MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Computer Science and Information Theory

Egalitarian solutions in the stable roommates problem

Attila Juhos

Supervisors: Ágnes Cseh and Tamás Fleiner
MTA KRTK KTI

MTA KRTK **KTI**

Budapest, 14 November 2018

Abstract

Egalitarian solutions in the stable roommates problem

The analysis of stable matchings in one-sided matching markets

Matching markets naturally occur in any field of life, most relevant examples including the admission of pupils to schools or matching patients with kidney failure to appropriate donors. It is common that the allocation is undertaken by a central scheme, a mechanism that yields an equitable, optimal matching. For instance, the Hungarian university admission system is based on such considerations.

Applications modelled by a two-sided matching market usually involve preferences that agents on either side of the market form about the counter-market party. The goal is to determine a stable matching. The classical showpieces in the field are the high school and university admission systems. The standard representation of the matching problem is a bipartite graph which encodes pupils and universities into vertices and an edge links a pupil and a university if and only if the pupil actually applies to the institution in question. While pupil-side preferences are determined by the application order, the preferences of the universities are implicitly determined by the score of applicants from their previous studies. The aim is to centrally compute a matching, where university quotas are filled such that no pupil is rejected from a highly ranked university unless all the accepted applicants possess a higher score. This property is called stability.

One-sided markets attempt to pair up agents of a single object-space with stability in mind. A typical application in the flesh is the dormitory quota-distribution system used by the Technion Israel Institute of Technology [17]. Students within the same living facility are paired up into rooms based on their personal preferences about each other. The very name of this market problem, the Stable Roommates problem, originates from this example. The goal is to achieve a matching, in which there are no two students that were not paired up, yet they mutually prefer each other to the assigned partners.

Fair, optimal stable matchings

In the roommates-problem, the mean rank of partners taken over all agents may well differ among stable matchings. The aim of our research was to select an optimal, or so called, egalitarian solution from the possible stable matchings. Notwithstanding the fact that the task is proven to be NP-hard, there are significant results concerning approximation algorithms, which, by definition, compute efficiently an almost optimal solution. Our main purpose is the enhancement of such algorithms by designing new techniques. By the end of the work, we will manage to improve the well-founded 2-approximation in general case to a 9/7-approximation in a special case, where the length of lists are limited.

Contents

Chapter 1

Introduction

As already stated in the Abstract, the knowledge of matching markets and matching problems involving preferences are increasingly useful when it comes to large, even nation-wide matching schemes. With the advent of high performance information technology, matching tasks, previously done on local scale, "computed" on paper, are now often centralised and computed on global scale. For instance, schemes in China are required to distribute over 10 million students among higher education institutes through a centralised process [16]. Similar matching systems are used, for instance, in New York to allocate over 60 thousand pupils to high schools [16], or in Hungary to assign school-leavers to universities [1, 18].

Matching theory itself is a widely studied area, see for example [14]. This area studies mostly independent sets of edges within graphs, in a bare model, where edges are considered equally good. Nevertheless, the problem becomes more enthralling as soon as preferences are introduced. Sticking to practical examples, pupils do not only provide a set of universities they wish to be admitted to, but exact preferences are determined as well. Another hailed example is the assigning process of doctors and residents to hospital-positions implemented by the National Resident Matching Program (NRMP) [19, 24] in the US. According to the 2017 annual report [23], in 2017 just over 43.000 students applied via the NRMP system for 31.757 positions.

While the latter examples define two-sided matching markets, many applications have at their basis a one-sided market as a model. Traditionally, this is called the Stable Roommates problem. Probably the most relevant and implemented application in one-sided matching theory is the living kidney exchange program [15], where patients waiting for kidney transplant are looking for appropriate donors. The patients usually provide a donor who is most often physiologically unsuitable for them. They form a pair and are introduced to a central matching scheme that matches the patient-donor pairs if the donors are proper for the other pair's patient. The system may also consider other physiological aspects that make some donors more acceptable than others. A further field of application involves the study of P2P systems with stability criteria in mind. Lebedev et al. [13] investigated collaboration choices in P2P networks and derived an evolution model from stable roommates aspects. This result could bare fruit in future network design, since service providers wish to complete file transactions among system participants as efficiently as possible. Moving on, the existing literature handles bipartite problems with single-sided preferences too, although this report will no longer reflect on this area (the reader is referred to [16]). Example applications include the allocation of dormitory facilities to students [17].

The goal in every case is to compute a matching that is optimal, *stable* with respect to some rigorous definition. The literature defines multiple types of stability, more specifically: weak, strong and super [10]. This report omits these refined notions of stability because the simpler model that is analysed here incorporates the three notions into the same definition of stability. The game theory and economics literature (see e.g. [21]) move further on to analyse strategy-proof or truthful mechanisms that ensure the most preferred strategy of agents being that of revealing their true preferences. Computational social choice theory (e.g. [2]) addresses problems where decision is reached collectively by multiple agents and which decision will surely yield winners and losers. Economic and social choice aspects consist also a significant contribution of stable matching theory, which could not be presented here, so the reader is once again referred to [16].

Due to the typical size of applications, the “manual” computation of stable matchings is infeasible. Therefore, the problems are solved by advanced information technology and algorithms that automate the computation. Nevertheless, running time being a factor of paramount importance, algorithm designers still face a serious challenge considering, once again, the increasing size of problems. This report intends to discuss the stable marriage problem and its variants from an algorithmic point of view and, thus, to contribute to the immense existing knowledge in this and other parts of the field [8].

A further concern of the field is the study of different solutions of the same problem-instance. It is easily seen that the number of solutions can be exponentially bigger than the input size itself. Moreover, the solutions may even differ with respect to further optimality criteria. Therefore, primitive algorithms, previously used for finding and producing at least on of the solutions, are continuously tried to be enhanced to produce specific solutions with specific properties. Again, there are multiple optimality criteria defined by frontrunners, yet we are to focus on probably the most important of them: *egalitarianism*. Unfortunately, we must get disillusioned because of the computational hardness of finding egalitarian stable matchings [4]. Nonetheless, we do attempt, with success, to approximate these with efficient algorithms, starting from an already existing, but not entirely correct result of Cseh et al. [3].

The rest of the report is organised according to the points raised in previous paragraphs. Chapter 2 revisits preliminary notions from other fields and then introduces the specific vocabulary of our area along with much of the upcoming notation. Chapter 3 confines itself to present fundamental structural results about stable matchings, with accent mostly put on the connection between multiple such matchings. Chapter 4 is the first chapter to relate the algorithmic point of view by presenting the two most elementary and most celebrated algorithms in the field that solve the stable marriage and the stable roommate problem. Both the algorithms' correctness are proved in detail. Chapter 5 aims to introduce egalitarianism and the weighted stable matching problem. This chapter also invites us to think on what will be the main focus of Chapter 6 and the entire report. The short-listed configuration of the problem is introduced along with the main result of this work. Chapter 6, section 6.1 introduces definitions and necessary notions to understand proofs, after which section 6.2 presents the new algorithm of the author and his supervisor, which can efficiently compute a “near optimal” stable matching from a stable roommate instance with short lists. Subsections 6.2.1, 6.2.2 and 6.2.3 contain the proof of correctness for different cases. In Chapter 7 the final conclusion is drawn and some further open questions are outlined, too.

Chapter 2

Notions, notation, models

The existing literature distinguishes between multiple matching market models, as suggested by Manlove in [16] and in Chapter 1. This report is going to focus on two of them: two-sided matching problems (or more commonly, bipartite matching problems with two-sided preferences, or simpler: bipartite matching problems) and one-sided matching problems (or non-bipartite matching problems with preferences).

2.1 Preliminaries

Before entering the discussion, we introduce some common notation and basic notions from graph theory, relation theory and algorithm theory.

2.1.1 Graph theory

Let us be given a graph $G = (V, E)$ with vertex set V and edge set E. Unless stated else, from now on any such graph will be considered simple, i.e. neither having multiple edges nor loops. Also, unless stated else, m will denote $|E|$ and n will denote $|V|$. We denote the number of edges incident to an arbitrary vertex $v \in V$ by $d(v)$ and call it the *degree* of v. The adjacent vertices of $v \in V$ are the vertices reachable from v via a single edge, i.e. u is adjacent to v if and only if there exists $e = uv \in E$. The set of all adjacent vertices form the *neighbourhood* of v and is denoted by $\mathcal{N}(v)$. More generally, given a subset X of the vertices, the neighbourhood of X is: $\mathcal{N}(X) = \{u \in V : \exists v \in X \text{ such that } uv \in E\}$. Particularly, $\mathcal{N}(\{v\}) = \mathcal{N}(v)$.

A set of edges $M \subseteq E$ is called a matching, if no two edges in M are incident to the same vertex, or equivalently, each vertex $v \in V$ is covered by at most one edge in M. If, for vertices u and v, $uv \in M$, then we say that M matches u and v. We also define $M(v)$ to be u, if M matches u and v. A matching M is called *maximal*, if it isn't a strict subset of any other matching, or equivalently: there is no edge $e \in E \setminus M$, such that $M \cup \{e\}$ is a matching.

A graph is called *bipartite*, if the vertex set V may be decomposed into vertex-sets U and W (i.e. $V = U \cup W$ and $U \cap W = \emptyset$) such that all edges are incident to exactly one vertex from U and exactly one vertex from W. Occasionally, we will use the following notation: $n_U = |U|$ and $n_W = |W|$.

2.1.2 Relation theory

Let us be given a set A. A set $R \subseteq A \times A$ is called a *relation (defined) on A*. If $(a, b) \in R$, then it is said that a is in relation with b or, more frequently, a is *greater* than b and it is denoted as aRb or $a \prec b$. A relation is

- reflexive, if $a \prec a$, for any $a \in A$;
- antisymmetric, if for any $a, b \in A$, $a \prec b$ and $b \prec a$ implicates that $a = b$ (a and b denote the same element of A);
- transitive, if for any $a, b, c \in A$, $a \prec b$ and $b \prec c$ implicates that $a \prec c$.

A relation R on A is called a *total or linear ordering of A*, if it is reflexive, antisymmetric, transitive and for any $a, b \in A$, either $a \prec b$ or $b \prec a$ holds. Note that in this case, R actually orders the elements of A into a list where the greater element is in front of the lower element. In case of linear orderings we define the relation $\preccurlyeq$: $a \preccurlyeq b$ if and only if either $a \prec b$ or $a = b$.

2.1.3 Algorithm theory

Let Σ denote a set of characters and call it *abc*. *Words* are finite lists of characters and the set of all words is denoted by Σ^*. An arbitrary $\mathcal{L}$ subset of Σ^* is called *language*. A language $\mathcal{L}$ is said to be *efficiently recognisable*, if there exists an algorithm that for any word w decides whether w is an element of $\mathcal{L}$ or not and the number of steps in the execution of the algorithm is *polynomial* in $|w|$, the length of the input-word. The set of all efficiently recognisable languages is denoted by $\mathbf{P}$. Hence, $\mathcal{L}$ is recognisable if and only if $\mathcal{L} \in \mathbf{P}$.

Moving on, there are the languages who are efficiently recognisable under the circumstances that we provide a short enough certificate. More precisely, for any word $x \in \mathcal{L}$ there exists a certificate $y_x \in \Sigma^*$ such that the length of y_x is polynomial in $|x|$ and $\mathcal{L}' = \{(x, y_x) : x \in \mathcal{L}\} \in \mathbf{P}$. The set of all such languages is denoted by $\mathbf{NP}$. Since languages in $\mathbf{P}$ are recognisable with a 0-long certificate, $\mathbf{P} \subseteq \mathbf{NP}$.

Furthermore, any function $f : \Sigma^* \to \Sigma^*$ is said to be efficiently computable, if there exists an algorithm that computes from x the value of $f(x)$ and the number of steps is polynomial in $|x|$. The set of such functions is denoted by $\mathbf{FP}$. Many times, these functions appear as some algorithmic tasks, e.g. colour the vertices of a graph with at most two colours, if possible, or otherwise claim that it is impossible to do it. In this example f maps a graph-input to a colouring or a claim of impossibility. Other, invalid inputs are mapped to an error-message. It happens frequently that such a problem is said to be in $\mathbf{P}$(instead of being in $\mathbf{FP}$). In most cases a problem f may be reformulated in a decisional problem, e.g. in the example discussed the decisional version is the problem to answer whether an arbitrary graph admits a 2-colouring, but the justifying colouring is not interesting, the computation of it may even be omitted. Such a decisional problem is actually a language $\mathcal{L}_f$ defined by the inputs, for which the answer is affirmative. Obviously, $f \in \mathbf{FP}$ implicates that $\mathcal{L}_f \in \mathbf{P}$ and many times the reversed statement holds as well.

Moving one, given languages $\mathcal{L}_1$ and $\mathcal{L}_2$, it is said that $\mathcal{L}_1$ *may be reduced to* $\mathcal{L}_2$ if there exists a function $f \in \mathbf{FP}$ such that $x \in \mathcal{L}_1$ if and only if $f(x) \in \mathcal{L}_2$. In this case it is said that $\mathcal{L}_2$ is computationally at least as hard as $\mathcal{L}_1$, since an instance of the latter problem may always be solved by transforming an input with an efficient function to an input of the $\mathcal{L}_2$-problem. In notation, we many times write that $\mathcal{L}_1 \prec \mathcal{L}_2$. Besides that, such a function f is called a *reduction* from $\mathcal{L}_1$ to $\mathcal{L}_2$. The relation $\prec$ is reflexive, transitive. If $\mathcal{L}_1 \prec \mathcal{L}_2$ and

- if $\mathcal{L}_2 \in \mathbf{P}$, then $\mathcal{L}_1 \in \mathbf{P}$;
- if $\mathcal{L}_2 \in \mathbf{NP}$, then $\mathcal{L}_1 \in \mathbf{NP}$.

A language is said to be (in) **NP**-hard, if all languages in **NP** may be reduced to it: $\mathcal{L} \in \mathbf{NP}$-hard if and only if $\mathcal{L}' \prec \mathcal{L}$, for any $\mathcal{L}' \in \mathbf{NP}$. An **NP**-hard-language that itself is in **NP** is said to be (in) **NP**-complete. The following holds: if $\mathcal{L} \in \mathbf{NP}, \mathcal{L}_c \in \mathbf{NP}$-complete and $\mathcal{L}_c \prec \mathcal{L}$, then $\mathcal{L} \in \mathbf{NP}$-complete. **NP**-complete-problems are the hardest decisional problems inside **NP**. Also, a function f (most of the time appearing as a problem) is usually said (in a not too precise manner) to be **NP**-complete if the decisional version $\mathcal{L}_f$ is **NP**-complete.

2.2 Problem instances

2.2.1 Bipartite matching problem instances

The classical example of bipartite matching problems was introduced by Gale and Shapley [6]. In this example men and women formed their preferences about a subset of the other gender-class. These preferences form a linear ordering on the corresponding subset. The construction given this way is called an instance of the stable marriage problem with incomplete lists.

Definition 2.2.1. Let $G = (U \cup W, E)$ be a bipartite graph. For each $v \in U \cup W$, let $\prec_v$ be a linear ordering of $\mathcal{N}(v)$. Then $\prec_v$ is called the preference-relation of v. For any vertex-class $C \in \{U, W\}$ let $\mathcal{R}_C$ denote the the set of linear orderings on C: $\mathcal{R}_C = \{\prec_v \colon v \in C\}$. Let $\mathcal{R} = \mathcal{R}_U \cup \mathcal{R}_W$. The pair $\mathcal{I} = \langle G, \mathcal{R} \rangle$ is called *an instance of the stable marriage problem with incomplete lists.* The set of all instances of the stable marriage problem with incomplete lists is denoted by $\mathcal{SMI}$-INS. Vertices in U(and W) are usually called *men* (and *women*).

It is fair to imagine that not any matching between men and women could work out. There could exist a man and a woman who were not paired up, yet they mutually prefer each other to their assigned partners. Such an edge is called *blocking.* The motivation of the stability criteria, as it is to be defined soon, is exactly that of eliminating such couples.

2.2.2 Non-bipartite matching problem instances

In this model agents belong to the same object-space and there is no constraint on who they may be paired up. Similarly, preferences form a linear ordering. Due the already aforementioned application of the Technion Israel Institute of Technology [17], where students within the same dormitory facility are paired up based on personal preferences, this kind of problem is called the *stable roommates problem* and instances of the problem are called instances of the stable roommates problem with incomplete lists.

Definition 2.2.2. Let $G = (V, E)$ be a graph. For each $v \in V$, let $\prec_v$ be a linear ordering of $\mathcal{N}(v)$. Then $\prec_v$ is called the preference-relation of v. Let $\mathcal{R}$ be the set of all linear orderings defined on the vertices: $\mathcal{R} = \{\prec_v : v \in V\}$. The pair $\mathcal{I} = \langle G, \mathcal{R} \rangle$ is called *an instance of the stable roommates problem with incomplete lists*. The set of all instances of the stable roommates problem with incomplete lists is denoted by $\mathcal{SRI}$-INS. Vertices in $V$are usually called *agents*.

This case defines the same stability criteria as the one before. It should not happen in a stable matching that non-matched agents would prefer sharing rooms with each other to sharing it with their allocated partners.

2.3 Stability

It is noteworthy that every instance of the stable marriage problem is an instance of the stable roommates problem, or equivalently, $\mathcal{SMI}$-INS $\subset$ $\mathcal{SRI}$-INS. The roommates problem generalises the marriage problem. Furthermore, the definitions of blocking edge and stability are exactly the same in the two cases.

Definition 2.3.1. Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in \mathcal{SRI}$-INS and let M be a matching in G. Then M is called a matching of $\mathcal{I}$.

1. An edge $uv \in E$ is called a *blocking edge with respect to* M, if $uv \notin M$ and either u is unmatched in M or $v \prec_u M(u)$ and either v is unmatched in M or $u \prec_v M(v)$. We also say that *uv blocks the matching M*.
2. Matching M is called *stable* if there is no blocking edge with respect to M.

Having introduced all necessary notions, now we are able to formulate questions. Chapter 4 is going to investigate the problem of determining whether an instance of the stable marriage (roommates) problem admits a stable matching and whether this problem is computationally easy to answer. We conclude this chapter by introducing some further notation, after which Chapter 3 summarises some basic structural results concerning stable matchings.

2.4 Further notions and notation

Let $\mathcal{I} = \langle G, \mathcal{R} \rangle$ be a stable roommates instance.

- In a stable marriage or roommates problem, for any vertex v, the set $\mathcal{N}(v)$ is usually referred to as the *acceptable partners to v*.
- Let M be an arbitrary matching. A_M denotes the set of vertices covered by M. According to Corollary 3.2.1, stable matchings of the same problem-instance cover the same set of vertices. The set of these vertices is going to be denoted by A.
- Let the set of all stable matchings of $\mathcal{I}$ be denoted by $\mathcal{SM}(\mathcal{I})$.
- If $\mathcal{I}$ admits a stable matchings, i.e. $\mathcal{SM}(\mathcal{I}) \neq \emptyset$, then $\mathcal{I}$ is said to be a *solvable* instance. The set of solvable instances of the stable marriage (roommates) problem with incomplete lists is denoted by SOLV-$\mathcal{SMI}$-INS(SOLV-$\mathcal{SRI}$-INS). The main point of Theorem 4.1.1 will be that any stable marriage instance is solvable, hence SOLV-$\mathcal{SMI}$-INS $=$ $\mathcal{SMI}$-INS.

– If $\mathcal{I} \in$ SOLV-$\mathcal{SRI}$-INS, then an edge found in at least one of the stable matchings is called *stable edge*. Edges that are found in all stable matchings are called *fixed edges*. Obviously, fixed edges are stable edges as well.

– If the length of preference lists (the degree of vertices in G) in a stable roommates problem is upper-bounded by d, then the instance belongs to the d-$\mathcal{SRI}$-problem. The set of the instances of the d-$\mathcal{SRI}$-problem is d-$\mathcal{SRI}$-INS, and the set of solvable instances is SOLV-d-$\mathcal{SRI}$-INS.

Chapter 3

Basic structural results

In this chapter we display a handful of structural results on stable matching that are gripping in themselves and, besides that, serve as crucial background-statements for upcoming theorems. Since the report is incapable to cover the wide range of structural properties addressed by the literature, the reader may wish to further deepen into details in [8].

3.1 Union of stable matchings

The simplest observation is that stable matchings are maximal as well, for otherwise the amending edge is blocking.

Proposition 3.1.1. *Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in$ SOLV-$\mathcal{SRI}$-INS and let $M \in \mathcal{SM}(\mathcal{I})$. Then, M is maximal as a matching.*

The lemma that stands at the basis of most of the upcoming claims is related to the union of matchings (not necessarily stable) within the same graph.

Lemma 3.1.2. *Let G be a graph (the reader is reminded that since Chapter 2 any graph is considered to be a simple graph) and let M_1 and M_2 be two matchings of G. Then the symmetric difference $M_1 \triangle M_2$ consists of cycles and paths whose edges are alternately taken from M_1 and M_2. We also say that $M_1 \triangle M_2$ consists of* alternating *cycles and paths.*

Proof. Let us take $S = M_1 \triangle M_2$ and let $H = (V(G), S)$, i.e. the graph composed by the edges in S. Since any vertex v may be covered by at most one edge from both matchings M_1 and M_2, the degree of any vertex in H is at most 2. Such a graph may consist of cycles and paths. Furthermore, in any such path or cycle no vertex with degree 2 may lie on edges taken from the same matching, hence edges are taken alternately from the two matchings. □

Remark. Therefore, $M_1 \cup M_2$ consists of *coincident edges* (edges contained by both M_1 and M_2) and of alternating cycles and paths.

The following statements discuss the structure of stable matchings and the connection between them. An interesting result about the structure of the stable matchings of an instance is that they cover exactly the same set of vertices. This a direct consequence of the fact that the symmetric difference of two stable matchings may only contain alternating cycles.

Claim 3.1.3. *Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in$ SOLV-$\mathcal{SRI}$-INS and let $M_1, M_2 \in \mathcal{SM}(\mathcal{I})$. Then $M_1 \triangle M_2$ consists of alternating cycles.*

Proof. M_1 and M_2 are matchings, hence, because of Lemma 3.1.2, $M_1 \triangle M_2$ consists of alternating cycles and paths. We show that the latter is excluded. Let us suppose that $M_1 \triangle M_2$ contains a maximal (not expandable) alternating path $v_0 v_1 \dots v_r$, where $v_{2i-1} v_{2i} \in M_1$ and $v_{2i} v_{2i+1} \in M_2$, for all possible i. For the sake of explanation, for any edge e contained by exactly one of the matchings M_1 or M_2, we call the other matching the *counter-matching* of e.

It is easy to see that either $v_0 v_1$ or $v_{r-1} v_r$ blocks the counter-matching. Let us suppose that none of them do so. It is not possible that $r = 1$, for otherwise v_0 and v_1 are unmatched by the counter-matching of $v_0 v_1$, so $v_0 v_1$ blocks it. Neither may $r = 2$ hold, for otherwise v_1 together with the preferred vertex from $\{v_0, v_2\}$ blocks the counter-matching. Hence, $r \geq 3$. $v_0 v_1$ or $v_{r-1} v_r$ are not blocking the counter-matchings, so $v_2 \prec_{v_1} v_0$ and $v_{r-2} \prec_{v_{r-1}} v_r$. Therefore, there exists $k \in \{1, 2, \dots, r-2\}$ such that $v_{k+1} \prec_{v_k} v_{k-1}$ and $v_k \prec_{v_{k+1}} v_{k+2}$. Thus, $v_k v_{k+1}$ blocks the counter-matching, contradiction.

Therefore, either $v_0 v_1$ or $v_{r-1} v_r$ blocks the corresponding counter-matching, which contradicts the first assumption. Thus, $M_1 \triangle M_2$ may only consist of alternating cycles. □

3.2 Connection of stable matchings. The Rural Hospitals Theorem

Corollary 3.2.1 (Rural Hospitals Theorem [20])**.** *In an $\mathcal{I} = \langle G, \mathcal{R} \rangle$ solvable stable roommates instance all the stable matchings cover the same set of vertices. Equivalently, $A_M = A_S$, for any $M, S \in \mathcal{SM}(\mathcal{I})$.*

Remark. From now on $A_M = A_S$ is going to be denoted by A.

From the proof of Claim 3.1.3 one may easily deduce that in the symmetric difference of two stable matchings, in any alternating cycle preference-relations "point into the same direction" in the sense how Claim 3.2.2 formalises it.

Claim 3.2.2. *Let $C = (v_1 v_2 \dots v_r)$ be an alternating cycle in $M_1 \triangle M_2$. Then either $v_{k-1} \prec_{v_k} v_{k+1}$ for all $k = \overline{1, r}$ or $v_{k+1} \prec_{v_k} v_{k-1}$ for all $k = \overline{1, r}$, where indexes are taken mod r.*

The proof of this claim is trivially constructable from the second paragraph of the proof of Claim 3.1.3. Furthermore, a relatively simple statement follows from this claim as well:

Corollary 3.2.3. *Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in$ SOLV-$\mathcal{SRI}$-INS and let $M_1, M_2 \in \mathcal{SM}(\mathcal{I})$. Let us suppose that v and w are two adjacent vertices in G such that $vw \in M_1$, but $vw \notin M_2$. Then exactly one of the vertices v and w must be better off and exactly one of them must be worse off in M_2. Formally, the following holds: either $M(v) \prec_v w \wedge v \prec_w M(w)$ or $w \prec_v M(v) \wedge M(w) \prec_w v$.*

Proof. Since $vw \in M_1$ and $vw \notin M_2$, then $vw \in M_1 \triangle M_2$. Let C be the alternating cycle containing vw. Then the statement follows from Claim 3.2.2. □

Chapter 4

Fundamental algorithms

In the current chapter we are to investigate two algorithms that decide whether an instance of the stable marriage and stable roommates problem admits a solution and that produce a solution in affirmative situation. We are going to show that the stable marriage problem is so much simpler than the more general configuration that any instance admits a stable matching.

4.1 Gale-Shapley Algorithm

Theorem 4.1.1 (Gale and Shapley, 1962 [6])**.** *Any arbitrary* $\mathcal{I} \in \mathcal{SMI}$-INS *admits a stable matching and there exists an* $\mathcal{O}(m)$ *time algorithm that computes and outputs a stable matching of* $\mathcal{I}$*. Hence,* SOLV-$\mathcal{SMI}$-INS $= \mathcal{SMI}$-INS*.*

The theorem is proven by designing an algorithm that fulfils the requirements. The components of Algorithm 1 represent a compilation of all previous descriptions and interpretations [6, 12] of the original algorithm, the so called "deferred acceptance" procedure.

Algorithm 1 Gale-Shapley

Input: $\mathcal{I} = \langle G(U \cup W, E), \mathcal{R} \rangle \in \mathcal{SMI}$-INS

```
1:  while there exists a man u ∈ U whose all proposals, if any made, have been rejected and
    whose list is non-empty do
2:      let w be the first woman on u's list
3:      u proposes to w
4:      for each man u' such that u ≺_w u' do
5:          if w was proposed to u' then
6:              reject the proposal
7:          end if
8:          withdraw the edge u'w from G
9:      end for
10: end while
11: let M be the set of proposal edges
12: STOP and OUTPUT M
```

The best way to understand the algorithm is to learn from nature. The men try to propose to the best possible woman on their list (lines 2 and 3). Whenever a man u *proposes* to a woman w, uw becomes a *proposal edge*. In this case both u and w *have a proposal*. The women naturally withdraw any man from their list found lower than the proposing one (lines 4 and 8). In this case, the edge itself is deleted and the woman in question is withdrawn from the list of these men. The procedure continues until each man is either proposed to a woman or has run out of possible spouses (line 1).

Proof of Theorem 4.1.1. We prove the theorem by showing that after the execution of Algorithm 1 on $\mathcal{I}$, the edge-set M is a stable matching. First and foremost, M is a matching. Indeed, no man can have multiple proposals, since they only went on proposing whenever the previous one has been turned down and deleted. Moreover, women only keep the last proposal, any previous proposal is implicitly rejected and deleted.

Secondly, let us take an edge $uw \notin M$ such that u either finished with a less preferred woman w' or has run out of chances. Since he moved down his list further than w, there are two possibilities: either he proposed to w, but this proposal was rejected as a consequence of w being proposed to by a preferred man u'; or even before u could have moved down to w, uw was deleted because of the same reason. In either cases, u was dominated by a proposing man u'. Note that in the algorithm an already proposed woman always holds a proposal and only changes it at the sight and proposal of a better partner. Hence w has a partner u'' in M, and $u'' \preccurlyeq_w u' \prec_w u$. Hence, uw cannot be blocking and the matching M is stable. $\square$

4.2 Irving Algorithm

Contrary to the previous situation, not all stable roommate instances admit solution. For instance see the example in Figure 4.1. There are tree agents numbered with $1, 2$ and 3, respectively, and all agents find all the others acceptable. Preferences form a cycle. Any matching leaves at least one vertex uncovered. However, one of the matched vertices will prefer this one to its partner, hence the matching is unstable.

$$\begin{array}{c|c} 1 & 2\ 3 \\ 2 & 3\ 1 \\ 3 & 1\ 2 \end{array}$$

Figure 4.1. An $\mathcal{SRI}$-instance that admits no stable matching.

Nevertheless, it is computationally easy to determine the existence of a solution and to produce one in affirmative situation. This result belongs to Irving [9] and the algorithm described here is the adaptation of the one given in Irving's paper with some notions borrowed from Gusfield [7].

Theorem 4.2.1 (Irving, 1985 [9]). *There exists an $\mathcal{O}(m)$ time algorithm that for any $\mathcal{I} \in \mathcal{SRI}$-*INS *either computes a stable matching of $\mathcal{I}$ or provides a proof for its non-existence.*

We are going to prove the theorem by proving the correctness of Algorithm 2 (which uses Algorithm 3 as a subroutine). The main routine breaks down into two phases, which we investigate separately.

4.2.1 Description of the Algorithm

4.2.1.1 The first phase

The first phase resembles the Gale-Shapley-type "deferred acceptance" procedure presented earlier. For the sake of grasping the essence better, we adopt the terminology related to *proposals*. Here, everyone will *propose* (see line 13), as opposed to the previous settings, where only men engaged women. However, rejections happen with the same mechanism. Whenever a person receives a better proposal (for becoming the roommate of the proposer), he turns down the previous rejection and withdraw any worse acceptable partner on their list (lines 14 to 20). Similarly to the previous case, the withdrawal of a person q from p's list involves withdrawing p from q's list and the subsequent deletion of edge pq. The instance is truncated.

After the proposal-rejection process is over, some of the agents run out of entries on their lists. These agents are removed from the instance (line 23). We will see later that these are the agents who are unmatched in any stable matching, if there is one (keep in mind Corollary 3.2.1). By deleting isolated agents, the instance is brought into a special state: for any agents $a \neq b$, b is first on a's list if and only if a is last on b's list (Lemma 4.2.2). Throughout the second phase the execution keeps this property invariant. Let us call this property *proposal-consistency*.

The goal of the algorithm is to pair up agents. Line 25 and later, in the second phase, line 33 match agents whose list contain a single entry. Notice that this can be done, because from line 25 till the end of the execution the reduced instance $\mathcal{I}_R$ is proposal-consistent. See details in Lemma 4.2.2 and Corollary 4.2.3. The purpose of the second phase will be to cut down the rest of the agents' lists to one element. Note that exactly because of proposal-consistency, if a has multiple entries with b being first, then b's list may not be lead by a, hence agents with multiple entries cannot be paired up.

4.2.1.2 The second phase. Rotations

The second phase reduces the lists by *eliminating rotations* [7]. Let us assume that after executing the first phase and zero or more times the second phase, there is a sequence of agents $a_1, a_2, \ldots, a_r$ with multiple entries on their list, such that the second element of a_i's list is the same as a_{i+1}'s first element and a_r's second element is the same as a_1's first element. Alternatively: there exists another sequence of agents $b_1, b_2, \ldots, b_r$ such that a_i's list starts with b_i, b_{i+1} and a_r's list starts with b_r, b_1. For such sequences the ordered set R of agent-triples written in the form $(a_i : b_i, b_{i+1})$ is called a *rotation* (Gusfield [7] calls it *exposed* rotation. In his paper he conducts a deep investigation into rotations and their structure. Since this is out of our purpose, we use the shortened term). The member a_i is called *list-owner*, b_i is called the *list-heading* and b_{i+1} is the *list-second* (with respect to a_i's list, obviously). Lemma 4.2.4 will justify that there is a rotation if and only if there is an agent with multiply inhabited list, but it is actually straightforward from Algorithm 3, which is designed clearly for the purpose of manufacturing rotations. One further convention is that the last element of a list is called *list-ending*.

Elimination of rotations The *elimination* of a rotation involves the following: for each i, b_i is forced to reject a_i, who in return proposes to b_{i+1} (lines 28 to 30). The natural rejection process learned from the first phase is also repeated: b_{i+1} withdraws partners strictly worse than a_i (including a_{i+1}) and rejected partners also withdraw b_{i+1} from their list (line 31). However, as opposed to the first, Gale-Shapley-type phase, this does not induce a proposal-rejection avalanche, since b_{i+1}'s previous proposal came from a_i. Hence, the only agents whose proposals are turned down are the a_i's who immediately propose to the b_{i+1}'s, respectively.

At the end of each execution of the second phase, agents with single-entry lists are paired up. Since edges are deleted in each iteration, the process ends with either all the agents of the reduced instance being paired up or at least one agent running out of possible partners. The former case outputs the solution, while the latter one claims the non-existence of a stable matching.

Algorithm 2 Irving

Input: $\mathcal{I} = \langle G(V, E), \mathcal{R} \rangle \in \mathcal{SRI}$-INS

PHASE 1

while there exists an agent $u \in V$ whose all proposals, if any made, have been rejected and whose list is non-empty **do**
    let w be the first agent on u's list
    u proposes to w
    for each agent u' such that $u \prec_w u'$ **do**
        if w has previously accepted the proposal from u' **then**
            reject the proposal
        end if
        withdraw the edge $u'w$ from G
    end for
end while

remove agents whose list became empty
let $\mathcal{I}_R$ be the reduced instance
pair up the agents whose lists have only one entry each

PHASE 2

while there exists no agent with empty list and there exists an agent a whose list has at least two entries **do**
    let $R := \textsc{FindRotation}(\mathcal{I}_R, a)$
    for each list-owner a_i in R **do**
        list-heading b_i rejects a_i
        a_i proposes to b_{i+1}
        b_{i+1} deletes everyone from its list below a_i and these agents delete b_{i+1} on their list
    end for
    pair up the agents whose lists have only one entry each
end while

OUTPUT

if there is an agent with an empty list **then**
    STOP and OUTPUT "There is no stable matching."
else
    let M be the set of pairs formed throughout the execution
    STOP and OUTPUT M and "There is a stable matching."
end if

Algorithm 3 FINDROTATION

Input: $\mathcal{I}_R$, a reduced instance and a, an agent in $\mathcal{I}_R$ with more than one entry on his list

let S be an ordered set
let $a_0 = a$, b_0 and b_1 the first two persons on a_0's list
insert at the end of S the element $(a_0 : b_0, b_1)$
let a_1 be b_1's list-ending
let $j = 1$
while there is no $k < j$ such that $a_j = a_k$ **do**
    let b_j and b_{j+1} be the first two agents on a_j's list
    insert at the end of S the element $(a_j : b_j, b_{j+1})$
    let a_{j+1} be the list-ending of b_{j+1}'s list
    let $j := j + 1$
end while
let $k < j$ such that $a_j = a_k$
remove elements from S inserted before $(a_k : b_k, b_{k+1})$
return S

4.2.2 Formal proof of correctness

Most of the lemmas that lead to the complete proof of Theorem 4.2.1 are derived from Irving's paper [9], but most probably with different terms, in an altered sequence, and with different perspective. First of all we shall see some of the structural properties that hold. Let $\mathcal{I} \in \mathcal{SRI}$-INS and execute Algorithm 2 taking $\mathcal{I}$ as input.

Lemma 4.2.2. *After the execution of the first phase and zero or more times the execution of the second phase, or more precisely, whenever lines 25 and 33 are reached:*

1. *a is on b's list if and only if b is on a's list;*

2. *the reduced instance is proposal-consistent, i.e. a is last on b's list if and only if b is first on a's list.*

Proof. It suffices to prove that these properties are invariant with respect to one iteration of the second phase and that they hold initially, after the execution of the first phase.

1. At the beginning of the algorithm this property holds by definition of the instance. The rejection and withdrawal mechanism (in the first and the second phase) is deliberately designed in a way that it keeps this property. The only way that a list shortens is that an agent b rejects an agent a. Nevertheless, at the same time a withdraws b from his list.

2. On one hand, in the first phase a will only ever have b as the list-heading if a was b's last proposal. However, at the time of the proposal b rejected everyone ranked lower on his list.

   On the other hand, assume that at the end of the first phase b's list-ending is a. The process terminated because everyone either run out of entries or managed to propose to somebody. Because of the first statement, empty-listed people are not proposed to. Besides that, different agents may only propose to different agents at the same moment. To conclude, the agents left with a successful proposal propose to exactly the set of agents determined by themselves. Hence, exactly the agents with non-empty lists are the proposed ones. Agent b's list is non-empty, so b is proposed to, but the only proposer can be a.

This argument is true for the first and the second phases as well, so the property is proven. □

Corollary 4.2.3. *Arriving to lines 25 or 33, a is the only person on b's list if and only if b is the only person on a's list. Hence, they can be paired up indeed.*

Lemma 4.2.4. *Let $\mathcal{I}_R$ be the reduced instance at the start of some iteration of the second phase. If there is an agent a who has multiple entries on his list, then there exists a rotation in $\mathcal{I}_R$ and such a rotation can be computed by Algorithm 3 taking $(\mathcal{I}_R, a)$ as input.*

Proof. As initiated in lines 41 to 44, let $a_0 = a$ and the first two elements of a_0 are b_0, b_1. However, at each step, the list of b_{i+1} has at least two elements due to Lemma 4.2.2. On one side, b_{i+1} is on a_i's list, so a_i is on b_{i+1}'s list. On the other side, b_{i+1} is not the only element of the list of a_i, so b_{i+1}'s list is inhabited by multiple agents. Denote the last one of them by a_{i+1}. Then the list-heading of a_{i+1} is b_{i+1}. Since b_{i+1} has multiple elements, a_{i+1} should have multiple elements, too. Let us call the list-second b_{i+2}. This process goes on until for some j there exists $k < j$ such that $a_j = a_k$. In that case the ordered set of elements $(a_i : b_i, b_{i+1})$, where $i = \overline{k, j-1}$ is a rotation. Algorithm 3 returns this rotation. □

Corollary 4.2.5. *The execution of Algorithm 2 on $\mathcal{I} \in \mathcal{SRI}$-INS terminates after $\mathcal{O}(m)$ steps.*

Proof. Since the number of edges is finite, the first phase will halt after finite steps. Besides that, each iteration of the second phase eliminates a rotation, thus deletes at least 2 more edges. Hence, the second phase is iterated finite times. Thus, the execution reaches an end after finite steps.

As far as time complexity is concerned, the cost of the algorithm is estimated by the number of accesses to the data structures representing the input. An instance $\mathcal{I}$ may be described entirely by a graph represented by the adjacency lists of vertices, where agent v's adjacency list reflects his preferences over acceptable partners. We assume that every operation on any adjacency list has cost $\mathcal{O}(1)$.

The size of the input is, thus, $\Omega(n + m)$. Every single edge may be proposed and deleted at most once. Therefore the cost induced by proposals and deletions is upper-bounded as $\mathcal{O}(2 \cdot m) = \mathcal{O}(m)$. As for finding rotations, Algorithm 3 imposes a cost of at most $\mathcal{O}(n')$, where n' is the number of agents with non-empty lists, since each person comes for at most once, and the number of data structure accesses for each agent is at most $\mathcal{O}(1)$. Consequently, the cost of the execution is $\mathcal{O}(n' + m) = \mathcal{O}(m)$. □

The aim of the following lemmas and theorems is to prove that instance $\mathcal{I}$ admits a solution if and only if the algorithm outputs a stable matching.

Lemma 4.2.6. *If the algorithm outputs a matching M, then it is stable, and, hence, the instance $\mathcal{I}$ is solvable.*

Proof. Let us take an edge $ab \notin M$ and presume that either a is not matched in M or $b \prec_a M(a)$. This means that at a certain point in the execution b withdrew a from his list no matter they had a common proposal or not. This could only happen because an agent a' such that $a' \prec_b a$ proposed to b.

Furthermore, notice that agents who have ever received a proposal are matched in M. Firstly, in the first phase no proposed agent will turn down the proposal without the advent of a better partner. Secondly, the algorithm would never have outputted a matching if somebody in the second phase had run out of possible partners (see lines 26 and 35). Hence, b has a partner in M. Moreover, another feature of the algorithm is that agents only reject others for better partners, so the final partner of b, a'' say, is such that, $a'' \preccurlyeq a' \prec a$. Therefore ab is non-blocking, the matching M is stable. □

Lemma 4.2.7. *In the first phase of the algorithm no stable edge is ever deleted.*

Proof. Let us assume that for instance $\mathcal{I}$, the execution of the first phase deletes some stable edges, the first of which is uw, as a consequence of w having received a proposal from a preferred agent u'. Let the corresponding stable matching be M. Since M is stable and $u'w \notin M$, $u'w$ may not block M. Nonetheless, $u' \prec_w u$, so u' must have a partner w' in M, hence $w' \prec_{u'} w$. Since during the execution of the algorithm u' proposed to w, the edge $u'w'$ must have been deleted prior to uw's deletion. This contradicts the fact that uw was the first to be withdrawn. □

As we already defined, the reduced instance $\mathcal{I}_R$ is the truncated instance formed from the initial instance by applying the first phase of the algorithm and zero or more times the second phase. If $\mathcal{I}$ admits a solution, then let M be a stable matching. We say that M *is contained in* the reduced instance $\mathcal{I}_R$ if all the edges of M are found in $\mathcal{I}_R$.

Remark. Lemma 4.2.7 essentially claims that all stable matchings are contained in the reduced instance after the execution of the first phase.

Lemma 4.2.8. *For $\mathcal{I} \in$ SOLV-$\mathcal{SRI}$-INS let $\mathcal{I}_R$ be a reduced instance and let*

$$R = \{(a_i : b_i, b_{i+1}) : i = \overline{1, r}, \textit{ where index is taken modulo } r\}$$

be a rotation in $\mathcal{I}_R$. Let M be a stable matching contained in $\mathcal{I}_R$. Then:

- *either $M(a_i) = b_i$, for all i,*
- *or $M(a_i) \neq b_i$, for all i.*

Proof. Let us assume that $M(a_i) = b_i$, for some i. Agent b_i is also found on a_{i-1}'s list (see Figure 4.2). However, since b_i is a_i's list-heading, because of Lemma 4.2.2, a_i is the list-ending of b_i. Thus, $a_{i-1} \prec_{b_i} a_i$. Therefore, a_{i-1} is matched in M, for otherwise $a_{i-1}b_i$ blocks M, and in order for M to be stable, $M(a_{i-1}) \prec_{a_{i-1}} b_i$. M is contained in $\mathcal{I}_R$, so the only possibility is that $M(a_{i-1}) = b_{i-1}$. Following the same argument, we find that $M(a_i) = b_i$, for all i. □

$$\begin{array}{r|lll} a_{i-1} & b_{i-1} & b_i & \dots \\ a_i & b_i & b_{i+1} & \dots \end{array} \qquad \begin{array}{r|lllll} b_i & \dots & a_{i-1} & \dots & a_i \end{array}$$

Figure 4.2. The list of agents a_{i-1}, a_i, b_i.

Lemma 4.2.9. *If there is a stable matching M contained in $\mathcal{I}_R$ such that $M(a_i) = b_i$, for all i, then there is a stable matching M' contained in $\mathcal{I}_R$ such that $M'(a_i) \neq b_i$.*

Remark. The idea of the proof is to eliminate the rotation R and let the a_i's match up with b_{i+1}'s, respectively. In order to do that, we first need to check whether this matching is actually feasible. If the set of list-owners is not disjoint from the the set of list-headings, then the matching's existence is questionable. However, this cannot happen.

Proof. Let $A = \{a_1, a_2, \dots, a_r\}$ and $B = \{b_1, b_2, \dots, b_r\}$. Suppose $A \cap B \neq \emptyset$. Then there exist indices j, k such that $a_j = b_k$. It is obvious that a_j's list has at least two entries and so does b_k's. In M a_j is paired up with his list-heading. Now, b_k is matched with a_k and b_k is a_k's list-heading, so a_k is b_k's list-ending. Thus a_j is matched to multiple agents on his list, a contradiction.

Let M' be the matching, in which $M'(a_i) = b_{i+1}$ for all i. This can be done, since the previous proposers of elements in B were the elements of A, so elements of B actually permute proposers amongst themselves. All other elements outside $A \cup B$ are paired with their partner in M: for all $c \in \overline{A \cup B}$, $M'(c) = M(c)$. M' is claimed to be a stable matching with the desired property.

All agents, except for the list-owners (A) have received the same partner or a strictly better one. Therefore edges among these may not block M', for otherwise they would have blocked M. Hence any blocking edge involves one of the members of A. Assume that an edge $a_i c$ blocks the matching M', where $c \neq b_{i+1} = M'(a_i)$. Then c is in front of b_{i+1} in a_i's list. There are two cases to consider:

- $c = b_i$. However, b_i received a strictly better partner then a_i, contradicting that $a_i c$ is blocking M';
- $c \prec_{a_i} b_{i+1}$ in the original list, but c is not present on a_i's list any more. This could only happen, because c withdrew a_i from his list (we remind the reader that the elimination of rotations was also interpreted as list-headings being forced to reject their proposals). At the time of that withdrawal c must have received a better proposal than a_i. Consequently, c "survived" the first phase, i.e. it is part of the reduced instance. However, the algorithm did not halt before starting the new iteration of the second phase, hence, c's list is sill non-empty, so c is still proposed. The monotonicity of the preference for proposals causes that $M'(c) \prec_c a_i$, once again contradicting the fact that $a_i c$ is blocking M'. □

Corollary 4.2.10. *If the original problem admits a solution (i.e. $\mathcal{I} \in$ SOLV-$\mathcal{SRI}$-INS), then Algorithm 2 will output a stable matching of $\mathcal{I}$.*

Proof. According to Lemma 4.2.7, no stable edge is deleted during the first phase. So after zero iterations of the second phase, the reduced instance $\mathcal{I}_R$ contains a stable matching. We prove that at the end of each iteration of the second phase, $\mathcal{I}_R$ will contain a stable matching.

Suppose that after $i-1$ iterations M is a stable matching of $\mathcal{I}_R$. In the i^{th} iteration of the second phase we eliminate a rotation R_i. Due to Lemma 4.2.8, in M either all the list-owners of R_i are paired up with their list-headings or none of them. In the latter case, after the elimination of R_i, $\mathcal{I}_R$ still contains M. (Note that because of a_i being paired up in M with b_{i+1} or worse, b_{i+1} may not be paired up with someone to whom he prefers a_i. Thus, the deletion of non-proposal edges does not affect M.) In the former case, courtesy of Lemma 4.2.9, there exists a stable matching M' after the elimination as well.

It is here that we also see that none of the lists may become empty after such an elimination (obviously with respect to the reduced instance). If an agent u's list empties, and his list-ending (or in other words, his proposer) before the elimination was w, then uw certainly blocks M', since w could not possibly receive a better partner in M'. Contradiction.

Therefore, the execution continues until everybody's list remains with one entry. Due to Corollary 4.2.5, the execution will terminate. Because of Corollary 4.2.3, the agents can be paired up forming a matching. The property that the reduced instance always contains a stable matching induces that this sole matching is stable and it is outputted. □

Chapter 5

Egalitarian stable matchings

Theorem 4.2.1 showed that deciding whether an instance of the stable roommates problem admits a solution can be done in polynomial time (hence SOLV-$\mathcal{SRI}$-INS $\in$ **P**). However, stable matchings may differ in a number of aspects. The purpose of this and Chapter 6 is to investigate a particular optimality concept: egalitarianism. In section 5.1 we are going to define the rank of an agent with respect to another agent. Afterwards, the egalitarian stable roommates problem will be defined, whose aim is to minimise the average rank over all stable matchings. Section 5.2 introduces the weighted stable matching problem and celebrates Teo and Sethuraman's [22] result on approximating optimal matchings within such a problem-instance. Not only is this result thought-provoking in itself, but it represents the basis for our result, outlined later in section 5.3 and in Chapter 6.

5.1 Introduction to egalitarian stable matchings and their algorithmics

Definition 5.1.1.

Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in \mathcal{SRI}$-INS be a stable roommates problem.

1. If $u, v \in V$ are two acceptable partners, then the position of v in u's preference list is called the *rank of v with respect to u* and is denoted by $r_u(v)$.

Further, assume that $\mathcal{I} \in$ SOLV-$\mathcal{SRI}$-INS.

2. Courtesy of Corollary 3.2.1, all stable matching cover the same set of vertices, denoted by A. In this case
$$c(M) := \sum_{v \in A} r_v\big(M(v)\big)$$
is called the *cost of the stable matching M*.

3. $M_{egal} \in \mathcal{SM}(\mathcal{I})$ is called an *egalitarian stable matching of $\mathcal{I}$*, if
$$c(M_{egal}) \leq c(M), \text{ for all } M \in \mathcal{SM}(\mathcal{I}).$$

Remark. Note that instances are finite in the number of edges, hence the number of possible matchings is finite as well. Thus, there always exists an egalitarian among them.

Our obvious goal is to determine for any problem-instance the egalitarian stable matching. The naivest approach, the brute force search for the egalitarian (or any stable matching with a specific property) among all the stable matchings is undermined by the simple observation that the number of stable matchings alone can be exponential in the size of the input. As for example, see Figure 5.1. This is a stable marriage instance constructed from n blocks, each of them consisting of two men and two women. Within a block there are 2 distinct stable matchings. However, any combination of "local" stable matchings adds up to a "global" one, since there are no interconnecting edges. Thus, there are 2^n stable matchings, whereas the size of the input is $\Theta(4n + 4n) = \Theta(n)$.

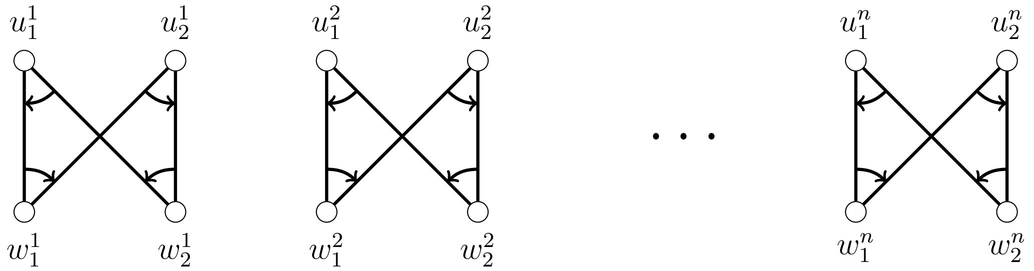


Figure 5.1. An $\mathcal{SMI}$-instance that contains $4n$ people "grouped" into blocks with 4 members each. There are two local stable matchings within each group, hence there are 2^n stable matchings in the instance.

Nonetheless, an egalitarian stable matching within a stable marriage problem is computationally feasible (see [11]). However, according to Feder [4, 5], there is most probably no sophisticated, polynomial-complexity algorithm that would compute an egalitarian stable matching from any stable roommates input, since the problem is **NP**-hard. Feder also gave a 2-approximation algorithm for the problem, which means that there is a polynomial time algorithm that produces for any instance a stable matching whose cost is at most twice the cost the egalitarians'. It is the contribution of Feder, once again, that probably there is no approximation with a ratio strictly smaller than 2 for the general case, unless a widely believed conjecture is false. These results are summarised in the upcoming theorems.

Definition 5.1.2.

1. Let the problem of determining an egalitarian stable matching in a solvable instance of the stable roommates (marriage) problem be denoted by EGAL-$\mathcal{SRI}$ (EGAL-$\mathcal{SMI}$).
2. Let the decisional version of the problem of determining an egalitarian stable matching be denoted by EGAL-$\mathcal{SRI}$-DEC (and EGAL-$\mathcal{SMI}$-DEC). Formally:

$$\text{EGAL-}\mathcal{SRI}\text{-DEC} = \{(\mathcal{I}, k) : \mathcal{I} \in \text{SOLV-}\mathcal{SRI}\text{-INS}, \exists M \in \mathcal{SM}(\mathcal{I}) \text{ such that } c(M) \leq k\}$$

$$\text{EGAL-}\mathcal{SMI}\text{-DEC} = \{(\mathcal{I}, k) : \mathcal{I} \in \mathcal{SMI}\text{-INS}, \exists M \in \mathcal{SM}(\mathcal{I}) \text{ such that } c(M) \leq k\}$$

Remark. Note that in Definition 5.1.2, item 1 required the problem-instance to be solvable, i.e. in SOLV-$\mathcal{SRI}$-INS. The reason for this is that the existence of solutions isn't a computational question any more due to Theorem 4.2.1. Thanks to Theorem 4.1.1, the definition of EGAL-$\mathcal{SMI}$-DEC didn't even require formally the instance to be solvable, since it automatically admits solutions.

Theorem 5.1.3 (Irving, Leather, Gusfield, 1987 [11]). EGAL-$\mathcal{SMI}$-DEC $\in$ **P**. *There exists a polynomial time algorithm that computes an egalitarian stable matching from an instance of the stable marriage problem.*

Theorem 5.1.4 (Feder, 1992 [4] and 1994 [5]).

1. EGAL-$\mathcal{SRI}$-DEC $\in$ **NP**-*complete*
2. EGAL-$\mathcal{SRI}$ *is 2-approximable, i.e. there exists a polynomial time algorithm that, for any instance of the stable roommates problem, approximates the egalitarian stable matching within a factor of* 2.
3. EGAL-$\mathcal{SRI}$ $\in$ **UGC**-*hard, i.e. assuming the Unique Games Conjecture, the* EGAL-$\mathcal{SRI}$ *problem cannot be approximated within* $2 - \varepsilon$*, for any* $\varepsilon > 0$.

Remark. Later in section 5.2, we will see that Teo and Sethuraman [22] generalised this result, although the 2-approximation wasn't improved.

5.2 The weighted stable matching problem

The MIN WEIGHT-$\mathcal{SRI}$ is the problem of providing a minimum weight stable matching in an instance $\mathcal{I}' = \langle G, \mathcal{R}, w \rangle$, where $\mathcal{I} = \langle G, \mathcal{R} \rangle$ would be a usual stable roommates instance that admits a solution, but it now has an added weight function defined on the edges: $w : E(G) \to \mathbb{R}$. Obviously, there is a natural reduction from EGAL-$\mathcal{SRI}$-DEC to MIN WEIGHT-$\mathcal{SRI}$-DEC with the weight function $w(uv) = r_u(v) + r_v(u)$, for all $uv \in E(G)$, hence MIN WEIGHT-$\mathcal{SRI}$-DEC $\in$ **NP**-complete.

Nevertheless, according to the results published in [22], Teo and Sethuraman designed a polynomial time algorithm, which produces a stable matching from a MIN WEIGHT-$\mathcal{SRI}$ instance, whose weight approximates the weight of the optimal stable matching within a factor of 2 on the account of the weight function being of a special form. Let us discuss this form.

Definition 5.2.1. Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in \mathcal{SRI}$-INS be a stable roommates instance. A function $f_u : \mathcal{N}(u) \to [0, +\infty)$ on the neighbourhood of any $u \in V(G)$ is called *U-shaped* if there is a neighbour q of u such that f_u is monotone decreasing on neighbours in order of u's preference list until q and f_u is monotone increasing on neighbours in order of u's preference list from q.

Theorem 5.2.2 (Teo, Sethuraman, 1997 [22]). *Let* $\mathcal{I}' = \langle G, \mathcal{R}, w \rangle$ *be a* MIN WEIGHT-$\mathcal{SRI}$-*instance with weight function* w *having the following property: for each* $u \in V(G)$ *there exists a U-shaped function* f_u *such that* $w(uv) = f_u(v) + f_v(u)$ *holds for every* $uv \in E(G)$*. In this case there exists a polynomial time algorithm that determines a stable matching of* $\mathcal{I}'$*, which approximates the optimal weight within a factor of 2.*

5.3 Approximation of EGAL-$\mathcal{SRI}$ in special cases

5.3.1 Short lists

This section aims to give insight into the existing literature about approximation-attempts in the past, but it also provides the disclaimer for Chapter 6. Cseh, Irving and Manlove [3] investigated the problem where preference lists of agents are upper-bounded by different constants, namely $3, 4, 5$ and, in each case, designed a polynomial time algorithm that calculates a stable matching approximating the egalitarian solution within factors strictly lower than 2. Our inspection revealed that the proof was incorrect. The main result of the author and his supervisor is the reassessment of these cases. The approximation results neither increase nor decrease, but stay intact. Nonetheless, we incline to believe that these results could be improved. The main theorem:

Theorem 5.3.1. *There exists a polynomial time algorithm that, given a solvable stable roommates instance with preferences lists consisting of at most d items ($d \in \{3, 4, 5\}$), produces a stable matching whose cost approximates the cost of egalitarian stable matchings within a factor of $\frac{2d+3}{7}$, i.e. for values $3, 4, 5$ we have approximations within $\frac{9}{7}, \frac{11}{7}, \frac{13}{7}$.*

The proof of the Theorem 5.3.1 is outlined and fully detailed in Chapter 6. In the rest of this section a subset of flaws in the proof of Cseh et al. [3] are outlined that we discovered. For the sake of notions and notation of the article, the reader is referred to Chapter 6, since these are kept from the original article with some additions. We even recommend the reader to return to subsection 5.3.2 after having understood the flow of the proof in Chapter 6 and to ignore subsection 5.3.2 until then. No term or statement builds on this subsection in the rest of the report.

5.3.2 Flaws in the original proof

5.3.2.1 Preprocessing phase of [3, Theorem 6]

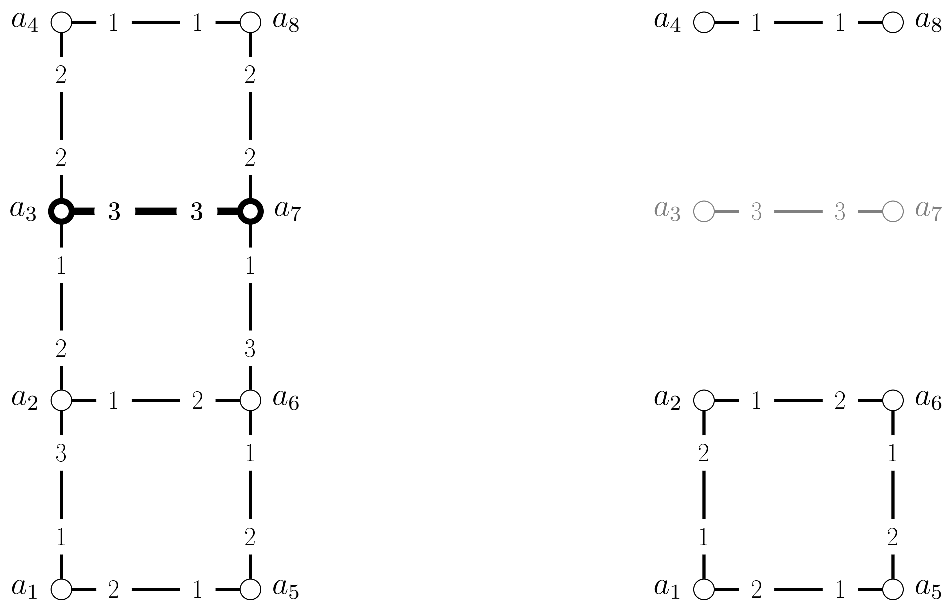
The very first observation is a rather unclear and error-prone explanation regarding a preprocessing phase for the authors' algorithm in [3]. The following statement originates from the paper, more precisely from the proof of Theorem 6:

> 'To simplify our proof, we execute some basic pre-processing of the input graph. If there are any $(1, 1)$-pairs in G, then these can be fixed, because they occur in every stable matching and thus can only lower the approximation ratio. Similarly, if an arbitrary stable matching contains a $(3, 3)$-pair, then this edge appears in all stable matchings and thus we can fix it. Those $(3, 3)$-pairs that do not belong to the set of stable edges can be deleted from the graph. From this point on, we assume that no edge is ranked first or last by both of its end vertices in G and prove the approximation ratio for such graphs.
>
> Take the following weight function on all $uv \in E$:
>
> $$w(uv) = \begin{cases} 0, & \text{if } uv \text{ is a } (1,2)\text{-pair,} \\ 1, & \text{otherwise.'} \end{cases}$$

First and foremost, from this description it is not obvious how the modified instance is related to the initial instance in terms of weights and egalitarianism. Indeed, $(1, 1)$-pairs may be fixed in the sense that the stable matchings of the original instance and that of the instance reduced by such vertices are essentially the same as far as $(1, 1)$-pairs are neglected (see for details Lemma 6.1.8). Nevertheless, the rank of agents change and the weight w applied by the proof may apply a weight 1 on edges that have been $(1, 2)$-pairs in the original instance and a weight 0 on edges that have had cost at least 4 in the initial instance. It is debatable, whether the solution revealed by their algorithm on the truncated instance will indeed hold the same approximation-ratio in the original instance when $(1,1)$ pairs are added again. Neither could we refute nor prove the statement. Instead of that, the corrected algorithm in our report (see Algorithm 4) changes the order of these operations in the first place.

Secondly, the statement of being “allowed” to remove the fix $(3,3)$-pairs is vague, too. In fact, removing these edges along with vertices may even bring in new solutions into the truncated instance, see for example Figure 5.2. It is straightforward that the initial instance (Figure 5.2a) admits a single stable matching, namely $\{a_1a_5, a_2a_6, a_3a_7, a_4a_8\}$. However, after removing the a_3a_7 edge, which is a $(3,3)$-pair, the truncated instance (Figure 5.2b) admits two distinct stable matchings: $\{a_1a_5, a_2a_6, a_4a_8\}$ and $\{a_1a_2, a_5a_6, a_4a_8\}$. By adding back the edge a_3a_7, the first of these solutions converts into the original solution, but the the second solution converts into an unstable matching, since a_2a_3 blocks $\{a_1a_2, a_5a_6, a_3a_7, a_4a_8\}$.

On one hand, this example shows that such a preprocessing step is not allowed, since the solution computed from the truncated instance is not even necessarily convertible into a solution of the original problem, let alone fulfilling requirements on the approximation ratio. On the other hand, this transformation is not a necessity and, as one can check it out, the execution and the proof of correctness of Algorithm 4 in Chapter 6 do not build on this operation.



(a) Initial instance.

(b) After cutting $(3,3)$-pairs.

Figure 5.2. A 3-$\mathcal{SRI}$-instance that contains at the beginning one single stable matching, yet after cutting off the stable $(3,3)$-pair, the truncated instance admits a new solution as well.

5.3.2.2 Second part of the proof of [3, Theorem 6]

The second, more important error in the proof concerns the second part of the proof of Theorem 6, which resembles our proof of Lemma 6.2.6.

The following is quoted from [3]:

‘$2|M_{egal}^{(1,2)}| - |M| > 0$

Let us denote $2|M_{egal}^{(1,2)}| - |M| > 0$ by $\hat{x}$. Notice that $|M_{egal}^{(1,2)}| = \frac{\hat{x}+|M|}{2}$. We can now express the number of edges with cost 3, and at least 4 in M_{egal}.

$$\begin{aligned} c(M_{egal}) &\geq 3 \cdot \frac{\hat{x} + |M|}{2} + 4 \cdot \left(|M| - \frac{\hat{x} + |M|}{2} \right) \\ &= 3.5|M| - 0.5\hat{x} \end{aligned}$$

Let $|M'^{(1,2)}| = z_1$. Then exactly z_1 edges in M' have cost 3. It follows from (1) that $z_1 \geq \hat{x}$. Suppose that $z_2 \leq z_1$ edges in $M^{(1,2)}$ correspond to edges in $M_{egal}^{(1,2)}$. Recall that $|M_{egal}^{(1,2)}| = \frac{\hat{x}+|M|}{2}$. The remaining $\frac{\hat{x}+|M|}{2} - z_2$ edges in $M_{egal}^{(1,2)}$ have cost at most 4 in M'. This leaves $|M| - |M_{egal}^{(1,2)}| - (z_1 - z_2) = \frac{|M|-\hat{x}}{2} - z_1 + z_2$ edges in M_{egal} that are as yet unaccounted for; these have cost at most 5 in both M_{egal} and M'.'

Inequality (1) looks like '$M'^{(1,2)} \geq 2|M_{egal}^{(1,2)}| - |M|$'.

Now, the meaning of the statement that 'The remaining $\frac{\hat{x}+|M|}{2} - z_2$ edges in $M_{egal}^{(1,2)}$ have cost at most 4 in M'' is questionable. The authors most probably wished to emphasize that M' and $M_{egal}^{(1,2)}$ commonly have z_2 edges of type $(1,2)$ and that the rest of the $(1,2)$-pairs in M_{egal} 'have a low cost' in M'. However, paper [3] doest not feature any statement that concerns a bijection between the edges of M_{egal} and M' (note our Claim 6.1.7 addressing this problem). Paper [3] only states and proves what is called in our report Claim 6.2.3, which only discusses the change in the cost contribution of vertices that are paired up in M_{egal}. It is true that the cost contribution of endpoints of those $\frac{\hat{x}+|M|}{2} - z_2$ edges becomes at most 4 in M', but these vertices do not necessarily pair up into $\frac{\hat{x}+|M|}{2} - z_2$ edges in M'. Hence, the statement 'This leaves $|M| - |M_{egal}^{(1,2)}| - (z_1 - z_2) = \frac{|M|-\hat{x}}{2} - z_1 + z_2$ edges in M_{egal} that are as yet unaccounted for' becomes meaningless.

Chapter 6

Egalitarian stable matchings in $\mathcal{SRI}$ with short lists

This chapter is designated to prove Theorem 5.3.1, already studied in [3], but published with an incorrect proof. Section 6.1 aims to introduce necessary definitions and presents, once again, the main theorem in question along with some auxiliary theorems that surround and justify the existence and necessity of our theorem. Section 6.2 unfolds the new algorithm that computes the stable matchings with the claimed approximation-ratios and proves general statements that apply for all situations when d is $3, 4$ or 5. After this, subsections 6.2.1, 6.2.2 and 6.2.3 close the proofs for individual cases. The latter sections concentrate mostly on the specific statements that hold for these particular cases.

6.1 Preliminaries

6.1.1 Definitions, main theorems

First, we are going to define the problem in question, together with the decisional version:

Definition 6.1.1.

1. Let the problem of finding an egalitarian stable matching in a solvable stable roommates instance, where the preference list of each agent is upper-bounded by a positive integer d, be denoted by EGAL-d-$\mathcal{SRI}$.

2. Let the decisional version of EGAL-d-$\mathcal{SRI}$ be denoted by EGAL-d-$\mathcal{SRI}$-DEC, or formally:

$$\text{EGAL-}d\text{-}\mathcal{SRI}\text{-DEC} = \{(\mathcal{I}, k) : \mathcal{I} \in \text{SOLV-}\mathcal{SRI}\text{-INS}, d(v) \leq d \text{ for all agents } v,$$
$$\exists M \in \mathcal{SM}(\mathcal{I}) \text{ such that } c(M) \leq k\}$$

Our attention will drop on particular cases, when $d \in \{3, 4, 5\}$, since the barrier between tractability and intractability is between the $d = 2$ and $d = 3$ cases. This dual result is conveyed, as follows:

Theorem 6.1.2 (Cseh, Irving, Manlove, 2017 [3]).

1. EGAL-2-$\mathcal{SRI}$-DEC $\in$ **P** *and there exists a polynomial time algorithm that computes an egalitarian stable matching from an* EGAL-2-$\mathcal{SRI}$ *problem-instance.*
2. EGAL-3-$\mathcal{SRI}$-DEC $\in$ **NP**-*complete. Consequently,* EGAL-d-$\mathcal{SRI}$-DEC $\in$ **NP**-*complete, for any* $d \geq 3$.

Proof.

1. In an instance $\mathcal{I} = \langle G, \mathcal{R} \rangle \in$ SOLV-2-$\mathcal{SRI}$-INS, graph G is the union of cycles and paths. Any stable matching of the instance is a "local" stable matching in these components and the cost of the "global" stable matching is computed as the sum of the costs of local stable matchings. Notice that because of Proposition 3.1.1, any stable matching in $\mathcal{I}$ is maximal, thus each cycle and path contains at most 2 stable matchings. We only need two find *component-wise* the stable matchings with lowest cost. The union of these will be an egalitarian stable matching. Thus, we designed an $\mathcal{O}(m)$ time algorithm for EGAL-2-$\mathcal{SRI}$.
2. For the **NP**-completeness proof the reader is referred to [3]. The second part of the statement is trivial from the fact that there is natural reduction from EGAL-3-$\mathcal{SRI}$-DEC to EGAL-d-$\mathcal{SRI}$-DEC, for any $d \geq 3$. □

The main result of this chapter and of this entire report is that the egalitarian stable matchings can be approximated within factors of $\frac{9}{7}, \frac{11}{7}, \frac{13}{7}$ for cases $d = 3, 4, 5$, already stated in Theorem 5.3.1. We repeat the theorem once again here, with the introduction of the new notation.

Theorem 6.1.3. EGAL-d-$\mathcal{SRI}$ *is approximable within a factor of* $\frac{2d+3}{7}$, *if* $d \in \{3, 4, 5\}$. *Particularly:*

1. EGAL-3-$\mathcal{SRI}$ *is approximable within* $\frac{9}{7}$.
2. EGAL-4-$\mathcal{SRI}$ *is approximable within* $\frac{11}{7}$.
3. EGAL-5-$\mathcal{SRI}$ *is approximable within* $\frac{13}{7}$.

The rest of the section introduces two possibilities (subsections 6.1.2 and 6.1.3) to investigate the cost of stable matchings by changing from one such matching to another. Moreover, a so-called preprocessing lemma (subsection 6.1.4) is presented that will aid the execution and the proof of correctness of Algorithm 4.

6.1.2 Change in cost through constraint on preference-changes

First of all, Cseh et al. [3] contributed a very simple definition along with a straightforward observation. Claim 6.1.5 basically provides us with constraints on how the cost contribution of connected vertices in a stable matching changes when another stable matching is considered.

Definition 6.1.4. Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in \mathcal{SRI}$-INS. Call an edge $vw \in E(G)$ an (i, j)-pair if w is v's i^{th} choice and v is w's j^{th} choice, i.e. $r_v(w) = i$ and $r_w(v) = j$.

Claim 6.1.5. *Let* $\mathcal{I} = \langle G, \mathcal{R} \rangle \in$ SOLV-d-$\mathcal{SRI}$-INS, *where* $d \geq 3$, *and take two stable matchings* $M, M' \in \mathcal{SM}(\mathcal{I})$. *Take an edge* $uw \in M$.

1. *If* $r_u(w) = r_w(u) = 1$, *then* $uw \in M'$ *(uw is a fix pair).*
2. *If* $r_u(w) = r_w(u) = d$, *then* $uw \in M'$ *(uw is a fix pair).*
3. *If* $r_u(w) < r_w(u)$ *or there is equality, but* $r_u(w) = r_w(u) \notin \{1, d\}$, *then* $r_u(M'(u)) + r_w(M'(w)) \leq d + r_w(u) - 1$.

Consequently,

$$\frac{r_u(M'(u)) + r_w(M'(w))}{r_u(w) + r_w(u)} \leq \frac{d+1}{3}, \tag{6.1.1}$$

and, for the equality to hold, it is necessary that uw is of type $(1, 2)$.

Proof.

1. $(1, 1)$-pairs are obviously found in any stable matching, for otherwise they block the matching.
2. Let us assume that $uw \notin M'$. According to Corollary 3.2.3, exactly one of u and w must be better-off and exactly one of them must be worse-off in M'. Nevertheless, none of u or w can be paired up with a worse partner. Hence, $uw \in M'$.

   Remark. Note that this only means that stable (d, d)-pairs are fixed, but there could well be non-stable (d, d)-pairs.
3. A simple consequence, once again, of Corollary 3.2.3. We reach the greatest expansion of u and w's combined cost by allowing w to improve his partner by 1 position and pair up u with his least preferred partner. Thus, $r_u(M'(u)) + r_w(M'(w)) \leq d + (r_w(u) - 1)$.

For the proof of the consequence let us denote $r_u(w) = a, r_w(u) = b$. In cases 1 and 2 the ratio is 1, which is obviously smaller than the expression on the right. In case 3 we have that $a \leq b$ and if $a = 1$, then $b \geq 2$ and $a \neq d$. Then, according to item 3,

$$\frac{r_u(M'(u)) + r_w(M'(w))}{r_u(w) + r_w(u)} \leq \frac{d+b-1}{a+b}.$$

Now, function $g(a, b) = \frac{d+b-1}{a+b}$, where $a \in \{1, 2, \ldots, d-1\}$ and $b \in \{2, 3, \ldots d\}$ and $d \geq 3$ is monotone decreasing in both of its variables. For variable a this is trivial. For variable b, we transform g as follows:

$$g(a, b) = 1 + \frac{d-a-1}{a+b},$$

which is monotone decreasing in b, since $d - a - 1 \geq 0$. Then, we find that

$$g(a, b) \leq g(1, 2) = \frac{d+1}{3}.$$

Since g is strictly decreasing in a, for equality $a = 1$ is a necessity. In that case $d - a - 1 > 0$, thus $g(1, b)$ is strictly decreasing in b. So $b = 2$ is a necessity as well. □

Claim 6.1.5 has an impact on the worst case cost of stable matchings in a d-$\mathcal{SRI}$-instance. Namely:

Corollary 6.1.6. *In an instance $\mathcal{I} \in$ SOLV-d-$\mathcal{SRI}$-INS, where $d \geq 3$, the cost of any stable matching approximates the cost of egalitarian stable matchings within a factor of $\frac{d+1}{3}$.*

Proof. Let M_{egal} be one of the egalitarian stable matchings in $\mathcal{I}$ and let $M \in \mathcal{SM}(\mathcal{I})$ be an arbitrary stable matching. We know that

$$c(M_{egal}) = \sum_{v \in A} r_v(M_{egal}(v)) = \sum_{vw \in M_{egal}} (r_v(w) + r_w(v)).$$

From Corollary 3.2.1 M covers the same vertices as M_{egal}, therefore

$$c(M) = \sum_{v \in A} r_v(M(v)) = \sum_{vw \in M_{egal}} (r_v(M(v)) + r_w(M(w))).$$

According to Claim 6.1.5, however,

$$r_v(M(v)) + r_w(M(w)) \leq \frac{d+1}{3} \cdot (r_v(w) + r_w(v)), \text{ for any } vw \in M_{egal}.$$

Hence,

$$\frac{c(M)}{c(M_{egal})} \leq \frac{\sum_{vw \in M_{egal}} \frac{d+1}{3} \cdot (r_v(w) + r_w(v))}{\sum_{vw \in M_{egal}} (r_v(w) + r_w(v))} = \frac{d+1}{3}.$$

□

6.1.3 Change in cost through bijection between M_{egal} and M

Let M_{egal} be one of the egalitarian stable matchings in $\mathcal{I}$ and let $M \in \mathcal{SM}(\mathcal{I})$ be an arbitrary stable matching. Claim 6.1.5 in subsection 6.1.2 showed how the cost contribution of vertices connected in M_{egal} change when we switch to matching M. In order to prove Theorem 6.1.3 another technique is constructed, which allows the investigation of M. We would like to design a bijection from M_{egal} to M so that we could examine M through the lens of the bijection.

Let $\varphi : M_{egal} \to M$ such that for any $e \in M_{egal} \cap M$, $\varphi(e) = e$. Note that $(1,1)$- and fix (d,d)-pairs fall under these circumstances. The rest of the mapping is discussed in the following paragraphs.

It is known from Lemma 3.1.2 and Claim 3.1.3 that the union of two stable matchings consists of common edges and even cycles, whose edges are alternately taken from the two matchings. From Claim 3.2.2 and Corollary 3.2.3 it is also clear that within an alternating cycle the preferences at each vertex point at the same direction.

Now, the symmetric difference $M_{egal} \triangle M$ consists only of disjoint alternating cycles. In each cycle φ maps M_{egal}-edges to their neighbours. Obviously, this can only be done in two ways in each cycle: to one of the M_{egal}-edges we assign one of its two neighbours, then the rest of the mapping is specified. φ should always map into the direction of preference, in the sense that if there are consecutive vertices v_{i-1}, v_i, v_{i+1} in an alternating cycle such that $v_{i-1}v_i \in M_{egal}$ (obviously $v_i v_{i+1} \in M$) and $v_{i+1} \prec_{v_i} v_{i-1}$, then $\varphi(v_{i-1}v_i) = v_i v_{i+1}$.

Remark that if in a cycle there exists an edge $v_i v_{i+1} = e_i \in M_{egal}^{(1,2)}$ such that $r_{v_i}(v_{i+1}) = 1$ and $r_{v_i+1}(v_i) = 2$, then φ assigns e_i the neighbour on the side of vertex v_{i+1}. Also, if there exist two neighbouring edges $e \in M_{egal}$, $f \in M$ of type $(1,2)$, then $\varphi(e) = f$.

A very similar statement to Claim 6.1.5 may be developed by the means of this bijection. We extend the previously described φ to a bijection between arbitrary stable matchings, since neither the construction, nor the argumentation depended on the egalitarianism of the domain matching. We are going to call φ the *standard bijection* between M and M'.

Claim 6.1.7. *Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in$ SOLV-d-$\mathcal{SRI}$-INS, where $d \geq 3$, and take arbitrary stable matchings $M, M' \in \mathcal{SM}(\mathcal{I})$. Let $\varphi : M \to M'$ be the standard bijection between them as explained above. Take an edge $uw \in M$. Let $\varphi(uw) = pq \in M'$.*

1. *If $r_u(w) = r_w(u) = 1$, then $pq = uw \in M'$ (uw is a fix pair).*
2. *If $r_u(w) = r_w(u) = d$, then $pq = uw \in M'$ (uw is a fix pair).*
3. *If $r_u(w) < r_w(u)$ or there is equality, but $r_u(w) = r_w(u) \notin \{1, d\}$, then $r_p(q) + r_q(p) \leq d + r_w(u) - 1$.*

Consequently,

$$\frac{r_p(q) + r_q(p)}{r_u(w) + r_w(u)} \leq \frac{d+1}{3}, \tag{6.1.2}$$

and for the equality to hold, it is necessary that uw is of type $(1, 2)$.

Proof. 1 and 2 should be clear. As for 3, either $uw = pq$ is fix pair, in which case the statement holds, because $r_u(w) = r_w(u) = d$ cannot hold; or uw is non-fix, but $\{p, q\} \cap \{u, w\} \neq \emptyset$. The rank at the common agent may only improve by the definition of φ, so this rank is at most $r_w(u) - 1$ and we are done. For the consequence the very same proof applies as the one outlined in the proof of Claim 6.1.5. □

6.1.4 Preprocessing lemma

The following lemma presents a basic preprocessing method by cutting of edges of type $(1, 1)$. By this, smaller instances are obtained and approximation algorithms make smaller absolute and relative errors. The lemma basically claims that edges of type $(1, 1)$ are part of all stable matchings and can be removed without influencing the "effective" part of the stable matching.

Lemma 6.1.8. *Let $\mathcal{I} = \langle G, \mathcal{R} \rangle \in \mathcal{SRI}$-INS and let M_f be the set of edges of type $(1, 1)$. Let us denote by $\mathcal{I}'$ the stable roommates instance created from $\mathcal{I}$ by removing from instance $\mathcal{I}$ the endpoints of edges in M_f along with all the edges incident to these vertices. Preference lists are kept with the exception that removed edges shorten the corresponding lists. In this case the stable matchings of $\mathcal{I}'$ are exactly those of $\mathcal{I}$ without the edges in M_f. Formally:*

$$\mathcal{SM}(\mathcal{I}') = \{M' \subseteq E(G) \setminus M_f : M' \cup M_f \in \mathcal{SM}(\mathcal{I})\}.$$

Proof. On one hand, let M' be a stable matching in $\mathcal{I}'$ and let $M = M' \cup M_f$. Since not only the edges of M_f, but endpoints were cut off from $\mathcal{I}$, M is indeed a matching. Furthermore, the endpoints of edges in M_f are all matched to their favourite choice, therefore none of them participates in a blocking edge. Hence, the endpoints of a possible blocking edge are from $\mathcal{I}'$. However, M' was stable in $\mathcal{I}'$, so there cannot be any blocking edge. Consequently, M is stable in $\mathcal{I}$.

On the other hand, let M be a stable matching in $\mathcal{I}$ and let $M' = M \setminus M_f$. Due to the fact that $M_f \subseteq M$, M' does not cover any vertex incident to an edge from M_f, therefore M' is a matching in $\mathcal{I}'$. Since instance $\mathcal{I}$ was only reduced by cutting off edges, the stability of M' could not be compromised. Hence, the proof is complete. □

6.2 New algorithm

We are now completely prepared to present the main algorithm of this report. It is noteworthy that the same algorithm is applied in cases EGAL-3-$\mathcal{SRI}$, EGAL-4-$\mathcal{SRI}$, EGAL-5-$\mathcal{SRI}$, but it obviously yields different results. The most relevant impetus for such an algorithm is suggested by the consequence of Claim 6.1.5, stating that the blame for an explosion in the cost of stable matchings should be put on edges of type $(1, 2)$ in an egalitarian stable matching whose cost convert into $d + 1$ in an arbitrary matching. In order to minimise the preponderance of such transformations, our algorithm aims to accumulate as many $(1, 2)$-pairs as possible. This is achieved by applying the known results about the MIN WEIGHT-$\mathcal{SRI}$ problem. We label $(1, 2)$-pairs with 0 weight, facilitating an optimal matching with an abundance in such pairs.

Algorithm 4 Approximation of EGAL-d-$\mathcal{SRI}$

Input: $\mathcal{I} = \langle G, \mathcal{R} \rangle \in$ SOLV-d-$\mathcal{SRI}$-INS, where $d \geq 3$ is arbitrary, but fixed

55: Apply the following weight function on the edges of G:

$$w : E(G) \to [0, +\infty), w(uv) = \begin{cases} 0, & \text{if } uv \text{ is a } (1,2)\text{-pair,} \\ 1, & \text{otherwise.} \end{cases} \tag{6.2.1}$$

56: Let M_f denote the edges of G of type $(1, 1)$. Remove the vertices connected with these edges along with all the edges incident to these vertices. Preferences lists and edge-weights, baring removed edges, are kept intact.

57: Apply the algorithm of Teo and Sethuraman [22] to the newly created $\mathcal{I}' = \langle G', \mathcal{R}, w \rangle$ MIN WEIGHT-$\mathcal{SRI}$ input. This yields an M' stable matching.

58: Let $M = M' \cup M_f$.

59: STOP and OUTPUT M.

We claim that the Algorithm 4 justifies Theorem 6.1.3 and in all three cases we will have the right approximations. First, we prove the generally applicable statements.

Let us introduce the following notation: $f = |M_f|$, $\mu = |M|$, $\mu' = |M'|$, where M' is the matching computed in line 57 and M is $M' \cup M_f$, computed in line 58. Obviously, $\mu = \mu' + f$ holds. Also, for an arbitrary stable matching N, let $N^{(1,2)}$ denote the set of edges in N of type $(1, 2)$. Let M_{egal} denote one of the egalitarian stable matchings of $\mathcal{I}$.

Lemma 6.2.1. *Algorithm 4 eventually halts and runs in polynomial time.*

Proof. The size of the input is $\Theta(n)$, since the preference lists are of length at most d. The description of the weight function and the removal of $(1, 1)$ edges in line 56 should be proportional to the number of edges, thus to n. Because of Lemma 6.1.8, instance $\mathcal{I}' \in$ SOLV-$\mathcal{SRI}$-INS.

It is also clear that the weight function applied on $\mathcal{I}$ meets the condition of Theorem 5.2.2. Indeed, since edges of type (i, j) may only be assigned weight 0, if $i, j \leq 2$, therefore the weights found on edges incident to an arbitrary, but fixed vertex u, in order of the preference list of u, are of form $(a, b, 1, \ldots, 1)$, where $a, b \in \{0, 1\}$. In each combination, the weight function itself is U-shaped (considering only edges pointing from u). This means that by taking $f_u(v) = \frac{w(uv)}{2}$, $(\forall) uv \in E(G)$, for each $u \in V(G)$, f_u is U-shaped and for each $uv \in E(G)$, $w(uv) = \frac{w(uv)}{2} + \frac{w(uv)}{2} = f_u(v) + f_v(u)$. Moving on, notice that in line 56 edges of type $(1, 1)$ are cut off. However, this means that some lists may shorten by losing their first element.

Nevertheless, a U-shaped list keeps its property even after the withdrawal of its first element. Thus, the instance $\mathcal{I}' = \langle G', \mathcal{R}, w \rangle$ prepared for line 57 fulfils all conditions of Theorem 5.2.2, according to which line 57 computes indeed a stable matching in polynomial time. Thanks to Lemma 6.1.8, M is stable in the original instance.

The costs of the addition in line 58 and the outputting of M are also linear in n. □

Lemma 6.2.2. *After the execution of Algorithm 4, the following holds:*

$$|M^{(1,2)}| \geq 2 \cdot |M^{(1,2)}_{egal}| - \mu'. \tag{6.2.2}$$

Proof. As already stated in the proof of Lemma 6.2.1 the weight function applied on $\mathcal{I}'$ fulfils the condition of Theorem 5.2.2. Note that the weights assigned to edges in line 55 do not change when edges are removed in line 56 (i.e. after cutting edges one may spot edges of type $(1,1)$ with weight 0 and $(1,2)$-pairs with weight 1). Due to Theorem 5.2.2, the yielded M' stable matching approximates the optimal weight within a factor of a 2. According to Lemma 6.1.8, M is a stable matching in $\mathcal{I}$ and $M'_{egal} = M_{egal} \setminus M_f$ is a stable matching in $\mathcal{I}'$.

Remark. Although it does not influence the string of thoughts, keep in mind that M'_{egal} is generally not an egalitarian stable matching in $\mathcal{I}'$.

Let M'_{opt} be a weight-optimal stable matching in $\mathcal{I}'$. It is clear from the definition of M'_{opt} that

$$w(M') \leq 2 \cdot w(M'_{opt}) \leq 2 \cdot w(M'_{egal}), \tag{6.2.3}$$

where $w(N)$ denotes the weight of N: $w(N) = \sum_{uv \in N} w(uv)$.

The construction of the weight function implies that $w(N') = \mu' - |N^{(1,2)}|$, if $N \in \mathcal{SM}(\mathcal{I})$, $N' \in \mathcal{SM}(\mathcal{I}')$ and $N = N' \cup M_f$, because

$$w(N') = |\{e \in N' : w(e) \neq 0\}| = |\{e \in N' : e \notin N^{(1,2)} \, (\subseteq N')\}|.$$

Hence, (6.2.3) can be transformed:

$$\mu' - |M^{(1,2)}| \leq 2 \cdot \left(\mu' - |M^{(1,2)}_{egal}|\right), \text{or}$$

$$|M^{(1,2)}| \geq 2 \cdot |M^{(1,2)}_{egal}| - \mu'.$$ □

Lemma 6.2.2 provides a first clue on how Algorithm 4 intends to maximise the number of $(1,2)$-pairs in the output and, thus, minimising the cost of the stable matching. Based on the sign of the right hand side of inequality 6.2.2, we are going to distinguish two cases. In negative case M_{egal} contains only a handful $(1,2)$-pairs. In this case inequality (6.2.2) conveys no information, but the fact that not even the half of edges in M_{egal} are $(1,2)$-pairs makes M_{egal} so expensive that all the stable matchings' costs stay in our reach. In positive case, the matching outputted by the algorithm contains a large number of $(1,2)$-pairs and will be proper for our purposes. From now on we discuss the special cases for $d = 3, 4, 5$ in separate subsections.

type of uw	$max(r_u(M'(u)) + r_w(M'(w)))$ and $max(r_p(q) + r_q(p))$	max ratio
$(1, 1)$	$1 + 1$	1
$(1, 2)$	$3 + 1$	4/3
$(1, 3)$	$3 + 2$	5/4
$(2, 2)$	$1 + 3$, $3 + 1$, $2 + 2$	1
$(2, 3)$	$2 + 3$, $3 + 2$	1
$(3, 3)$	$3 + 3$	1

Table 6.1. Maximum cost of change from M to M' in 3-$\mathcal{SRI}$.

6.2.1 EGAL-3-$\mathcal{SRI}$

This section finishes the proof of Theorem 6.1.3 for case $d = 3$.

Claim 6.2.3 (Application of Claims 6.1.5 and 6.1.7 for $d = 3$)**.** *Let $\mathcal{I} \in$ SOLV-3-$\mathcal{SRI}$-INS and $M, M' \in \mathcal{SM}(\mathcal{I})$ and let $\varphi : M \to M'$ be the standard bijection. Take $uw \in M$ and let $\varphi(uw) = pq$. The maximum of $r_u(M'(u)) + r_w(M'(w))$ and of $r_p(q) + r_q(p)$ varies according to Table 6.1.*

Claim 6.2.4 (Application of Corollary 6.1.6 for $d = 3$)**.** *In $\mathcal{I} \in$ SOLV-3-$\mathcal{SRI}$-INS any stable matching approximates the egalitarian stable matching within a factor of 4/3.*

As already stated in Theorem 6.1.5, edges of type $(3, 3)$ are either found in all stable matchings or in none of them. Let us denote the number of fix $(3, 3)$-pairs by y. Now two cases are distinguished depending on the sign of the term on the right of inequality (6.2.2).

Lemma 6.2.5. *Suppose that $2 \cdot |M_{egal}^{(1,2)}| - \mu' \leq 0$. In that case, any matching $M^* \in \mathcal{SM}(\mathcal{I})$ is within a factor of 9/7 compared to M_{egal}, i.e. $c(M^*) \leq \frac{9}{7} c(M_{egal})$. Since the approximation ratio is true for any $M^* \in \mathcal{SM}(\mathcal{I})$, it is also true for M.*

Proof. Denote $0 \leq \mu' - 2 \cdot |M_{egal}^{(1,2)}| = x$. Equivalently, we have $|M_{egal}^{(1,2)}| = \frac{\mu' - x}{2}$, where $x \geq 0$. In this case, in M_{egal} there are f $(1, 1)$-pairs, y $(3, 3)$-pairs, $\frac{\mu'-x}{2}$ $(1, 2)$-pairs and the rest of the edges is of cost at least 4 (see Claim 6.2.3). Thus,

$$\begin{aligned} c(M_{egal}) &\geq 2f + 3 \cdot \frac{\mu' - x}{2} + 4 \cdot \left(\mu' - \frac{\mu' - x}{2} - y\right) + 6y = \\ &= 2f + 2y + 3.5\mu' + 0.5x. \end{aligned} \tag{6.2.4}$$

Let M^* be an arbitrary stable matching in $\mathcal{I}$. M^* must contain exactly the same f $(1, 1)$-pairs and exactly the same y $(3, 3)$-pairs as M_{egal}. Nonetheless, M^* and M_{egal} cover the same set of vertices. We saw that there were $\frac{\mu'-x}{2}$ $(1, 2)$-pairs and $\left(\mu' - \frac{\mu'-x}{2} - y\right)$ edges of types $(1, 3), (2, 2), (2, 3)$ in M_{egal}. Once again, courtesy of Claim 6.2.3, the contribution of the endpoints of $(1, 2)$-pairs is at most 4 and the contribution of endpoints of edges of types $(1, 3), (2, 2), (2, 3)$ is at most 5 in stable matching M^*. Therefore, it holds that

$$\begin{aligned} c(M^*) &\leq 2f + 4 \cdot \frac{\mu' - x}{2} + 5 \cdot \left(\mu' - \frac{\mu' - x}{2} - y\right) + 6y = \\ &= 2f + y + 4.5\mu' + 0.5x. \end{aligned} \tag{6.2.5}$$

From (6.2.4) and (6.2.5) we conclude that

$$\frac{c(M^*)}{c(M_{egal})} \leq \frac{2f + y + 4.5\mu' + 0.5x}{2f + 2y + 3.5\mu' + 0.5x}. \tag{6.2.6}$$

If $\mu' = 0$, then all the edges of any stable matching in $\mathcal{I}$ are $(1,1)$-pairs. In that case $M^* = M_{egal}$ and $c(M^*)/c(M_{egal}) = 1$. If $\mu' \neq 0$, then

$$\frac{c(M^*)}{c(M_{egal})} \leq \frac{2f + y + 4.5\mu' + 0.5x}{2f + y + 3.5\mu' + 0.5x} \leq \frac{4.5\mu'}{3.5\mu'} = \frac{9}{7}.$$

□

Lemma 6.2.6. *Suppose that $2 \cdot |M_{egal}^{(1,2)}| - \mu' > 0$. In that case M approximates M_{egal} within a factor of $9/7$, i.e. $c(M) \leq \frac{9}{7}c(M_{egal})$.*

Proof. Let us denote $2 \cdot |M_{egal}^{(1,2)}| - \mu' = \hat{x} > 0$. Hence, $|M_{egal}^{(1,2)}| = \frac{\mu'+\hat{x}}{2}$.

Similarly to the previous case, we observe that M_{egal} has f $(1,1)$-pairs, y $(3,3)$-pairs, $\frac{\mu'+\hat{x}}{2}$ $(1,2)$-pairs and the weight of the rest of the edges is at least 4. Therefore,

$$\begin{aligned} c(M_{egal}) &\geq 2f + 3 \cdot \frac{\mu' + \hat{x}}{2} + 4 \cdot \left(\mu' - \frac{\mu' + \hat{x}}{2} - y\right) + 6y = \\ &= 2f + 2y + 3.5\mu' - 0.5\hat{x}. \end{aligned} \tag{6.2.7}$$

Moreover, let $|M^{(1,2)}| = z_1$. According to inequality (6.2.2), $z_1 \geq \hat{x}$. Let φ be the standard bijection from M_{egal} to M and let

$$C = \left\{e \in M_{egal}^{(1,2)} | \varphi(e) \in M^{(1,2)}\right\} \text{ and } |C| = z_2.$$

Obviously $z_2 \leq z_1$ and $z_2 \leq |M_{egal}^{(1,2)}|$. Now we analyse edges found in $M_{egal}^{(1,2)} \setminus C$. Let $vw = e \in M_{egal}^{(1,2)} \setminus C$. Since $\varphi(e) \notin M^{(1,2)}$, Claim 6.2.3 leaves us with the only possibility that $\varphi(e)$ is a $(1,3)$-pair. Hence, the cost of edge $\varphi(e)$ is exactly 4.

Subsequently, in M there are exactly f $(1,1)$-pairs, y $(3,3)$-pairs, $|M^{(1,2)}| = z_1$ edges are of type $(1,2)$. At least $|\varphi(M_{egal}^{(1,2)} \setminus C)| = |M_{egal}^{(1,2)}| - z_2$ edges are of type $(1,3)$, i.e. of cost 4. The rest of the edges have cost at most 5. Thus,

$$\begin{aligned} c(M) &\leq 2f + 3z_1 + 4 \cdot \left(\frac{\mu' + \hat{x}}{2} - z_2\right) + 5 \cdot \left(\mu' - z_1 - \left(\frac{\mu' + \hat{x}}{2} - z_2\right) - y\right) + 6y = \\ &= 2f + y + 4.5\mu' - 0.5\hat{x} - 2z_1 + z_2. \end{aligned}$$

Furthermore, because of $z_2 \leq z_1$ and $\hat{x} \leq z_1$, we can deduce that

$$\begin{aligned} c(M) &\leq 2f + y + 4.5\mu' - 0.5\hat{x} - 2z_1 + z_1 \leq \\ &\leq 2f + y + 4.5\mu' - 1.5\hat{x}. \end{aligned} \tag{6.2.8}$$

From (6.2.7) and (6.2.8) we conclude that

$$\frac{c(M)}{c(M_{egal})} \leq \frac{2f + y + 4.5\mu' - 1.5\hat{x}}{2f + 2y + 3.5\mu' - 0.5\hat{x}}. \tag{6.2.9}$$

It is easy to see from $|M_{egal}^{(1,2)}| = \frac{\mu'+\hat{x}}{2}$ and $|M_{egal}^{(1,2)}| = |\varphi(M_{egal}^{(1,2)})| \leq |M'| = \mu'$, that $\hat{x} \leq \mu'$. Hence, by applying similar transformations to the previous case it results, that for $\mu' \neq 0$

$$\frac{c(M)}{c(M_{egal})} \leq \frac{4.5\mu' - 1.5\hat{x}}{3.5\mu' - 0.5\hat{x}} < \frac{9}{7},$$

for any $\hat{x} > 0$, which was the assumption of this case. In case $\mu' = 0$, similarly to other case, all edges of any stable matching are of type $(1,1)$ and all stable matchings are egalitarian.

□

To summarise, in all cases stable matching M has a cost within a factor of $9/7$ compared to any egalitarian stable matching. Now the proof of Theorem 6.1.3, case $d = 3$ is closed.

6.2.2 EGAL-4-$\mathcal{SRI}$

In this section we are to finish the proof of Theorem 6.1.3 for case $d = 4$.

Claim 6.2.7 (Application of Claims 6.1.5 and 6.1.7 for $d = 4$)**.** *Let $\mathcal{I} \in$ SOLV-4-$\mathcal{SRI}$-INS and $M, M' \in \mathcal{SM}(\mathcal{I})$ and let $\varphi : M \to M'$ be the standard bijection. Take $uw \in M$ and let $\varphi(uw) = pq$. The maximum of $r_u(M'(u)) + r_w(M'(w))$ and of $r_p(q) + r_q(p)$ varies according to Table 6.2.*

type of uw	$max(r_u(M'(u)) + r_w(M'(w)))$ and $max(r_p(q) + r_q(p))$	max ratio
$(1,1)$	$1+1$	1
$(1,2)$	$4+1$	$5/3$
$(1,3)$	$4+2$	$6/4 = 3/2$
$(1,4)$	$4+3$	$7/5$
$(2,2)$	$1+4,\ 4+1$	$5/4$
$(2,3)$	$4+2$	$6/5$
$(2,4)$	$4+3$	$7/6$
$(3,3)$	$2+4, 4+2, 3+3$	1
$(3,4)$	$3+4, 4+3$	1
$(4,4)$	$4+4$	1

Table 6.2. Maximum cost of change from M to M' in 4-$\mathcal{SRI}$.

Claim 6.2.8 (Application of Corollary 6.1.6 for $d = 4$)**.** *In $\mathcal{I} \in$ SOLV-4-$\mathcal{SRI}$-INS any stable matching approximates the egalitarian stable matching within a factor of* $5/3$.

Let us denote the number of $(4,4)$-pairs by y. Based on the sign of the right-hand side of inequality (6.2.2), two cases are once again distinguished.

Lemma 6.2.9. *Suppose that $2 \cdot |M_{egal}^{(1,2)}| - \mu' \leq 0$. In that case, any matching $M^* \in \mathcal{SM}(\mathcal{I})$ is within a factor of* $11/7$ *compared to M_{egal}, i.e. $c(M^*) \leq \frac{11}{7}c(M_{egal})$. Since the approximation ratio is true for any $M^* \in \mathcal{SM}(\mathcal{I})$, it is also true for M.*

Proof. Similarly to the proof of Lemma 6.2.5, we have $|M_{egal}^{(1,2)}| = \frac{\mu'-x}{2}$, where $x \geq 0$. Furthermore, denote the number of edges in M_{egal} with cost at least 5 by t. In summary, M_{egal} contains f $(1,1)$s, y $(4,4)$s, $\frac{\mu'-x}{2}$ $(1,2)$s, t edges with cost at least 5 and the rest of edges have cost exactly 4 (these are exactly the $(1,3)$- and $(2,2)$-pairs).

$$\begin{aligned} c(M_{egal}) &\geq 2f + 3 \cdot \frac{\mu' - x}{2} + 4 \cdot \left(\mu' - \frac{\mu' - x}{2} - t - y\right) + 5t + 8y = \\ &= 2f + t + 4y + 3.5\mu' + 0.5x. \end{aligned} \tag{6.2.10}$$

Let M^* be an arbitrary stable matching in $\mathcal{I}$. M^* must contain exactly the same f $(1,1)$-pairs and exactly the same y $(4,4)$-pairs as M_{egal}. Thanks to Claim 6.2.7, the total cost contribution of endpoints of $(1,2)$s in M_{egal} may become maximum 5 in M^*. For edges with cost 4 and at least 5 this maximum is 6 and 7, respectively (see Table 6.2). Consequently,

$$\begin{aligned} c(M^*) &\leq 2f + 5 \cdot \frac{\mu' - x}{2} + 6 \cdot \left(\mu' - \frac{\mu' - x}{2} - t - y\right) + 7t + 8y = \\ &= 2f + t + 2y + 5.5\mu' + 0.5x. \end{aligned} \tag{6.2.11}$$

From (6.2.10) and (6.2.11):

$$\frac{c(M^*)}{c(M_{egal})} \leq \frac{2f + t + 2y + 5.5\mu' + 0.5x}{2f + t + 4y + 3.5\mu' + 0.5x} \leq \frac{11}{7},$$

with the same argumentation, as previously.

□

Lemma 6.2.10. *Suppose that $2 \cdot |M_{egal}^{(1,2)}| - \mu' > 0$. In that case M approximates M_{egal} within a factor of $11/7$, i.e. $c(M) \leq \frac{11}{7} c(M_{egal})$.*

Proof. Denote $2 \cdot |M_{egal}^{(1,2)}| - \mu' = \hat{x} > 0$. Hence, $|M_{egal}^{(1,2)}| = \frac{\mu'+\hat{x}}{2}$. Furthermore, denote the number of edges in M_{egal} with cost at least 5 by t. In summary, M_{egal} contains f $(1,1)$s, y $(4,4)$s, $\frac{\mu'+\hat{x}}{2}$ $(1,2)$s, t edges with cost at least 5 and the rest of edges have cost exactly 4 (these are exactly the $(1,3)$- and $(2,2)$-pairs).

$$\begin{aligned} c(M_{egal}) &\geq 2f + 3 \cdot \frac{\mu' + \hat{x}}{2} + 4 \cdot \left(\mu' - \frac{\mu' + \hat{x}}{2} - t - y\right) + 5t + 8y = \\ &= 2f + t + 4y + 3.5\mu' - 0.5\hat{x}. \end{aligned} \tag{6.2.12}$$

Similarly to the proof of Lemma 6.2.6, let $|M^{(1,2)}| = z_1$. According to inequality (6.2.2), $z_1 \geq \hat{x}$. By taking φ as the standard bijection from M_{egal} to M and

$$z_2 = \left| M_{egal}^{(1,2)} \cap \varphi^{-1}\left(M^{(1,2)}\right) \right|,$$

we have that $z_1 \geq z_2$ and $|M_{egal}^{(1,2)}| - z_2$ edges are of type $(1,3)$ or $(1,4)$, hence with cost at most 5.

Consequently, the image of edges in M_{egal} of type $(1,1)$, $(4,4)$ and $(1,2)$ are all accounted for. Assume that a_2 edges from the t edges in M_{egal} with cost at least 5 map to $(1,2)$-pairs in M.

The rest of them convert into edges with weight at most 7 (see Table 6.2). Also suppose that a_1 edges from the ones in M_{egal} with cost exactly 4 map to $(1,2)$-pairs and the rest of them convert into cost at most 6. Hence,

$$\begin{aligned} c(M) &\leq 2f + 3z_1 + 5 \cdot \left(\frac{\mu' + \hat{x}}{2} - z_2 \right) + 6 \cdot \left(\mu' - \frac{\mu' + \hat{x}}{2} - t - y - a_1 \right) + 7(t - a_2) + 8y = \\ &= 2f + t + 2y + 5.5\mu' - 0.5\hat{x} + 3z_1 - 5z_2 - 6a_1 - 7a_2. \end{aligned}$$

What is more, outside the $z_2 + a_1 + a_2$ edges that converted into $(1,2)$-pairs, no other edge can map to a $(1,2)$, hence $z_1 = z_2 + a_1 + a_2$, or $z_1 - z_2 = a_1 + a_2$. Combined with the fact that $\hat{x}, z_2 \leq z_1$, we find that:

$$\begin{aligned} 3z_1 - 5z_2 - 6a_1 - 7a_2 &= 3z_1 - 5z_2 - 6(z_1 - z_2) - a_2 = \\ &= -3z_1 + z_2 - a_2 \leq \\ &\leq -3z_1 + z_1 = -2z_1 \leq \\ &\leq -2\hat{x}. \end{aligned} \tag{6.2.13}$$

Hence, we deduce that

$$c(M) \leq 2f + t + 2y + 5.5\mu' - 2.5\hat{x}. \tag{6.2.14}$$

From (6.2.12) and (6.2.14) it can be concluded that

$$\frac{c(M)}{c(M_{egal})} \leq \frac{2f + t + 2y + 5.5\mu' - 2.5\hat{x}}{2f + t + 4y + 3.5\mu' - 0.5\hat{x}}. \tag{6.2.15}$$

Again, from $|M_{egal}^{(1,2)}| = \frac{\mu' + \hat{x}}{2}$ and $|M_{egal}^{(1,2)}| = |\varphi(M_{egal}^{(1,2)})| \leq |M'| = \mu'$ it is obvious that $\hat{x} \leq \mu'$. Hence, for $\mu' \neq 0$

$$\frac{c(M)}{c(M_{egal})} \leq \frac{5.5\mu' - 2.5\hat{x}}{3.5\mu' - 0.5\hat{x}} < \frac{11}{7},$$

for any $\hat{x} > 0$. In case $\mu' = 0$, all edges of any stable matching are of type $(1,1)$ and all stable matchings are egalitarian.

□

6.2.3 EGAL-5-$\mathcal{SRI}$

Claim 6.2.11 (Application of Claims 6.1.5 and 6.1.7 for $d = 5$)**.** *Let $\mathcal{I} \in$ SOLV-5-$\mathcal{SRI}$-INS and $M, M' \in \mathcal{SM}(\mathcal{I})$ and let $\varphi : M \to M'$ be the standard bijection. Take $uw \in M$ and let $\varphi(uw) = pq$. The maximum of $r_u(M'(u)) + r_w(M'(w))$ and of $r_p(q) + r_q(p)$ varies according to Table 6.3.*

Claim 6.2.12 (Application of Corollary 6.1.6 for $d = 5$)**.** *In $\mathcal{I} \in$ SOLV-5-$\mathcal{SRI}$-INS any stable matching approximates the egalitarian stable matching within a factor of $6/3 = 2$.*

Denote the number of $(5,5)$-pairs by y. Based on the sign of the right-hand side of inequality (6.2.2), two cases are again distinguished.

type of uw	$max(r_u(M'(u)) + r_w(M'(w)))$ and $max(r_p(q) + r_q(p))$	max ratio
$(1,1)$	$1+1$	1
$(1,2)$	$5+1$	$6/3 = 2$
$(1,3)$	$5+2$	$7/4$
$(1,4)$	$5+3$	$8/5$
$(1,5)$	$5+4$	$9/6 = 3/2$
$(2,2)$	$1+5,\ 5+1$	$6/4 = 3/2$
$(2,3)$	$5+2$	$7/5$
$(2,4)$	$5+3$	$8/6 = 4/3$
$(2,5)$	$5+4$	$9/7$
$(3,3)$	$2+5, 5+2$	$7/6$
$(3,4)$	$5+3$	$8/7$
$(3,5)$	$5+4$	$9/8$
$(4,4)$	$3+5, 5+3, 4+4$	1
$(4,5)$	$5+4, 4+5$	1
$(5,5)$	$5+5$	1

Table 6.3. Maximum cost of change from M to M' in 5-$\mathcal{SRI}$.

Lemma 6.2.13. *Suppose that $2 \cdot |M_{egal}^{(1,2)}| - \mu' \leq 0$. In that case, any matching $M^* \in \mathcal{SM}(\mathcal{I})$ is within a factor of $13/7$ compared to M_{egal}, i.e. $c(M^*) \leq \frac{13}{7}c(M_{egal})$. Since the approximation ratio is true for any $M^* \in \mathcal{SM}(\mathcal{I})$, it is also true for M.*

Proof. Similarly to the previous proofs, we have $|M_{egal}^{(1,2)}| = \frac{\mu'-x}{2}$, where $x \geq 0$. Furthermore, denote the number of edges in M_{egal} with cost exactly 5 by t and the number of edges with cost at least 6 by w. In summary, M_{egal} contains f $(1,1)$s, y $(5,5)$s, $\frac{\mu'-x}{2}$ $(1,2)$s, t edges with cost exactly 5, w edges with cost at least 6 and the rest of edges have cost exactly 4 (these are exactly the $(1,3)$- and $(2,2)$-pairs).

$$\begin{aligned} c(M_{egal}) &\geq 2f + 3 \cdot \frac{\mu'-x}{2} + 4 \cdot \left(\mu' - \frac{\mu'-x}{2} - t - w - y\right) + 5t + 6w + 10y = \\ &= 2f + t + 2w + 6y + 3.5\mu' + 0.5x. \end{aligned} \tag{6.2.16}$$

Let M^* be an arbitrary stable matching in $\mathcal{I}$. M^* must contain exactly the same f $(1,1)$-pairs and exactly the same y $(5,5)$-pairs as M_{egal}. Due to Claim 6.2.11, the total cost contribution of endpoints of $(1,2)$s in M_{egal} may become maximum 6 in M^*. For edges with cost 4, 5 and at least 6 this maximum is 7,8 and 9, respectively (see Table 6.3). Thus,

$$\begin{aligned} c(M^*) &\leq 2f + 6 \cdot \frac{\mu'-x}{2} + 7 \cdot \left(\mu' - \frac{\mu'-x}{2} - t - w - y\right) + 8t + 9w + 10y = \\ &= 2f + t + 2w + 3y + 6.5\mu' + 0.5x. \end{aligned} \tag{6.2.17}$$

From (6.2.16) and (6.2.17):

$$\frac{c(M^*)}{c(M_{egal})} \leq \frac{2f + t + 2w + 3y + 6.5\mu' + 0.5x}{2f + t + 2w + 6y + 3.5\mu' + 0.5x} \leq \frac{13}{7}.$$

□

Lemma 6.2.14. *Suppose that $2 \cdot |M_{egal}^{(1,2)}| - \mu' > 0$. In that case M approximates M_{egal} within a factor of $13/7$, i.e. $c(M) \leq \frac{13}{7} c(M_{egal})$.*

Proof. The proof follows the same route as the proofs of Lemmas 6.2.6 and 6.2.10. First, we introduce the occurring notation.

Denote $2 \cdot |M_{egal}^{(1,2)}| - \mu' = \hat{x} > 0$. Hence, $|M_{egal}^{(1,2)}| = \frac{\mu'+\hat{x}}{2}$. Furthermore, denote the number of edges in M_{egal} with cost exactly 5 by t and the number of edges with cost at least 6 by w. Summarising, M_{egal} contains f $(1,1)$s, y $(5,5)$s, $\frac{\mu'+\hat{x}}{2}$ $(1,2)$s, t edges with cost exactly 5, w edges with cost at least 6 and the rest of edges have cost exactly 4 (these are exactly the $(1,3)$- and $(2,2)$-pairs).

$$\begin{aligned} c(M_{egal}) &\geq 2f + 3 \cdot \frac{\mu'+\hat{x}}{2} + 4 \cdot \left(\mu' - \frac{\mu'+\hat{x}}{2} - t - w - y\right) + 5t + 6w + 10y = \\ &= 2f + t + 2w + 6y + 3.5\mu' - 0.5\hat{x}. \end{aligned} \tag{6.2.18}$$

The z_1 $(1,2)$-pairs of M are assigned through φ to z_2 $(1,2)$-pairs, a_1 edges of cost exactly 4, a_2 edges of cost exactly 5 and a_3 edges of cost at least 6. Hence, $z_1 - z_2 = a_1 + a_2 + a_3$. Also, in M_{egal} $\frac{\mu'+\hat{x}}{2} - z_2$ edges convert to cost at most 6, $t - a_2$ to cost at most 8, $w - a_3$ to cost at most 9 and the rest of cost-4 edges (except those a_1 edges) to cost 7 in M. Hence,

$$\begin{aligned} c(M) \leq\ & 2f + 3z_1 + 6 \cdot \left(\frac{\mu'+\hat{x}}{2} - z_2\right) + \\ & + 7 \cdot \left(\mu' - \frac{\mu'+\hat{x}}{2} - t - w - y - a_1\right) + 8(t - a_2) + 9(w - a_3) + 10y = \\ =\ & 2f + t + 2w + 3y + 6.5\mu' - 0.5\hat{x} + 3z_1 - 6z_2 - 7a_1 - 8a_2 - 9a_3. \end{aligned}$$

Nevertheless,

$$\begin{aligned} 3z_1 - 6z_2 - 7a_1 - 8a_2 - 9a_3 &= 3z_1 - 6z_2 - 7(z_1 - z_2) - a_2 - 2a_3 = \\ &= -4z_1 + z_2 - a_2 - 2a_3 \leq -4z_1 + z_1 = -3z_1 \leq \\ &\leq -3\hat{x}. \end{aligned} \tag{6.2.19}$$

Consequently,

$$c(M) \leq 2f + t + 2w + 3y + 6.5\mu' - 3.5\hat{x}. \tag{6.2.20}$$

From (6.2.18) and (6.2.20) it can be concluded that

$$\frac{c(M)}{c(M_{egal})} \leq \frac{2f + t + 2w + 3y + 6.5\mu' - 3.5\hat{x}}{2f + t + 2w + 6y + 3.5\mu' - 0.5\hat{x}} < \frac{13}{7},$$

with the same arguments, as in the previous proofs of Lemmas 6.2.6 and 6.2.10.

□

Chapter 7

Conclusion and open questions

We investigated stable matchings within non-bipartite matching markets, which model kidney exchange programs or dormitory admission schemes. Special attention was paid to matchings with a particular form of optimality, namely egalitarianism, where the average rank of assignees takes its minimum. We proved Theorem 6.1.3, which was inspected prior to us in [3], yet the proof had flaws. Theorem 6.1.3 and Algorithm 4 provide us with a tool to approximate the egalitarian solution in the case of short lists within better factors then the ones known previously. Nevertheless, most applications rely on permitting users to specify a limited number of preferences, hence our result could bear fruit any such field of application (for instance, university admission schemes operated on nation-wide level).

However, there is much room for improvement in the area. Theorem 5.1.4 reminds us on the **UGC**-hardness of the problem of approximating egalitarian stable matchings within a factor of $2 - \varepsilon$ ($\varepsilon > 0$) among general instances. Yet, there are no previous known attempts on proving approximation-hardness for short list cases. Moreover, Theorem 5.2.2 provides us with a 2-approximation of the MIN WEIGHT-$\mathcal{SRI}$ problem in the general setting. Our representation, though, only counted on the approximation of short listed MIN WEIGHT-$\mathcal{SRI}$-instances. Hence, any enhancement on Teo and Sethuraman's algorithm could immediately lead to improvements in our result. In addition, we may not restrict only to constant length-limits, but to instances, too, where the length of lists are bounded by some increasing functions of n, the number of agents. In general, there could be $\Theta(n)$ acceptable partners for each agent. By restricting this to, e.g. $\mathcal{O}(\log n)$ we have a special case, yet large enough lists. Last, but not least, it would be interesting to extend egalitarian stable matchings to instances with generalised preference structures that involve ties as well. For instance, universities may rank school-leavers equally based on an integer-valued admission score.

Acknowledgements

This work was supported by the Cooperation of Excellences Grant (KEP-6/2017), by the Ministry of Human Capacities under its New National Excellence Program (ÚNKP-18-1-I-BME-309).

The author is pleased to thank his supervisor, Ágnes Cseh, on the everlasting support and the continuous mentorship throughout their entire work, only one product of which being this comprehensive report.

Bibliography

[1] P. Biró and S. Kiselgof. College admissions with stable score-limits. *Central European Journal of Operations Research*, 23(4):727–741, 2015.

[2] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *Proceedings of SOFSEM '07: the 33rd International Conference on Current Trends in Theory and Practice of Computer Science*, volume 4362 of *Lecture Notes in Computer Science*, pages 51–69. Springer, 2007.

[3] Á. Cseh, R. W. Irving, and D. F. Manlove. The stable roommates problem with short lists. *Theory of Computing Systems*, Sep 2017.

[4] T. Feder. A new fixed point approach for stable networks and stable marriages. *Journal of Computer and System Sciences*, 45:233–284, 1992.

[5] T. Feder. Network flow and 2-satisfiability. *Algorithmica*, 11:291–319, 1994.

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

[7] D. Gusfield. The structure of the stable roommate problem – efficient representation and enumeration of all stable assignments. *SIAM Journal on Computing*, 17(4):742–769, 1988.

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

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

[10] R. W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48:261–272, 1994.

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

[12] D. Knuth. *Mariages Stables.* Les Presses de L'Université de Montréal, 1976. English translation in *Stable Marriage and its Relation to Other Combinatorial Problems*, volume 10 of CRM Proceedings and Lecture Notes, American Mathematical Society, 1997.

[13] D. Lebedev, F. Mathieu, L. Viennot, A.-T. Gai, J. Reynier, and F. de Montgolfier. On using matching theory to understand P2P network design. In *Proceedings of INOC '07: International Network Optimization Conference*, 2007.

[14] L. Lovász and M. D. Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[15] M. Lucan, P. Rotariu, D. Neculoiu, and G. Iacob. Kidney exchange program: a viable alternative in countries with low rate of cadaver harvesting. *Transplantation Proceedings*, 35:933–934, 2003.

[16] D. F. Manlove. *Algorithmics of Matching Under Preferences.* World Scientific, 2013.

[17] N. Perach, J. Polak, and U. G. Rothblum. A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the Technion. *International Journal of Game Theory*, 36:519–535, 2008.

[18] I. Ríos, T. Larroucau, G. Parra, and R. Cominetti. College admissions problem with ties and flexible quotas. Technical report, SSRN, 2014.

[19] A. E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92:991–1016, 1984.

[20] A. E. Roth. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica*, 54:425–427, 1986.

[21] A. E. Roth and M. A. O. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, volume 18 of *Econometric Society Monographs.* Cambridge University Press, 1990.

[22] C.-P. Teo and J. Sethuraman. LP based approach to optimal stable matchings. In M. E. Saks, editor, *Proceedings of SODA '97: the 8th ACM-SIAM Symposium on Discrete Algorithms*, pages 710–719. ACM-SIAM, 1997.

[23] National Resident Matching Program. 2017 annual report. Web document available at `http://annualreport.nrmp.org/`. Accessed 23 October 2018.

[24] National Resident Matching Program. About the NRMP. Web document available at `http://www.nrmp.org/about-nrmp/`. Accessed 23 October 2018.