-1}, $\mathsf{signature}(M'_j) \succ \mathsf{signature}(\mathsf{OPT})$; hence by Lemma 3.2.1, M'_j is r^*-optimal and while OPT has only 0 in its last $r-r^*$ coordinates, M'_j has some positive values in its last $r-r^*$ coordinates. This is a contradiction to the optimality of OPT.

□

In fact, based on the proof of Lemma 3.3, it follows that if $w'_j(M'_j) = 0$, then any $(j-1)$-optimal matching is weight-maximal. By Lemma 3.2, the graph G_{r^*+1} is the subgraph G' and the subset K_{r^*} of vertices is the critical set V' stated in Theorem 1.5.

We now show that our algorithm has running time as claimed in Theorem 1.3. Each iteration, barring Steps 2(0), 2(a) and 2(b), can be done easily in $O(m)$ time. In the next section, we show how to solve the primal program in $O(\sqrt{n}m\log n)$ time, the dual in $O(\sqrt{n}m)$ time, or both in $\tilde{O}(n^\omega)$ time with high probability. Thus Theorem 1.3 follows.

Rank-maximal/maximum cardinality rank-maximal/fair matchings. It is now straightforward to see that rank-maximal matchings and maximum cardinality rank-maximal matchings can be computed in time $O(r^*\sqrt{n}m\log n)$ or $\tilde{O}(r^*n^\omega)$ with high probability, as promised in Theorem 1.4. However for fair matchings, a little more work is needed because we need to know the value of r^* (the worst rank used in a fair matching) right at the beginning: we will need to use weight functions $w_1,\ldots,w_{r^*}$, where for $1 \le i \le r^*$, w_i is defined as: for any edge $e=(a,b)$, $w_i(e)$ is 2 if both a and b rank each other as rank $\le r^*-i+1$ neighbors, it is 1 if exactly one of $\{a,b\}$ ranks the other as a rank $\le r^*-i+1$ neighbor, otherwise it is 0.

The value r^* can be easily computed right at the start of our algorithm as follows. Let M^* be a maximum cardinality matching in G. The value r^* is the smallest index j such that the subgraph $\bar{G}_j$ admits a matching of size $|M^*|$, where $\bar{G}_j$ is obtained by deleting all edges $e=(a,b)$ from G where either a or b (or both) ranks the other as a rank $> j$ neighbor. We compute r^* by first computing M^* and then computing a maximum cardinality matching in $\bar{G}_1, \bar{G}_2, \ldots$ and so on till we see a subgraph $\bar{G}_j$ that admits a matching of size $|M^*|$. This index $j = r^*$ and it can be found in $O(r^*\sqrt{n}m)$ time [10] or in $O(r^*n^\omega)$ time [9, 18]. Thus Theorem 1.4 follows.

4 Solving the primal and dual programs

Let $G_j = (A \cup B, E_j)$ be the subgraph that we work with in the j-th iteration and let $K_{j-1} \subseteq A \cup B$ be the critical set of vertices in the j-th iteration. Recall that for each $e \in E_j$, we have $w_j(e) \in \{0,\cdots,W\}$.

The primal program can be solved by the following folklore technique: create a new graph $\tilde{G}_j$ by taking two copies of G_j and making the two copies of a vertex $u \notin K_{j-1}$ adjacent using an edge of weight 0. A maximum weight *perfect* matching in $\tilde{G}_j$ yields a maximum weight matching in G_j that matches all vertices in K_{j-1}, i.e., an optimal solution to the primal program of the j-th iteration. Note that since W is some constant, a maximum weight *perfect* matching in $\tilde{G}_j$ can be found in $O(\sqrt{n}m\log n)$ time by the fastest bipartite matching algorithms [1, 2, 5], or in $\tilde{O}(n^\omega)$ time with high probability by Sankowski's algorithm [20].

Let M_j be the optimal solution of the primal program. We now discuss how to use it to solve the dual program. Our idea is built upon that of Iri [11] (who solved the special case of $K_{j-1} = \emptyset$). Recall that if any vertex v is unmatched in M_j, then $v \notin K_{j-1}$.

- Add a new vertex z to A and let $A' = A \cup \{z\}$. Add an edge of weight 0 from z to each vertex in $B \setminus K_{j-1}$. For convenience, we call the edges from z to these vertices "virtual" edges. Note that after this transformation, M_j is still the optimal feasible solution.

  [As there can be only $O(n)$ virtual edges, the time complexity is not changed when later we apply the single-source shortest-paths algorithm.]

- Next direct all edges $e \in E_j \setminus M_j$ from A' to B and let the distance $d(e) = -w_j(e)$; also direct all edges in M_j from B to A' and let the distance $d(e) = w_j(e)$.

- Create a *source vertex* s and add a directed edge from s to each *unmatched* vertex in $A' \setminus K_{j-1}$. Let $\mathcal{R}$ denote the set of all vertices reachable from s.

Lemma 4.1 *By the above transformation,*

(1) $B \setminus K_{j-1} \subseteq \mathcal{R}$.
(2) There is no edge between $A' \cap \mathcal{R}$ *and* $B \setminus \mathcal{R}$.
(3) M_j *projects on to a perfect matching between* $A' \setminus \mathcal{R}$ *and* $B \setminus \mathcal{R}$.

Proof. (1) holds because there is a directed edge from s to z and directed edges from z to every vertex in $B \setminus K_{j-1}$. To show (2), it is trivial to see that there can be no edge from $A' \cap R$ to $B \setminus R$ (by the definition of $B \setminus R$). If there is an edge (b, a) from $B \setminus R$ to $A' \cap R$, then this has to be an edge in M_j and hence it is a's only incoming edge. So for a to be reachable from s, it has to be the case that b is reachable from s, contradicting that $b \in B \setminus R$.

For (3), observe that if $b \in B \setminus \mathcal{R}$ is unmatched in M_j, then $b \notin K_{j-1}$ and such a vertex can be reached via z, contradicting the assumption that $b \in B \setminus \mathcal{R}$. If $a \in A' \setminus \mathcal{R}$ is unmatched in M_j, then such a vertex can be reached from s, contradicting the assumption that $a \in A' \setminus \mathcal{R}$. So all vertices in $(A' \cup B) \setminus \mathcal{R}$ are matched in M_j. By (2), a vertex $b \in B \setminus \mathcal{R}$ cannot be matched to vertices in $A' \cap \mathcal{R}$. If a vertex $a \in A' \setminus \mathcal{R}$ is matched to a vertex $B \in \mathcal{R}$, then a is also in $\mathcal{R}$, a contradiction. This proves (3).

□

Note that there may exist some edges in $E_j \setminus M_j$ that are directed from $A' \setminus R$ to $B \cap \mathcal{R}$. Furthermore, some vertices of $A \setminus K_{j-1}$ can be contained in $A \setminus \mathcal{R}$.

Delete all edges from $A' \setminus \mathcal{R}$ to $B \cap \mathcal{R}$ from G_j; let H_j denote the resulting graph. By Lemma 4.1.3, no edge of M_j has been deleted, thus M_j belongs to H_j. Note that M_j is still the optimal matching in the graph H_j. Moreover, H_j is split into two parts: one part is $(A' \cup B) \cap \mathcal{R}$, which is isolated from the second part $(A' \cup B) \setminus \mathcal{R}$.

Next add a directed edge from the source vertex s to each vertex in $B \setminus \mathcal{R}$. Each of these edges e has distance $d(e) = 0$. By Lemma 4.1.3, all vertices can be reached from s now. Also note that there can be no negative-length cycle, otherwise, we can augment M_j along this cycle to get a matching of larger weight while still keeping the same set of vertices matched, which leads to a contradiction to the optimality of M_j.

Apply the single-source shortest paths algorithm [8, 19, 20, 21] from the source vertex s in this graph H_j where edge weights or edge distances are given by $d(\cdot)$. Such algorithms take $O(\sqrt{n}m)$ time or $\tilde{O}(n^\omega)$ time. Let d_v be the distance label of vertex $v \in A' \cup B$.

We define an initial vertex cover as follows. If $a \in A'$, let $\tilde{y}_a := d_a$; if $b \in B$, let $\tilde{y}_b := -d_b$. (We will adjust this cover further later.)

Lemma 4.2 *The constructed initial vertex cover $\{\tilde{y}_v\}_{v \in A' \cup B}$ for the graph H_j satisfies the following:*

(1) For each vertex $v \in ((A \cup B) \cap \mathcal{R}) \setminus K_{j-1}$, $\tilde{y}_v \geq 0$.
(2) If $v \in (A \cup B) \setminus K_{j-1}$ is unmatched in M_j, then $\tilde{y}_v = 0$.
(3) For each edge $e = (a, b) \in H_j$, we have $\tilde{y}_a + \tilde{y}_b \geq w_e^j$.
(4) For each edge $e = (a, b) \in M_j$, we have $\tilde{y}_a + \tilde{y}_b = w_e^j$.

Proof. For (1), suppose that $a \in (A \cap \mathcal{R}) \setminus K_{j-1}$ and $\tilde{y}_a < 0$. By Lemma 4.1.2 and the fact that all edges from $A' \setminus \mathcal{R}$ to $B \cap \mathcal{R}$ are absent, the shortest path from r to a cannot go through $(A \cup B) \setminus \mathcal{R}$. So there exists an alternating path P (of even length) starting from some *unmatched* vertex $a' \in (A' \cap \mathcal{R}) \setminus K_{j-1}$ and ending at a. The distance from a' to a along path P must be negative, since $d_a = \tilde{y}_a < 0$. Therefore,

$$\sum_{e \in M_j \cap P} w_e < \sum_{e \in P \setminus M_j} w_e.$$

Note that it is possible that the first edge $e = (a', b) \in P$ is a virtual edge, i.e., $a' = z$ and the first edge e connects z to some vertex $b \in (B \cap \mathcal{R}) \setminus K_{j-1}$. In this case, such an edge has distance $d_e = 0$ and b is not part of the critical set K_{j-1}. Therefore, irrespective of whether the first edge is virtual or not, we can replace the matching M_j by $M_j \oplus P$ (ignoring the first edge in P if it is virtual), thereby creating a feasible matching with larger weight than M_j, a contradiction.

So we are left to worry about the vertex $b \in (B \cap \mathcal{R}) \setminus K_{j-1}$. Recall that $\tilde{y}_b = -d_b$. We claim that $d_b \leq 0$. Suppose not. Then the shortest distance from s to b is strictly larger than 0. But this cannot be, since there is a path composed of edges (s, z) and (z, b), and such a path has total distance of exactly 0. This completes the proof of (1).

To show (2), by Lemma 4.1.3, an unmatched vertex must be in $\mathcal{R}$. First, assume that this unmatched vertex is $a \in (A \cap \mathcal{R}) \setminus K_{j-1}$. By our construction, there is only one path from s to a, which is simply the directed edge from s to a and its distance is 0. So $y_a = d_a = 0$. Next assume that this unmatched vertex is $b \in (B \cap \mathcal{R}) \setminus K_{j-1}$. Suppose that $\tilde{y}_b > 0$. Then $d_b = -\tilde{y}_b < 0$. By Lemma 4.1.2 and the fact that all edges from $A' \setminus \mathcal{R}$ to $B \cap \mathcal{R}$ have been deleted, the shortest path from s to b cannot go through $(A \cup B) \setminus \mathcal{R}$. So the shortest path from s to b must consist of the edge from s to some unmatched vertex $a \in (A' \cap \mathcal{R}) \setminus K_{j-1}$, followed by an augmenting path P (of odd length) ending at b. As in the proof of (1), we can replace M_j by $M_j \oplus P$ (irrespective of whether the first edge in P is virtual or not) so as to get a matching of larger weight while preserving the feasibility of the matching, a contradiction. This proves (2).

For (3) and (4), first consider an edge $e = (a, b)$ outside M_j in H_j. Such an edge is directed from a to b. So $\tilde{y}_a - w_e^j = d_a + d(e) \geq d_b = -\tilde{y}_b$. This proves (3). Next consider an edge $e = (a, b) \in M_j$. Such an edge is directed from b to a. Furthermore, e is the only incoming edge of a, implying that e is part of the shortest path tree rooted at s. As a result, $-\tilde{y}_b + w_e^j = d_b + d(e) = d_a = \tilde{y}_a$. This shows (4). This completes the proof of Lemma 4.2. □

Modifying the initial vertex cover. At this point, we possibly still do not have a valid cover for the dual program due to the following two reasons.

- Some vertex $a \in A \setminus K_{j-1}$ has $\tilde{y}_a < 0$. (However there is no worry that some vertex $b \in B \setminus K_{j-1}$ has $\tilde{y}_b < 0$, since Lemma 4.1.1 states that such a vertex is in $\mathcal{R}$ and Lemma 4.2.1 states that $\tilde{y}_b$ must be non-negative.)

- The edges deleted from G_j (to form H_j) are not properly covered by the initial vertex cover $\{\tilde{y}_v\}_{v\in A\cup B}$.

We can remedy these two defects as follows. Define $\delta = \max\{\delta_1, \delta_2\}$ as follows,

$$\text{where } \delta_1 = \max_{e=(a,b):\, \tilde{y}_a+\tilde{y}_b<w_e^j} \{w_e^j - \tilde{y}_a - \tilde{y}_b\} \quad \text{and} \quad \delta_2 = \max_{a\in A\setminus K_{j-1}:\, \tilde{y}_a<0} \{-\tilde{y}_a\}.$$

In $O(n+m)$ time, we can find such a δ or conclude that the initial vertex cover is already a valid solution for the dual program. In the following, we assume that δ exists (if the initial cover is already a valid solution for the dual program, then the proof that it is also optimal is just the same as Theorem 4.3.) We build the final vertex cover as follows.

1. For each vertex $u \in (A\cup B)\cap \mathcal{R}$, let $y_u = \tilde{y}_u$;
2. For each vertex $a \in A\setminus \mathcal{R}$, let $y_a = \tilde{y}_a + \delta$;
3. For each vertex $b \in B\setminus \mathcal{R}$, let $y_b = \tilde{y}_b - \delta$;

Theorem 4.3 *The final vertex cover $\{y_v\}_{v\in A\cup B}$ is an optimal solution for the dual program.*

Proof. We first argue that $\{y_v\}_{v\in A\cup B}$ is a feasible dual solution. By Lemma 4.2.1 and the choice of δ, all vertices $a \in A\setminus K_{j-1}$ have $y_a \geq 0$. By Lemma 4.1.1 and Lemma 4.2.1, all vertices $b \in B\setminus K_{j-1}$ have $y_b \geq 0$. Also by Lemma 4.1.2 and Lemma 4.2.3, and the choice of δ, all edges in E_j are properly covered. So $\{y_v\}_{v\in A\cup B}$ is feasible.

Now observe that

$$\begin{aligned} w_j(M_j) = \sum_{e\in M_j} w_e^j &= \sum_{e=(a,b)\in M_j,\, b\in\mathcal{R}} \tilde{y}_a + \tilde{y}_b + \sum_{e=(a,b)\in M_j,\, b\notin\mathcal{R}} (\tilde{y}_a+\delta) + (\tilde{y}_b - \delta) \\ &= \sum_{e=(a,b)\in M_j} y_a + y_b \\ &\geq \sum_{u\in A\cup B} y_u, \end{aligned}$$

where the last inequality holds because if a vertex u is unmatched, Lemma 4.2.2 states that $\tilde{y}_u = 0$ and since u must be in $\mathcal{R}$, we have $y_u = \tilde{y}_u = 0$.

Now by the linear programming duality theorem, we conclude that the cover $\{y_v\}_{v\in A\cup B}$ is optimal. □

We thus conclude that the dual problem can be solved in time $O(\sqrt{n}m)$ or $\tilde{O}(n^\omega)$ in each iteration.

References

[1] J. B. Orlin and R. K. Ahuja. New scaling algorithms for the assignment and minimum mean cycle problems In *Mathematical Programming* 54(1): 41-56, 1992.

[2] R. Duan and H.-H. Su. A Scaling Algorithm for Maximum Weight Matchings in Bipartite Graphs. In 23rd *SODA*: 1413-1424, 2012.

[3] A. Dulmage and N. Mendelsohn. Coverings of bipartite graphs. In *Canadian J. Math.* 10: 517-534, 1958.

[4] M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. In *J.ACM* 34(3), 596-615, 1987.

[5] H. Gabow and R. Tarjan. Faster scaling algorithms for network problems. In *SIAM J. Comput.* 18: 1013-1036, 1989.

[6] D. Gale and L.S. Shapley. College admissions and the stability of marriage. In *American Mathematical Monthly* 69: 9-15, 1962.

[7] D. Gusfield and R.W. Irving. The stable marriage problem: structure and algorithms. MIT Press, 1989.

[8] A. Goldberg. Scaling Algorithms for the Shortest Paths Problem. In *SIAM J. Comput.* 24(3): 494-504, 1995. Conference version in SODA 1993.

[9] N. J. A. Harvey. Algebraic Structures and Algorithms for Matching and Matroid Problems. In *SIAM J. Comput.* 39(2): 679-702, 2009. Conference version in FOCS 2006.

[10] J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. In *SIAM J. Comput.* 2: 225-231, 1973. Conference Version in 12th Symposium on Switching and Automata Theory.

[11] M. Iri. A new method of solving transportation-network problems. In *Journal of the Operations Research Society of Japan* 3: 27-87, 1960.

[12] R. W. Irving. Greedy Matchings. University of Glasgow, Computing Science Department Research Report, TR-2003-136, 2003.

[13] R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail and K. E. Paluch. Rank-maximal matchings. In *ACM Transactions on Algorithms* 2(4): 602-610, 2006. Conference version in SODA 2004.

[14] T. Kavitha and C. D. Shah Efficient Algorithms for Weighted Rank-Maximal Matchings and Related Problems. In *International Symposium on Algorithms and Computation (ISAAC)*: 153-162, 2006.

[15] H.W. Kuhn. The Hungarian method for the assignment problems. In *Naval Research Logistics Quarterly* 2: 83-97, 1955.

[16] K. Mehlhorn and D. Michail. Network Problems with Non-Polynomial Weights and Applications. Available at `www.mpi-sb.mpg.de/~mehlhorn/ftp/HugeWeights.ps`

[17] D. Michail. Reducing rank-maximal to maximum weight matching. In *Theoretical Computer Science* 389(1-2): 125-132, 2007.

[18] M. Mucha and P. Sankowski. Maximum Matchings via Gaussian Elimination. In the Proceedings of the 45th IEEE Symp. Foundations of Computer Science: 248-255, 2004.

[19] P. Sankowski. Shortest Paths in Matrix Multiplication Time. In the Proceedings of the 13th Annual European Symposium on Algorithms: 770-778, 2005.

[20] P. Sankowski. Maximum weight bipartite matching in matrix multiplication Time. In *Theoretical Computer Science* 410: 4480-4488, 2009. Conference version in ICALP 2006.

[21] R. Yuster and U. Zwick. Answering distance queries in directed graphs using fast matrix multiplication. In *46th FOCS*: 90-100, 2005.

Linear time local approximation algorithm for maximum stable marriage

Zoltán Király[*]

Abstract

We consider a two-sided market under incomplete preference lists with ties, where the goal is to find a maximum size stable matching. The problem is APX-hard, a 3/2-approximation was given by McDermid [M 2009]. This algorithm has non-linear running time, and, more importantly needs a global knowledge about all preference lists.

We give a very natural, economically reasonable, local, linear time algorithm with the same ratio, using some ideas of Paluch [P 2010]. In this algorithm every people make decisions using only their own list, and some information asked from members of this lists (like during the famous algorithm of Gale and Shapley).

Some consequences to the Hospitals/Residents problem are also discussed.

Keywords: stable marriage, Gale-Shapley algorithm, approximation, Hospitals/Residents problem.

1 Introduction

In 1962 Gale and Shapley [GS 1962] gave their famous simple deferred acceptance algorithm, that always finds a stable matching in a two-sided market. If incomplete lists and ties are allowed in the preference lists, their algorithm is still working, and gives some stable matching. However in this case we are usually interested in not only finding some stable matching, but one with maximum size. This problem (usually called MAX-SMTI) is APX-hard, and probably cannot be approximated within factor of 4/3. McDermid [M 2009] gave the first 3/2-approximation. In this paper we give a much simpler algorithm than he gave, which is only a slight modification of the historical algorithm of Gale and Shapley. This also gives 3/2-approximation, but in **linear** time. The proof of the approximation ratio is uncomplicated, thus it serves as a good example for teaching purposes.

In this paper we make difference between *global* and *local* algorithms. When we speak about a global algorithm, we assume that there is a centralized decision mechanism, having all possible information (the preference list of all participants). A global algorithm is called linear if the number of steps it takes is some constant times the size of the input (usually the total length of the preference lists).

A local algorithm is quite restricted compared to a global algorithm. Here every participant must have his/her own algorithm and no global information is available. Besides his/her own information, a participant can only ask information from his/her acceptable partners. In this model we assume everyone has his/her own list sorted at the beginning, and a local algorithm

[*] Department of Computer Science and MTA-ELTE Egerváry Research Group, Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary. Research was supported by grants (no. CNK 77780 and no. CK 80124) from the National Development Agency of Hungary, based on a source from the Research and Technology Innovation Fund, and by TÁMOP grant 4.2.1./B- 09/1/KMR-2010-0003.

called linear if the number of steps any participant needs is some constant times the size of the input (length of the preference list) of that particular participant.

We will give both linear-time global and linear-time local algorithms for approximating the maximum size stable marriage problem, moreover in our local algorithm every participant of the two-sided market will make only very natural, economically reasonable decisions.

An instance of the stable marriage problem consists of a set U of men, a set W of women, and a preference list for each person, that is a weak linear order (ties are allowed) on some participants (thus, in this paper, we are dealing with only the practical situation of incomplete lists) of the opposite gender. If $m \in U$ and $w \in W$, then a pair mw 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 $G = (U, W, E)$, (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). Note, that total length of the lists is proportional to $|E|$. A *matching* in this graph consists of mutually disjoint acceptable pairs.

We store the weak order of the lists as priorities. For an acceptable pair mw, let $\text{pri}(w, m)$ be an integer from 1 up to $|U|$ representing the priority of m for w. We say that $w \in W$ *strictly prefers* $m_1 \in U$ to $m_2 \in U$ if both m_1 and m_2 are acceptable for w, and $\text{pri}(w, m_1) > \text{pri}(w, m_2)$. Ties are represented by equal priorities, e.g., if m_1 and m_2 are tied in w's list, then $\text{pri}(w, m_1) = \text{pri}(w, m_2)$. We will later break up some ties, dividing the men in a tie into two groups, and we will say, that w prefers one group to the other. So there will be a case, that *w prefers m_1* to m_2, but not strictly prefers, and in this situation we mean that $\text{pri}(w, m_1) = \text{pri}(w, m_2)$, but m_1 is in the preferred group and m_2 is not. If both of them are in the same group then we say m_1 and m_2 are *alike* for w.

We define $\text{pri}(m, w)$ similarly, of course, $\text{pri}(m, w)$ is not related to $\text{pri}(w, m)$. We represent these priorities in the figures by writing $\text{pri}(m, w)$ and $\text{pri}(w, m)$ close to the corresponding endvertex of edge mw ($\text{pri}(m, w)$ is written next to m, while $\text{pri}(w, m)$ is written next to w). For a man m we will modify his list in either of two ways. Sometimes we delete a woman from the list. Sometimes (at a given point, when his list is empty) we will restore the original list of m. We mean by "*favorite woman*" of m, the mostly preferred woman still on m's list; if there are more alike women on the top of the list, we choose one of them arbitrarily.

Remark. The author apologizes that the text of this paper is not PC. Just like the one of Gale and Shapley [GS 1962], or the phrases used in many other papers on this topic. It would indeed be possible to change the terminology, but we find this approach pretty well-mannered.

Let M be a matching. If m is *matched* in M, or in other words, if m is *engaged* then we denote m's fiancée by $M(m)$. Similarly we use $M(w)$ for the fiancé of an engaged woman w.

Definition 1 *A pair mw is* blocking M, *if $mw \in E \setminus M$ (they are an acceptable pair and they are not matched) and*

- *w is either not engaged or w strictly prefers m to her fiancé,* **and**
- *m is either not engaged or m strictly prefers w to his fiancée.*

Definition 2 *A matching is called* stable *if there is no blocking pair.*

While there may be objections to the connotations of these words, for simplicity we use the term "**lad**" to represent a man who still has some women on his list, whom he did not propose to so far, and we use the term "**old bachelor**" to represent a man who was refused by all acceptable women and decides to become inactive forever. Moreover we use the term "**maiden**" to represent a woman who did not get any proposal so far. Later we need to use some other similar terms, defined there.

It is well-known that a stable matching always exists and can be found in linear time. The celebrated algorithm of Gale and Shapley [GS 1962] is the following.

A man can either be a *lad* or an *old bachelor.* A lad can either be *active* or *engaged.* A woman can either be *maiden* or *engaged.* At the beginning every man is an active lad and every woman is a maiden.

> *Algorithm GS*
>
> While there exists an active man m, he proposes to his favorite woman w. If w accepts his proposal, they become engaged. If w refuses him, m will delete w from his list, and will remain active.
> When a woman w gets a new proposal from man m, she always accepts this proposal, if she is a maiden. She also accepts this proposal, if she prefers m to her current fiancé. Otherwise she refuses m.
> If w accepted m, then she refuses her previous fiancé, if there was one (breaks off her engagement), and becomes engaged to m.
> If a man m was engaged to a woman w, and later w refuses him, then m becomes active again, and deletes w from his list.
> If the list of m becomes empty, he will turn into an old bachelor and will remain inactive forever.

After Algorithm GS finishes, the engaged pairs make up the output matching M (we may imagine that this time all the engaged pairs get married).

Theorem 1 (Gale and Shapley [GS 1962]) *Algorithm GS always ends in a stable matching M. This algorithm runs in $O(|E|)$ time.*

An interesting problem, motivated by applications, is to find a stable matching of maximum size. As the applications of this problem are important (see e.g., in [IM 2007, IM 2008], where detailed lists of known and possible applications are given, that motivate investigating refined approximations), researchers started to develop good approximation algorithms in the past six years. We say that an algorithm is r-approximating, if it gives a stable matching M with size $|M| \geq (1/r) \cdot |M_{\mathrm{opt}}|$, where M_{opt} is a stable matching of maximum size. Observe, that after a run of GS no unmarried man and unmarried woman can form an acceptable pair. Consequently Algorithm GS gives a 2-approximation, and for complete bipartite graphs (every woman-man pair is acceptable), it gives the optimum. The first non-trivial approximation algorithm was given by Halldórsson et al. [HIMY 2007], where they gave a 13/7-approximation if all ties are of length two. The breakthrough was achieved by Iwama, Miyazaki and Yamauchi [IMY 2007], who gave a 15/8-approximation (for any length of ties). This was later improved by Irving and Manlove [IM 2007] to a 5/3-approximation for the special case, where ties are allowed on one side only, and moreover only at the ends of the lists. Their algorithm also applies to the Hospitals/Residents problem (see later) if residents have strictly ordered lists. If, moreover, ties are of size 2, Halldórsson et al. [HIMY 2007] gave an 8/5-approximation and in [HIMY 2004] they described a randomized algorithm for this special case with expected ratio of 10/7.
This problem is known to be NP-hard for even very restricted cases [IMMM 1999, MIIMM 2002]. Moreover, it is APX-hard [HIIMMMS 2003] and, supposing P≠NP, it cannot be approximated within a factor of strictly less than 21/19, even if ties occur in the preference lists on one side only, furthermore, if every list is either totally ordered or consists of a single tied pair [HIMY 2007].

Moreover, refining the ideas of [HIMY 2007], Yanagisawa [Y 2007] proved, that an approximation within a factor of $4/3-\varepsilon$ implies $(2-\varepsilon')$-approximation of vertex cover, and this result also applies to the case when each tie has length two. If, moreover, ties occur only in the preference lists on one side only, it was proved in [HIMY 2007], that an approximation within a factor of $5/4-\varepsilon$ has the same implication. We note that interestingly the minimization version (where we are looking for a stable matching of minimum size) is also APX-hard [HIIMMMS 2003].

We proposed a simple linear-time 5/3-approximating algorithm (called GSA2) for the general case of this problem, first presented at the first MATCH-UP workshop (see in [K1 2008, K2 2008, K 2009]), and at the same time, an even simpler 3/2-approximation (called GSA1) was given for the special case where ties are allowed on one side only. For this algorithm, the proof was also very short, see Section 2. This is also valid for the practically important "Hospitals/Residents" problem, where the lists of the Residents are strict, see Section 5.

During the talks given at the 1st MATCH-UP workshop in Reykjavík and at ESA in Karlsruhe, we posed several questions, conjectures and open problems. Many of them were answered in the meanwhile.

The conjecture stating that the performance ratio proved for GSA2 is sharp proved to be true by Yanagisawa [Y 2008], who gave a simple example where GSA2 really gives a matching of size exactly $\frac{3}{5}\cdot M_{\mathrm{opt}}$.

Irving and Manlove [IM 2009] implemented a basic version of our algorithm for the one-sided-ties Hospitals/Residents problem and gave a detailed comparison with their best heuristic (which is not a local algorithm, it needs some max-flow computation on an auxiliary graph). They tested carefully the algorithms with real-life and artificial data. They concluded that for the most cases their best heuristic executed the best, but, on the average, our algorithm also gave a stable assignment of size at least 99.41% of their best one. We do not know too many other examples, where an algorithm with a guaranteed approximation ratio is so close in the practice to the best known heuristic.

For the one-sided-ties case, Iwama, Miyazaki, and Yanagisawa [IMY 2010] gave a $25/17\approx 1.47$-approximation. They solved the relaxed version of an appropriate ILP formulation, and used the fractional optimum to guide the tie-breaking process. Besides this, their algorithm is similar to GSA1, but the analysis is much deeper. Of course, as they have to solve an LP, it gives non-linear running time, and needs global information, so does not yield a local algorithm. However we consider this result important, they could break the 3/2-barrier.

We had a conjecture given forth in Reykjavík [K1 2008], stating that a simple modification and repetition of GSA2 gives 3/2-approximation for the general problem. It was (partially) answered by McDermid [M 2009], who gave the first 3/2-approximation for the general case. He used GSA1 (and not GSA2), but not with simple repetitions, at some points he stopped the main algorithm, constructed an auxiliary graph, and solved a maximum matching problem on it. In short, he used novel and rather complicated techniques, and so his algorithm needs $O(n\sqrt{n}|E|)$ running time (where $n=|U|+|W|$), and his algorithm also needs global information, so cannot be converted to a local algorithm.

Recently Paluch [P 2010][1] gave a new 3/2-approximation algorithm for the general case, claiming (without a proof) a linear running time. Her algorithm was still quite complicated, uses many concepts, and the analysis was also lengthy. It was not shown that her algorithm is local, but it can really be converted to a linear-time local algorithm with some more efforts, like as in Section 4.

[1] Parallel to this work she made significant simplifications, and after releasing the technical report version [K-tech 2011] of our algorithm, she made some further changes, see the newest version of Paluch's algorithm in [P 2011].

In Section 2 we reformulate Algorithm GSA1 of [K1 2008, K2 2008, K 2009], then in Section 3 we give a simple linear-time 3/2-approximation, with a simple proof of correctness. We lean on two important ideas of Paluch. Our new algorithm is a slightly modified version of GSA1 given in the next section, thus it is also very reminiscent of the traditional Algorithm GS. In Section 4 we detail how our new algorithm can be implemented to run in linear time, both for the global and the local version. Finally, in Section 5 we reformulate this algorithm for the Hospitals/Residents problem.

2 Men have strictly ordered lists

In this section we suppose that the lists of men are strictly ordered. From now on a man can be a lad, a **bachelor** or an old bachelor, where we use the term "bachelor" for a man who was refused by all acceptable women once, but in this setup he remains active and starts again to propose every woman on his recovered list. If there are two men, m_1 and m_2 with the same priority on a woman w's list, and m_1 is a lad but m_2 is a bachelor, then w **prefers bachelor** m_2 to **lad** m_1. In the description of the algorithm, differences from Algorithm GS are set in boldface.

Algorithm GSA1

While there exists an active man m, he proposes to his favorite woman w. If w accepts his proposal, they become engaged. If w refuses him, m will delete w from his list, and will remain active.
When a woman w gets a new proposal from man m, she always accepts this proposal, if she is a maiden. She also accepts this new proposal, if she prefers m to her current fiancé. Otherwise she refuses m.
If w accepted m, then she refuses her previous fiancé, if there was one (breaks off her engagement), and becomes engaged to m.
If m was engaged to a woman w and later w refuses him, then m becomes active again, and deletes w from his list.
If the list of m **becomes empty for the first time, he turns into a bachelor, his original list is recovered, and he reactivates himself. If the list of** m **becomes empty for the second time**, he will turn into an old bachelor and will remain inactive forever.

This simple algorithm runs in $O(|E|)$ time, as there are at most $2|E|$ proposals altogether[2]. It is easy to see that Algorithm GSA1 gives a stable matching M.

Theorem 2 (2008) *If men have strictly ordered preference lists, M is the output of Algorithm GSA1 and M_{opt} is a maximum size stable matching then $|M_{\text{opt}}| \leq \frac{3}{2} \cdot |M|$.*

Proof. 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. An alternating path component is called augmenting path if both end-edges are in M_{opt}. An augmenting path is called short, if it consists of 3 edges (see Figure 1). 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 a short augmenting path.

We claim that a short augmenting path 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' is a maiden,

[2] but see Section 4 for the details

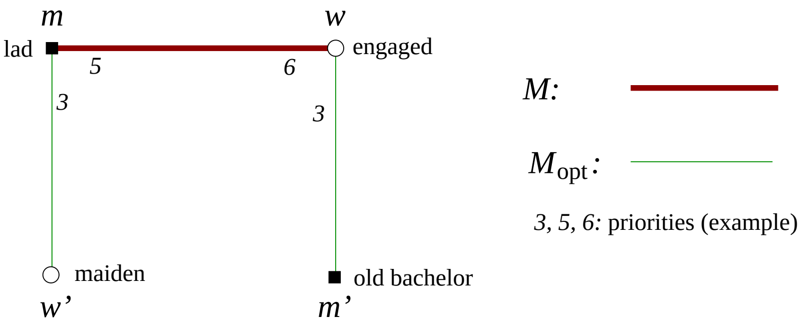


Figure 1: A short augmenting path

thus she never got a proposal during Algorithm GSA1. Consequently m is a lad, who prefers w to w'. As the algorithm finished, m' is an old bachelor, so he proposed to w also as a bachelor (see Figure 1), but w preferred m to m'. Consequently w strictly prefers m to m'. However, in this case edge mw blocks M_{opt}, a contradiction. □

3 The new algorithm for general stable marriage

For the new algorithm we use the following terms, most of which are familiar to the reader. A man can either be a *lad*, or a *bachelor*, or an *old bachelor.* A lad or a bachelor can either be *active* or *engaged.* If women w_1 and w_2 have the same priority on m's list, and w_1 is maiden but w_2 is engaged, then m **prefers maiden** w_1 to **engaged** w_2. An engaged man is **uncertain**, if his list contains a woman he prefers to his actual fiancée (this can happen, if there were two maidens with the same highest priority on m's list, and m became engaged to one of them).

A woman can either be *maiden* or *engaged.* An engaged woman is **flighty**, if her fiancé is *uncertain.* If there are two men, m_1 and m_2 with the same priority on a woman w's list, and m_1 is a lad, but m_2 is a bachelor, then w **prefers bachelor** m_2 to **lad** m_1.

At the beginning every man is a lad and every woman is a maiden. In the description of the algorithm, differences from GSA1 are set in boldface.

New Algorithm

While there exists an active man m, he proposes to his favorite woman w. If w accepts his proposal, they become engaged. If w refuses him, m will delete w from his list, and will remain active.
When a woman w gets a new proposal from man m, she always accepts this proposal, if she is a maiden **or a flighty engaged woman**. She also accepts this proposal, if she prefers m to her current fiancé. Otherwise she refuses m.
If w accepted m, then she refuses her previous fiancé, if there was one (breaks off her engagement), and becomes engaged to m.
If m was engaged to a woman w and later w refuses him, then m becomes active again, and deletes w from his list, **except if** m **is uncertain, in this case** m **keeps** w **on the list**.
If the list of m becomes empty for the first time, he turns into a bachelor, his original list is recovered, and he reactivates himself. If the list of m becomes empty for the second time, he will turn into an old bachelor and will remain inactive forever.

After the algorithm finishes, the engaged pairs get married and form matching M.

Lemma 1 *When a woman gets the first proposal, she becomes engaged, and will never become maiden again. A woman can become flighty only after the first proposal she got. After the second proposal a woman can never be flighty. If a woman changes her fiancé, then she always prefers the new fiancé to the previous one, except if she was flighty, when she may refuse a preferred man, but in this case she remains on the refused man's list.*

Proof. The only statement that needs a proof, is that after getting the second proposal a woman w cannot be flighty. However, in this case she got the last proposal as an engaged woman, so either she kept her first fiancé, consequently she was not flighty, or her new fiancé m could not be uncertain. (If there is a maiden w' on the list of m with the same priority as w, then m would prefer w' to w, so he would propose to w' first.) □

Lemma 2 *The matching M given by the new algorithm is stable.*

Proof. For suppose mw is a blocking pair. If w is maiden then she did not get any proposals, so m did not reach her when he processed his list, consequently he is engaged to a preferred woman w' (though it can be the case, that w' is flighty and now m would prefer w to w', but he does not strictly prefer her, so pair mw cannot be blocking).

If m is not married then he is an old bachelor, consequently he proposed to w at least twice. By the previous lemma the priority of the fiancé of woman w is monotonically increasing after the second proposal she got. So the husband of w cannot be strictly less preferred than m.

Finally suppose both m and w are married, the wife of m is w', and the husband of w is m'. As pair mw is blocking, m strictly prefers w to w', so m also proposed to w and she refused him. If at the time of this refusal w was not flighty, then she got finally a husband not worse than m. If she was flighty that time, then she remained on the list of m, so m proposed to her again before w'. In all cases we came to a contradiction. □

Lemma 3 *There is no short augmenting path.*

Proof. Suppose $m'wmw'$ is a short augmenting path (see Figure 1). The algorithm finished, so m' is an old bachelor, but m is a lad, because w' remained a maiden. As M_{opt} is a stable matching, edge mw does not block it, so either w does not strictly prefer m to m', or m does not strictly prefer w to w'. In the second case – as w' was always a maiden – after m proposed to w and got engaged to her, he became uncertain, thus w became flighty. It is impossible that later w refuses m, because after this m would propose to the preferred maiden w' before he proposes again to w. So w remained flighty until the end, meaning that she got only one proposal, but this is impossible, because m' proposed to her twice.

Assume w does not strictly prefer m to m'. Observe, that when m becomes a bachelor, he has no maidens on his recovered list, consequently an uncertain bachelor cannot exist. Take the moment, when m' proposed to w as a bachelor. If w refused m' at this time, then she was not flighty, because a flighty woman never refuses a proposal. Thus by Lemma 1, husband m of w is not less preferred, but this is a contradiction, as a woman prefers bachelors to lads. □

These lemmas together with the next section give:

Theorem 3 *The new algorithm always gives a stable matching in linear time for the general problem, and is $3/2$-approximating.*

4 Implementation and running time

If the reader is not interested in implementation details and having truly linear-time algorithms, it is advised to skip this section.

The original Algorithm GS is thought to be a linear-time local algorithm by its definition, but that is not obvious at all, as when getting a proposal from a man m, a woman w must look up in some dictionary, what is the value of $\text{pri}(w, m)$. In order to implement Algorithm GS as a linear-time local algorithm, we first assume that the lists of men are sorted (this assumption can be weakened to requiring that the priorities are natural numbers not exceeding the length of the list, because in this case a man can do bucket (or counting) sort in linear-time), and moreover we have to make **any** one of following assumptions.

- The system is "wired" along acceptable pairs, which means here that when a man m sends a proposal to a woman w, she sees on which wire this call is coming in, and the priority $\text{pri}(m, w)$ is written on that wire. Or, equivalently, better fitting to our mobile phone centralized world, there are no wires, but when an accessible man m calls woman w, then not only his phone number (his index in U) is shown, but also his position in the phone-book of w, such that his index in w's array.
- Women can through dice, and so they can use the perfect hashing approach of [FKS 1984].
- Women has a black-box procedure, which on input m outputs in constant time $\text{pri}(w, m)$.
- Men has some extra knowledge, for each acceptable woman w they know their own position in the list of w, such that their index in w's array.

Remark. Gusfield and Irving in [GI 1989] made a stronger assumption, the existence of ranking arrays. For complete preference lists it is still equivalent to the above assumptions.

Here we may assume any of these assumptions, but we will concentrate (and use) the fourth one, because that fits the best in our description of the algorithm. To make the new algorithm linear-time and local, we must define the communication between acceptable pairs, as well as the data structure needed for the participants. In our local implementation of the new algorithm all men run the same algorithm, as well as all women.

First we describe the algorithm of an arbitrary woman w. She stores 3 non-changing arrays, her own status (maiden or engaged), and if she is engaged then she also stores the name, priority and status of her fiancé. The first array U_w contains all the acceptable men in arbitrary order of priorities. The second array PR_w contains the corresponding priorities at w. The third array I_w stores the relative positions, such that if w is in the ith position in the list of man $U_w(j)$ then $I_w(j) = i$ (this list is initialized at the beginning with accepting messages from men).

Woman w gets proposals in the form $(m, i, \text{status}(m))$, where $\text{status}(m)$ can be one of lad or bachelor, and i is the index in w's array U_w where she stores m. If w is maiden and gets a proposal from man m then she changes her status to engaged, she stores $(m, \text{pri}(w, m), \text{status}(m))$, and she tells all the men in her list about her new status (engaged), together with her index stored in I_w. If she later gets a proposal from man m' then she first asks his fiancé m, whether he is uncertain. Now she has all the information needed to make a decision, she stores the name, priority and status of the new fiancé, and she sends the message "refused" to the non-preferred man.

An algorithm of a man m is slightly more complicated due to continuous reordering needed in his list, and the fact that he must always know whether he is uncertain. He stores 3 non-changing arrays (see bellow), a changing Boolean array B_m, a dequeue (double-ended queue), two pointers, the number P_m of preferred women (relative to his fiancée) and his status (originally lad). The first array W_m contains all the acceptable women in non-increasing order of priorities. The second array PR_m contains the corresponding priorities at m. The third array I_m stores the relative positions, such that if m is in the ith position in the list of woman $W_m(j)$ then $I_m(j) = i$ (by our assumption these are known at the beginning). He keeps $B_m(j) =$True, if $W_m(j)$ is a maiden, and False otherwise. This array is initialized to all-True, and an item is changed to False, when he gets the corresponding message from a woman.

The first pointer f_m points to the first woman of the current tie, and the second one n_m points to the first woman on m's list, who has (strictly) lower priority than f_m. At the beginning, for all acceptable women $w = W_m(j)$, man m sends a message (m, i, j) to w, where $i = I_m(j)$ (and woman w stores $I_w(i) = j$). When m is a lad, he considers the first woman on his list, f_m points to her (such that $f_m = 1$), and remember her priority p. He scans his list until finding a woman with lower priority than p, and changes n_m pointing to this woman (if he reaches the end of the list then he define $n_m = \infty$). While scanning he checks every woman w whether she is a maiden. If yes then he puts w in front of his dequeue, else (if w is engaged) he puts w at the end of his dequeue. Meanwhile he counts the maidens in the dequeue and stores this number in P_m. Whenever he gets a message "engaged" and has to change the corresponding value of B_m, he also checks whether the sender is in the dequeue (such that for her index j we have $f_m \leq j < n_m$), and if yes, then he decreases P_m. Whenever he is asked whether he is uncertain, he returns yes, if and only if $P_m > 0$. Then he takes the first element out of the dequeue, and proposes to this woman.

Whenever man m gets a refusal from w, he first checks whether $P_m > 0$, if yes, he puts back w at the end of the dequeue, and he takes the first element out of the dequeue and checks whether she is a maiden. If this is not the case then he puts back this woman at the end of his dequeue, and takes the next one from the front. Otherwise he proposes to this woman.

Otherwise, if $P_m = 0$ then he checks whether the dequeue is empty. If not then he simply takes the first woman from the dequeue and proposes to her.

When the dequeue is empty, he rather proceeds as follows. He changes f_m to n_m and starting from this new f_m he makes a new dequeue the same way as above (calculating P_m meanwhile), and adjust his pointer n_m. Then he processes the new dequeue as above. If $f_m = \infty$ at the first time then he changes his status to bachelor, and reset f_m to 1. If $f_m = \infty$ at the second time then he changes his status to old bachelor, and finishes.

From this local algorithm the linear-time global algorithm also follows easily, only we have to get rid of our assumptions. This can be done using the famous linear-time radix sort, as follows. First we sort triplets of form $(\text{pri}(m, w), m, w)$ for all acceptable pairs mw (this set of triplets can be collected from men). After sorting we scan this list, and build up the sorted arrays W_m of men easily.

Next every woman w subscribe each quadruplet $(m, w, 0, i)$ to the center, where i is the index of man m in w's list. And every man m subscribe each quadruplet $(m, w, j, 0)$ to the center, where j is the index of woman w in m's list. The center sorts these quadruplets and fuses neighboring pairs getting a list of quadruplets (m, w, j, i), and stores $I_m(j) = i$.

After these detailed descriptions it is obvious that both the local and the global algorithms run in linear-time.

5 Generalizations to the Hospitals/Residents problem

It is well-known, that algorithms for the one-to-one model can be easily converted to corresponding algorithms (for example, by cloning) for the many-to-one problems (see e.g., [K 2009]). This new algorithm can be also generalized to the many-to-many model, but we leave this generalization to the full version, so here we only detail the many-to-one case, because of the great importance and many practical usage of this model.

In the *Hospitals/Residents* problem, (also called Colleges/Students problem, and by many other names) 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 at most one in F, and each hospital w has degree at most $c(w)$ in F. 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 hospital w is *full* if $|F(w)| = c(w)$, and otherwise *under-subscribed*. Here a pair mw 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 unassigned or $\mathrm{pri}(m, w) > \mathrm{pri}(m, F(m))$, and
- w is either under-subscribed or $\mathrm{pri}(w, m) > \mathrm{pri}(w, m')$ for at least one resident $m' \in F(w)$.

An assignment is *stable* if there is no blocking pair.

First we consider the case when ties can only reside on hospitals lists. In [K1 2008, K2 2008, K 2009] it was shown, that if the preference lists of residents are strictly ordered then an easy modification of GSA1 (called HRGSA1) gives a 3/2-approximation in linear time. If we consider the resident-proposal version of the new algorithm, we can observe, that as there are no ties on the resident's side, no uncertain resident exists, consequently the new algorithm runs equivalently to HRGSA1. This algorithm is the same as that of Gale and Shapley with only one modifications: if a resident is refused by all hospitals for the first time, he/she gets an extra score of a half point (raising the priority by one half at every hospital; the same effect, as when a man becomes a bachelor), the list is recovered, reactivates himself/herself and starts making applications from the beginning of his/her list.

However the new algorithm makes possible to run the Hospital-proposal version as well. It will also give now a 3/2 approximation, and expectedly it results in a stable assignment that is better for the hospitals than the result of the resident-proposal scheme. This statement should be tested by some empirical future work. From now on hospitals play the role of men and residents play the role of women. We detail this algorithm below.

At the beginning all hospitals have the sorted list of its applicants. A hospital w prefers resident m to resident m', if m has strictly higher priority, or they have the same priority ($\mathrm{pri}(w, m) = \mathrm{pri}(w, m')$) and m' has got some offer, but m has not. A hospital w is active, if it is under-subscribed, in this case $f(w)$ denotes the number of free places (capacity minus the number of non-refused offers it made). In this case it makes an offer to the favorite resident on its list. A hospital m is **uncertain about the offer for** m, if there is a resident m' still on its list, whom it prefers to m.

A resident m is either **unoffered** or **offered**, if offered, he/she is called **precarious**, if his/her current offer is uncertain. A resident always accepts an offer, if either he/she is unoffered, or he/she is precarious, or if the new offer is better for him/her than the previous one.

Hospital-proposing Algorithm

While there exists an active hospital w, it offers to his favorite resident m. If m refuses this offer, w will delete m from its list, and will remain active.
When a resident m gets a new offer from hospital w, he/she always accepts this offer, if he/she is unoffered or precarious. He/she also accepts this offer, if he/she prefers w to the hospital he/she is actually assigned. Otherwise he/she refuses w.
If resident m accepted w, then he/she refuses his/her previous offer, if there was one.
If hospital w had offered to resident m and later m refuses it, then w deletes m from its list, except if m was precarious, in this case w keeps m on the list.
If the list of hospital w becomes empty then it remains inactive forever.

After the algorithm finishes, the assignment is made along the non-refused offers.

Similarly to Section 4, this algorithm can also be implemented as a linear-time local or global algorithm, we leave the details to the full version.

Unfortunately these algorithms do not give better approximation ratio than HRGSA1. We think that no linear time local algorithm can give better approximation ratio than 3/2.

Needless to say that our two algorithms (the Resident-proposal and the Hospital-proposal ones) equipped with the full machinery of the new algorithm, also can give a 3/2-approximation, when we allow ties on both sides. This also has some applications (for example, when residents have no preferences, only a list of acceptable hospitals). The only change we have to make, that after the list of a hospital gets empty the first time, it should become to an advantaged hospital, and starts proposing from its recovered list. And naturally, if for a resident m two hospitals are originally alike, then he/she prefers the advantaged one.

References

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

[FKS 1984] M. L. Fredman, J. Komlós, E. Szemerédi, Storing a Sparse Table with 0(1) Worst Case Access Time *J. ACM* **31** 3 (1984) pp. 538–544.

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

[HIIMMMS 2003] 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.

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

[HIMY 2007] 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.

[IM 2007] 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

[IM 2009] R. W. Irving, D. F. Manlove, Finding Large Stable Matchings *Journal of Experimental Algorithmics* **14** (2009) DOI: 10.1145/1498698.1537595

[IMMM 1999] 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* **LNCS 1644** (1999) pp. 443–452.

[IMY 2007] 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.

[IM 2008] K. Iwama, S. Miyazaki, A survey of the stable marriage problem and its variants. *Proceedings of the International Conference on Informatics Education and Research for Knowledge-Circulating Society*, IEEE Computer Society, Washington (2008) pp. 131–136.

[IMY 2010] K. Iwama, S. Miyazaki, H. Yanagisawa, A 25/17-Approximation Algorithm for the Stable Marriage Problem with One-Sided Ties *ESA '10, LNCS* **6347** (2010) pp. 135–146, DOI: 10.1007/978-3-642-15781-3_12

[K1 2008] Z. Király, Better and simpler approximation algorithms for the stable marriage problem *Proceedings of MATCH-UP 2008: Matching Under Preferences – Algorithms and Complexity, Satellite workshop of ICALP, July 6, 2008, Reykjavík, Iceland* pp. 36–45.

[K2 2008] Z. Király, Better and Simpler Approximation Algorithms for the Stable Marriage Problem *ESA 2008, Lecture Notes in Computer Science* **5193** (2008) pp. 623–634.

[K 2009] Z. Király, Better and Simpler Approximation Algorithms for the Stable Marriage Problem *Algorithmica* **60** (1), DOI: 10.1007/s00453-009-9371-7, (2009 (online), 2011), pp. 3–20.

[K-tech 2011] Z. Király, Approximation of Maximum Stable Marriage *Egres Technical Report TR-2011-03*, www.cs.elte.hu/egres/

[MIIMM 2002] 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.

[M 2009] E. J. McDermid A $\frac{3}{2}$-approximation algorithm for general stable marriage *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece* **LNCS 5555** (2009) pp. 689–700.

[P 2010] K. Paluch Faster and simpler approximation of stable matchings arXiv:0911.5660v2

[P 2011] K. Paluch Faster and simpler approximation of stable matchings arXiv:0911.5660v4

[Y 2007] H. Yanagisawa, Approximation Algorithms for Stable Marriage Problems *PhD Thesis* www.lab2.kuis.kyoto-u.ac.jp/~yanagis/thesis_yanagis.pdf (2007)

[Y 2008] H. Yanagisawa, Personal communication (2008)

Strong stability in contractual networks and matching markets

Alexander Teytelboym[†]

1 Introduction

Matching markets and networks are omnipresent in economic activity.[1] A network is a structure of connections between agents. A social network can describe friendships and a research network can show how firms collaborate on research and development (R&D). A two-sided matching market, however, consists of two types of heterogeneous agents - firms and workers, doctors and hospitals, or financiers and inventors - who have preferences over productive matches with agents of the other type. Matching markets can be one-to-one (where one inventor may be matched to one financier), many-to-one (where a financier may match with several inventors, but an inventor may match with only one financier) or even many-to-many (where several inventors may match with several financiers). Here we will consider multilateral matching markets where the same set of agents may form contracts to participate in separate ventures (Teytelboym (2012) illustrates this).

Although networks and matching markets have a common graph-theoretic structure (a network is essentially a one-sided matching market), the two concepts are addressed in separate literatures. Yet there is key theme in both literatures: stability. Understanding stability is crucial to predicting what network or matching is most likely to form. Here we focus on a stability concept, which allows for arbitrary group deviations from a particular contract allocation. In particular, we will say that a network or contract allocation in the matching market (a 'matching' if there are no contracts) is *strongly stable* if no group of agents can deviate, drop some of their contracts, form new contracts among themselves, and as a result all be made strictly better off. The main advantage of using this stability concept is that if a matching or a network is strongly stable then we can expect this contract allocation will persist.

The network and matching literatures approach the question of stability differently. Network theorists attempt to find conditions for the ways in which agents can allocate divisible surplus on the network. Matching theorists deal with allocation of indivisible objects (such as contracts) so they impose conditions on the individual preferences of agents, which guarantee a stable matching. The most important such condition, called 'substitutability', was introduced by Kelso and Crawford (1982). The following example illustrates this concept. Suppose a venture capitalist (VC) faces two inventors A and B and ideally she would fund both. Now suppose inventor B goes bankrupt and is no longer available. We say that the VC's preferences are substitutable if she would still be willing to fund inventor A. A large literature (discussed below) shows that in order to ensure stability

[†] Department of Economics, University of Oxford. I owe an enormous debt of gratitude to my supervisor Vincent Crawford for his patience, guidance, and support. I would also like to thank Marek Pycia, Francis Dennig, and the discussants at the Nuffield College *Learning, Games and Networks* workshop for their comments. Please email any comments to alexander.teytelboym@economics.ox.ac.uk

[1] Roth and Sotomayor (1990) offer an excellent treatment of the classic results in matching theory. Jackson (2008) and Goyal (2009) present recent surveys of the networks literature.

in any matching market the preferences of agents on both sides of the market (i.e. preferences of the VC and the inventors) must be substitutable. Yet substitutability is a rather strong condition on individual preferences. It would rule out the following case: the VC still wants to fund both projects, but when inventor B drops out, she wants to fund neither. This could be because the VC now regards inventors A and B as complementary: the technologies they developed are only marketable when sold together. Without substitutability stable matchings often cease to exist. However, even if agents' preferences are substitutable, but they care about agents on their side of the market ('peer effects'), stable matchings are no longer guaranteed (see example 2.1 in Echenique and Yenmez (2007)). In our example, if inventor B wants to work with the VC as long as the VC does not fund inventor A, peer effects will be present. Preferences over agents on the same side of the market can be viewed as a type of externality. The prevalence complementarities and peer effects in matching markets and networks is illustrated by the following examples.

Knowledge spillovers and complementarities in production networks

As economies grow and specialize, they exhibit more complements in production and more substitutes in consumption. In *The Wealth of Nations* Adam Smith estimated that in a pin factory ten specialized workers produced 4,800 pins a day each, whereas any one of them working alone would produce:

> certainly, not the two hundred and fortieth, perhaps not the four thousand eight hundredth part of what they are at present capable of performing, in consequence of a proper division and combination of their different operations (Smith (1776), Book 1, Chapter, 1)

Modern production of sophisticated technology involves assembly of various complementary components, which may be manufactured by different firms around the world.[2] Ostrovsky (2008, p. 914) notes that:

> In some industries (e.g., construction), firms along supply chains combine several complementary inputs to produce final goods, with inputs themselves consisting of multiple complementary parts, many of them heterogeneous, complex, and an important part of the final cost of the outputs.

When firms innovate and specialize, their technologies "spill over" as their competitors imitate and improve them. Hence, firms may make their R&D decisions strategically depending on their position in the production network (Baker et al., 2008). Venture capital funds exploit these features of modern technologies by providing complementary inputs of capital and managerial experience across a portfolio of start-ups, which are encouraged to share each other's technologies (Hellman, 1998). The way profits are divided among collaborating firms is crucial for stability of alliances.

[2] For example, Bloomberg reported in March 2011 that the Japanese earthquake "meaningfully impaired" Apple Inc's California-based assembly of consumer electronics. (Bloomberg, 29 March 2011, "Apple's Production of iPads and iPhones May Be Hurt by Japan Earthquake" by Ian King and Adam Satariano)

Peer effects in schools and organisations

Peer effects are ubiquitous in many organisational networks and matching markets. Academics do not simply pick the university that offers the highest salary - they care about how prolific the other members of the department are. Parents want to make sure that at school their children are surrounded by good peers. Workers in a firm may be assigned to work simultaneously in several teams. They want to pick the tasks that give them the highest chance of promotion, but, other things being equal, they prefer to be in a team with friendly colleagues. Complementarities and peer effects are often present simultaneously in organisations. No football team can win without a goalkeeper and no goalkeeper wants to play for a team without a good defender.

The current literature offers only partial solutions to the problem of existence of strongly stable matchings and networks when complementarities and externalities are present. The contribution of the present paper is twofold. First, we present a necessary and sufficient condition for the formation of *strongly stable* networks. The condition, called *strong pairwise alignment*, states that for any two networks, preferences of agents who are members of every contract on both networks must be identical. This condition subsumes and extends known sufficient conditions for strong stability on networks and allows for complementarities and externalities.

We then apply our results to a multilateral matching market with finite contracts. We demonstrate that if contractual language is sufficiently fine (e.g. agents can specify contractual terms sufficiently finely to accommodate any pattern of strict preferences), then strong pairwise alignment is also necessary and sufficient for strongly stable allocation in multilateral matching markets. This challenges some recent results in matching theory and shows that a) substitutability of preferences is not necessary for a strongly stable matching and b) stable matchings are not preserved whenever contractual language becomes more coarse. The rest of this paper is organised as follows. Section 2 presents the networks model and gives two motivating examples. Section 3 extends this to a multilateral matching market model with finite contracts. Section 4 concludes. Proofs, a literature review, and further examples can be found in Teytelboym (2012).

2 Networks

We begin by considering contractual networks between partners engaged in research.

2.1 Ingredients

Agents form a finite set N. We define a network (a hypergraph) as a pair $H \equiv (N, Y)$ where $Y \subseteq X \equiv 2^N \setminus \emptyset$. Any $y \in Y$, such that $|y| \geq 2$, is called a hyperedge, which represents a separate *contractual agreement* (or *contract*) between the agents $i \in y$.[3] We use hypergraphs in order to capture that fact that some of the agents already linked by a contractual commitment may want to form separate contractual links.

[3] Hypergraphs are simple generalization of graphs, where all hyperedges Y have cardinality 2 and represent edges connecting vertices. All our results apply to graphs. See examples below.

The set of all possible hypergraphs on N is $\mathcal{H}$. Any two agents i and j, who belong to the same contract y are called partners. Denote by $a(y) \equiv y \subseteq N$ the agents associated with a contract Y. Then $a(Y) \equiv Y \equiv \bigcap_{y \in Y} \{a(y)\}$ would denote the set of agents who belong to every contract in Y and by $Y_i = \{y \in Y | i \in y\}$ the set of contractual agreements which agent i is part of. The reason why $a(y) \equiv y$ in the networks framework is that every contract is defined by the set of agents who are part of it. This assumption will be relaxed in the next section, but we keep the notation consistent. The set of contracts that every member of $S \subseteq N$ is part of is denoted $Y_S = \bigcap_{i \in S} Y_i$ and the set of all contracts that S is part of is $\mathcal{Y}_S = \bigcup_{i \in S} Y_i$. Hence, $\{i, j\} = S$ are partners if $Y_S \neq \emptyset$.

Any two agents are connected if there exists a path $(i, i_1, i_2 \dots i_k, j)$ such that $i, i_1 \in y^0$, $i_1, i_2 \in y^1$ and $j \in y^k$ for some $y^0 \dots y^k \in Y$. If any two distinct agents are connected then the hypergraph is connected. If the hypergraph is not connected, then it can be partitioned into maximally connected components $\pi \in \Pi(H)$ (elements of the partition). A component is trivial if it consists of one agent. If all components are trivial, the hypergraph is empty. $H(\pi)$ denotes a subhypergraph H on π.

A value function v defines the surplus (profit) of each contractual agreement on a network:

$$v : \mathcal{H} \to [0, \infty)$$

where $v(\emptyset) = 0$. The set of all value functions is $\mathcal{V}$. The payoffs of the agents are determined by a strictly increasing, continuous allocation rule, $\gamma_i(H, v)$ where $\gamma : \mathcal{H} \times \mathcal{V} \to \mathbb{R}^{|N|}$ is the set of all allocation rules. Thus the allocation rule determines how the surplus (profit) from a set of contractual agreements is shared between partners. Define $q : \mathcal{H} \times [0, \infty) \to \mathbb{R}^{|N|}$, such that $q(H, v(H)) = \gamma(H, v)$ for all $H \in \mathcal{H}$ and $v \in \mathcal{V}$. We assume that $lim_{w \to \infty} q_i(H, w) = \infty$ for all $w = v(H)$.

2.2 Stability

We now extend the definition of strong stability to networks represented by hypergraphs. Consider a group deviation which *induces* $H' \equiv (N, Y')$ by $S \subseteq N$ from $H \equiv (N, Y)$ such that:

- New agreements are formed by deviating agents only: $Y_T \in Y'$ and $Y_T \notin Y \implies T \subseteq S$
- Agents can sever any existing contractual agreement: $Y_T \in Y$ and $Y_T \notin Y' \implies T \bigcap S \neq \emptyset$. But that means agents cannot break any contracts with those outside S.

Definition 1 A network H is **strongly stable** with respect to the allocation rule γ and value function v if there is no group deviation such that for all $i \in S$, $\gamma_i(H', v) > \gamma_i(H, v)$ where H' is a network induced by a group deviation S from H.

This definition of stability allows for any strictly profitable deviation by any group of agents. It generalises the definition of strongly stable networks found in Dutta and Mutuswami (1997) to hypergraphs.

2.3 Example

In this section, we illustrate why strong pairwise alignment is important for network stability. Suppose that four firms $N = \{1, 2, 3, 4\}$ can collaborate on research. Expected profits from each collaboration are known ex ante. Once the contracts are signed, the agreed profit allocation rule is enforced when the profits are received. How should the firms allocate expected profits to ensure that a strongly stable research network can be found regardless of the distribution profits that is possible in the network? In these examples, we will restrict ourselves to the standard case studied in the network literature where agents can be in one research collaboration at a time ($|Y_i| = 1$, see Figure 1). We will also assume (without any loss of generality) that any connected component, which contains the same researchers, has the same value.

The profits from any network which represents a research collaborations are represented by the following (superadditive and monotonic) value function $v \in \mathcal{V}$:

$v(\{1\}) = v(\{1, 3\}) = v(\{2, 3\}) = v(\{3, 4\}) = 30$; $v(\{1, 2\}) = 40$; $v(\{4\}) = 5$; $v(\{3\}) = 0$; $v(\{2, 4\}) = 50$; $v(\{1, 4\}) = v(\{1, 3, 4\}) = 60$; $v(\{1, 2, 4\}) = 80$; $v(\{2, 3, 4\}) = 66$; $v(\{1, 2, 3\}) = 93$; $v(\{1, 2, 3, 4\}) = 98$.

The payoff structure reflects the fact that there may be complementarities and technological spillovers in the research network.

2.3.1 Instability - Shapley value

We first suppose that firms agree (ex ante) to allocate the profits according to the Shapley value (Shapley, 1953) of each network component. It is well known that the Shapley value does not always produce a surplus allocation that accommodates a strongly stable network. Table 1 shows how the payoffs are allocated in some productive networks.

Table 1: Values and profits in a strongly unstable network

	Firm profit (γ_i)			
Network value/Agent	**1**	**2**	**3**	**4**
$v(\{1, 4\}) = 60$	$42\frac{1}{2}$	.	.	$17\frac{1}{2}$
$v(\{2, 4\}) = 50$	.	$27\frac{1}{2}$	.	$22\frac{1}{2}$
$v(\{1, 2, 3\}) = 93$	41	31	21	.
$v(\{1, 2, 3, 4\}) = 98$	$35\frac{1}{3}$	29	14	$19\frac{2}{3}$

In this case there is no strongly stable network. If firms 1 and 4 form a network, 2 and 4 would have an incentive to deviate. However, then firms 1, 2 and 3 would want to deviate to obtain a payoff of 5. But firms 1 and 4 would want to deviate back to their original network, forming a cycle. It can be easily checked that no other network is strongly stable.[4]

[4] According to Shapley value, the remaining network payoffs are as follows: $\gamma_1(\{1, 2\}) = 30$, $\gamma_2(\{1, 2\}) = 0$,

2.3.2 Stability - equal sharing

Let us now consider a network with the same value function, but where firms agree to share profits equally (Jackson and van den Nouweland, 2005).[5] See Table 2.

Table 2: Values and payoffs in a strongly stable network

	Firm profit (γ_i)			
Network value/Agent	**1**	**2**	**3**	**4**
$v(\{1,4\}) = 60$	30	.	.	30
$v(\{2,4\}) = 50$	.	25	.	25
$v(\{1,2,3\}) = 93$	31	31	31	.
$v(\{1,2,3,4\}) = 98$	$24\frac{1}{2}$	$24\frac{1}{2}$	$24\frac{1}{2}$	$24\frac{1}{2}$

The strongly stable network in this case is $\{\{1,2,3\},\{4\}\}$ and the firms' expected profit vector is $(30, 12\frac{1}{2}, 0, 12\frac{1}{2})$.[6] In this case the strongly stable allocation is also efficient, but as Dutta and Mutuswami (1997) this is not always the case (see footnote 5). Since the contract specifies the allocation of profit, once profits are received they cannot be transferred between agents.

2.4 Assumption on preferences

We now explain why the allocation according to Shapley value did not accommodate a strongly stable network, but equal sharing did. In order to do this, we now formally state our main assumption on the profit allocation rule.

Definition 2 For a given value function v, an allocation rule γ satisfies **strong pairwise alignment** if for any agents i and j who are partners in contracts $Y_{i,j} \in Y$ and $Y'_{i,j} \in Y'$ in $Y, Y' \subseteq X$ then $\gamma_i(H, v) \geq \gamma_i(H', v) \Longleftrightarrow \gamma_j(H, v) \geq \gamma_j(H', v)$.

The assumption states that whenever we take two different networks (sets of contracts represented as hypergraphs) with the same value function and any set of agents who are partners in

$\gamma_1(\{1,3\}) = 30$, $\gamma_3(\{1,3\}) = 0$, $\gamma_2(\{2,3\}) = 20$, $\gamma_3(\{2,3\}) = 10$, $\gamma_3(\{3,4\}) = 12.5$, $\gamma_4(\{3,4\}) = 17.5$, $\gamma_1(\{1,2,4\}) = 34\frac{1}{6}$, $\gamma_2(\{1,2,4\}) = 19\frac{1}{6}$, $\gamma_4(\{1,2,4\}) = 26\frac{2}{3}$, $\gamma_2(\{2,3,4\}) = 27\frac{5}{6}$, $\gamma_3(\{2,3,4\}) = 12\frac{5}{6}$, $\gamma_4(\{2,3,4\}) = 25\frac{1}{3}$, $\gamma_1(\{1,3,4\}) = 34\frac{1}{6}$, $\gamma_3(\{1,3,4\}) = 4\frac{1}{6}$, $\gamma_4(\{1,3,4\}) = 21\frac{2}{3}$. Hence, for the remaining networks: $\gamma_{1,2}(\{1,2,3\}) > \gamma_{1,2}(\{1,2\})$, $\gamma_1(\{1,4\}) > \gamma_1(\{1,3\})$, $\gamma_{2,3}(\{1,2,3\}) > \gamma_{2,3}(\{2,3\})$, $\gamma_{3,4}(\{1,2,3,4\}) > \gamma_{3,4}(\{3,4\})$, $\gamma_{1,2}(\{1,2,3\}) > \gamma_{1,2}(\{1,2,4\})$, $\gamma_{2,3}(\{1,2,3\}) > \gamma_{2,3}(\{2,3,4\})$, $\gamma_{1,3}(\{1,2,3\}) > \gamma(\{1,3,4\})$ and $\gamma_{1,2,3}(\{1,2,3\}) > \gamma_{1,2,3}(\{1,2,3,4\})$.

[5] The intuition would be the same if firms competed to gain monopoly rights to the technology they develop. Suppose that if firms spend c_i on research, they have probability $\frac{c_i}{\sum_i c_i}$ of getting the technology. Each firm expects to obtain $\frac{c_i}{\sum_i c_i} v(Y_i) - c_i$ and in a Nash equilibrium each firm's expected payoff is $\gamma_i(H, v) = v(Y_i)/|Y_i|^2$ (Tullock, 1980). The strongly stable network in this case is $\{\{1\},\{2,4\},\{3\}\}$ and the firms' expected profit vector is $(30, 12\frac{1}{2}, 0, 12\frac{1}{2})$. The resulting strongly stable allocation does not maximise surplus.

[6] If surplus is shared equally, the remaining network payoffs are as follows: $\gamma_4(\{3,4\}) = \gamma_3(\{3,4\}) = \gamma_3(\{2,3\}) = \gamma_2(\{2,3\}) = \gamma_1(\{1,2\}) = \gamma_1(\{1,3\}) = \gamma_3(\{1,3\}) = \gamma_2(\{1,2\}) = 15$, $\gamma_1(\{1,2,4\}) = \gamma_2(\{1,2,4\}) = \gamma_4(\{1,2,4\}) = 80/9$, $\gamma_2(\{2,3,4\}) = \gamma_3(\{2,3,4\}) = \gamma_4(\{2,3,4\}) = 22/3$, $\gamma_1(\{1,3,4\}) = \gamma_3(\{1,3,4\}) = \gamma_4(\{1,3,4\}) = 20/3$.

every contract in both networks, their preferences over the networks should be the same. In the first example, we can see that profit allocation according to the Shapley value does not satisfy strong pairwise alignment. Firm 1 prefers contractual agreement $\{1,2,3,4\}$ to $\{1,2,4\}$, but firm 4 has the opposite preference. In the second example strong pairwise alignment is satisfied for any two networks. For example, firms 1, 2 and 3 prefer contractual agreement $\{1,2,3\}$ to $\{1,2,3,4\}$; firms 1 and 2 both prefer contract $\{1,2,3,4\}$ to contract $\{1,2\}$, and so on (see footnote 6).

2.5 Stability result

We can now state our first stability result for networks.

Proposition 1 *A network $H \in \mathcal{H}$ is strongly stable with respect to the allocation rule γ and the value function $v \in \mathcal{V}$ if and only if the allocation rule satisfies strong pairwise alignment.*

Unlike Dutta and Mutuswami (1997) and Jackson and van den Nouweland (2005), we do not assume anything about component additivity of the value function or the component-balancedness, anonymity or decomposability of the allocation rule and their results are subsumed in Proposition 1.[7] In order to illustrate this, let us simply assume that the profit firms get equals to the sum of the contracts from separate contractual agreement it signs. Hence, the allocation rule is *contract-additive* if $\gamma_i(H,v) = \sum_{y \in Y_i} \gamma_i(H(N,y),v)$.

Corollary 1 *Suppose the allocation rule γ is contract-additive. A network $H \in \mathcal{H}$ is strongly stable with respect to γ and the value function $v \in \mathcal{V}$ if and only if the allocation rule satisfies strong pairwise alignment for any two contractual agreements.*

This makes it clear how the results from Pycia (2012) translate into a network structure. If the allocation rule is contract-additive, then equal sharing or Nash bargaining (with exogenous bargaining powers and fixed outside options) for profits within each contractual agreement will satisfy pairwise alignment, whereas Shapley (or Myerson) value division or Kalai-Smorodinsky bargaining would not (see further examples in the Appendix).

Finally, similarly to Dutta and Mutuswami (1997) and Pycia (2012), we can note that all strongly stable allocations can be supported by a strong Nash equilibrium.

3 Multilateral matching markets

Here we extend the network model to a matching market, where there are two types of agents, let us call them financiers and inventors. In order to produce profitable ventures inventors must be matched to financiers. Each financier can enter into several ventures with any group of inventors, but each venture will require a separate contractual agreement. We allow agents to have general

[7] In the hypergraph context a value function is component-additive if $v(H) = \sum_{\pi \in \Pi(H)} v(\pi)$ i.e. there are no externalities between components. An allocation rule is component-balanced if $\sum_{i \in \pi} \gamma_i(H,v) = v(\pi)$ (value allocated fully to the component members) and component decomposable $\gamma_i(H,v) = \gamma_i(H(\pi),v)$ for $\pi \in \Pi(H)$, and $i \in \pi$ (allocation in component is independent of the structure of other components).

preferences over their contractual agreements and partners. The most important distinction between the two models is that the contracts that agents can sign are now assumed to be finite.

3.1 Ingredients

We introduce a multilateral matching model with contracts, following Hatfield and Kominers (2011b). For concreteness, we could partition the finite set of agents $N = F \bigcup G$ into a set F of financiers and a set G of inventors. Let X denote a set of all contracts. An *contracual allocation* is a set of contracts $Y \subseteq X$. Each multilateral contract $y \in Y \subseteq X$ is now associated with at most one financier and at least one inventor. It is worth noting that unlike in the network model, the same set of agents can now sign several contracts. Denote by $a(y) \subseteq N$ the set of agents in contract y and so if $f \in a(y)$ the financier f is associated with a contract y. Then $a(Y) \equiv \bigcap_{y \in Y} \{a(y)\}$ would denote the set of agents who belong to every contract in Y. As before, $Y_g \equiv \{y \in Y | g \in a(y)\}$ is the set of contracts in Y associated with inventor g and for any $S \subseteq N$, $Y_S = \bigcap_{i \in S} Y_i$, and $\mathcal{Y}_S = \bigcup_{i \in S} Y_i$.[8]

The main distinction between a network model and a multilateral matching model is the restriction on how contractual agreements can be formed. In the network model a contractual agreement could be signed between any two agents. In the matching market, we preserve its two-sided nature by imposing the following limits on contractual feasibility.

Assumption 1 *No contract includes more than one financier.*

Assumption 2 *Every financier can sign a contract with at least one inventor.*

Assumption 3 *A financier cannot sign a single contract with every inventor.*

Define $\precsim_i$ is a weak preference relation (a complete preorder) over sets of contracts in 2^{Y_i} involving i. For a set of contracts Y_i written in a given contractual language X (see next section) this preference relation is denoted $\precsim_i^X \in \mathcal{R}_i^X$ where $\mathcal{R}_i^X$ is the domain of all possible preference ordering on X for i. The preference profile of all agents over X is then $(\precsim_i^X)_{i \in N} = \precsim_N^X$. Hence, $\precsim_N^X$ is an element of $\mathcal{R}^X = \Pi_{i \in N} \mathcal{R}_i^X$ - the preference domain under contractual language X. Note that the preference domain depends entirely on the cardinality of X i.e. how many possible different contracts agents can sign.

Definition 3 Contract allocation satisfies **strong pairwise alignment** with respect to a contractual language X if for any agents $i, j \in N$ and any allocations $Y, Y' \subseteq X$ where some sets of contracts $Y_{i,j} \in Y$ and $Y'_{i,j} \in Y'$ contain both i and j, then , $Y_{i,j} \precsim_i^X Y'_{i,j} \Longleftrightarrow Y_{i,j} \precsim_j^X Y'_{i,j}$.

This immediately generalizes our definition of strong pairwise alignment over networks. Whenever there is a choice of between two different sets of contracts, both of which include a financier and an inventor, the financier and the inventor must prefer the same contract. A multilateral partnership, studied in a potential function framework by Page Jr. and Wooders (2010) is an appropriate

[8] This multilateral matching market is a natural extension of the many-to-many matching model with bilateral contracts discussed by Hatfield and Kominers (2011a) and Hatfield et al. (2011).

example. Financiers and inventors are partners in the project, so they aim to maximize the value of the project. Similarly, if an inventor wants to pursue a management strategy, the financier of this venture will support her and vice versa.

3.2 Contractual language

In the multilateral matching market we no longer assume that contracts simply specify the shares of perfectly divisible profit. Instead we assume that contracts form a finite set. The rationale for this was given by Roth (1984): “salary cannot be specified more precisely than to the nearest penny, hours to the nearest second”. Contracts, however, can specify many aspects of a relationship between financiers and inventors. A contract would stipulate exactly how much equity each party gets, whether they get a seat on the management board and so on. In this paper, we will assume that financiers and inventors can determine the terms contractual relationship sufficiently precisely.[9]

Definition 4 Contractual language X is **fine** if for any $\precsim_N^X \in \mathcal{R}^X$, any agent $i \in N$, and three different sets of contracts $Y, Y', Y'' \subseteq X$ if $Y' \precsim_i Y''$ and $i \in a(Y)$, then there is a preference profile $\precsim_N^{*X} \in \mathcal{R}^X$ such that $Y' \precsim_i^{*X} Y \precsim_i^{*X} Y''$, and all agents' $\precsim_N^{*X}$-preferences between sets of contracts not including Y are the same as their $\precsim_N^X$-preferences.

Definition 5 Contractual language X is **very fine** if it is fine and:

- For any $\precsim_N^X \in \mathcal{R}^X$, any agent $i \in N$, and two different sets of contracts $Y, Y' \subseteq X$, there is a preference profile $\precsim_N^{*X} \in \mathcal{R}^X$ such that $Y \prec_i^{*X} Y'$ for all $i \in a(Y \bigcap Y')$ and all agents' $\precsim_N^{*X}$-preferences between sets of contracts not including Y are the same as their $\precsim_N^X$-preferences.
- For any $\precsim_N^X \in \mathcal{R}^X$, any agents i and j, and three different sets of contracts $Y, Y', Y'' \subseteq X$, $Y' \prec_i Y \sim_j Y''$, then there is a preference profile $\precsim_N^{*X} \in \mathcal{R}^X$ such that $Y' \prec_i^{*X} Y \prec_j^{*X} Y''$ and all agents' $\precsim_N^{*X}$-preferences between sets of contracts not including Y are the same as their $\precsim_N^X$-preferences.

The assumptions on contractual language stipulate that if the wage dimension of the contract is no longer divisible (the profit division is already specified to the neartest penny), there will be another contractual dimension (e.g. working hours) which will allow us to ‘separate’ agents' preferences between any two contract allocations. While most of the matching literature assumes that all

[9] The fineness condition is similar to saying that contracts between agents can be represented by simplicial complexes. We could also apply a theory of contractual language developed by Hatfield and Kominers (2011a), which extended the work of Roth (1984), to a many-to-many matching market. A contractual primitive is any possible agreement between a financier and an inventor. The set of contractual primitives between a financier $f \in F$ and a set of inventors $\mathcal{G} \subseteq G$ as $\omega(f, \mathcal{G})$ and $\Omega_f \equiv \bigcup_{f \in F} \omega(f, \mathcal{G})$ is the set of contractual primitives associated with financier f. A primitive allocation $\Lambda \subseteq \bigcup_{(f,\mathcal{G}) \in f \times (2^{\mathcal{G}} \setminus \emptyset)} \omega(f, \mathcal{G})$, where $\omega(f, \mathcal{G}) \bigcap \omega(f', \mathcal{G}') = \emptyset$ for $(f, \mathcal{G}) \neq (f', \mathcal{G}')$. Define the power set $\mathcal{P}(\omega(f, \mathcal{G})) = 2^{\omega(f,\mathcal{G})}$ as a collection of primitives between f and $\mathcal{G}$. A contract between f and $\mathcal{G}$ is an element of $\mathcal{P}(\omega(f, \mathcal{G})) - \emptyset$. For example, the primitives of the contract could specify the amount of equity ($e \in E$), management board structure ($m \in M$) and advertising strategy ($a \in A$) so $\omega(f, \mathcal{G}) = \{e, m, a\}$. Contract language $X \equiv \bigcup_{(f,\mathcal{G}) \in f \times (2^{\mathcal{G}} \setminus \emptyset)} X_{(f,\mathcal{G})}$ where $X_{(f,\mathcal{G})} \subseteq \mathcal{P}(\omega(f, \mathcal{G})) - \emptyset$ is a contract language for any financier and any subset of inventors. Hence, a contract language is a union of all possible relationships between all financiers and inventors. Λ is expressible in contract language X if there exists some $Y \subseteq X$ such that $\Lambda = \bigcup_{y \in Y} y$. If a contractual language cannot express an allocation, this means that this allocation is not permitted under this language.

preference relations between contracts are strict, our conditions on preferences are substantially less restrictive. Contractual language is very fine for the domain of all strict preference profiles. Hence, in the case of a strictly increasing, continuous allocation rule used to divide profits in the network model (described in the previous section) contractual language is very fine.

3.3 Stability

We say a contract allocation is strongly stable if no agent wants to drop his contracts and no group of agents can deviate, form contracts only among themselves, drop any of their existing contracts and strictly prefer the new allocation. It mimics the strong stability notion we used for contractual networks. It is essentially the strongest stability concept used in the matching market literature. If a strong group stable matching exists, then it is difficult to imagine how any deviation from could it be rational.

Definition 6 A contract allocation $Y \subseteq X$ is **strongly stable** with respect to contractual language X if it is:

1. Individually rational: for all $i \in a(Y)$, $Y_i = max_{\precsim_i^X}\{U \subseteq Y | y \in U \Rightarrow i \in a(y)\}$
2. There does not exist a non-empty, feasible $Z \subseteq X$ such that:

- $Z \bigcap Y = \emptyset$, and
- for all $j \in a(Z)$, there exists a $W_j \subseteq Z \bigcup Y$ such that $Z \subseteq W_j$ and $W_j \succ_j^X Y_j$

3.4 Main results

The two main results of this paper are the following:

Theorem 1 *If contractual language X is fine and all preference profiles in $\mathcal{R}^X$ satisfy strong pairwise alignment, a strongly stable contract allocation $Y \subseteq X$ exists.*

Theorem 2 *If contractual language X is very fine and a strongly stable contract allocation exists $Y \subseteq X$ then all preference profiles in $\mathcal{R}^X$ satisfy strong pairwise alignment.*

Teytelboym (2012) provides the proofs. These results show that a strongly stable contract allocation can exist for a matching market with complementarities, peer effects (or technological spillovers) and finite contracts as long as the preference profile of agents is suitably constrained. Proposition 1 is a clear consequence of Theorems 1 and 2 where $F = \emptyset$ because we have not ruled out that a contract has no financiers.

Our setting challenges several recent results in the matching literature.[10] First, in a doctor-hospital matching setting Hatfield and Kominers (2011a) shows that if preferences for one agent are

[10] Pycia (2012) shows that in a many-to-one setting that the usual comparative statics outlined in Crawford (1991) do not hold.

not substitutable, then "there exist substitutable preferences for the other doctors and hospitals such that no many-to-one stable allocation exists" (Theorem 12). While our results do not contradict it, we showed that substitutability of preferences is not necessary (in the usual mathematical sense) in a matching market as Hatfield and Kominers imply. Second, Hatfield and Kominers (2011a) show that a coarser language will support a strongly stable many-to-many matching whenever it exists (Theorem 3). The spirit of our results suggests the opposite. Without a fine contractual language a strongly stable contract allocation is not guaranteed when strong pairwise alignment is satisfied.

4 Conclusions and extensions

In this paper, we showed that a necessary and sufficient condition for the existence of strongly stable networks and contract allocations in two-sided matching markets is strong pairwise alignment. First, this condition generalized some of the results from the networks literature. Second, contrary to conventional wisdom in the matching literature, it showed that neither substitutability of preferences nor a coarse contractual language is essential for a strongly stable contract allocation. This paper also shows that an integrated approach to matching markets and networks can be very fruitful. It also suggests that there may be a deeper relationship between contractual language and production technologies in matching markets than previously thought.

An interesting extension of this model would be to design an efficient P-complete algorithm to find the strongly stable allocation where strong pairwise alignment is satisfied. Since most efficient algorithms rely on substitutability of preferences (Echenique and Oviedo, 2006) or on a many-to-one market structure (Echenique and Yenmez, 2007), they cannot be applied directly to this model. Finally, the results in this paper say nothing about the conflict between efficiency and stability common in the networks literature. This avenue of research could also be explored further.

References

Baker, G. P., R. Gibbons, and K. J. Murphy (2008). Strategic alliances: Bridges between "islands of conscious power". *Journal of the Japanese and International Economies 22*(2), 146–163.

Crawford, V. P. (1991). Comparative statics in matching markets. *Journal of Economic Theory 54*(2), 389–400.

Dutta, B. and S. Mutuswami (1997). Stable networks. *Journal of Economic Theory 76*(2), 322–344.

Echenique, F. and J. Oviedo (2006). A theory of stability in many-to-many matching markets. *Theoretical Economics 1*(1), 233–273.

Echenique, F. and M. B. Yenmez (2007). A solution to matching with preferences over colleagues. *Games and Economic Behavior 59*(1), 46–71.

Goyal, S. (2009). *Connections: An Introduction to the Economics of Networks.* Princeton University Press.

Hatfield, J. W. and S. D. Kominers (2011a). Contract design and stability in matching markets. Working paper, Harvard Business School.

Hatfield, J. W. and S. D. Kominers (2011b). Multilateral matching. Mimeo, Stanford GSB.

Hatfield, J. W., S. D. Kominers, A. Nichifor, M. Ostrovsky, and A. Westkamp (2011). Stability and competitive equilibrium in trading networks. Mimeo, Stanford GSB.

Hellman, T. (1998). The allocation of control rights in venture capital contracts. *RAND Journal of Economics 29*(1), 57–76.

Jackson, M. O. (2008). *Social and Economic Networks.* Princeton University Press.

Jackson, M. O. and A. van den Nouweland (2005). Strongly stable networks. *Games and Economic Behavior 51*(2), 420–444.

Kelso, A. S. and V. P. Crawford (1982). Job matching, coalition formation, and gross substitutes. *Econometrica 50*(6), 1483–1504.

Ostrovsky, M. (2008). Stability in supply chain networks. *American Economic Review 98*(3), 897–923.

Page Jr., F. H. and M. Wooders (2010). Club networks with multiple memberships and noncooperative stability. *Games and Economic Behavior 70*(1), 12–20.

Pycia, M. (2012). Stability and preference alignment in matching and coalition formation. *Econometrica 80*(1), 323–362.

Roth, A. E. (1984). Stability and polarization of interests in job matching. *Econometrica 52*(1), 47–58.

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

Shapley, L. S. (1953). A Value for n-Person Games. In H. W. H. Kuhn and A. W. Tucker (Eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press.

Smith, A. (1970, 1776). *An Inquiry into the Nature and Causes of The Wealth of Nations.* Penguin.

Teytelboym, A. (2012). Strong stability in contractual networks and matching markets. Mimeo, University of Oxford.

Tullock, G. (1980). Efficient rent seeking. In J. Buchanan, R. Tollison, and G. Tullock (Eds.), *Toward a theory of the rent-seeking society*, pp. 97–112. Texas A and M University Press, College Station.

Format B paper abstracts

A SUPPLY AND DEMAND FRAMEWORK FOR TWO-SIDED MATCHING MARKETS

EDUARDO M. AZEVEDO AND JACOB D. LESHNO

Two-sided matching markets are markets where agents care about whom they interact with in the other side. This includes several real world examples, such as colleges and students, or Internet content providers and advertisers. In these, and several other prominent examples, each agent on one side is matched to a large number of agents on the other side. However, little is known about special properties of such markets. To investigate this, we propose a variation of the Gale and Shapley college admissions model, where a finite number of colleges is matched to a continuum of students. Colleges' preferences over students are represented by a score. A student's score at different colleges need not be the same.

In both the continuum and the traditional discrete models, stable matchings have a very simple representation. Any stable matching can be described by a vector of threshold scores at each college, which we term a cutoff. We define a student's demand given cutoffs to be her favorite college out of the colleges for which her score is above the threshold. A set of cutoffs is said to clear the market if demand for colleges equals the supply of seats, or leaves empty seats in colleges with a cutoff equal to zero (the minimal score). The cutoff lemma guarantees that each stable matching corresponds to a market clearing cutoff, and vice versa. Cutoffs allow us to find the set of stable matchings by solving a set of simple market clearing equations.

We find that for almost every continuum matching market the following holds: (i) There is a unique stable matching. (ii) For any sequence of approximating discrete economies, the diameter of the set of stable matchings converges to 0. This complements previous findings that in large matching markets all stable matchings are very similar ([5], [4]).(iii) Moreover, the set of stable matchings converges to the stable matching of the continuum limit. (iv) Stable matchings of the continuum economy vary continuously with the parameters of the model. This theoretically supports using data and simulations to inform market design (as in [5]). (v) Stable matchings of discrete economies randomly generated from the distribution of agents in the limit converge almost surely to the limit economy's stable matching. This implies a simple characterization of the asymptotic behavior of several mechanisms, generalizing previous work on the random serial dictatorship mechanism.

References

[1] E. Azevedo. Imperfect competition in two-sided matching markets. Mimeo, 2010.

[2] E. Azevedo and J. Leshno. Can we make school choice more efficient? An example. Mimeo, 2010.

[3] D. Gale and L. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

[4] N. Immorlica and M. Mahdian. Marriage, honesty, and stability. In *SODA* 2005.

[5] A. Roth and E. Peranson. The redesign of the matching market for american physicians: Some engineering aspects of economic design. *The American Economic Review*, 89(4):748–780, 1999.

Extended Abstract. A full version of this paper is available at `http://www.people.fas.harvard.edu/~jleshno/papers/College_continuum.pdf`. Azevedo: Harvard University, azevedo@fas.harvard.edu. Leshno: Harvard University and Harvard Business School, leshno@hbs.edu.

Solutions for the Stable Roommates Problem with Payments

Péter Biró[1], Matthijs Bomhoff[2], Petr A. Golovach[3],
Walter Kern[2], and Daniël Paulusma[3]

[1] Institute of Economics, Hungarian Academy of Sciences; H-1112, Budaörsi út 45,
Budapest, Hungary, birop@econ.core.hu***

[2] Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, P.O.Box 217, NL-7500 AE Enschede
{m.j.bomhoff,w.kern}@math.utwente.nl

[3] School of Engineering and Computing Sciences, Durham University,
Science Laboratories, South Road, Durham DH1 3LE, UK
{petr.golovach,daniel.paulusma}@durham.ac.uk†

The stable roommates problem with payments has as input a graph $G = (V, E)$ with an edge weighting $w : E \to \mathbb{R}_+$ and the problem is to find a stable solution. A *solution* is a matching M with a vector $p \in \mathbb{R}_+^V$ that satisfies $p_u + p_v = w(uv)$ for all $uv \in M$ and $p_u = 0$ for all u unmatched in M. A solution is *stable* if it prevents *blocking* pairs, i.e., pairs of adjacent vertices u and v with $p_u + p_v < w(uv)$.

By pinpointing a relationship to the accessibility of the coalition structure core of matching games, we give a simple constructive proof for showing that every yes-instance of the stable roommates problem with payments allows a path of linear length that starts in an arbitrary unstable solution and that ends in a stable solution. Our result generalizes a result of Chen, Fujishige and Yang [1] for bipartite instances to general instances. We also show that the problems Blocking Pairs and Blocking Value, which are to find a solution with a minimum number of blocking pairs or a minimum total blocking value, respectively, are NP-hard. Finally, we prove that the variant of the first problem, in which the number of blocking pairs must be minimized with respect to some fixed matching, is NP-hard, whereas this variant of the second problem is polynomial-time solvable.

We pose the following two open problems. What is the computational complexity of Blocking Pairs and Blocking Value restricted to input graphs with unit edge weights?

References

1. B. Chen, S. Fujishige and Z. Yang. Decentralized Market Processes to Stable Job Matchings with Competitive Salaries. *Working paper, Kyoto University, RIMS-1715*, 2011.

*** Supported by the Hungarian Academy of Sciences under its Momemtum Programme (LD-004/2010).

† Supported by EPSRC Grant EP/G043434/1.

Flexibility of Transfers and Unraveling in Matching Markets

Songzi Du[*] Yair Livne[†]

May 2012

Abstract

We show that without flexible transfers, the timing of transactions is difficult to coordinate in large matching markets. In our model, some agents have the option of matching early before others arrive. We compare two regimes. In the first regime, transfers which divide surpluses created between the two sides of the market are *exogenously* fixed, perhaps due to some institutional constraints. Then even with a centralized mechanism that implements a stable matching after all agents arrive, some agents have incentives to match early. We prove that in this setting, as the market gets large, on average approximately one quarter of all agents have strict incentives to match early. Moreover, as the market gets large, with probability tending to 1 there is no early matching scheme that is dynamically stable. On the other hand, in the second regime in which agents can freely negotiate transfers, a stable matching after all agents arrive eliminates all incentives to match early and is dynamically stable.

[*] Graduate School of Business, Stanford University. Email: songzidu@stanford.edu

[†] Graduate School of Business, Stanford University. Email: ylivne@stanford.edu

Tuition Exchange[*]

Umut Mert Dur[†]
University of Texas

M. Utku Ünver [‡]
Boston College

June 28, 2012

Abstract

In this paper we introduce a new class of matching problems which mimics tuition exchanges programs used by colleges in US as a benefit to their faculty members. The most important benefit of participating to the tuition exchange program is that colleges strengthen their compensation package to their faculty and staff at a very nominal cost. Participating colleges find The Tuition Exchange can serve as a strong incentive for top job candidates to accept their offers. Hence, the tuition exchange programs help level the playing field for small colleges in hiring and retaining promising faculty. In tuition exchange programs, each college ranks its own faculty members according to the length of the employment of the faculty. Based on this ranking each college determines the set of eligible dependents of faculty who can participate the scholarship program. Then, the eligible students (dependents) are awarded with scholarship according to the preferences of colleges over eligible students, preferences of eligible students over colleges and the number of available slot in each college. The main concern for each colleges is maintaining a balance between the number of students certified as eligible by that institution (exports) and the number of scholarships awarded to students certified as eligible by other member colleges enrolling at that institution (imports). We propose a new mechanism, two sided top trading mechanism (2S-TTC), which is a variant of well-known top trading cycle mechanism . To our knowledge this is the first time such that a variant of TTC mechanism is used in a market in which both sides (colleges and students) are strategic. We show that 2S-TTC mechanism selects balanced matching which is not dominated by another balanced matching. Moreover, it cannot be manipulated by students and it respects the internal rankings of colleges. We also show that it is the unique mechanism holding these features.

[*] We thank John Duffy for initial discussions.

[†] Address: The University of Texas at Austin, Department of Economics ; e-mail: umutdur@gmail.com; web page: https://sites.google.com/site/umutdur/

[‡] Address: Boston College, Department of Economics, 140 Commonwealth Ave., Chestnut Hill, MA, 02467; e-mail: unver@bc.edu; web page: http://www2.bc.edu/~unver

On the structural characteristics of the Stable Marriage polytope *

Pavlos Eirinakis[1], Dimitrios Magos[2], Ioannis Mourtos[1], Panayiotis 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 *Stable Marriage* problem asks for a matching of men to women that is stable under given preferences. It has been observed that some man-woman pairs have the property that although they are non-stable (i.e., they participate in no solution), they cannot be removed from the preference lists; such a removal would alter the set of solutions. However, these pairs have not been characterized yet. Likewise, some of the fundamental characteristics of the Stable Marriage polytope have not been established. In the current work, we show that these two seemingly distant open issues are closely related. We identify the pairs with the above-mentioned property and present a polynomial algorithm for producing them. This is accomplished by using the partial order defined on *rotations*, representable by the *rotation-poset graph* G, and its *transitive reduction* G^-. Utilizing that result, we derive the dimension of the Stable Marriage polytope P and all alternative minimal linear descriptions. More specifically, we establish that the dimension of P equals the number of nodes of G (i.e., the number of rotations). Further, we establish the minimal equation system and show that non-removable non-stable pairs induce some of the facets of P. The remaining facets of P are also identified with the use of the graph G^-. Hence, we obtain a minimal linear description of P. In fact, we derive all alternative such descriptions as different inequalities may define the same facet and several equations may take each other's place in the minimal equation system.

*This research has been co-funded by the European Union (European Social Fund) and Greek national resources under the framework of the "Archimedes III: Funding of Research Groups in TEI of Athens" project of the "Education & Lifelong Learning" Operational Programme.

An Equilibrium Analysis of the Probabilistic Serial Mechanism[*]

Özgün Ekici[†] Onur Kesten[‡]

May, 2012

Abstract

The prominent mechanism of the recent literature in the assignment problem is the *probabilistic serial* (PS). Under PS, the truthful (preference) profile always constitutes an ordinal Nash Equilibrium, inducing a random assignment that satisfies the appealing ordinal efficiency and envy-freeness properties. We show that both properties may fail to be satisfied by a random assignment induced in an ordinal Nash Equilibrium where one or more agents are non-truthful. Worse still, the truthful profile may not constitute a Nash Equilibrium, and every non-truthful profile that constitutes a Nash Equilibrium may lead to a random assignment which is not ordinally efficient, not even weakly envy-free, and which admits an ex-post inefficient decomposition. A strong ordinal Nash Equilibrium may not exist, but when it exists, any profile that constitutes a strong ordinal Nash Equilibrium induces the random assignment induced under the truthful profile. The results of our equilibrium analysis of PS call for caution when implementing it in small assignment problems.

JEL Classification Numbers: C70, D61, D63

Keywords: random assignment, probabilistic serial, equilibrium, Nash Equilibrium, ordinal Nash Equilibrium, strong ordinal Nash Equilibrium, ordinal efficiency, envy-freeness.

[*] We would like to thank Utku Ünver and Morimutsu Kurino for useful discussions and comments.

[†] Özyeğin University, Istanbul, Turkey. Email: ozgun.ekici@ozyegin.edu.tr

[‡] Tepper School of Business, Carnegie Mellon University, PA 15213, USA. Email: okesten@andrew.cmu.edu.

Unspecified donation in kidney exchange: when to end the chain

K.M. Glorie[1]*, M. de Klerk[2,3], A.P.M. Wagelmans[1], J.J. van de Klundert[4], W.C. Zuidema[2], F.H.J. Claas[5], W. Weimar[2,3].

Econometric Institute[1], Erasmus University Rotterdam, Dept of Internal Medicine – Transplantation[2], Erasmus Medical Center Rotterdam, Dutch Transplant Foundation[3], Leiden, Institute of Health Policy and Management[4], Erasmus University Rotterdam, Dept of Immunohematology and Bloodtransfusion[5], University Medical Center Leiden.

* Corresponding author. E-mail: glorie@ese.eur.nl

Abstract This paper studies participation of unspecified donors in kidney exchange through simultaneous domino paired donation (DPD) and non-simultaneous extended altruistic donor (NEAD) chains. In contrast to previous studies, we specifically investigate the termination of chains, the possibility of transplantation across the blood type barrier, and the impact of incentives in multi-center exchanges. Furthermore, we look into the effect of various configuration parameters such as the timing between exchanges. Our analysis is based on a simulation study that uses data of all 438 patient-donor pairs and 109 unspecified donors who were screened at Dutch transplant centers between 2003 and 2011. In order to clear large exchanges involving long DPD and NEAD chains, we developed an iterative multi-criteria branch-and-price solver that adheres all allocation criteria of the Dutch national kidney exchange program. Because multi-center coordination may raise incentive issues, special attention is paid to individually rational implementation. Our findings are as follows. Chains are best terminated when no further segment is part of an optimal exchange within 3 months. Transplantation across the blood type barrier allows for longer continuation of chains, more transplants and more equitability among patient groups. NEAD chains perform slightly better than DPD chains, provided that the renege rate is sufficiently low. The most substantial gains, however, are due to national individually rational coordination. Particularly highly sensitized and blood type O patients benefit. Appropriate timing between exchanges can further improve these results.

Matching with our Eyes Closed

Gagan Goel [*] Pushkar Tripathi [†]

June 9, 2012

Abstract

Motivated by an application in kidney exchange, we study the following query-commit problem: we are given the set of vertices of a non-bipartite graph G. The set of edges in this graph are not known ahead of time. We can *query* any pair of vertices to determine if they are adjacent. If the queried edge exists, we are *committed* to match the two endpoints. Our objective is to maximize the size of the matching.

This restriction in the amount of information available to the algorithm constraints us to implement myopic, greedy-like algorithms. A simple deterministic greedy algorithm achieves a factor 1/2 which is tight for deterministic algorithms. A big open question in this direction is to give a randomized greedy algorithm that has a significantly better approximation factor. This question was first asked almost 20 years ago by Dyer and Frieze and they showed that a natural randomized strategy of picking edges at random doesn't help and has an approximation factor of 1/2 + o(1). They left it as an open question to devise a better randomized greedy algorithm. In subsequent work, Aronson, Dyer, Frieze, and Suen gave a different randomized greedy algorithm and showed that it attains a factor $0.5 + \epsilon$ where ϵ is 0.0000025; thus showing what they quoted as *"a small triumph for randomization!"*.

In this paper we propose and analyze a new randomized greedy algorithm for finding a large matching in a general graph and use it to solve the query commit problem mentioned above. We show that our algorithm attains a factor of at least 0.56, a significant improvement over 0.50000025.

As for upper bounds, we show that no randomized algorithm can have an approximation factor better than 0.7916 for the query commit problem. We also study another intersting class of randomized algorithms called vertex-iterative algorithms. Both our algorithm and that by Aronson et. al. fall in this class. We show that no vertex iterative algorithm can have an approximation factor better than 0.75.

[*] Google Research, New York (E-mail: gagangoel@google.com).

[†] Algorithms Combinatorics and Optimization, Georgia Institute of Technology (E-mail: pushkar.tripathi@gatech.edu).

Two-sided matching with one-sided data

Guillaume Haeringer
Universitat Autònoma de Barcelona

Vincent Iehlé
Université Paris Dauphine

In most two-sided matching markets that utilize a stable mechanism agents typically do submit short preference lists. The purpose of this paper is to show that this feature add some additional information about the set of possible stable matchings that can be exploited to improve upon existing mechanisms.

The starting point of the paper is to consider what we call a *pre-matching problem*, which consists of two sets of agents (i.e., the two sides of the market), and for only one side a preference ordering over a subset of the agents from the other side. It is assumed that if a agent, say, a student, appears on the preference list of an agent from the other side, say, a school, then for any realization of the student's preferences that school will be considered as acceptable for that student. Conversely, a student not appearing on a school's preference list will consider that school as unacceptable. A pre-matching problem can be easily obtained from a "classical" matching problem (simply by deleting from each school's preferences the students that view the school as unacceptable). Clearly, if we changes students' preferences from a matching problem without modifying the set of acceptable schools the corresponding pre-matching problem will remain unchanged.

Our first result consist of characterizing a set of conditions for a pre-matching problem that says whether, for each student and each school, there exists a matching problem such that for *some* stable matching that student and that school are matched together. In case there does not exist a students' preference profile and a stable matching (with respect to those preferences) the student is said to be dummy for that school. We also provide an algorithm to check whether a student is a dummy for a school.

In the second part of the paper we consider the student-optimal stable mechanism. It is well known that this mechanism is not efficient. We propose a new mechanism where before running Gale and Shapley's Deferred Acceptance algorithm we first eliminate from school's preferences the dummy students. It is shown that by doing so the matching we obtain weakly Pareto dominates the student-optimal matching computed with the original preference profile. While this new mechanism is not strategyproof, we show however that for each student, given a set of schools she has decided to put in her preference list, it is a dominant strategy to put each school in the same order as in her true preferences.

Dynamic Matching in Overloaded Systems

Jacob D. Leshno*

Abstract

In many assignment problems items arrive stochastically over time. When items are scarce, agents form an overloaded waiting list and items are dynamically allocated as they arrive; two examples are public housing and organs for transplant. Even when all the scarce items are allocated, there is the efficiency question of how to assign the right items to the right agents. I develop a model in which impatient agents with heterogeneous preferences wait to be assigned scarce heterogeneous items that arrive stochastically over time. Social welfare is maximized when agents are appropriately matched to items, but an individual impatient agent may misreport her preferences to receive an earlier mismatched item. To incentivize an agent to avoid mismatch, the policy needs to provide the agent with a (stochastic) guarantee of future assignment, which I model as putting the agents in a priority buffer-queue. I first consider a standard queue-based allocation policy and derive its welfare properties. To determine the optimal policy, I formulate the dynamic assignment problem as a dynamic mechanism design problem without transfers. The resulting optimal incentive compatible policy uses a buffer-queue of a new queueing policy, the uniform wait queue, to minimize the probability of mismatching agents. Finally, I derive a policy which uses a simple rule: giving equal priority to every agent who declines a mismatched item (a SIRO buffer-queue). This policy is optimal in a class of robust mechanisms and has several good properties that make it a compelling market design policy recommendation.

*Harvard University and Harvard Business School. yarboz@gmail.com

Paired and Altruistic Kidney Donation in the UK: Algorithms and Experimentation*

David F. Manlove and Gregg O'Malley

School of Computing Science, University of Glasgow, Glasgow, UK.
Email: {david.manlove,gregg.omalley}@glasgow.ac.uk.

Abstract

We study the computational problem of identifying optimal sets of kidney exchanges in the UK. We show how to expand an integer programming-based formulation [1, 2] in order to model the criteria that constitute the UK definition of optimality. The software arising from this work has been used by the National Health Service Blood and Transplant to find optimal sets of kidney exchanges for their National Living Donor Kidney Sharing Schemes since July 2008. We report on the characteristics of the solutions that have been obtained in matching runs of the scheme since this time. We then present empirical results arising from the real datasets that stem from these matching runs, with the aim of establishing the extent to which the particular optimality criteria that are present in the UK influence the structure of the solutions that are ultimately computed. A key observation is that allowing 4-way exchanges would be likely to lead to a significant number of additional transplants.

References

[1] D.J. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *Proceedings of EC '07: the 8th ACM Conference on Electronic Commerce*, pages 295–304. ACM, 2007.

[2] A.E. Roth, T. Sönmez, and M.U. Ünver. Efficient kidney exchange: Coincidence of wants in a market with compatibility-based preferences. *American Economic Review*, 97(3):828–851, 2007.

*Research financially supported by the National Health Service Blood and Transplant (NHSBT) between 1 April 2010 – 30 June 2011, and between 1 January 2012 - 30 June 2013 (project 10-11-ODT). We gratefully acknowledge collaboration with Rachel Johnson, Joanne Allen and with other staff at NHSBT. The views expressed here are those of the authors and not necessarily those of NHSBT. Research also financially supported by a Knowledge Transfer Account from the Engineering and Physical Sciences Research Council between 1 July 2011 - 31 December 2011. Finally, we would like to thank Péter Biró, Rob Irving, Kirstin MacDonald and Ana Viana for valuable input into this work.

An Experimental Comparison of Single-Sided Preference Matching Algorithms

Dimitrios Michail
Dept. of Informatics and Telematics
Harokopio University of Athens, Greece
michail@hua.gr

Consider the scenario where a set of applicants $\mathcal{A}$ has an interest in obtaining a set of posts $\mathcal{P}$ and suppose that associated with each member of $\mathcal{A}$ is a preference list (possibly including ties) comprising a subset of elements of $\mathcal{P}$. A matching of $\mathcal{A}$ to $\mathcal{P}$ is an allocation of each applicant to at most one post such that each post is filled by at most one applicant. Stated differently, it is a matching in the bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$ where E consists of all pairs (a, p) where p belongs in the ordered preference list of a.

The main focus of this work is to experimentally study matchings computed by various one-sided preference matching algorithms with respect to their unpopularity. On the other hand, since it would be unfair to judge algorithms based solely on the unpopularity, we include additional quality measurements such as cardinality, total rank, maximum rank and running time. We compare several different algorithms for the computation of rank-maximal matchings [3, 4], the algorithm of [1] for the computation of popular matchings, and the algorithm of [2]. While popular matchings seem to be unrelated to rank-maximal matchings, the algorithmic techniques required in order to efficiently compute both types are very much related. Thus, all algorithms are implemented using similar heuristics and graph representations.

The experimental comparison of the aforementioned algorithms is performed on instances created by three random structured instance generators. All generated problem instances try to mimic different real life situations, while maintaining as few parameters as possible. Moreover, in addition to synthetic datasets, we experiment with two real-world datasets.

References

[1] David J. Abraham, Robert W. Irving, Telikepalli Kavitha, and Kurt Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37(4):1030–1045, 2007.

[2] Chien-Chung Huang, Telikepalli Kavitha, Dimitrios Michail, and Meghana Nasre. Bounded unpopularity matchings. *Algorithmica*, pages 1–20, 2010.

[3] Robert W. Irving, Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna E. Paluch. Rank-maximal matchings. *ACM Transactions on Algorithms*, 2(4):602–610, 2006.

[4] Dimitrios Michail. Reducing rank-maximal to maximum weight matching. *Theoretical Computer Science*, 389(1-2):125 – 132, 2007.

Two Simple Variations of Top Trading Cycles

Thayer Morrill, North Carolina State University (thayer_morrill@ncsu.edu)

Abstract:
Top Trading Cycles is widely regarded as the preferred method of assigning students to schools when the designer values efficiency over fairness. However, there is a flaw in Top Trading Cycles when objects may be assigned to more than one agent. If agent i's most preferred object a has a capacity of q_a, and i has one of the q_a highest priorities at a, then Top Trading Cycles will always assign i to a. However, until i has the highest priority at a, Top Trading Cycles allows i to trade her priority at other objects in order to receive a. Such a trade is not necessary and may cause a distortion in the fairness of the assignment. We introduce two simple variations of Top Trading Cycles in order to mitigate this problem. The first, Clinch and Trade, reduces the number of unnecessary trades but is bossy and depends on the order in which cycles are processed. The second, priority-adjusted TTC, is nonbossy and independent of the order in which cycles are processed, but allows more unnecessary trades than is necessary to be strategyproof and efficient.

Faster and simpler approximation of stable matchings

Katarzyna Paluch [*]

Institute of Computer Science, Wrocław University

Abstract. We give a $\frac{3}{2}$-approximation algorithm for stable matchings that runs in $O(m)$ time. The previously best known algorithm by McDermid has the same approximation ratio but runs in $O(n^{3/2}m)$ time, where n denotes the number of people and m is the total length of the preference lists in a given instance. Also the algorithm and the analysis are much simpler. We also give the extension of the algorithm for computing stable many-to-many matchings.

[*] Supported by MNiSW grant number N N206 1723 33, 2007-2010.

Hedonic Coalition Formation and Individual Preferences

Szilvia Pápai

We examine the hedonic coalition formation problem, in which players have preferences over the coalitions that they are in, and each player is a member of exactly one coalition. This problem is a generalization of well-known matching problems, such as the marriage and roommate problems. The core of a hedonic coalition formation problem may be empty, i.e., there may not exist a stable hedonic coalition structure for a given coalition formation problem (see Banerjee et al. (2001) *Soc. Choice Welfare* 18: 135-153 and Bogomolnaia and Jackson (2002) *Games Econ. Behav.* 38: 201-230, among others).

In this paper we focus on restrictions on individual preferences, following Alcalde and Romero-Medina (2006) *Soc. Choice Welfare* 27: 365-375, rather than on restrictions on the preference profile (Bogomolnaia and Jackson (2001), Banerjee et. al (2002)) or on feasible coalitions (Pápai (2004) *Games Econ. Behav.* 48: 337-354). What makes a preference restriction an individual preference restriction? A preference domain that satisfies a particular individual preference restriction is a Cartesian product of the agents' preferences satisfying this restriction, and we can allow each player to have a similar set of allowed preferences. Both preference profile restrictions and individual preference restrictions are important to investigate, as they complement each other. Profile restrictions are more descriptive in nature, as they can clarify whether there is a stable coalition structure in a particular situation, given the players' preferences. By contrast, individual preference restrictions gain their relevance if one asks the normative question of how to restrict players' preferences in order to guarantee the existence of a stable coalition structure. In addition, individual preference restrictions are typically easy to check, and an individual preference restriction that guarantees the existence of a stable coalition structure is also immune to population changes: for example, if new players arrive, it is still assured that a stable coalition structure exists, assuming that the new players' preferences also satisfy the individual preference restriction.

We introduce an individual preference restriction called Inclusion Restriction, and prove that under appropriate assumptions of what constitutes a rich domain that satisfies an individual preference restriction, this property is the only one that guarantees the existence of a stable coalition structure for each preference profile in this domain. Inclusion Restriction requires that if two individually rational coalitions have a superset of their union ranked below both of them, then their intersection is ranked above at least one of them. We also identify two sufficient conditions for the existence of stable coalition structures, Intersection Restriction and Union Restriction, both of which imply the Inclusion Restriction, and show that when comparable, given our assumption of strict preferences, all of the sufficient conditions in the literature are stronger than at least one or the other of our two sufficient conditions. We note that the main property of Inclusion Restriction is a substantial weakening of the already known sufficient conditions, which can be seen immediately from the definition itself, as well as from the algorithm that we provide in order to show the sufficiency of this property. The algorithm gives us a way to identify a stable coalition structure for each preference profile in the domain for which Inclusion Restriction holds. Furthermore, in contrast to previous papers, which provide sufficient conditions only, we give a characterization: Inclusion Restriction is not only a weaker sufficient condition than the ones provided previously, but it is also a necessary condition when the preference domain is minimally rich, as specified, subject to an individual preference restriction. Therefore, our result sheds light on how demanding the restriction on individual preferences need to be in order to ensure the existence of a stable coalition structure by restricting individual preferences.

Author Index