NYILATKOZAT

Név: Csáji Gergely Kál

ELTE Természettudományi Kar, szak: Matematikus MSc

NEPTUN azonosító: RU5HZ5

Szakdolgozat címe:
Complexity of Stable Matching Problems

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által irt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2022.05.25

a hallgató aláírása

Complexity of Stable Matching Problems

MSc Thesis
2022. May

Author:
Gergely Kál Csáji

Supervisors:

Tamás Király, associate professor at the Department of Operations Research, Eötvös Loránd University
and
Péter Biró, senior research fellow at the Institute of Economics, Hungarian Academy of Sciences

Faculty of Science, Eötvös Loránd University

Contents

Acknowledgements

I would like to thank my supervisor, Tamás Király for introducing me to this topic and guiding me to successfully start my research in the field. I am grateful for the many advice and remarks I received from him and looking forward to continue working together with him.

I would also like to thank Péter Biró, my other supervisor, for providing many interesting research topics to me and for allowing me to get a job joining his research group.

I am also thankful for Kirstóf Bérczi, who also included me in his research group and have worked together with me on many interesting research questions.

I thank God for the countless blessings I received in this time and for being the rock I can base my life on.

Finally, I would like to thank my girlfriend, my family and my friends for supporting me during my studies and always being there for me when I needed it.

1 Introduction

In this thesis we give a comprehensive review about stable matchings and some of their most interesting generalizations. The thesis follows a complexity theoretic viewpoint, therefore when we will deal with given problems, our main aim is to either show that the problem is hard, for example NP-hard, or to give a polynomial-time algorithm. For some problems, even more efficient algorithms are available than the ones mentioned in here, but from our complexity theoretic viewpoint, they are not as important. Instead, we try to include as many of the most interesting topics in the field and their complexity as we can.

Stable matchings were introduced by a seminal paper of Gale and Shapley [12] in 1962. The original problem describes two sided matching markets, where the agents have preferences on each other. The aim is to match them in a way such that there will be no pair of agents, who mutually would like to deviate from the outcome, e.g. both of them prefers the other to the one he/she got. These types of pairs would likely lead to a chain of deviations, destabilizing the market, which we are trying to avoid.

Since the introduction of stable matchings, an enormous amount of research has been done in the field and countless generalizations and special cases has been considered. For a quite comprehensive review of the current state and aims of the field, see the book of David Manlove [23]. For a more detailed review of the two original problems: the stable marriage and the stable roommates, see the book of Gusfield and Irving [13].

Stable matching problems have also many applications. Two of the most well known applications are the US National Resident Matching Program (NRMP) and the university application mechanisms in several countries, for example in Hungary. We will deal with those problems in much more detail in the following sections.

A similar model is also used to describe Kidney Exchange Programs (KEPs). Here, the agents are donor-recipient pairs. The motivation is that in many cases, there is a friend or relative to the patient who is willing to be a donor for the patient but for some medical reasons they are not compatible. Therefore, they search for a cycle of donor-recipient pairs such that each donor is compatible with the next recipient of the cycle. Here the graph in the background has a vertex for each donor-recipient pair, and the directed edges represent that the donor is compatible with the recipient. Also, based on how similar they are in terms of blood types, etc it is possible to give certain preferences for the vertices.

Apart from providing a comprehensive review of many generalizations of the stable matching problem, the thesis contains several new results too. For example, the theorems of Section 4.2 and 4.3, Section 6.2, and Section 7.2 are mainly new results that are also available in my working paper [7]. Also, the thesis contains simpler proofs for many known results, than the original ones.

The thesis is organized the following way. In Section 2, we introduce the original Stable Marriage problem (SM) and investigate its structural and algorithmic properties. We also deal with some simple generalizations, for example when the agents are allowed to be indifferent between some other agents in their preferences. In Section 3, we describe the other classical problem in the field, the Stable Roommates problem (SR) and describe the novel polynomial-time algorithm of Irving that solves it. Again, we also consider some simple extensions of the problem and their computational complexity. In Section 4, we move on to a the case, where the underlying structure is a hypergraph, called the Stable Hypergraph Matching problem. We also explore a fundamental result of Scarf, closely related to stable matching problems. In Section 5 and 7, we consider a generalization, that is motivated by real-life application, the Hospital Resident Couple problem (HRC). Finally, in Section 6, we move on to a slightly more abstract problem, called the Stable Flow problem (SF), where instead of a matching of the agents, we are searching for flows in a network.

2 The Stable Marriage problem

The original problem on the field, proposed by Gale and Shapley in their landmark paper in 1962 [12], is the so called stable marriage problem. The main difference here from the well known marriage problem is that here the agents on the two sides of the graph (the men and women respectively) have preferences on each other, which to be fair is a quite reasonable assumption in many applications. Formally, let $G = (U, W, E)$ be a bipartite graph. Suppose that for each woman $w \in W$ there is a strict preference list $>_w$ over the men, and for each man $m \in U$ there is a preference list $>_m$ over the women. If a man m is not on w's list, then we will say that m is *unacceptable* to w. We will always assume that acceptability is mutual, that is m finds w acceptable if and only if w finds m acceptable. This will be represented by an edge joining m and w in G.

Let M be a matching in G, that is, a subset of edges such that each vertex is incident to at most one edge. Denote the partner of an agent a in M by $p_M(a)$. Note that if a is unmatched, then $p_M(a) = \emptyset$. We will assume that the preferences of the agents satisfy that they prefer every acceptable partners to being left alone.

Definition 2.1. We say that an edge mw *blocks* M, if $m >_w p_M(w)$ and $w >_m p_M(m)$, that is both of them would prefer to switch from their current partners to each other.

Observe, that such pairs would likely lead to new connections, such that some agents would switch partners and some agents would be left alone. Such actions would destabilize the market, so we want to avoid having blocking pairs. This motivates the following definition:

Definition 2.2. A matching M is called *stable* if there is no pair blocking M.

The first question arising is that when does a stable matching exists and how can we find one? The answer is, as shown by Gale and Shapley in their original paper is that a stable matching always exists and it can be found by a simple

Algorithm 1 Gale-Shapley algorithm

```
Initialize M := ∅
prop(m) :=first woman in m's preference list
while There is a man m who is free in M and prop(m) ≠ ∅ do
    Let w = prop(m)
    if m >_w p_M(w) then
        M := M ∪ mw − p_M(w)w
    else
        M := M
        prop(m) :=next woman in m's preference list
    end if
end while
```

and elegant algorithm, see Algorithm 1 ($prop(m)$ denotes the best woman in m's list, he hasn't proposed to yet).

Now we show that the algorithm terminates in a stable matching in $\mathcal{O}(|E|)$ time.

Theorem 2.3. *(Gale, Shapley [12] The Gale-Shapley algorithm always returns a stable matching M with running time $\mathcal{O}(|E|)$.*

Proof. In each iteration there is a proposal happening and if a man m have proposed to a woman w, then he never proposes to her again. That means the number of iterations is at most the number of possible proposals, which is $|E|$.

Now let M be the output of the algorithm. It is clear that M is a matching. To prove stability let us suppose for the contrary that there is an edge mw that blocks M. This means that m prefers w to $p_M(m)$ (remember $p_M(m)$ can be the empty set if every woman rejected m). But then m had to propose to w during the algorithm, and w had to reject m (possibly at a later iteration). That means w got a better partner and since all women can only improve their position in the algorithm it follows that $p_M(w) >_w m$, contradiction. □

Lemma 2.4. *In each possible iteration of the (men proposing) Gale-Shapley algorithm, it returns the same matching M. Also, each man m gets the best possible partner he can get in any stable matching in M.*

Proof. Suppose one execution of the algorithm yielded a matching M but there is an other stable matching M', such that $w' = p_{M'}(m) >_m p_M(m) = w$. That means, that w' had to reject m, because she was engaged to a better partner m'. Furthermore, suppose that this rejection was the first rejection that happened between a man m and a woman w such that there is a stable matching containing mw. (We will call such pairs *stable pairs* in the future) Now, m' hasn't been rejected by any stable partner yet, so $w' >_{m'} p_M(m')$. But then $w'm'$ blocks M, contradiction. □

Notice that this lemma states quite a remarkable fact: if we assign every man the best possible stable partner he can get, than we not only obtain a matching, but a stable matching. In the next sections, we will see that there are several similar properties of stable matchings, for example they admit a nice lattice structure.

One can show in a similar way that, in this matching M each woman obtains the worst possible stable partner she can get. Also by symmetry, if we run the algorithm with women as proposals, than the resulting matching will be optimal for the woman and worst for the men. In the following sections, we will call this man optimal stable matching M_o and the woman optimal stable matching M_z.

2.1 Structure of stable matchings

In this section we discuss the basic structural properties of stable matchings. We will assume that the preference lists are complete, that is no man and woman consider each other unacceptable, and that the set of men and the set of woman have the same size, that is $|A| = |B|$. This can be assumed without loss of generality, by adding dummy agents to one side who are worst for everyone and extending the preference lists to complete by adding the unacceptable agents to the end. Then, a stable matching M' from the new instance gives a stable matching M in the original by deleting the edges from it that correspond to originally unacceptable relations. (for any possible blocking edge mw, at least one of them had a better partner in M', but if such an edge was deleted because it was an unacceptable edge, then since mw is worse, it is also an unacceptable edge, so it is deleted). For the other direction, a stable matching M can be extended to a stable matching M' in the new instance by finding a stable matching between the originally unassigned agents.

The section summarizes the corresponding results from the book of Gusfield and Irving [13], where a much more detailed version is available.

Theorem 2.5. *Let M and M' be two stable matchings with $mw \in M$ and $mw \notin M'$. Then precisely one of $\{m, w\}$ have a better partner in M than in M'. Furthermore, the number of people, who prefer M is the same as the number of people who prefer M'.*

Proof. Let U and W denote the sets of men and woman who prefer M to M', and U' and W' the ones who prefer M'. Then, since there can be no pair $(m, w) \in U \times W \cup U' \times W'$ (otherwise mw would block M or M'), it follows that $|U| \leq |W'|$ and $|U'| \leq |W|$. But $|U| + |U'| = |A| = |B| = |W| + |W'|$, so $|U| = |W|$ and $|U'| = |W'|$ □

Definition 2.6. We say that M *dominates* M', if every man has an at least as good partner in M then in M'. If there is a man, who obtains a strictly better partner, then we say that M *strictly dominates* M'.

This dominance defines a partial order $\preceq$ on the set of stable matchings. As we have seen, the unique largest element is M_o, while the unique smallest element is M_z, the man and woman optimal stable matchings respectively. Now we prove another interesting result that is a more general theorem than lemma

2.4: if we take any two stable matchings and assign to every man their better partner in the two matchings, then we get another stable matching.

Theorem 2.7. *Let M and M' be two different stable matching. Then, if we give each man his better partner in $M \cup M'$, then the result M'' is a stable matching.*

Proof. First suppose that there are two men $m \neq m'$ who receive the same partner in M''. Then one of them, say m prefers M and the other M' and $w = p_M(m) = p_{M'}(m')$. So by applying theorem 2.5 to mw we get that w prefers M' and to $m'w$ we get that w prefers M, contradiction. So M'' is indeed a matching.

Now suppose that there is a blocking edge mw. That means that m likes w better than both $p_M(m)$ and $p_{M'}(m)$. Also w likes m better than her current partner, that is either $p_M(w)$ or $p_{M'}(w)$. Either way, we get that mw blocks M or M', contradiction again. □

Note that these theorem has quite strong consequences too. It means by "adding" stable matchings, we can obtain new stable matchings that dominate both the original ones. Also, by symmetry it is easy to see that we get a stable matching if each man obtains his worse partner too.

Definition 2.8. Let $M \wedge M'$ denote the matching where each man gets his better partner and $M \vee M'$ the matching where they get the worse one.

Let $\mathcal{M}$ denote the set of stable matchings. Now it's easy to see that $\wedge_{M \in \mathcal{M}} M = M_o$ and $\vee_{M \in \mathcal{M}} M = M_z$. We will show that the set of stable matchings form a distributive lattice.

Definition 2.9. A partial order $(P, \preceq)$ forms a distributive lattice if

1. For every $a, b \in P$ there is an element $a \wedge b$ called the *meet*, such that $a \wedge b \preceq a$, $a \wedge b \preceq b$ and for every $c \in P$ such that $c \preceq a$ and $c \preceq b$ it follows that $c \preceq a \wedge b$.

2. For every $a, b \in P$ there is an element $a \vee b$ called the *join*, such that $a \preceq a \vee b$, $b \preceq a \vee b$ and for every $c \in P$ such that $a \preceq c$ and $b \preceq c$ it follows that $a \vee b \preceq c$.

3. For every $a, b, c \in P$ it holds that $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ and $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$.

Theorem 2.10. *In the stable marriage problem the partial order $(\mathcal{M}, \preceq)$ forms a distributive lattice, with $M \wedge M'$ being the meet and $M \vee M'$ being the join.*

Proof. $M \wedge M' \preceq M, M'$ is immediate from the definitions. Now let N be a matching such that $N \preceq M$ and $N \preceq M'$. That means that in N each man gets an at most as good partner as their worse partner in $M \cup M'$. So $N \preceq M \wedge M'$ by definition. The other case is similar.

Finally, let M, N, L be stable matchings. It is easy to verify that each man gets the same partner in $M \wedge (N \vee L)$ and in $(M \wedge N) \vee (M \wedge L)$ and similarly with the other property. □

Now we return to the case where the preference list might be incomplete and the two sides can have different cardinalities. Here of course a stable matching may leave some agents without a partner. The interesting fact is, that every stable matching matches the same set of agents, so if an agent is unassigned in one stable matching, then he/she is unassigned in all of them.

Theorem 2.11. *Every Stable matching M matches the same set of agents.*

Proof. Suppose there is a man m who is matched in a stable matching M but is not in M'. Let w be his partner in M. Now, w has to be matched by M', otherwise mw would block M', and by lemma 2.5 she prefers M'. Let m' be her partner. Again m' has to be matched in M,... etc we obtain an infinite chain of agents that cannot get back to m, since m is alone in M', contradiction. □

2.2 The rotation poset and its applications

In the previous section we have seen that the set of stable matchings have a really nice structure. In this section we describe a quite powerful representation of the set of stable matchings that has many algorithmic benefits for several problems. Furthermore, we show that this representation is quite compact and even better, it can be explicitly constructed in $\mathcal{O}(|V|^2)$ time.

Definition 2.12. Let $\mathcal{M}(m, w)$ denote the set of stable matchings that contain the edge mw.

It is easy to see that if M and M' are in $\mathcal{M}(m, w)$, then so are $M \wedge M'$ and $M \vee M'$, so $\mathcal{M}(m, w)$ is a (possibly empty) sublattice of $\mathcal{M}$. So we can define the man optimal matching in $\mathcal{M}(m, w)$, let this be $M(m, w)$.

Definition 2.13. A stable matching M is called *irreducible*, if $M = M(m, w)$ for some edge mw.

Denote the set of irreducible matchings by $I(\mathcal{M})$ and let $(I(\mathcal{M}), \preceq)$ be the partial order on $I(\mathcal{M})$ defined by the same domination relation.

Definition 2.14. Let $(R, \preceq)$ be a partial order and $S \subset R$. We say that S is a *closed subset*, if no element in $R \setminus S$ precedes any element in S.

Definition 2.15. Let M be a stable matching. The irreducible support of M, denoted by $U(M)$ is $\{M(m, w) : (m, w) \in M\}$.

Now we show that a stable matching M can be generated by assigning each man the worst partner he has in any stable matching among $U(M)$.

Lemma 2.16. *For any stable matching M, $M = \vee U(M)$.*

Proof. If there would be an edge $m_1 w_1 \in M \setminus \vee U(M)$, then since $M(m_1, w_1) \in U(M)$, there is an edge $m_2 w_2 \in M$, such that in $M(m_2, w_2)$ m_1 gets a strictly worse partner than w_1. But $M(m_2, w_2)$ is the man optimal stable matching containing $m_2 w_2$, so it dominates M, contradiction. □

Let $\overline{U(M)}$ be the set of irreducible matchings that dominate some matching in $U(M)$, which is the closure of $U(M)$. Then, similarly $M = \vee\overline{U(M)}$.

Lemma 2.17. *There is a set S of stable matchings not containing M, such that $M = \vee S$, if and only if $M \notin I(\mathcal{M})$*

Proof. The if part has already been shown in the previous lemma. For the other direction, suppose M is in $I(\mathcal{M})$. But, since $M \notin S$ and every matching in S strictly dominates M, for any $mw \in M$ there is the matching $M(m,w) \in S$ strictly dominating M, contradiction. □

Lemma 2.18. *Different closed subsets of $I(\mathcal{M})$ yield different stable matchings.*

Proof. Suppose $S \neq T$ are closed subsets, but $\vee T = \vee S$. Since S and T are closed, there is a matching $M = M(m,w)$ in $S \setminus T$ (by symmetry), such that M does not dominate any matching in T. Clearly, the partner of m in $\vee S$ is no better than w. Suppose the same holds for $\vee T$. Then, there is a matching $M' \in T$, such that $p_{M'}(m) \leq_m w$. But this means that M', which is dominated by $M' \wedge M(m,w)$ is dominated by $M(m,w)$ too, since $mw \in M' \wedge M(m,w)$, contradiction. □

From these lemmas the following nice characterization theorem follows:

Theorem 2.19. *The stable matchings and the closed subsets of $(I(\mathcal{M}), \preceq)$ are in one to one correspondence, such that $M = \vee S$, where S is the set of stable matchings that dominate M. Furthermore if S and S' are two closed subsets of $(I(\mathcal{M}), \preceq)$, then $M = \vee S$ dominates $M' = \vee S'$, if and only if $S \subset S'$.*

Also the set $I(\mathcal{M})$ can be constructed in polynomial time. We only need to modify the Gale-Shapley algorithm the following way, to find $M(m,w)$ or report that no stable matching containing mw exist: During the algorithm, every woman other than w rejects m, and w rejects any man other than m. It is easy to see that this algorithm returns $M(m,w)$, if it exists.

There is an other representation of stable matchings, that is even more useful, that uses so called rotations. First, we introduce the notion of rotations. For a matching M, let $s_M(m)$ denote the first woman w on m's list, such that w strictly prefers m to her partner $p_M(w)$ in M, if there is any.

Definition 2.20. $\rho = (m_0, w_0), .., (m_{r-1}, w_{r-1})$ is called a *rotation (exposed) in M*, if $w_{i+1} = s_M(m_i)$ and $w_i = p_M(m_i)$ for each $i = 0, .., r-1$ (mod r).

If ρ is a rotation exposed in a stable matching M, then the elimination of ρ from M, denoted by M/ρ is the matching where we delete the edges $m_i w_i$ from M and add the edges $m_i w_{i+1}$.

Lemma 2.21. *If M is a stable matching and $\rho = (m_0, w_0), .., (m_{r-1}, w_{r-1})$ is a rotation exposed in M, then $M' = M/\rho$ is also a stable matching and M dominates M'.*

Proof. First we show that M' is stable. Suppose there is a blocking edge mw. Since M was stable, only edges adjacent to some m_i or w_i agent in ρ could become blocking. But each w_i got strictly better, so any blocking edge must contain an $m_i \in \rho$. To block, w has to be better for m_i than $w_{i+1} = s_M(m_i)$, but by the definition of $s_M(m_i)$, w likes her partner in M better, so she also likes her partner in M' better than m.

The domination follows from the fact that each woman got weakly better in M', therefore by theorem 2.5 each man got weakly worse, so M' is dominated by M. □

By a chain of Theorems and Lemmas similar to the ones before, one can prove that the set $\Pi(\mathcal{M})$ consisting of the rotations has a nice partial order such that it is isomorphic to $I(\mathcal{M})$ after the removal of the man optimal matching M_o, which results in the following two characterization theorems. A detailed approach can be found in the book of Gusfield and Irving [13].

Theorem 2.22. *[13] A rotation ρ' precedes ρ in $\Pi(\mathcal{M})$ if and only if ρ' has to eliminated in any chain of rotation elimination starting from M_o that results in a stable matching M, where ρ becomes exposed.*

Theorem 2.23. *[13] The following statements are true:*

- *There is a one to one correspondence between the closed subsets of $\Pi(\mathcal{M})$ and the stable matching.*
- *A closed set $S \subset \Pi(\mathcal{M})$ corresponds to a stable matching M, if and only if M is obtained from the man-optimal stable matching M_o, by eliminating the rotations of S in any (possible) order. Also that is the only way to obtain M from M_o by rotation eliminations.*
- *If S and S' are the set of rotations corresponding to stable matchings M and M', then M dominates M' if and only if $S \subset S'$.*

Furthermore, it is also possible to construct a graph $G(\mathcal{M})$, whose vertices are the rotations, such that its transitive closure is exactly $\Pi(\mathcal{M})$.

This rotation poset can be used for several purposes. For example we can use it to find minimal weight stable matchings or stable matchings with forced and forbidden edges. However, these kind of problems will have a much easier solution after we have described the stable matching polytope, that is the polytope whose vertices are exactly the characteristic vectors of stable matchings, so we do not describe them here.

An interesting application however, that uses rotations but there is no known algorithm for it using the stable matching polytope is robust matchings. A *k-robust matching* is a matching M such that not only M is stable, but if we make k swaps in the preferences of the agents in any possible way (where a swap stands for interchanging two adjacent agents in a preference list), then M still remains stable. There is a novel algorithm from Chen et al. [6] that can decide whether there exist a k-robust matching and if yes, it can find one in polynomial-time. The algorithm's heart is the nice structure of the rotations we have described in the previous theorems.

Therefore, while some problems are much easier to solve using the stable matching polytope, in many cases the rotation poset can be an at least as powerful tool too.

2.3 The Stable Matching Polytope

In this section we introduce perhaps the most powerful characterization of stable matchings, called the stable matching polytope. It is a polytope whose vertices correspond to the characteristic vectors of stable marriages.

The first one to characterize the stable matching polytope was Vande Vate [36]. However, this characterization only worked for the case when the graph was a complete bipartite graph. Later, Rothblum [32] gave an alternative description of the stable matching polytope, that worked for any bipartite graph. In this thesis, we follow a simplified proof by Roth et al. [31]. In the beginning of this section we recall two well known theorems in operation research. The first is the Duality theorem. It relates the so called primal and dual programs. The primal linear program is of the form

$$\begin{gathered}\max(cx):\\ Ax \leq b\\ x \geq 0,\end{gathered}$$

where A is an $n \times m$ matrix, $c \in \mathbb{R}^m$ is the cost function and $b \in \mathbb{R}^n$ is the bounding vector.

The Dual program of such an instance is:

$$\begin{gathered}\min(yb):\\ yA \geq c\\ y \geq 0.\end{gathered}$$

Theorem 2.24. *(Duality theorem) The primal program has a finite optimum value, if and only if the dual has too. Also, in this case the two values are equal, so* $\max cx = \min yb$.

The other theorem we will use is about the relation of the optimal primal-dual solution pairs.

Theorem 2.25. *Let x be an optimum solution of the primal, and y an optimum solution of the dual. Then the two vectors satisfy the complementary slackness conditions, that are:*

1. $(Ax - b)_i y_i = 0$*, for* $i = 1, .., n$*, where* v_i *denotes the* i*-th component of* v*, and*

2. $(yA - c)_j x_j = 0$ *for* $j = 1, .., m$*.*

Reversely, if x and y are two solutions satisfying these conditions, then $cx = yb$, so they are optimal solutions.

Definition 2.26. Let $G = (U, W, E)$ be a bipartite graph with preferences on the vertices. Let **P** be the polytope defined by

$$\{x \in \mathbb{R}^{m \times n} : \ x \geq 0, x(E(v)) \leq 1 \ \forall v \in U \cup W, \ x(\phi(e)) \geq 1 \ \forall e \in E \ x_{mw} = 0 \ \forall mw \notin E\},$$

where $E(z)$ denotes the edges incident to z in E and $\phi(uv) = \{f : \ f \geq_u uv$ or $f \geq_v uv\}$.

First, we state a Lemma about the primal and dual solutions of P. Observe that the dual of $max(\sum_{m,w} x_{mw}) : x \in P$ is

$$\begin{gathered} min(\textstyle\sum_{m \in U} y_m + \sum_{w \in W} y_w - \sum_{m,w} y_{mw}) : \\ y_m + y_w - \textstyle\sum_{w' <_m w} y_{mw'} - \sum_{m' \leq_w m} y_{m'w} \geq 1, \\ y \geq 0. \end{gathered}$$

So in our case, complementary slackness for an optimal primal-dual pair x, y is equivalent to the following:

1. if $x(E(v)) < 1$, then $y_v = 0$,
2. if $x(\phi(mw)) > 1$ then $y_{mw} = 0$
3. if $y_m + y_w - \sum_{w' < w} y_{mw'} - \sum_{m' < m} y_{m'w} > 1$, then $x_{mw} = 0$.

Lemma 2.27. *Each $x \in P$ is an optimal solution of $max(\sum_{mw \in E} x_{mw})$: $x \in P$. Also, letting $y_m = \sum_{w \in W} x_{mw}$, $y_w = \sum_{m \in U} x_{mw}$ and $y_{mw} = x_{mw}$, y is an optimal dual solution.*

Proof. Let $x \in P$ be an arbitrary solution and let y be as stated. y is a feasible dual solution, since $y_m + y_w - \sum_{w' <_m w} y_{mw'} - \sum_{m' \leq_w m} y_{m'w} = \sum_{w \in W} x_{mw} + \sum_{m \in U} x_{mw} - \sum_{w' <_m w} y_{mw'} - \sum_{m' \leq_w m} y_{m'w} = \sum_{w' >_m w} x_{mw'} + \sum_{m' >_w m} x_{m'w} \geq 1$ and also $y \geq 0$.

Furthermore $\sum_{m \in U} y_m + \sum_{w \in W} y_w - \sum_{m,w} y_{mw} = 2 \sum_{m,w} x_{mw} - \sum_{m,w} x_{mw} = \sum_{m,w} x_{mw}$, so the objective functions cx and yb have the same value, so by the duality theorem both x and y are optimal primal and dual solutions respectively. □

Remark 2.28. This lemma implies that for each $x \in P$, if $\sum_{w \in W} x_{mw} > 0$, then there is an optimal dual solution with $y_m > 0$. Then, by complementary slackness, each optimal solution, so each $x' \in P$ must satisfy that $\sum_{w \in W} x'_{mw} = 1$. Also, if $x_{mw} > 0$, then $\sum_{w' >_m w} x'_{mw'} + \sum_{m' >_w m} x'_{m'w} + x'_{mw} = 1$ for every $x' \in P$ by complementary slackness, since $\sum_{w' >_m w} x'_{mw'} + \sum_{m' >_w m} x'_{m'w} + x'_{mw} = y'_m + y'_w - \sum_{w' < w} y'_{mw'} - \sum_{m' < m} y'_{m'w}$.

Now we state the main theorem of this section.

Theorem 2.29. *The polytope P is exactly the convex hull of the characteristic vectors of stable matchings.*

Proof. First, let M be a stable matching. Then, the characteristic vector of M, denoted by χ_M is inside P. The first two set of constraints are obviously satisfied since M is a matching, and if there would be an edge e with $x(\phi(e)) < 1$, then by the integrality of χ_m $x(\phi(uv)) = 0$, meaning $uv \notin M$ and also, no edge better than uv for u or v is in M, so uv blocks M, contradiction.

Similarly, if x is an integer solution of P, then it is a characteristic vector of a matching M, and it is stable, since a blocking edge uv would mean that $x(\phi(uv)) = 0$.

It remains to show that any vector $x \in P$ can be written as a convex combination of characteristic vectors of stable matchings. First, we state a lemma:

Lemma 2.30. *Let $x \in P$ be a vector. For each man $m \in U$, we assign m the best possible partner among the ones with $x_{mw} > 0$. If no such woman exists for m, then m remains free. Denote the resulting edge set M_x. Then, M_x is a stable matching. Also, each woman w gets the worst possible partner among the ones with $x_{mw} > 0$.*

Proof. First we show that M_x is a matching. Suppose there is two men m and m' who obtain the same partner w in M_x and assume by symmetry that $m >_w m'$. Then, by the construction of M_x : $\sum_{w':w'>_m w} x_{mw'} = 0$, since m obtained its best partner with $x_{mw'} > 0$. So by x satisfying $x(\phi(mw)) \geq 1$ it must hold that $\sum_{m'':m'' \geq_w m} x_{m''w} = 1$. But $m' <_w m$ and $x_{m'w} > 0$, contradicting $x(E(w)) \leq 1$. So M_x is a matching.

Suppose mw blocks M_x. Then, m prefers w to his partner, meaning $\sum_{w' \geq_m w} x_{mw'} = 0$. This implies $\sum_{m' >_w m} x_{m'w} = 1$, so w must have a partner she prefers to m, contradiction.

Now we prove that each woman with $\sum_{m \in U} x_{mw} > 0$ obtains her worst partner in M_x. Let w be a woman and m be the worst partner of w with $x_{mw} > 0$. Then,

$$\sum_{m' >_w m} x_{m'w} < 1 = \sum_{m' \in U} x_{m'w} = \sum_{m' \geq_w m} x_{mw},$$

by remark 2.28 about complementary slackness. Also, $\sum_{w' >_m w} x_{mw'} + \sum_{m' >_w m} x_{m'w} + x_{mw} = 1$.

This means that $\sum_{w' >_m w} x_{mw'} = 0$, meaning $mw \in M_x$. □

Finally, we prove the theorem. Let $x \in P$ be an arbitrary non integer vector and let z be the incidence vector of M_x. Since z is integral, $z \neq x$. We will show that there is a $0 < \delta < 1$, such that $y^\delta = \frac{x - \delta z}{1 - \delta} \in P$. Then, since $x = \delta z + (1 - \delta) y^\delta$, we can write x as a convex combination of two different vectors in P, meaning x is not an extreme point of P.

First we show that, for δ small enough, $y^\delta \geq 0$. But this follows from the fact that $z_{mw} = 0$, if $x_{mw} = 0$, by the definition of M_x.

To see that $y^\delta(E(v)) \leq 1$ for any $v \in U \cup W$ notice that if $\sum_{w \in W} x_{mw} = 1$, then $\sum_{w \in W} z_{mw} = 1$, and similarly for woman, so for any δ, the condition is satisfied. And as we have observed, if $\sum_{w\in W} x_{mw} < 1$, then it is 0 and so is $\sum_{w\in W} z_{mw}$, so again, any δ suffices.

Lastly, we need to show that $y^\delta(\phi(mw)) = \sum_{w' \geq_m w} y^\delta_{mw'} + \sum_{m' >_w m} y^\delta_{m'w} \geq 1$ for all $mw \in E$. Writing

$$\sum_{w' \geq_m w} y^\delta_{mw'} + \sum_{m' >_w m} y^\delta_{m'w} = \frac{1}{1-\delta}(x(\phi(mw)) - \delta z(\phi(mw))$$

we see that the situation would not hold for arbitrary small positive δ, if and only if $x(\phi(mw)) = 1$, but $z(\phi(mw)) > 1$. But $z(\phi(mw)) > 1$ means that both m and w is matched to someone they prefer more than each other in M_x. This implies $\sum_{w' >_m w} x_{mw'} > 0$. By Lemma 2.30, w is matched to her worst choice among the man with $x_{mw} > 0$, so we get that $\sum_{m' >_w m} x_{m'w} = 1$. But combining these two conditions, we get that $x(\phi(mw)) > 1$, contradiction.

□

Being able to characterize stable matching by the extreme points of a polytope allows us to compute efficiently stable matchings of minimal/maximal weight for any cost function.

For example, by taking $c(mw) = r_m(w) + r_w(m)$ (where $r_m(w) = k$, if w is the k-th worse woman for m) we can find the so called egalitarian stable matching that maximizes the sum of the "satisfaction" of the agents in both side.

Theorem 2.31. *For any cost function c on the edges, a minimum weight stable matching can be found in polynomial-time.*

Another very interesting problem is the stable matching problem with forced and forbidden edges. Here we are given a set of forbidden edges $F_1 \subset E$ and a set of forced edges $F_2 \subset E$ and we want to find a stable matching containing all forced edges and none of the forbidden edges, if any.

But notice, that now with the LP description of stable matchings, we can just find the extreme point of P, minimizing the cost function c, such that $c(e) = 1$, for $e \in F_1$, $c(e) = -1$ for $e \in F_2$, and $c(e) = 0$ otherwise. It is straightforward to verify, that if a minimum cost matching has cost less than or equal to $-|F_2|$, then it satisfies the requirements and otherwise no such stable matching exists. These problems only have complex combinatorial algorithms. Some of them may be faster then solving the above linear program, but for our complexity theoretic viewpoint, they are not very important, the main takeaway here is that thanks to this characterization, all these problems can be easily solved in polynomial time.

Corollary 2.32. A stable matching that contains a given set of edges F_1 and excludes another set of edges F_2 can be found in polynomial-time, if there is any.

2.4 Stable Marriage with Ties

Now suppose the preference lists are not strict, so there can be agents who are indifferent between some other agents. This is represented by ties in the preference lists. If there are ties, then different notions of blocking and stability arises. These are the following. Let $(G, <)$ be a bipartite graph with preferences and let M be a matching.

Definition 2.33. A pair *mw weakly blocks*, if $m \geq_w p_M(w)$, $w \geq_m P_M(m)$ and at least one of them strictly prefers the other to his/her partner. A pair *mw strongly blocks* if $m >_w p_M(w)$ and $w >_m p_M(m)$.

Definition 2.34. M is *weakly stable* if there is no strongly blocking pair to M. M is *strongly stable* if there is no weakly blocking pair.

The most common notion in the literature in the case of ties is weak stability, so we are considering that in this thesis also. From here on, when saying stability in a preference structure with ties, we will always mean weak stability.

If the task is to only find a (weakly) stable matching, then the problem is easy. We can just break the ties arbitrarily and run the Gale-Shapley algorithm. We know that for any breaking of ties there always exists a stable matching and this matching is stable in the original instance too, since if there would be a strongly blocking pair, than that would block in the instance with no ties too.

So at first everything looks just as nice as in the strict case. However, there are some properties that no longer hold if we allow ties. For example it will no longer be true that each stable matching covers the same set of agents: just consider a $K_{1,2}$ graph, where the agent who is alone ranks the other two agents in a tie. Then, both possible matchings are stable, but cover different vertices.

Therefore an important question that arises is that can we find a maximum cardinality stable matching when ties are allowed? Sadly, the answer seems to be no, as the problem is NP-complete, which was shown by Manlove et al. [24]. At least, the NP-completeness of this problem allows for some nice and elegant reductions for other hard variants of the stable matching problem. Note that if the preference lists are complete, than the Gale-Shapley algorithm returns a complete stable matching, therefore in that case, the problem is solvable. So we suppose that the preference lists can be incomplete.

COM-SMTI
Input: A bipartite graph G and preference profile $\mathcal{P}$ with ties
Question: Is there a complete weakly stable matching?

Theorem 2.35. *[24] COM-SMTI even if the ties only occur on one side of the graph.*

Proof. NP-containment is trivial.

We reduce from the EXACT-MAXIMAL-MATCHING proven to be NP-complete by Horton et al. [14]. In this problem we are given a graph G, an integer number k and the task is to determine whether there exists an inclusionwise maximal matching M, such that $|M| = k$. They also showed NP-hardness for the case when the graph G is a subdivision graph, that is G can be obtained from another graph G' by substituting each edge with a path of length 2 (or equivalently divide each edge with a point). Therefore we can suppose that $G = (U, W, E)$ is bipartite and one side of the vertices all have degree 2. We can assume that G was connected and not a tree, so $|U| \geq |W|$. Furthermore, we can assume that $|U| = |W|$, because otherwise if $|U| = |W| + r$, then we can add r distinct $K_{1,2}$-s to the graph, such that the two sides will have the same size and there is a maximal matching of size k in the original if and only if there is one of size $k + r$ in the new one. Also, this way we preserve that one side of the vertices all have degree 2.

So let $G = (U, W, E)$, $|U| = |W| = n$, and $k \in \mathbb{Z}$ be an instance of EXACT-MAXIMAL-MATCHING. We construct a new bipartite graph $H = (A, B, F)$, where $A = U \cup U' \cup X$, $U = \{m_1, .., m_n\}$, $U' = \{m'_1, .., m'_n\}$, $X = \{x_1, .., x_{n-k}\}$ and $B = W \cup Y \cup Z$, $W = \{w_1, .., w_n\}$, $Y = \{y_1, ..y_n\}$ and $Z = \{z_1, .., z_{n-k}\}$. The graph between U and W is the same as G, so we can suppose that each man in U has exactly 2 neighbours in W. Let this two neighbours of m_i be denoted by w_{j_i} and w_{k_i}, $j_i < k_i$. We define for each $w_j \in W$ a set M_j consisting of the neighbours of w_j in U. Also, we define a set M'_j consisting of those m'_i-s, such that $m_i w_j \in E$ and $j = k_i$.

The preferences of the agents are the following:

$$\begin{aligned} m_i: &\quad y_i > w_{j_i} > w_{k_i} > [Z] \\ m'_i: &\quad y_i > w_{k_i} \\ x_i: &\quad [W] \\ w_j: &\quad (M_j \cup M'_j) > (X) \\ y_j: &\quad (m_j, m'_j) \\ z_j: &\quad (U) \end{aligned}$$

for each $i = 1, .., n$ and $j = 1, ..n$. The notation $[S]$ for a set S denotes an arbitrary strict order on the elements of S and (S) denotes that the agents in S are tied. The edges of F are exactly between those pairs that consider each other acceptable.

Now let us suppose that we have a maximal matching M of size k. Then, we make a matching M' as follows: For each edge $m_i w_j \in M$, if $j = j_i$, then we add $m_i w_{j_i}$ and $m'_i y_i$ to M', otherwise we add $m_i y_i$ and $m'_i w_{k_i}$. Those $n - k$ men m'_i that are still unmatched are matched to the corresponding y_i in M'. The $n - k$ unmatched w_j-s are paired with the agents from X arbitrarily and the $n - k$ unmatched m_i-s are paired with the agents from Z arbitrarily.

It is easy to see that M' is a complete matching. Suppose there is a strictly blocking pair to M'. Then, the woman in the pair has to be from W, since the other women are indifferent between their partners. Also, the man has to be from U (since a woman from W could only improve with someone from $U \cup U'$, but from U' every man is either with its first choice, or their only better choice y_j is with someone better than them), so suppose the blocking pair is (m_i, w_j). But this means that m_i is with a woman from Z, so m'_i has to be with y_i and w_j

is with a man from X. From this we get that m_i and w_j are both not covered by M, but they are adjacent, contradiction.

For the other direction suppose M' is a complete weakly stable matching. Then each y_i is paired with one of m_i or m'_i, so the edge set M that we obtain by adding those $m_i w_j$ edges such that m_i or m'_i is with w_j in M' is a matching. Since M' is complete, k of the w_j-s are matched to agents in $U \cup U'$, so $|M| = k$. Suppose that M is not maximal. Then there is an edge $m_i w_j$ that can be added to M. That means that m'_i was matched to y_i, m_i to someone from Z and w_j to someone from X. But then the pair (m_i, w_j) would block M', contradiction. $\square$

Actually, it is possible to show by a similar reduction that the following stronger theorem is also true:

Theorem 2.36. *COM-SMTI is NP-complete, even if only the women's lists contain ties, and if a woman's list contain a tie, than her preference list only consist of a single tie of length 2.*

Let MAX-SMTI be the problem of finding a maximal cardinality weakly stable matching. By theorem 2.36, it is NP-hard to find one. However, there is a simple $\frac{3}{2}$-approximation by Király [19] for the case when the ties are on one side only. He also gave a $\frac{5}{3}$-approximation algorithm for the general case, that has been improved by McDermind [25] to a $\frac{3}{2}$-approximation algorithm later. For the one-sided tie case, the current best algorithm is a $(1 + \frac{1}{e})$ approximation by Lam and Plaxton [21]. Here, we only describe the simple $\frac{3}{2}$-approximation algorithm for the one-sided ties case.

Suppose ties are only in the preference lists of the women.

Algorithm 2 (Király)

Initialize $M := \emptyset$, each $m \in U$ is free and unpromoted
prop(m) :=first woman in m's preference list
while there is a man m who is free in M and *prop*$(m) \neq \emptyset$ **do**
  Let $w = prop(m)$
  if $m >_w p_M(w)$ or ($m \sim_w p_M(w)$, m is promoted but $p_M(w)$ is not) **then**
    $M := M \cup mw - p_M(w)w$
  else
    $M := M$
    prop(m) :=next woman in m's preference list
    if *prop*$(m) = \emptyset$ and m in unpromoted **then**
      m becomes promoted
      prop(m) :=first woman in m's preference list
    end if
  end if
end while

Informally, the algorithm is really similar to a standard Gale-Shapley algorithm, with the following difference: Each man m, if he has been rejected by all women, becomes promoted and starts to propose to the women again. Now that he is promoted, any woman who would be indifferent between him and an other man, if he is promoted but the other is not, she will prefer him and reject the other. It is easy to see that this algorithm terminates in at most $2|E|$ iteration, therefore it is also polynomial.

Theorem 2.37. *Algorithm 2 produces a (weakly) stable matching M, such that if M_{OPT} is a maximal cardinality stable matching, then $|M_{OPT}| \leq \frac{3}{2}|M|$.*

Proof. Let M be the matching given by the algorithm. First we show that M is stable.

Suppose there is an edge mw (strongly) blocking M. Than, m prefers w to his partner (if there is any), therefore he has been proposed to w. But w rejected m, which is only possible if w already had an at least as good partner as m and since each woman's position weakly improves during the algorithm, this also holds in M, therefore mw is not strictly blocking, contradiction.

Now we show that if M_{OPT} is a maximal cardinality stable matching, then $|M_{OPT}| \leq \frac{3}{2}|M|$. Take the symmetric difference of M and M_{OPT}. Every component of it is either a path or a cycle. Clearly, every component has at least two edges, since both matchings must be inclusionwise maximal, since they are stable. We show that there is no component such that it has more than $\frac{3}{2}$ as many edges from M_{OPT}. To show this, we only have to show that there is no component with 2 edge from M_{OPT} and one edge from M. Suppose there is a component C line that. Let the vertices of C be $\{m_1, m_2, w_1, w_2\}$ and suppose $m_1w_1, m_2, w_2 \in M_{OPT}$ and $m_1w_2 \in M$.

m_1w_1 does not block M and w_1 is free in M, therefore m_1 prefers w_2 to w_1. m_1w_2 does not block M_{OPT}, therefore w_2 weakly prefers m_2 to m_1.

During the algorithm, since w_1 is unassigned in M (so noone proposed to her), m_1 cannot have proposed to everyone in his preference lists and so he cannot become promoted. But m_2 is unassinged M too, so he must got promoted and then rejected again by w_2, which could only happen, if w_2 strictly prefers m_1. But then m_1w_2 strongly blocks M_{OPT}, contradiction.

□

2.5 The Stable Allocation problem

In this section we describe a generalization of the stable matching problem, where both the vertices and the edges are allowed to have capacities, introduced by Baiou and Balinski [2], which is called the *stable allocation* problem.

Let $G = (U, W, E)$ be a bipartite graph and suppose each $v \in U \cup W$ has a strict ranking $>_v$ on the adjacent edges. Furthermore, suppose that we are given a capacity $b(v)$ for each $v \in U \cup W$ and a capacity $c(e)$ for each $e \in E$. An allocation x is called *stable*, if there is no edge uw, such that $x_{uw} < c(u, w)$, $\sum_{w' \in W} x_{uw'} < b(u)$, or there is a w', with $x_{uw'} > 0$ and $w' <_u w$, and similarly $\sum_{u' \in U} x_{u'w} < b(w)$ or there is a u' with $x_{u'w} > 0$ and $u' <_w u$.

By a straightforward adaptation of the Gale-Shapley algorithm, one can always find a stable solution in pseudo-polynomial time, but it is not at all trivial how to solve this problem in truly polynomial time.

Here, we describe briefly an algorithm by Dean and Munshi [10]. Since the two sides have capacities now, it is more convenient to call them jobs and machines. For ease of notation, we refer to the jobs simply as $i \in U$ and to the machines as $j \in W$.

We add a dummy job, indexed as job 1, that has capacity large enough in order for all the original machines to be matched to it. Also, its adjacent edges have large enough capacity too. Job 1 ranks every machine in an arbitrary order, while every machine ranks job 1 as the worst acceptable partner. Then, we also add a dummy machine, indexed as machine 1, which we add to the end of the preference list of each job (even job 1). Machine 1 ranks the jobs in an arbitrary order, ending with job 1 and has capacity large enough such that each original job can be fully assigned to it. Furthermore, we make this capacity in a way such that the sum of capacities are the same for both sides.

The reason behind this is to force that in any stable solution all agents are fully matched.

Given an assignment x, for each machine $j \in W$, we denote by r_j the least preferred job i, with $x_{ij} > 0$. Also, for each job i, denote by q_i the most preferred machine with $x_{ij} < c(i,j)$. q_i will be the proposal pointer pointing to the current best partner of i, while r_j will be the refusal pointer, pointing to the edge where the machine j would reject in the Gale-Shapley algorithm too.

We also define a graph $G(x)$, such that $G(x)$ has vertex set $U \cup W$ and the edge set of $G(x)$ is $\{jr_j \mid j \in W\} \cup \{q_i i \mid i \in U - \{1\}\}$. Let C be a component of $G(x)$. We say that component C is *fully assigned*, if each job $i \in C$ is saturated in x. It is also easy to see from the definition of $G(x)$, that each component has the same number of edges and vertices, so it contains a unique cycle, except the one containing the dummy machine being a tree.

Now we describe the algorithm. The algorithm starts with the assignment x in which every job but the dummy is unassigned, and every machine is fully assigned to job 1. Then, in each step, we choose a component C in the graph $G(x)$ that is not fully assigned. Then, if C does not contain the dummy machine, then take the unique cycle D_C in C and increase the value of x by $\delta(D_C) = \min\{x_{r_j j} \mid j \in D_C \cap W, c(i,q_i) - x_{iq_i} \mid i \in D_C \cap U\}$ on the edges of the form iq_i and decrease it by $\delta(D_C)$ on the edges of the form $r_j j$. (note that the cycle must be alternating between these two kind of edges).

If C is the tree component containing machine 1, then take any job i that is not fully assigned. Then, take the unique path P_C from i to machine 1. Job i proposes $\delta(P_C) = \min\{x_{r_j j} \mid j \in P_C \cap W; c(i,q_i) - x_{iq_i} \mid i \in P_C \cap U; b(i) - \sum_{j \in W} x_{ij}\}$ to r_i. Then, if q_i becomes oversaturated, it refuses $\sum_{i' \in U} x_{i'q_i} + \delta(P_C) - c(q_i)$ on the edge $r_{q_i}q_i$, if it is greater than 0. (by the choice of $\delta(P_C)$, we know that the $x_{r_{q_i}q_i}$ value is at least as big as the refused value). Then we iterate proposals and rejection on the path P_C, until we arrive at a machine that does not become oversaturated. Notice, that the dummy machine can receive even all the assignments from the original jobs, therefore we surely arrive at such a machine in P_C.

Then, we update the q_i-s, the r_j-s, and $G(x)$ and start again, until all components are fully assigned in $G(x)$.

Theorem 2.38. *[10] The algorithm computes a stable solution in* $\mathcal{O}(|E|\log(|V|))$ *time.*

Proof. The proof of the running time requires a special data structure, called dynamic trees, which we omit here. We only prove that the running time is polynomial in $|V|, |E|$. Notice, that during the algorithm it never happens that the x value of an edge increases and then decreases, or decreases and then increases. Therefore, if an edge leaves $G(x)$, it never becomes a part of it again. Also, in each iteration, an edge leaves $G(x)$ (and an other one enters) or a job becomes fully saturated, therefore the number of iterations is bounded by $|E|+|V|$. It is also clear that each iteration can be executed in polynomial time, so the algorithm is polynomial.

To show that the algorithm computes a stable allocation, we show that it induces a set of proposals and rejections that a modified Gale-Shapley algorithm also could have done. If we augment on a path, then it is true, however if the augmentation is on a cycle, then in a Gale-Shapley execution, no job would have proposed, because they were saturated.

Therefore, we modify the original instance as follows. Let x denote the solution output by the algorithm and x_o denote the job optimal stable assignment. For each job $i \in U$, let $l(i)$ denote the least-preferred machine in i's preference list for which $x_{il(i)} > 0$. For every $j \geq_i l(i)$ except the dummy machine, set $c(i,j) = x_{ij}$. Finally, increase $b(i)$ by one for all jobs $i \in U$.

The crucial observation is that the job-optimal assignment for the new instance is almost identical to x, except with the one extra unit for each job being assigned to the dummy machine. Also, it is possible that the Gale-Shapley algorithm applied to the new instance I' only uses the proposals to the dummy machine at the very end of its execution. This means that the difficulty with cycle components no longer remains, as every job in I' is only partially assigned through the entire process of building x_o. Therefore, the augmentations performed by the algorithm correspond to a set of aggregated proposals and rejections that the GS algorithm could have performed on I' with the right order of proposals, meaning $x = x_o$. This proves that x is a (job optimal) stable assignment.

□

3 The Stable Roommates problem

The stable roommates (SR) problem is the version of the stable matching problem, where the underlying graph is not bipartite. So the agents cannot be separated into two types, like in the stable marriage problem. The problem is called stable roommates problem, because it can be interpreted as the agents being students who are assigned to rooms of size two. Each student has a ranking of the other students that represents his/her preference over who he/she would like to share a room with.

Formally, we are given a graph $G = (V, E)$ and a ranking $>_v$ for each vertex $v \in V$. We would like to find a matching M that is stable in the same sense as before, so there is no blocking pair uv, with both of them preferring each other to their current partner in M.

3.1 The algorithm of Irving

Irving's algorithm [15] was the first efficient algorithm for the stable roommates problem. The algorithm checks whether there is a complete stable matching in an SR instance, and if yes it finds one. The algorithm works for the case when the preference lists are complete, so each participants ranks all of the others or in other words, the underlying graph is complete. Obviously, in such an instance only a complete matching can be stable if $|V|$ is even, and a matching leaving out exactly one vertex, if $|V|$ is odd.

But as we show in the following lemma, we can find a stable matching in any SR instance if there is any by reducing it to an instance where the preferences are complete and $|V|$ is even, so we can use Irving's algorithm on this new instance.

Lemma 3.1. *Let $G = (V, E)$ be a stable roommates instance. Then, there is an other instance $G' = (V', E')$, such that $|V'|$ is even, G' is complete and there is a complete stable matching M' in G', if and only if there is a stable matching M in G, which can be found by taking $M = M' \cap E$.*

Proof. Suppose that $|V|$ is even and fix an ordering $v_1, .., v_n$ on the vertices. Make the graph G complete by adding all the remaining edges and extend the preferences by adding the originally unacceptable agents to the end of the preference lists in the order we fixed in the beginning.

Suppose M is a stable matching of G. Let the vertices that are not matched in M be $U = \{v_{i_1}, ..., v_{i_k}\}$ with $i_1 < i_2 < .. < i_k$. Clearly, no two agents in U can be acceptable to each other in G, since then they would block M.

We make a matching M' by adding the edges $v_{i_1}v_{i_2}, ..., v_{i_{k-1}}v_{i_k}$ to M.

Suppose there a blocking edge ab to M' in G'. ab cannot be an original edge by the stability of M. If a or b is assigned in M, say a, then by the construction of the preferences, a prefers $p_M(a) = p_{M'}(a)$ to b, contradiction.

Otherwise, if $a, b \in U$, then let $a = v_i$, $b = v_j$, $i < j$. Then, by the construction of M', a is matched to an agent who is before b in the order of the vertices, so again a prefers its partner to b, contradiction.

So if there is a stable matching in G, then there is a complete stable matching in G', and furthermore $M' \cap E = M$.

Now suppose there is a complete stable matching M' in G' and let $M = M' \cap E$. So we take the edges of M' that are edges in G.

Suppose there is a blocking edge $ab \in E$ to M. ab did not block M', so one of them, let's say a had a better partner in M'. But since b is already acceptable to a, so is his/her partner in M' meaning the edge is also in M, so a has a partner better than b in M, contradiction.

If the number of vertices is not even, then we add an additional vertex, that will be the worst choice for every original agent, whose ranking will be the same as the ordering of the vertices. Then, in the same way we can show that any stable matching M can be extended to a complete stable matching in G' and any complete stable matching in G' defines a stable matching in G. □

Now we describe the algorithm of Irving. It consists of two phases. The first phase has proposals and rejections like the Gale-Shapley algorithm, while in the second phase, we reduce the preference lists further by eliminating rotations until each preference list only consists of a single entry.

First we introduce some notations and definitions.

Definition 3.2. A *preference table* T is a table whose rows are the agents, and the entries in row v are the other agents in the order of v's preference list.

We assume that in the beginning each agent finds every other acceptable and throughout the algorithm we will maintain that agent u is in v's row if and only if v is in u's row. For an agent v let $f_T(v)$, $s_T(v)$ and $l_T(v)$ denote the first, second and last entries in v's row in the preference table T.

Phase 1: In the beginning of phase 1, each agent is free. Then, while there is a free agent v who has a nonempty list in T, it proposes to its first entry u in T and becomes semiengaged to u. Then, for every agent z who were already semiengaged to u we assign them to be free again and lastly, for each agent z who is worse than v for u, we delete each pair (u, z) from the preference table T (so we delete the corresponding entry in both agent's row). Let us call the reduced preference table T remaining at the end of phase 1, a *phase 1-table*.

Remark 3.3. If T is a phase 1-table, then

1. $v = f_T(u)$ if and only if $u = l_T(v)$
2. (u, v) is deleted from T if and only if $l_T(u) >_u v$ or $l_T(v) >_v u$.

We will call a table *stable*, if it satisfies 1. and 2., and no agent has an empty list.

Lemma 3.4. *Let T be a phase 1-table. Then all stable pairs are included in T*

Proof. Suppose (u, v) is the first stable pair deleted during phase 1. Suppose it happened because an agent z proposed to u. Then, u prefers z to v, so z can have no stable partner better than u, because (u, v) was the first stable pair deleted. But then, any stable matching containing the edge uv is blocked by uz, contradiction. □

Corollary 3.5. If an agent v has an empty list in T, then there is no stable matching.

Lemma 3.6. *If every agent in T has only one entry, then the entries define a stable matching.*

Proof. By remark 3.3 no pair (u, v) not in T can block any matching with all pairs included in T. Also if each agent has exactly one entry in T, then these entries define a unique matching, containing all pairs of T, so it is stable. □

Because of this, in the case when there is an agent with an empty list remaining, the algorithm terminates saying there is no stable matching. Also, if each agent has exactly one entry, than the algorithm outputs the unique matching defined by them, which we have seen is stable.

Otherwise, we proceed to the second phase of the algorithm.

Definition 3.7. Let T be a phase 1-table. $\rho = (u_0, v_0), ..., (u_{r-1}, v_{r-1})$ is called a *rotation exposed in* T, if $v_i = f_T(u_i)$ and $v_{i+1} = s_T(u_i)$ for each i. $\{u_0, .., u_{r-1}\}$ is called the U *set* of ρ and $\{v_0, .., v_{r-1}\}$ the V *set* of ρ.

Since $u_{i+1} = l_T(v_{i+1}) = l_T(s_T(u_i))$, any u_i uniquely determines a rotation as does any v_i.

Lemma 3.8. *If T is a phase 1-table, such that no agent's list is empty and there is a list that have more than one entry, then there exists a rotation ρ exposed in T.*

Proof. If an agent v has only one entry u in its list, then by remark 3.3 it is the only one entry in u's list. So for each agent u with more than one entry, $s_T(u)$, $f_T(u)$ and $l_T(u)$ has also more than one entry. So starting from one such agent u and then always stepping to $l_T(s_T(u))$, sooner or later we find a cycle that corresponds to the U set of a rotation. Letting $v_i = f_T(u_i)$ and $v_{i+1} = s_T(u_i)$ we can find a rotation ρ exposed in T. □

Phase 2: Iteratively, find a rotation ρ exposed in the current table T and eliminate it, which means that we delete every entry worse than u_{i-1} in v_i's row. We denote the table obtained from T by eliminating ρ, T/ρ.

Lemma 3.9. *Suppose ρ is a rotation exposed in T and that there are no empty list in T/ρ. Then*

1. $f_{T/\rho}(u_i) = v_{i+1}$ *for every* i
2. $l_{T/\rho}(v_i) = u_{i-1}$ *for every* i *and*
3. $f_{T/\rho}(u) = f_T(u)$ *for each agent not in the* U *set and* $l_{T/\rho}(v) = l_T(v)$ *for each agent not in the* V *set of* ρ.

Proof. Since $u_i = l_T(v_i)$, (u_i, v_i) is deleted. Also (u_i, v_{i+1}) cannot be deleted, since otherwise it could only be because of u_i, meaning that $u_i = v_j$ for some j and it prefers u_{j-1} to v_{i+1}, which means that every entry in u_i's row got deleted (since v_{i+1} was the second best $u_{j-1} = v_i$, that is also deleted), contradiction.

All entries worse than u_{i-1} for v_i are deleted. If (u_{i-1}, v_i) were deleted too, then $u_{i-1} = v_j$ and prefers u_{j-1} to v_i, so v_j's row would become empty, contradiction.

The first entry in any agent's row outside of U can't be in the V set of ρ, so they are not deleted. Similarly for the last entries for the agents not in V. □

Lemma 3.10. *Let ρ be a rotation exposed in a stable table T. If T/ρ contains no empty lists, then T/ρ is also stable.*

Proof. Condition 1 follows directly from the previous lemma. To see that if $(u, v) \notin T/\rho$ if and only if $l_{T/\rho}(u) >_u v$ or $l_{T/\rho}(v) >_v u$, notice that if (u, v) got deleted during the elimination of ρ, then one of the agents say u was in the V set of ρ, so every agent remaining in u's list is better than v. □

Now we arrive at the main theorem which is the heart of Irving's roommates algorithm.

Theorem 3.11. *If table T contains a stable matching and ρ is a rotation exposed in T, then T/ρ also contains a stable matching.*

Proof. Let M be a stable matching in T. Suppose there is a pair $(u_i, v_i) \in \rho$ that is not matched in M. Then, u_i is matched to $v_{i+1} = s_T(u_i)$ or someone worse. By the stability of M, this implies that v_{i+1} is matched to u_i or someone better than u_i. Since $u_{i+1} = l_T(v_{i+1})$, this means that $u_{i+1}v_{i+1} \notin M$ and v_{i+2} is matched to u_{i+1} or someone better,..., every v_j is matched to u_{j-1} or someone better, so no edges of M are deleted when eliminating ρ, meaning M is a stable matching in T/ρ also.

Now suppose all of the pairs (u_i, v_i) are matched in M. Then, the sets U and V of ρ are disjoint. Because otherwise, suppose $u_i = v_j$ for some i, j. Then $f_T(u_i) = v_i = u_j = l_T(v_j) = l_T(u_i)$, meaning u_i's row contains only one entry, contradiction.

So, if we assign the pairs (u_i, v_{i+1}), $i = 0, .., r-1$ to each other instead, we obtain another matching M'. Also, these edges all remain after the elimination of ρ.

Now suppose that M' is not stable and there is a blocking edge ab.

If one of them, say $a = u_i$ for some i, then $b = v_i$. But v_i is matched to u_{i-1}, who it prefers to u_i, contradiction.

If $a = v_i$ for some i and $b \neq u_i$, then there is an agent x, who prefers v_i to its current partner and v_i prefers x to u_{i-1}. But then, if $x \neq v_j$ for any j, then since $u_{i-1} >_{v_i} u_i$, the edge $v_i x$ would block M, which was stable, contradiction. And if $x = v_j$ for some j, then again, the edge $v_i v_j$ would block M, since both of them were assigned to their worst choice in M.

If a, b are not in the U or V sets of ρ, then both a and b receive the same partner in M' as in M, so ab would block M too, contradiction.

So M' is a stable matching that is contained in T/ρ. □

Summerizing the results, we get that during the second phase of the algorithm, either the table contains an agent with an empty list and then there is no stable solutions; each agent's list contain exactly one entry which defines a stable matching, or if not, then we can always find a rotation ρ, and by eliminating ρ we obtain a table with strictly less entries, such that this reduced table also contains at least one stable matching, if the original contained one. So the algorithm indeed produces a correct result and finds a stable matching if there is any and can be implemented in polynomial-time.

For the sake of completeness, we give a pseudocode of the algorithm in Algorithm 3.

Algorithm 3 Irving's algorithm

```
Initialize each agent v to be free and T to be a complete table
Phase 1:
while ∃x, such that x is free and has a nonempty preference list do
    y := f_T(x)
    for each z who was semiengaged to y do
        z becomes free
    end for
    x becomes semiengaged to y
    for each v who is worse for y than x do
        delete (y, v) from T
    end for
end while
Phase 2:
while There is an agent whose list has at least 2 entries in T and no list in
T is empty do
    Find a rotation ρ exposed in T
    T := T/ρ
end while
if Some list is empty in T then
    OUTPUT: No stable matching
else
    OUTPUT: T, which corresponds to a stable matching
end if
```

3.2 Tan's algorithm and stable partitions

It is important to note that there is an even more powerful similar algorithm for the stable roommates problem, in the sense that it can be used for finding a stable half integral matching even if there is no integral one. This algorithm is due to Tan [35]. It finds a so called stable partition of a stable roommates problem which is described the following way:

Definition 3.12. Let $(G, >)$ be a stable roommates instance. A *stable partition of* $(G, >)$ is a permutation $\pi : V \to V$ such that for each $v_i \in V$:

1. if $\pi(v_i) \neq \pi^{-1}(v_i)$, then $v_i\pi(v_i), v_i\pi^{-1}(v_i) \in E$ and $\pi(v_i) >_{v_i} \pi^{-1}(v_i)$ and

2. For each v_j adjacent to v_i if $\pi(v_i) = v_i$ or $v_j >_{v_i} \pi^{-1}(v_i)$, then $\pi^{-1}(v_j) >_{v_j} v_i$.

We call an agent v a *singleton in* π, if $\pi(v) = v$.

Theorem 3.13. *(Tan) [35] Let $(G, >)$ be a stable roommates instance. Then, it admits a stable partition π, which can be found in $\mathcal{O}(|V|^2)$ time. Furthermore each stable partition has the same set of singleton agents and the same set of odd cycles. Finally, there exists a stable matching if and only if there exist a stable partition without odd cycles of length ≥ 3.*

Now we show that this stable partition can be used to find a stable half-integral matching. Suppose we have a stable partition π. Define a half-integral matching M^π as follows:

- if v_i is a singleton then it is unassigned in M^π,
- otherwise, if $\pi(v_i) = \pi^{-1}(v_i)$, then $M^\pi(v_i\pi(v_i)) = 1$,
- for every other v_i, $M^\pi(v_i\pi(v_i)) = M^\pi(v_i\pi^{-1}(v_i)) = 0.5$,
- for every other edge $M^\pi(e) = 0$.

Lemma 3.14. *If π is a stable partition, then M^π is a stable half-integral matching.*

Proof. Suppose there is a blocking edge v_iv_j.

By (1) of definition 3.12, no edge of the form $v_i\pi(v_i)$ blocks, so $v_j \notin \{\pi(v_i), \pi^{-1}(v_i)\}$.

But then, if v_i is unassigned (meaning $\pi(v_i) = v_i$) or $v_j >_{v_i} \pi^{-1}(v_i)$ (note that $\pi^{-1}(v_i)$ is the worse edge in M^π for v_i if it has two different half edges), then by (2) of definition 3.12, $\pi^{-1}(v_j) >_{v_j} v_i$, so v_j is saturated with better agents than v_i, contradiction.

□

Stable partitions are useful for many kind of problems. One such problem is, when we are given a stable roommates instance, and we want to remove a minimal number of agents such that the remaining graph admits a stable matching. It has been shown by Tan [34], that this number is exactly the number of odd cycles in (any) stable partition.

Here, we give a new, much simpler proof to show that removing one agent from each odd cycle of a stable partition is optimal. Let us call a set $S \subset V$ *removable*, if $G - S$ admits a stable matching. We describe a simple algorithm that first computes a stable partition and then deletes an agent from each odd cycle and matches the remaining agents, presented as Algorithm 4; and then we prove its optimality. Note, however, that a removable set does not necessarily contain a vertex from each odd cycle, so optimality is not obvious. We show the correctness of the algorithm in the proof of the theorem below.

Theorem 3.15. *Algorithm 4 runs in in polynomial time and finds a smallest removable set.*

Algorithm 4

Input: A graph $G = (V, E)$ with strict preferences $>_v$ on the vertices.
Output: A smallest removable set $S \subseteq V$ and a stable matching M in $G - S$.

Set $S \leftarrow \emptyset$ and $M \leftarrow \emptyset$.
Find a stable partition π using Theorem 3.13.
Let $C_1, \ldots, C_k$ be the set of odd cycles of π of length ≥ 3, and let $C_{k+1}, \ldots, C_l$ be the set of even cycles of π. Let $\{v_0^i, \ldots, v_{j_i}^i\}$ denote the vertices of C_i, where $\pi(v_{j_i}^i) = v_0^i$, and $\pi(v_j^i) = v_{j+1}^i$ otherwise.
for $i = 1, \ldots, k$ **do**
    Set $S \leftarrow S + v_0^i$ and $M \leftarrow M \cup \{v_{2j-1}v_{2j} \mid 1 \leq j \leq j_i/2\}$.
end for
for $i = k+1, \ldots, l$ **do**
    Set $M \leftarrow M \cup \{v_{2j-2}v_{2j-1} \mid 1 \leq j \leq j_i/2\}$.
end for
return S, M

Proof. By Theorem 3.13, the algorithm has polynomial running time. We prove the correctness in two steps. First, we show that Algorithm 4 outputs a removable set.

Lemma 3.16. *Algorithm 4 outputs a set $S \subseteq V$ and a matching M that is stable in $G - S$.*

Proof. Let uv be an edge of $G - S$ not in M; we show that uv is not a blocking edge. Recall that π denotes the stable partition of $(G, >)$ obtained by applying Tan's algorithm.

Assume first that at least one of u and v is a singleton in π, say, $\pi(u) = u$. Then for any of its neighbors $w \in N(u)$, we have $\pi(w) \neq w$, $\pi(w) >_w u$, and $\pi^{-1}(w) >_w u$ since π is a stable partition. In particular, this holds for v. Since v is matched to either $\pi(v)$ or $\pi^{-1}(v)$ in M, uv is not a blocking edge.

Consider the case when $\pi(u) \neq u$ and $\pi(v) \neq v$. Since every vertex in $V \setminus S$ that is not a singleton in π gets a partner in M, both u and v are matched by the algorithm. If $u \neq \pi(v)$ and $v \neq \pi(u)$, then $\pi(v) >_v u, \pi^{-1}(v) >_v u$ or $\pi(u) >_u v, \pi^{-1}(u) >_u v$ hold as π is a stable partition. As u is matched to one of the vertices $\pi(u), \pi^{-1}(u)$ and v is matched to one of the vertices $\pi(v), \pi^{-1}(v)$, uv is not a blocking edge. Therefore we may assume that $\pi(u) = v$. However, as u and v are not in S and uv is not in M, v is matched to $\pi(v)$ in M. By $\pi(v) >_v u$, uv is not a blocking edge. □

It remains to show that the size of a removable set cannot be smaller than the number of odd cycles in a stable partition.

Lemma 3.17. *The minimum size of a removable set is equal to the number of odd cycles of length at least 3 in any stable partition π.*

Proof. Let S^* be a removable set, and let M^* be a stable matching in $G - S^*$. Since π is a stable partition, every odd cycle of length at least 3 in π must either have a vertex in S^* or a vertex u matched in M^* to a partner better than $\pi^{-1}(u)$. Indeed, if this does not hold for a cycle, then there is a vertex v in the cycle that is unmatched or matched to a worse partner than $v\pi(v)$ and $v\pi^{-1}(v)$. But then $\pi^{-1}(v)v$ blocks, since $\pi^{-1}(v)$ is not matched to anyone that is better than v by our assumption. We call a vertex $v \in V \setminus S^*$ *out-dominated* if it is matched in M^* to a vertex $u \neq \pi(v)$ with $u >_v \pi^{-1}(v)$. A vertex u is an *in-dominator* if it is the partner in M^* of an out-dominated vertex v. Note that u prefers $\pi(u)$ and $\pi^{-1}(u)$ to v because π is a stable partition.

Let $C = \{v_1, \ldots, v_t\}$ be an arbitrary cycle of length at least 2 in π such that $\pi(v_i) = v_{i+1}$, and let $v_{i_1}, \ldots, v_{i_z}$ be the in-dominator vertices in C. Observe that these vertices are pairwise non-adjacent on the cycle C as an adjacent pair would block M^*. Also, each of $v_{i_1-1}, \ldots, v_{i_z-1}$ must be either in S^* or out-dominated. Indeed, if for some $1 \leq j \leq z$ the vertex v_{i_j-1} is not in S^* and also not out-dominated, then the edge $v_{i_j-1}v_{i_j}$ blocks as v_{i_j} is an in-dominator.

Suppose now that t is odd. We claim that there is at least one vertex in $C \setminus \{v_{i_1-1}, \ldots, v_{i_z-1}\}$ that is either in S^* or is out-dominated. To see this, consider the paths obtained by removing the vertices $\{v_{i_1-1}, v_{i_1}, \ldots, v_{i_z-1}, v_{i_z}\}$ from C. Since t is odd, one of the paths contains an odd number of vertices; we may assume that this path is $v_1, \ldots, v_p$ for some odd p. If none of these vertices is deleted or out-dominated, then at least one them, say v_i, must be unmatched or matched in M^* to a vertex worse than $\pi^{-1}(v_i)$. If $i \neq 1$, then $v_{i-1}v_i$ blocks M^*, a contradiction. If $i = 1$, then $v_t = \pi^{-1}(v_1)$ is an in-dominator as v_1 is the first vertex of the path. This implies that v_tv_1 blocks M^*, a contradiction again. That is, at least one of the vertices $v_1, \ldots, v_p$ is deleted or out-dominated.

We conclude that the number of in-dominators in each odd cycle C is at most the sum of the numbers of out-dominated and deleted vertices in C minus one. If C is an even cycle, then the number of in-dominators is at most the sum of the numbers of out-dominated and deleted vertices. Finally, a singleton in π cannot be an in-dominator. It follows from the definitions that the number of in-dominators and the number of out-dominated vertices are the same. By combining these observations, we get that $|S^*|$ is at least the number of odd cycles in π. □

The theorem follows by Lemmas 3.16 and 3.17. □

3.3 Stable roommates with ties

In the previous section we have seen that the stable roommate problem can be solved in polynomial time. However, the algorithm of Irving only worked for the case when the preference lists did not contain any ties. Can we use the same algorithm to decide if there is a stable matching when ties are allowed? Here again, by stable we mean weakly stable, so there is no v_1v_2 edge, such that both of them strictly prefer each other to their partner.

SRT

Input: A complete graph G and $>_v$ weak preferences

Question: Is there a weakly stable matching M?

SRTI

Input: An arbitrary graph G and $>_v$ weak preferences

Question: Is there a weakly stable matching M?

In this case, even if we only want to find a stable matching and each preference list is complete, the problem becomes NP-hard, which was first shown by Ronn [30]. The trick that we used for the bipartite case, where we broke the ties arbitrarily does not work here, since there are exponentially many possible choices for that and some choice could lead to an instance that admits a stable matching but some not.

This problem is NP-complete even if each preference list is at most 3 long as shown by [8]. Here we give a different, simpler proof of their result. First we prove hardness for complete graph, but the same proof works for the short preference list case, simply by dropping all edges that are worse than the third for each agent.

Theorem 3.18. *SRT is NP-complete, even if each tie is of length 2 and are at the top of the preference lists.*

Proof. NP containment is trivial.

We reduce from the NP-complete problem (2,2)-E3-SAT, proven to be NP-complete by Berman et al. [3]. Here, we are given a CNF ϕ, such that each clause contains exactly three literals and each variable appears exactly twice in both negated and non negated form.

Let ϕ be an instance of (2,2)-E3-SAT, let $X_1, .., X_n$ be the variables and let $C_1, .., C_m$ be the clauses. We construct an SRT instance as follows:

For each clause C_j, we create 3 clause agents c_j^1, c_j^2 and c_j^3.

For each variable X_i we create 4 literal agents $x_i^1, x_i^2, \overline{x_i^1}, \overline{x_i^2}$.

Finally, for each i, we create two selector agents s_i^1 and s_i^2.

We define $f(c_j^k)$ to be x_i^l, if the k-th literal in C_j is the l-th appearance of X_i and to be $\overline{x_i^l}$, if the k-th literal in C_j is the l-th appearance of $\overline{X_i}$. Similarly, define $g(x_i^l)$ to be c_j^k, if the l-th occurrence of X_i is in the k-th place of clause C_j and $g(\overline{x_i^l})$ to be c_j^k, if the l-th occurrence of $\overline{X_i}$ is at the k-th place of C_j.

The preferences are the following:

$$
\begin{array}{ll}
c_j^1: & f(c_j^1) > c_j^2 > c_j^3 > ..., \; j = 1, ..., m \\
c_j^2: & f(c_j^2) > c_j^3 > c_j^1 > ... \; , \; j = 1, ..m \\
c_j^3: & f(c_j^3) > c_j^1 > c_j^2 > ..., \; j = 1, ...m \\
s_i^1: & (x_i^1, \; \overline{x_i^1}) > ..., \; i = 1, ..n \\
s_i^2: & (x_i^2, \; \overline{x_i^2}) > ..., \; i = 1, .., n \\
x_i^1: & s_i^1 > \overline{x_i^2} > g(x_i^1) > ..., \; i = 1, .., n \\
\overline{x_i^1}: & s_i^1 > x_i^2 > g(\overline{x_i^1}) > ..., \; i = 1, ..n \\
x_i^2: & s_i^2 > \overline{x_i^1} > g(x_i^2) > ..., \; i = 1, ..n \\
\overline{x_i^2}: & s_i^2 > x_i^1 > g(\overline{x_i^2}) > ..., \; i = 1, ..n,
\end{array}
$$

where $> \ldots$ means that the rest of the agents are ranked in (an almost) arbitrary order at the end. The only thing we care about this order is that for the c_j^k vertices the rest of the $c_{j'}^k$, $j' \neq j$ vertices are ranked in a way such that each $c_{j'}^k$ is better than each $c_{j''}^k$, if $j' < j''$.

Suppose there is a satisfying assignment to ϕ. Then, for each clause C_j there is at least one true literal. Create a matching M as follows: If X_i is true, than we add $\overline{x_i^1} s_i^1$ and $\overline{x_i^2} s_i^2$ to M and also $x_i^1 g(x_i^1)$, $x_i^2 g(x_i^2)$, otherwise we add $x_i^1 s_i^1$, $x_i^2 s_i^2$, $\overline{x_i^1} g(\overline{x_i^1})$ and $\overline{x_i^2} g(\overline{x_i^2})$.

Then, if there is a j, such that only one c_j^k is matched to a literal agent, then we match the other two. Finally, for those j, such that exactly two c_j^k-s are matched to literal agents, we match these clause agents among each other by ordering them with respect to j and then starting from the first, match each of them with the next one in the line.

Suppose there is a blocking pair to M. It cannot contain any agent that is matched with its first partner. Also it cannot contain any agent, such that their better partners are all with their first choices. Therefore, it could only contain agents of the type c_j^k, that are not matched to any of their first 3 choice. But among these, it is easy to check that M is a stable matching, since all their rankings are just the same as the order of their j indices.

Now suppose there is a stable matching M. Then, each s_i^l has to be matched to x_i^l or $\overline{x_i^l}$, otherwise they would block. Similarly, for each j, there has to be a c_j^k agent that is matched to a literal agent. Also, it cannot happen that x_i^1 and $\overline{x_i^2}$ or x_i^2 and $\overline{x_i^1}$ are both matched to a clause agent, since then they would block with each other.

So if we let the satisfying assignment be such that X_i is true, if and only if $\overline{x_i^1}$ or $\overline{x_i^2}$ is matched with s_i^1 or $s_i 2$, then this assignment is consistent, and each clause contains at least one true literal, so ϕ is satisfiable.

□

Theorem 3.19. *SRTI is NP-complete even if each preference list is at most 3 long and if it contains a tie, then it is of length 2 and there is no other entry in that list.*

3.4 The Stable Activities problem and the Stable b-matching problem.

In this section we consider some simple generalizations of the stable roommates problem, that can be reduced easily to the stable roommates problem, so they can be solved in polynomial-time.

The first such problem is the stable activities problem (SA), introduced by Cechlárová and Fleiner [5]. In SA, two agent can participate in several different activity and the rankings can depend on which activity they would spend with another agent. This can be represented as the underlying graph G having parallel edges, and the agents have preferences over these edges instead of the other agents. Here of course, we have to define blocking by an edge, not just an agent-pair.

SA

Input: A multigraph G and $>_v$ strict rankings on the adjacent edges
Question: Is there a stable matching M?

Now we show that such an instance can be reduced to a stable roommates instance efficiently.

Theorem 3.20. *(Cechlárová and Feliner)[5] Any instance* $(G,<)$ *of the stable activities problems can be reduced to an instance* $(G',<')$ *of the stable roommates problem, such that the stable solutions correspond to each other.*

Proof. The reduction used is the following. Substitute each edge $e = uv$ of G with a gadget G_{uv}. The vertices of G_{uv} will be $u_e^0, u_e^1, u_e^2, v_e^0, v_e^1, v_e^2$ and the edges are $uu_0^e, u_0^e u_1^e, u_1^e v_2^e, v_2^e v_0^e, v_0^e v_1^e, v_1^e u_2^e$ and $v_0^e v$. The preferences of the new vertices are:

$$\begin{array}{ll} u_0^e: & u_1^e > u > u_2^e \\ u_1^e: & v_2^e > u_0^e \\ u_2^e: & u_0^e > v_1^e \\ v_0^e: & v_1^e > v > v_2^e \\ v_1^e: & u_2^e > v_0^e \\ v_2^e: & v_0^e > u_1^e \end{array}$$

Also, we transform the rankings of the original agents $v \in G$ such that for each edge uv, we substitute e with v_0^e. Obviously, the new instance is a stable roommate instance, since there are no parallel edges.

Suppose M is a stable matching in $(G,<)$. Then, make M' by substituting each edge $e = uv$ with uu_0^e, vv_0^e, $u_1^e v_2^e$ and $u_2^e v_1^e$. For the edges uv not in M, if uv was dominated at u, we add the edges $u_1^e v_2^e, u_0^e u_2^e, v_0^e v_1^e$ otherwise the edges $u_0^e u_1^e, u_2^e v_1^e, v_0^e v_2^e$.

Assume that M' is not stable. Suppose the blocking edge is of type uu_0^e for some u. This means u prefers u_0^e to his/her partner in M'. But than, u_0^e was matched to his favourite partner, contradiction. Also, from any edge in the gadgets not in M', one of the endpoints have a strictly better partner, so those edges cannot block either.

Now suppose we have a stable matching M' in $(G',<)$. Then, if an edge uu_0^e is in M', then by the stability of M' $u_2^e v_1^e \in M'$, and also $u_1^e v_2^e, v_0^e v \in M'$, since otherwise one of $u_0^e u_2^e, u_2^e v_1^e$ or $v_0^e v_1^e$ would block M'. So $uu_0^e \in M'$ if and only if $vv_0^e \in M'$ and we can define M such that $e = uv \in M$ if and only if $\{uu_0^e, vv_0^e\} \subset M'$.

Suppose edge $e = uv$ blocks M. Then both u and v prefer e to their partner, so they prefer u_0^e and v_0^e to their partners in M' respectively. Since M' is stable, u_0 and v_0 must have their first partners in M'. But then, v_2^e must be unmatched, so $u_1^e v_2^e$ blocks M', contradiction.

□

Now we briefly talk about another generalization of the stable roommates problem, where the agents can have integer capacities. Let $G = (V, E)$ be a graph with $>_v$ preferences of the vertices. Furthermore, for each vertex v we are given a number $b(v)$ denoting the capacity of v.

Definition 3.21. An edge set M is called a *stable b-matching*, if for every $v \in V$: $|e \in M : v \in e| \leq b(v)$ and for each $f = uv \notin M$ edge it holds that at least one of $\{u, v\}$ are saturated (meaning $|e \in M : v \in e| = b(v)$) with strictly better agents.

The stable b-matching problem can easily be reduced to a simple stable roommates problem by making $b(v)$ copies of each vertex v and then all possible $b(u) \cdot b(v)$ copies of each $e = uv$ edge and finally setting all preferences of the copy vertices the same as the original, with the extension that the copies of the same vertex are ranked according to their indices. Since we can suppose without loss of generality that $b(v) \leq |E|$ for each $v \in V$, this reduction is polynomial and therefore the problem can be solved in polynomial-time.

There are more efficient algorithms of course for the stable b-matching problem, for example even Irving's algorithm can be generalized for it too, as described by Cechlárová and Fleiner in [5].

4 The Stable Hypergraph matching problem

A quite straightforward generalization of the standard stable matching problem is to consider hypergraph instead of graphs. This means that now we do not need to pair the agents, there can be coalitions of any sizes and the agents have preferences over these possible coalitions containing them. For example in the stable roommates problem , if the rooms are of size 3, then the possible allocations assign triplets of agents to rooms, so it can be seen as a stable matching problem in a 3-uniform hypergraph. Another example is the stable family problem, where the agents can be partitioned to three groups: the men, the women and the dogs. Each has preference lists over the possible pairs of the two different types of agents he or she finds acceptable. This problem is called the stable family problem or the 3-dimensional stable matching problem, which we will study in more detail in the next sections. Furthermore, we can have capacities on the vertices. Now we describe the stable hypergraph matching problem formally.

Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. For each $v \in V$ there is a capacity $k_v \in \mathbb{Z}$ and a strict preference list $>_v$ on the hyperedges containing v.

Definition 4.1. A fractional hyperedge matching $M : \mathcal{E} \to \mathbb{R}^+$ is called *feasible*, if $\sum_{e:v\in e} M(e) \leq k_v$ for all $v \in V$ and $M(e) \leq 1$ for all $e \in \mathcal{E}$.

Definition 4.2. Given a fractional matching M, a hyperedge f is called *blocking*, if $M(f) < k_f$ and for each $v \in f$, either v is unsaturated by M or there is an $f_v \in \mathcal{E}$, such that $v \in f_v$, $M(f_v) > 0$ and $f_v <_v f$.

Definition 4.3. A fractional matching M is called *stable*, if there are no blocking hyperedges.

FRACTIONAL HYPERGRAPH MATCHING
Input: A hypergraph $\mathcal{H} = (V, \mathcal{E})$, strict orderings $>_v$ and k_v capacities.
Output: A stable fractional matching M.

If the hypergraph is also 3-partite and 3-uniform, and each capacity is 1, then the instance corresponds to a stable family instance. So we define F-SFP as follows.

F-SFP
Input: A 3-partite, 3-uniform hypergraph $\mathcal{H} = (V, \mathcal{E})$ and strict orderings $>_v$.
Output: A stable fractional matching M.

The class PPAD

The class PPAD, introduced by Papadimitriou [28] is a complexity class consisting of search problems, where the existence of a solution is guaranteed. Generally, PPAD-hard problems are considered as hard problems, so efficient algorithms for them are unlikely to exist.

When considering polynomial-time reducibility among (search) problems in PPAD, we always mean two polynomial-time computable functions f and g, such that f maps a given instance I_A of problem A to an instance I_B of problem B and if y is a solution for $f(I_A)$ then $g(y, I_A)$ is a solution for I_A.

4.1 Scarf's Lemma and its connection with stable mathings

In this section we study a fundamental result proven by Scarf. It is a Lemma that plays a central role in the study of many variants of stable matchings, and can be used for countless existential proofs and complexity theoretic reductions. Although the Lemma originally was stated for so called non-transferable utility games to prove the existence of the fractional core (a game theoretic concept representing fairness), it can be stated in many equivalent forms. We will mainly need the following one:

Lemma 4.4. *[33] Let Q be an $n \times m$ nonnegative matrix, such that every column of Q has a nonzero element and let $q \in \mathbf{R}^{\mathbf{n}}_{+}$. Suppose every row i has an strict ordering $>_i$ on those columns j for which $Q_{ij} > 0$. Then there is an extreme point of $\{Qx \leq q,\ x \geq 0\}$, that dominates every column in some row, where we say that $x \geq 0$ dominates column j in row i, if $Q_i x = q_i$ and $j \leq_i k$ for all $k \in \{1, ..., m\}$, such that $Q_{ik} x_k > 0$.*

Another great feature of this Lemma is that its proof is algorithmic, so not only can we use this to prove the existence of fractional solutions in several versions of the stable matching problem, we can algorithmically find one too. Also, the algorithm is guaranteed to be finite, although its running time can be exponential, which is not surprising, since it is used to solve PPAD-hard problems.

Because of the remarkable importance of this theorem, we present here the original version too, with a slightly simplified proof by Aharoni and Holzman [1].

Lemma 4.5. *(Scarf [33]) Let $m < n \in \mathbb{N}$ and let B be an $m\times n$ real matrix that satisfies that the first m columns form an identity matrix. Let C be another $m\times n$ matrix such that $c_{ii} \le c_{ik} \le c_{ij}$ for all $i, j \le m$, $i \ne j$, $k > m$ and finally let b be a real m-dimensional vector such that $\{x : Bx = b\}$ is a bounded polyhedron. Then, there exists a subset $\mathcal{J} \subset [n]$ of size m for which the following hold:*

1. *$\exists x \in \mathbb{R}^n$: $Bx = b$ and $x_j = 0$ if $j \notin \mathcal{J}$*

2. *For every $1 \le k \le n$ $\exists i \in [m]$ such that $c_{ik} \le c_{ij}$ for all $j \in \mathcal{J}$.*

Proof. To simplify the proof we first introduce some definitions.

Definition 4.6. A column c^k is *$\mathcal{J}$-subordinated at i*, if $c_{ik} \le c_{ij}$ $\forall j \in \mathcal{J}$. c^k is *$\mathcal{J}$-subordinated* if there is an index i, where is $\mathcal{J}$-subordinated. $\mathcal{J}$ is *subordinating for C*, if every column of C is $\mathcal{J}$-subordinated.

Definition 4.7. $\mathcal{J} \subset [n]$ is a *feasible basis* for (B, b), if $|\mathcal{J}| = m$ and if the columns of B corresponding to the indices in $\mathcal{J}$ are linearly independent, then b belongs to the cone spanned by those columns.

Definition 4.8. We say that the pair (B, b) is *non-degenerate* if b is not in the cone spanned by fewer than m columns of B.

Definition 4.9. C is *ordinal-generic* if all the elements in each row of C are distinct.

It is easy to observe that subsets of subordinating sets are subordinating. Also, we can perturb b a little such that (B, b) becomes non-degenerate and every feasible basis of the perturbed instance is a feasible basis of the original too. Similarly we can perturb C a little such that the subordinating sets doesn't change and the conditions of the theorem still holds, so from now on we will assume that C is ordinal generic and (B, b) is non-degenerate. Notice that if we find a set $\mathcal{J}$ that is both subordinating and a feasible basis, than we are done. We will use the following, well-known lemma:

Lemma 4.10. *If $\mathcal{J}$ is a feasible basis and $k \notin \mathcal{J}$, then there is a unique index $j \in \mathcal{J}$ such that $\mathcal{J} - j + k$ is a feasible basis too.*

To prove the claims of the theorem, first we prove a technical lemma for it.

Lemma 4.11. *If $\mathcal{K}$ is a subordinating set of size $m - 1$, then*

1. *if $K \subset [m]$ (the first m columns), then there is a unique j, such that $\mathcal{K} + j$ is subordinating and*

2. *otherwise there are two such index j.*

Proof. Introduce a function f, such that $f(i)$ is the (unique) index in $k \in \mathcal{K}$ such that c_{ik} is minimal. Furthermore, because every column c^k is subordinated by $\mathcal{K}$ for some i, at row i c_{ik} is minimal among all c_{ij}, $j \in \mathcal{K}$, so f is surjective. Now, since $\mathcal{K}$ has size $m-1$, there is only one $h \in \mathcal{K}$ for which there are two $i_1, i_2 \in [m]$ such that $f(i_1) = f(i_2) = h$ and for every other there are precisely one. One can show similarly that if $\mathcal{K}+j$ is subordinating, then every column c^k $k \in \mathcal{K}+j$ there is an i, such that at row i, c_{ik} is minimal among all c_{il}, $l \in \mathcal{K}+j$. That means that there is an $a \in \{1,2\}$ such that $c_{i_a l} < c_{i_a k}$ for every l such that c^l is not $\mathcal{K}$-subordinated at any $i \neq i_a$. So let S_a denote the set of those $l \notin \mathcal{K}$ that are not subordinated at any $i \neq i_a$.

Observation: $\mathcal{K}+j$ is subordinating if and only if there is an $a \in \{1,2\}$ such that $j \in S_a$ and $c_{i_a j} \geq c_{i_a l}$, $l \in S_a$.

Suppose that $\mathcal{K} \subset [m]$. Then, since the first m columns of B form an identity matrix, $f(i) = i$ for all $i \in \mathcal{K}$, so precisely one of i_1 and i_2 belong to $\mathcal{K}$. Suppose it is i_1. Then $S_1 = \emptyset$, because the columns not in $\mathcal{K}$ are subordinated at i_2 and $S_2 = [n] \setminus \mathcal{K} \neq \emptyset$, which proves the first case, since there can only be one j such that $\mathcal{K}+j$ is subordinating by the above observation.

Otherwise $\mathcal{K} \setminus [m] \neq \emptyset$. Then neither i_1 nor i_2 belongs to $\mathcal{K}$, because otherwise $f(i_1) = f(i_2) = i_1$ (or i_2) but for $j \in \mathcal{K} \setminus [m]$, $c_{i_2 j} < c_{i_2 i_1}$ (by the conditions on the matrix C), contradiction. Hence, both S_1 and S_2 will be nonempty because $c_{i i_a} > c_{ij}$ for $i \neq i_a$ meaning c^{i_a} is not $\mathcal{K}$-subordinated at any $i \neq i_a$, so $i_a \in S_a$. Again, by the observation above, we are done.

□

Now finally we can prove Scarf's lemma. Make a bipartite graph $G = (A, B, E)$, where A is the set of the feasible bases containing 1 and B is the set of subordinating sets of size m not containing 1 and there is an edge between $F \in A$ and $S \in B$ if and only if $F \setminus S = \{1\}$. If $F \in A$ is subordinating, then we found a right basis. Otherwise suppose that F is not and that F is not isolated. By our Lemma, F can have degree one or two, and it can only be 1 if $F \subset [m]$, so $F = [m]$. Similarly if a set $S \in B$ is not a feasible basis and have positive degree, then by the definition of the edges $|S \setminus F| = 1$ for each neighbour of S. Let F be a neighbour and let the element in $S \setminus F$ be s. By lemma 4.10 there is a unique $f \in F$ such that $F' = F - f + s$ is a feasible basis. $f \neq 1$, since then $F = S$ would be a feasible subordinating basis. This means that S has degree two and its two neighbours are F and F'.

So we can conclude that every vertex of the graph that does not represent a feasible and subordinating basis has degree 0 or 2, and $F = [m]$ has degree 1. So by parity arguments, there is an other vertex X with degree 1, which can only happen if X is a feasible subordinating basis, so the proof is complete. □

Remark 4.12. Although the above proof does not explicitly describe the algorithm used for finding a solution, it basically just travels the path starting from $[m]$ in the graph G until it reaches a feasible subordinating basis.

SCARF
Input: A nonnegative matrix Q, a nonnegative vector q and strict orderings $>_i$ for each row.
Output: A dominating extreme point of $\{Qx \leq q,\ x \geq 0\}$.

Scarf's lemma can be used to prove the existence of stable fractional solutions in many problems. For example, to prove that there always exists a stable fractional solution in a Stable Hypergraph matching instance, we only have to reduce the problem to SCARF. But the reduction is quite simple: we take Q to be the incidence matrix of the hypergraph with the rows being the vertices and use the same rankings. The bounding vector consists of the capacities of the vertices. It is straightforward to check that a dominating solution will correspond to a stable fractional solution. Also, if the matrix is TU, then Scarf's lemma also guarantees the existence of an integral stable solution.

Theorem 4.13. *In any hypergraphic preference system, there always exists a stable fractional matching.*

4.2 Hardness results

In this section, building on the lemma of Scarf we will prove several hardness results about variants of the stable hypergraph matching problem and its fractional version FRACTIONAL HYPERGRAPH MATCHING. By theorem 5.6 we get the following result:

Theorem 4.14. *Deciding if there exists an integer stable hypergraph matching is NP-hard, even if each capacity is one and each vertex has degree at most two.*

It is also easy to see NP-containment, since the stability of a given matching can be checked in polynomial time, by going through the hyperedges and checking whether they block.

Now we turn our attention to the fractional case. Let FRACTIONAL HYPERGRAPH MATCHING denote the problem of finding a stable fractional matching in a hypergraphic preference system. The first ones to show the hardness of FRACTIONAL HYPERGRAPH MATCHING were Kintali et al. [17]. They also proved the PPAD-hardness of SCARF. (Note that FRACTIONAL HYPERGRAPH MATCHING can be considered a special case of SCARF).

Theorem 4.15. *(Kintali et al.) FRACTIONAL HYPERGRAPH MATCHING with unit capacities and SCARF are PPAD-complete.*

The proofs follow a long chain of reductions and are quite technical, so we omit them here. Instead, we will prove that even more restricted versions of the FRACTIONAL HYPERGRAPH MATCHING problem are PPAD-complete. The first such restriction we consider is the degree of the vertices. We have seen that the integral version of the problem is hard even if each degree is 2. Here we show a similar result of Ishizuka and Kamiyama [16], who proved the fractional problem is hard even if the degrees are at most three.

Theorem 4.16. *(Ishizuka and Kamiyama)* FRACTIONAL HYPERGRAPH MATCHING *is PPAD-complete, even if $d(v) \leq 3$ for each $v \in V$.*

Proof. We will reduce from the unit capacity general version of the FRACTIONAL HYPERGRAPH MATCHING problem. Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph and let $>_v$ denote the preferences of the vertices. Make a new hypergraph $\mathcal{H}' = (V', \mathcal{E}')$ the following way. The vertex set of $\mathcal{H}'$ will be $V' = \{v_i : v \in V, i = 1, ..., d(v)\} \cup \{v'_i : v \in V, i = 1, ..., d(v) - 1\}$. For each $e \in \mathcal{E}$ we make a hyperedge $e' = \{v_{r(v,e)} : v \in e\}$, where $r(v, e)$ is the rank of e in v's preference list, so it is 1, if e is v's best choice, 2 if it is v's second, ... and $d(v)$ if it is v's worst. So for a hyperedge e we make a hyperedge e' consisting of the $r(v, e)$-th copies of the v vertices in e. Let $E' = \{e' : e \in \mathcal{E}\}$. Then we define the edge set of $\mathcal{H}'$ as $\mathcal{E}' = E' \cup \{\{v_i, v'_i\}, \{v'_i, v_{i+1}\} : v \in V, i = 1, ..., d(v) - 1\}$.

Denote h^v_i the hyperedge in E' containing v_i. The rankings of the new instance will be the following.

$$
\begin{aligned}
v_1 :&\quad h^v_1 >_{v_1} \{v_1, v'_1\} \text{ for every } v \in V \\
v_i :&\quad \{v'_{i-1}, v_i\} >_{v_i} h^v_i >_{v_i} \{v_i, v'_i\} \text{ for every } v \in V \text{ and } i = 2, ..., d(v) - 1 \\
v_{d(v)} :&\quad \{v'_{d(v)-1}, v_{d(v)}\} >_{v_{d(v)}} h^v_{d(v)} \text{ for every } v \in V \\
v'_i :&\quad \{v_i, v'_i\} >_{v'_i} \{v'_i, v_{i+1}\} \text{ for each } v \in V,\ i = 1, ..., d(v) - 1
\end{aligned}
$$

Obviously, this hypergraph satisfies that each vertex has degree at most 3. Now we show that if we find a fractional stable hypergraph matching M' in this instance, then by taking $M(e) = M'(e')$ we get a feasible stable fractional matching in the original. For this we need two lemmas.

Lemma 4.17. *For any $v \in V$ and any $i = 1, ..., d(v)-1$ we have that $\sum_{j=1}^{i} M'(h^v_j) + M'(\{v_i, v'_i\}) = 1$*

Proof. We prove this by induction. Suppose $i = 1$. Then, since M' is a feasible matching, $M'(h^v_1) + M'(\{v_1, v'_1\}) \leq 1$. Suppose it is strictly less than 1. Then, v_1 is unsaturated in M' which would mean that $\{v_1, v'_1\}$ blocks, contradiction.

Now suppose we know the statement for $i < k$. By induction $\sum_{j=1}^{k} M'(h^v_j) = 1 - M'(\{v_{k-1}, v'_{k-1}\}) + M'(h^v_k)$, so the statement is equivalent to $M'(h^v_k) + +M'(\{v_k, v'_k\}) = M'(\{v_{k-1}, v'_{k-1}\})$.

Now, if $M'(\{v_k, v'_k\}) + M'(h^v_k) < M'(\{v_{k-1}, v'_{k-1}\}) \leq 1 - M'(\{v'_{k-1}, v_k\})$ (because M' is feasible), we have that v_k is unsaturated in M', so $\{v_k, v'_k\}$ blocks M', contradiction.

On the other hand, if $M'(\{v_{k-1}, v'_{k-1}\}) < M'(\{v_k, v'_k\}) + M'(h^v_k) \leq 1 - M'(\{v_k, v'_{k-1}\})$, then v'_{k-1} is unsaturated, so $\{v_k, v'_{k-1}\}$ blocks.

So it must hold that $M'(\{v_{k-1}, v'_{k-1}\}) = M'(\{v_k, v'_k\}) + M'(h^v_k)$ ☐

Lemma 4.18. *For any $v \in V$ and $i = 1, .., d(v)-1$ it holds that $M'(\{v'_i, v_{i+1}\} = \sum_{j=1}^{i} M'(h^v_j)$.*

Proof. By our previous Lemma $\sum_{j=1}^{i} M'(h^v_j) = 1 - M'(\{v_i, v'_i\})$, so it is enough to show that $M'(\{v'_i, v_{i+1}\}) + M'(\{v_i, v'_i\}) = 1$. It must be ≤ 1, since M' is feasible. If it would be strictly less than one, then v'_i would be unsaturated, so $\{v'_i, v_{i+1}\}$ would block, contradiction.

□

Now let $M(e) = M'(e')$. For every $v \in V$, we have that $\sum_{e:v\in e} M(e) = \sum_{j=1}^{d(v)} M'(h_j^v) = 1 - M'(\{v_{d(v)-1}, v'_{d(v)-1}\}) + M'(h_{d(v)}^v) = M'(\{v'_{d(v)-1}, v_{d(v)}\}) + M'(h_{d(v)}^v) \leq 1$, where we used the two lemmas and the feasibility of M'. So M is a feasible matching.

Suppose there is a blocking hyperedge f to M. Since M' was stable, f' does not block M'. This means that there is a $v \in f$ and $i \in \{1, ..., d(v)\}$ such that f' is dominated at v_i. This means that $M'(\{v_i, v'_{i-1}\}) = 1 - M(f')$. But by our lemmas, $M'(\{v_i, v'_{i-1}\}) = \sum_{j=1}^{i-1} M'(h_j^v)$, so f is dominated at v in M too, contradiction. □

Ishizuka and Kamiyama [16] also showed that the problem becomes polynomial-time solvable if the degrees of the vertices can be at most two. Their algorithm was the following: Let the hypergraph be $\mathcal{H} = (V, \mathcal{E})$. Make the edge graph of $\mathcal{H}$, that is the graph H with vertex set $U = \mathcal{E}$ and there is an edge between two vertices of the graph if the corresponding two hyperedges intersect. Now, make a superorientation of the edge graph. A superorientation is an orientation, where we are allowed to orient some edges both ways. If all common vertices of two hyperedge e and f prefer e to f, then orient the edge between e and f towards f. If all of them prefer f to e, then we orient is towards e. Otherwise we orient the edge ef in both directions.

Now, we iteratively check whether there are vertices in U such that they have no incoming edges (which can be either one-way or two-way edges). If there is a vertex $u \in U$, then we add the hyperedge corresponding to u in our matching with full weight and remove the vertex and its neighbours from H.

If there are no such vertices left in the edge graph, then we put the hyperedges corresponding to all of the remaining vertices to M with half weight.

Lemma 4.19. *This algorithm computes a stable fractional hypergraph matching.*

Proof. In the algorithm, if we include a hyperedge e containing a vertex v with weight one, then no other hyperedge containing v is added later, since we delete them from H (because they intersected e, so they were neighbours). Also, each degree is at most two, so adding both hyperedges containing v with half weight does not violate the capacity constraints either, so M is feasible.

Suppose there is a blocking edge f to M. If $M(f) = 0$, then f was deleted, so we added a hyperedge e, such that there was a vertex $v \in f \cap e$ who preferred e and $M(e) = 1$, contradiction. If $M(f) = \frac{1}{2}$, then since f had an incoming edge form a vertex, there is an edge e with $M(e) = \frac{1}{2}$, such that there is a vertex $v \in e \cap f$ preferring e, so v is saturated and f cannot block, contradiction. □

Now, we show that the problem remains hard even if we restrict both the degree and the size of the hyperedges to 3, which is a new result that I have proven in [7]. We need another theorem from the same paper, that we will only prove later in Section 7.

Theorem 4.20. *[7] The* F-SFP *problem is PPAD-hard.*

Theorem 4.21. *[7]* FRACTIONAL HYPERGRAPH MATCHING *is PPAD-complete even if each hyperedge has size at most 3, and each vertex has degree at most 3, each capacity is 1 and the hypergraph is polynomial-time 3 edge-colorable.*

Proof. For this we will use that since F-SFP is PPAD-complete, it is enough to apply the reduction used by Ishizuka and Kamiyama [16] to those hypergraphs that correspond to an SFP instance, because we can reduce any hypergraph to a stable family instance as a first step.

Then, we make the same construction as Ishizuka and Kamiyama. So let $V' = \{v_i : v \in V, i = 1, ..., d(v)\} \cup \{v'_i : v \in V, i = 1, ..., d(v) - 1\}$ and $\mathcal{E}' = E' \cup \{\{v_i, v'_i\}, \{v'_i, v_{i+1}\} : v \in V, i = 1, ..., d(v) - 1\}$ again.

Now, what is important to see is that because each $e \in \mathcal{E}$ had at most 3 vertices, so does E' and so does $\mathcal{E}'$, therefore both the degree and the hyperedge size constraints are satisfied in the new instance. It only remains to prove that it can be edge-colored by 3 colors in polynomial time. For this, we will use the following edge coloring: color each hyperedges in E' red, each hyperedge of the form $\{v_i, v'_i\}$ green, and each $\{v'_i, v_{i+1}\}$ yellow. Since for every vertex v only one hyperedge can be it is i-th in the ranking, we get that each vertex can be only adjacent to at most one red edge. It is easy to observe that each vertex can only be in one $\{v_i, v'_i\}$ edge and one $\{v'_i, v_{i+1}\}$ edge too, so the coloring is good indeed, and it can be done in linear time. □

4.3 Tractable cases of the stable hypergraph matching problem

In this section we show how the stable hypergraph matching, that is the integer version of FRACTIONAL HYPERGRAPH MATCHING relates to finding kernels in superorientations of graphs, as well as give some efficient algorithms for special cases of the problem.

Definition 4.22. A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is a *subtree-hypergraph*, if there is a tree $T = (V, E)$ on the ground set V, such that each hyperedge is a subtree of T.

Definition 4.23. A *superorientation* of a graph G is an orientation of G where each edge can be either simply oriented from one endpoint to the other, or in both ways.

Definition 4.24. A superorientation of G is *clique-acyclic*, if there are no directed cycles that contain only one-way edges and all of its vertices are part of a clique K in G.

Definition 4.25. A *kernel* K of a superoriented graph G is a set of points that is independent and absorbing, so from every vertex v not in K there is a directed edge or a two-way edge to some point in K from v.

Definition 4.26. A graph G is *chordal*, if any cycle of length ≥ 4 has a chord.

Theorem 4.27. *[7] If $\mathcal{H} = (V, \mathcal{E})$ is a subtree hypergraph, and each vertex capacity is 1, then there always exists a stable integral hypergraph matching and it can be found in polynomial time.*

Proof. We reduce the problem to finding a kernel in a clique-acyclic superorientation of a chordal graph that can be solved efficiently as proven by Pass-Lanneau et al. [29]. We simply consider the edge graph of H, that is, for each hyperedge F, we have a point v_F, and two vertices are connected if and only if the corresponding hyperedges intersect. Since H is a subtree hypergraph, the edge graph is chordal. Now orient the edges the following way: if $F_i \cap F_j \neq \emptyset$ and each vertex in $F_i \cap F_j$ prefers F_j to F_i, then we orient the edge $v_{F_i}v_{F_j}$ towards v_{F_j}. If there are vertices preferring F_i over F_j as well, then we orient the edge $v_{F_i}v_{F_j}$ in both directions. This results in a superorientation that is clique-acyclic, since the hyperedges of H have the the Helly property, so if any two of some hyperedges intersect, then there is a point contained in all of them. So any hyperedges that make a clique in the edge graph have a point in common, so there cannot be a directed cycle (containing only one-way edges) in it.

Let us suppose we have found a kernel K. Then, the corresponding edges form a stable matching: The points in K are independent, so the hyperedges do not intersect, hence the capacity constraints are not violated. And if there is a blocking hyperedge F, then it blocks each hyperedge in K (that it intersects) on their common vertices, so every edge between v_F and K points toward K, contradiction. □

A natural question to ask is that can it still be solved efficiently if we allow the vertices to have capacities greater than one. Sadly, the answer is no, even with very severe restrictions.

Theorem 4.28. *[7] Deciding if there exists an integer stable hypergraph matching is NP-complete, even if H is a subtree hypergraph, that tree is a star, only one vertex has capacity greater than 1, and all hyperedge sizes are at most 4.*

Proof. We will reduce from the stable family problem. Let the vertices of H be the vertices of the stable family problem and one additional vertex x. Extend each original hyperedge with x. The capacities will be the same, and let the capacity of x be the number of hyperedges. It is easy to check that the stable matchings of this instance will correspond to the stable matchings in the family problem an vice versa. □

Remark 4.29. Using the same construction, and the fact that the f-SFP is PPAD-complete, it is easy to see that the problem of finding a stable fractional solution in an instance of Theorem 4.28 is PPAD-complete too.

Finally, we construct a polynomial-time algorithm that finds an integral stable matching for any capacities, if the hypergraph $\mathcal{H} = (V, \mathcal{E})$ is laminar, that is, from any two intersecting hyperedges, one contains the other. The algorithm is the following:

Algorithm 5

```
S := ∅                                          ▷ S will be the actual matching
C := ∅                                    ▷ C will denote the checked hyperedges
while E \ C ≠ ∅ do
    Pick an inclusion-wise minimal hyperedge F in E \ C.
    if F is not blocking with respect to S then
        C := C ∪ {F}
        S := S
    else if  F is blocking with respect to S then
        C := C ∪ {F}
        S := S ∪ {F}
        If there were vertices in F that were saturated, then for each such
    vertex take its worst hyperedge.  Among these, remove the inclusion-wise
    maximal ones from S.
    end if
end while
```

Lemma 4.30. *The output $\mathcal{S}$ by the algorithm is a stable feasible hypergraph matching. Also, the running time is polynomial.*

Proof. Since every time a vertex becomes oversaturated we delete one edge containing it, the capacity constraints will be satisfied in the end, thus $\mathcal{S}$ is a feasible matching. Let us suppose that there is a blocking hyperedge F.

If F was not included in $\mathcal{S}$ when it was checked by the algorithm, then some vertex v of F had already been saturated with hyperedges better than F. First notice that, since we always pick inclusion-wise minimal hyperedges, every hyperedge that has already been checked at some point in the algorithm is either contained in the current hyperedge or is disjoint from it. So if a hyperedge F' forces v to throw out one of its previous hyperedges, then F' contains all of them and F' has to be better than one of them, so it is still better for v than F. Also, since we only throw out the inclusion-wise maximal worst hyperedges, that are all subsets of F', each vertex only loses at most one hyperedge, and if it loses one, then it gains one with F', so v remains saturated. Therefore, v will still be saturated with better hyperedges than F at the end of the algorithm, so F cannot block, contradiction.

If F was in $\mathcal{S}$, but got thrown out, then it had to be because it was the worst hyperedge of a saturated vertex v. So in that step the vertex v remained saturated, with strictly better hyperedges than F. Using the same argument as before, v will still dominate F at the end of the algorithm, contradiction.

In every step, a hyperedge that was not marked becomes marked, so there are $|\mathcal{E}|$ steps. A step consists of finding an inclusion-wise minimal unmarked hyperedge, checking whether it is blocking, and if it is, then computing the inclusion-wise maximal worst hyperedges of the saturated vertices. All of these can be done in polynomial time, so the running time is polynomial. □

5 The Hospital-Resident problem with couples

One of the first real life application of stable matching algorithms was the American National Resident Matching Program, NRMP for short. Actually, they used an algorithm similar to the Gale-Shapley algorithm even before the mathematical foundations of the field in 1962. In the NRMP the task was to assign residents to hospitals. Both the residents and the hospitals can submit preferences on each other into the system and like in the case of the stable marriage problem, we want to find an allocation of the residents such that there is no resident hospital pair (d, h) such that the resident would prefer to leave his or her current allocation and go to h and h would accept d, possibly by sending away a worse resident. Similarly to the stable marriage problem, these kind of pairs could start to launch a series of deviations that would destabilize the market. It is not hard to see, that the Hospital-Resident problem (HR) can be reduced to a simple stable marriage problem, we only need to make multiple copies of the hospitals each with unit capacity, that all have the same rankings as the original and extend the preferences of the residents over the hospitals by ranking the copies among each other arbitrarily.

Another problem similar to HR is the university admission problem, where there are universities instead of hospitals and students instead of residents. Also the ranking of the universities can include ties if two students have the same score, which makes the problem a little harder, but it is still possible to find stable solutions satisfying certain fair requirements.

Before we proceed to the case involving couples, we investigate the original problem a little deeper. First of all, we define the problem formally: let $\mathcal{H}$ be the set of hospitals and let $\mathcal{D}$ be the set of doctors/residents. Each hospital $h \in H$ has a strict preference list $>_h$ on the doctors and each doctor $d \in \mathcal{D}$ has a $>_d$ preference list over the hospitals. The hospitals also have $k_h \in \mathbb{Z}$ capacities.

Definition 5.1. Let M be a matching, so for each resident d, $\sum_{h\in\mathcal{H}} M(d,h) \leq 1$ and for each hospital h, $\sum_{d\in\mathcal{D}} M(d,h) \leq k_h$. A pair (d,h) *blocks* M, if $h >_d p_M(d)$ and either $\sum_{d\in\mathcal{D}} M(d,h) < k_h$ or there is a resident d' at h, such that $d >_h d'$. We say that M is *stable*, if there is no blocking resident-hospital pair.

We show that (not surprisingly) just like the stable matching problem, this problem also has nice structural properties. First we prove the so called Rural Hospital theorem.

Lemma 5.2. *Let M be the resident-optimal stable matching (it exists by the reduction to the stable marriage problem by making multiple copies of hospitals) and let M' be another stable matching. Suppose hospital h does not reach its capacity in M'. Then, every resident that is assigned to h in M is also assigned to h in M'.*

Proof. Suppose d is not at h in M' but at h in M. Then d prefers h to $p_{M'}(d)$ so (d,h) blocks M. □

Theorem 5.3. *(Rural Hospital theorem) In each Hospital-resident problem instance the following hold:*

1. *Each hospital has the same number of residents in any stable matching,*
2. *The same residents are unassigned in every stable matching*
3. *If a hospital is under its quota in any stable matching then it gets the same set of residents in every stable matching.*

Proof. 1 and 2 follow directly from the fact that we can reduce a HR instance to a stable matching instance by making k_h copies of h that rank the residents the same way, and the residents rank the copies among each other by their indices. Since in any stable matching the same set of vertices is matched, this proves 1. and 2. To see 3, we only need to use the previous lemma to see that undersubscribed hospitals must have exactly the same set of agents in every stable matching. □

Also, because it can be reduced to the stable matching problem, it is still true that if each resident picks their best choice hospital among their partner hospitals in two different stable matching, then this also results in a stable matching. And similarly, if each hospital picks the best (at most k_h) residents among their assigned residents in two different stable matchings, it also leads to a stable matching. So in this problem the stable matchings form a distributive lattice too.

Furthermore, again because of its simple reduction to the stable matching problem, we can find minimal/maximal weight stable matchings in this problem too.

5.1 Hardness results

Now we move on to the instance where couples are allowed too, called Hospital-Resident-Couple problem, HRC for short. Surprisingly, the problem becomes NP-hard even with this small additional condition, as we will soon show. In this setting there are three sets, the hospitals $\mathcal{H}$, the single residents $\mathcal{D}$ and the couples $\mathcal{C}$. The set of couples can be partitioned into two sets: the set of men $\mathcal{C}^m$ and the set of women $\mathcal{C}^w$. Each hospital $h \in \mathcal{H}$ has a capacity $k_h \in \mathbb{N}$ and a strict preference $>_h$ on the residents, that is the single residents and the members of couples. Similarly each single resident $d \in \mathcal{D}$ has a strict preference $>_d$ on the hospitals, and each couple $c \in \mathcal{C}$ has a strict preference list $>_c$ on the pairs of hospitals. These preference lists need not be complete; if a hospital/resident does not appear then we interpret it as being unacceptable. If the lengths of the preference lists are bounded by some constant, then we call it (α, β, γ)-HRC, where each single resident's preference list has length at most α, each couple's preference list has length at most β and each hospital's preference list has length at most γ. If there are no single residents, then α is omitted.

The blocking coalitions in this setting are defined the following way:

Definition 5.4. Let M be a matching in the HRC.

1. A single resident-hospital pair (d, h) *blocks*, if d prefers h to its current allocation in M (including the possibility that d is unassigned) and h is either under its quota or has a worse resident in M than d,

2. A coalition (c, h, h'), $h \neq h'$, $c = (c^w, c^m)$ *blocks*, if c prefers (h, h') to $M(c)$, h is unsaturated or has an at least as bad resident as c^w (since c^w may already be at h, if $h = M(c^w)$) and h' is unsaturated or has an at least as bad resident as c^m,

3. A coalition (c, h, h) *blocks* if $(h, h) >_c M(c)$ and both c_f and c_m are in the best k_h residents for h in the set $M(h) \cup \{c^w\} \cup \{c^m\}$.

Definition 5.5. A matching M is *stable* if it is feasible (so no hospital exceeds its quota and no resident has more than one position) and there are no coalitions that block M.

HRC
Input: A set of hospitals $\mathcal{H}$ with $>_h$ strict preferences over $\mathcal{D} \cup \mathcal{C}^m \cup \mathcal{C}^w$ and $k_h \in \mathbb{N}$ capacities, a set of single residents $\mathcal{D}$, with $>_d$, $d \in \mathcal{D}$ strict preferences over $\mathcal{H}$ and a set of couples $\mathcal{C}$ with $>_c$, $c \in \mathcal{C}$ strict preferences over $\mathcal{H} \times \mathcal{H}$.
Question: Is there a stable matching M?

Now we prove that even with the severe restrictions that there are no single residents, each capacity is 1 and the preference lists of all couples and hospitals are at most 2 long the problem is still NP-hard.

Theorem 5.6. *(Biró et al. [4]) It is NP-complete to decide if a given* $(2, 2)$*-HRC instance admits a stable matching or not, even if all capacities are 1.*

Proof. To show containment in NP we only need to see that we can check the stability of a given matching M by going through all the possible coalitions (which have size 3) and check if they are blocking, which can be done in polynomial time.

To prove NP-hardness we will reduce from the NP-complete problem (2,2)-E3-SAT. (The NP-completeness of this problem were shown by Berman et al.[3]). In this problem we have to decide whether a boolean formula φ in CNF has a satisfying assignment, where φ contains exactly 3 literals in every clause and every literal appears exactly twice in φ. Let $X = \{x_1, .., x_n\}$ be the set of variables and $C = \{C_1, ..., C_m\}$ be the set of clauses.

Then, we reduce it to an instance of the HRC as follows. The couples of the HRC instance will be $A \cup B$, where $A = \{(a_i^r, b_i^r) : \ r = 1, 2, \ i = 1, ..., n\}$ and $B = \{(c_j^s, d_j^s) : \ s = 1, 2, 3, \ j = 1, .., m\}$. The set of hospitals will be $H \cup T$, where $H = \{h_i^r : \ r = 1, .., 6, \ i = 1, ..., n\}$, $T = \{t_j^r : \ r = 1, .., 6, \ j = 1, .., m\}$.

Define $h(c_j^s)$ to be h_i^{2r+1} if the r-th appearance of x_i is at position s at C_j, and h_i^{2r+2}, if the r-th appearance of $\overline{x_i}$ is at position s at C_j. Similarly let $c(h_i^{2r+1})$ be c_j^s if the r-th occurrence of literal x_i is at the s-th position of C_j

and $c(h_i^{2r+2})$ be c_j^s if the r-th occurrence of literal $\overline{x_i}$ is at the s-th position of C_j. So the possible applications, with the order of the preferences are the following:

$$(a_i^1, b_i^1) : (h_i^1, h_i^3) > (h_i^2, h_i^4)$$
$$(a_i^2, b_i^2) : (h_i^2, h_i^5) > (h_i^1, h_i^6)$$
$$(c_j^1, d_j^1) : (h(c_j^1), t_j^4,) > (t_j^1, t_j^3)$$
$$(c_j^2, d_j^2) : (h(c_j^2), t_j^5) > (t_j^2, t_j^1)$$
$$(c_j^3, d_j^3) : (h(c_j^3), t_j^6) > (t_j^3, t_j^2).$$

The preferences of the hospitals are:

$$h_i^1 : a_i^2 > a_i^1$$
$$h_i^2 : a_i^1 > a_i^2$$
$$h_i^3 : b_i^1 > c(h_i^3)$$
$$h_i^4 : b_i^1 > c(h_i^4)$$
$$h_i^5 : b_i^2 > c(h_i^5)$$
$$h_i^6 : b_i^2 > c(h_i^6)$$
$$t_j^1 : c_j^1 > d_j^2$$
$$t_j^2 : c_j^2 > d_j^3$$
$$t_j^3 : c_j^3 > d_j^1$$
$$t_j^4 : d_j^1$$
$$t_j^5 : d_j^2$$
$$t_j^6 : d_j^3$$

We define two sets for each $1 \le i \le n$:
$T_i = \{(a_i^1, h_i^2), (a_i^2, h_i^1), (b_i^1, h_i^4), (b_i^2, h_i^6), (c(h_i^3), h_i^3), (c(h_i^5), h_i^5)\}$ and
$F_i = \{(a_i^1, h_i^1), (a_i^2, h_i^2), (b_i^1, h_i^3), (b_i^2, h_i^5), (c(h_i^4), h_i^4), (c(h_i^6), h_i^6)\}$.

First suppose that there is a satisfying assignment f to φ. We construct a matching M that will be stable. If x_i is true then we assign the pairs of T_i to each other, otherwise we assign the pairs of F_i. If c_j^s is assigned to $h(c_j^s)$ this way, then we match d_j^s to t_j^{s+3} accordingly. Then, for every clause C_j we do the following. If C_j has three true literals, then we do nothing, all couples corresponding to C_j are already matched. If C_j has one false literal at place s $(1 \le s \le 3)$ then we add the pairs $\{(c_j^s, t_j^s), (d_j^s, t_j^{s+2})\}$ to M. If C_j has only one true literal at position s then we add $\{(c_j^{s+1}, t_j^{s+1}), (d_j^{s+1}, t_j^{s+1})\}$ to M. That completes the construction of M.

Assume that M is not stable and there is a couple-hospital coalition blocking M.

The couples of the form (a_i^r, b_i^r) are never unassigned and if one of them is at their worse choice than one of their top choice hospital got a better resident, so these couples cannot block.

A couple (c_j^s, d_j^s) cannot block M with $(h(c_j^s), t_j^{s+3})$, since if they are not matched to $(h(c_j^s), t_j^{s+3})$, then since every hospital is full, $h(c_j^s)$ got a better applicant.

Finally, a couple (c_j^s, d_j^s) also cannot block M with (t_j^s, t_j^{s+2}), because that would mean that (c_j^s, d_j^s) is unassigned, which can only happen if there is only one true literal in C_j. But then (c_j^{s+2}, d_j^{s+2}) is at (t_j^{s+2}, t_j^{s+1}) so t_j^{s+2} has a better resident than d_j^s.

This shows that the matching M is stable.

For the other direction let us suppose that M is a stable matching. Then, both (a_i^1, b_i^1) and (a_i^2, b_i^2) must be matched for any i, since otherwise they would block with their second choice. So either both of them are at their first choice or both of them are at their second choice. By the construction of the instance it is easy to see that if there is no couple (c_j^s, d_j^s) for some j such that they are at their best option, then there can only be one couple from (c_j^1, d_j^1), (c_j^2, d_j^2), (c_j^3, d_j^3) that is assigned somewhere, but then one of the other two couples would block with its second choice hospitals. So for each j there is an s such that $\{(c_j^s, h(c_j^s), (d_j^s, t_j^s)\} \subset M$.

So let $h_i^r = h(c_j^s)$. Now we set the variables its truth values the following way: if r is 3 or 5, then let x_i be true, otherwise if r is 2 or 4, then let x_i be false. This guarantees that there is at least one true literal in each clause. We only need to check that this truth assignment is well-defined. But this follows from the fact that if there is a couple at (h_i^r, t_j^s), with $r = 3$ or $r = 5$, then (b_i^1, b_i^2) must be at (h_i^4, h_i^6), so none of them can be assigned a couple member of the set B and similarly for $r \in \{4, 6\}$. □

Before we move on to some other interesting aspects of this problem, like finding fractional or near feasible solutions, we make a detour to another generalization of the stable matching problem, which will surprisingly be also useful for some proofs about the Hospital-Resident-Couple problem.

6 The Stable Flow problem

In this section we study another generalization of the stable matching problem, the stable flow problem. It has been introduced by Fleiner [11]. It models a more complex market, where some kind of commodity is forwarded through agents, such that these agents can have preferences over who they buy from and over who they sell to. The precise formulation of the problem is the following:

Let $D = (V, A)$ be a directed graph with two special vertices s and t. s will be called the source and t the terminal. We assume no arc enters s and no arc leaves t. Furthermore, there is a capacity function on the edges $c : A \to \mathbb{R}_+$. Then the system (D, s, t, c) is called a *network*.

Definition 6.1. A *flow* of a network is a function $f : A \to R$ on the edges, such that $0 \leq f(e) \leq c(e)$ for each $e \in A$ and for each vertex $v \in V$, with $v \notin \{s, t\}$ it holds that $\sum_{uv\in A} f(uv) = \sum_{vu\in a} f(vu)$, so the inflow and the outflow are the same in every non source and non terminal vertex.

Now let us suppose that similarly to the other stable matching problems, there are strict preference relations $<_v$ given for the vertices v on the arcs incident to v. Then this network with preferences can be interpreted as a network of agents trading certain goods between each other, with the direction of the arcs representing the way of the trade and the edge capacities the maximum amount of goods an agent can sell to a certain other agent. Also, the agents may have other agents whom they prefer to buy from, which gives rise to the preferences on the edges.

What should then, in this new problem, the notion of stability mean? First of all, if there is an unsaturated st path in the network, then each agent could sell and receive more products along the path, so we want to avoid such situations. Also if there is an unsaturated path between two agents with starting agent being the source which can just send more, or one sending some flow to a worse agent and the other being the terminal or one receiving from a worse source, then they would like to send some amount of their trades along this path instead which again would lead to instability. This motivates the following definition.

Definition 6.2. A walk $P = (v_1, e_1, v_2, ..., e_{k-1}.v_k)$ *blocks* the flow f, if the following conditions hold:

- all the vertices are different, only v_1 and v_k can coincide,
- $f(e_i) < c(e_i)$, $i = 1, ..k$,
- $v_1 = s$ or there is an arc v_1u such that $f(v_1u) > 0$ and $v_1u <_{v_1} e_1$,
- $v_k = t$ or there is an arc uv_k such that $f(uv_k) > 0$ and $uv_k <_{v_k} e_{k-1}$.

Definition 6.3. A flow f is *stable* if it admits no blocking walk.

As we will see in the next section, a stable flow always exists and furthermore, if all capacities are integral, then there is a stable flow that is integral too.

6.1 Reduction to Stable Allocation

In this section we show how to find stable flows. As Fleiner showed [11], the problem can be reduced to a stable allocation problem, which can be solved in polynomial time, as we have seen.

Let us start with describing the reduction used by Fleiner. Let (D, s, t, c) be the network with preferences on the vertices. Let q be a large integer, such that for any vertex, the maximum size of flow that can travel through that vertex is strictly less than q. For example $q = \sum_{e\in A} c(e) + 1$ obviously suffices. Now we construct a bipartite graph $G_D = (U, W, E)$. For each $v \in V$ we make two vertices v^{in} and v^{out}, both with capacity q. Let $U = \{v^{in} | \ v \in V, v \neq s\}$ and $W = \{v^{out} | \ v \in V, v \neq t\}$. Then for each arc $uv \in A$ we make an edge $u^{out}v^{in}$ in G_D with capacity $c(uv)$. Furthermore add two edges between v^{in} and v^{out}, each with capacity q for every vertex $v \neq s, t$. We will refer to these edges as e_v and f_v. The preferences for the vertices are the following: Each vertex v^{in} ranks e_v best, then the other adjacent edges coming from the original network in the same order, followed by f_v at the bottom. The v^{out} vertices rank f_v first, then the network edges in their original order and e_v last. Now we show that the stable matchings of G_D and the stable flows are in an (efficiently computable) bijection with each other.

Theorem 6.4. *[11] Let (D, s, t, c) be a network with preferences. Then, a flow f is stable if and only if there is a stable allocation g of G_D with $g(u^{out}v^{in}) = f(uv)$ for every $u, v \in A$.*

Proof. Assume g is a stable allocation. Then, since no e_v and f_v edges are blocking at least one of v^{in} and v^{out} is saturated. Furthermore, we can see that both are saturated, since if for example v^{in} would be unsaturated then $g(f_v) < q$ and f_v would be a blocking edge. So all edges are saturated which means that the Kirchoff law is satisfied for $f(uv) = g(u^{out}v^{in})$. The capacity constraints are obviously satisfied too. Now let us suppose that f is not stable and there is a path $P = (v_1, a_1, .., v_k)$ blocking f. That means $f(a_1) < c(a_1)$ and there is an arc v_1u with positive f value that v_1 prefers more. This implies that $v_1^{out}v_2^{in}$ can only be dominated at v_2^{in}. This implies that v_2^{in} is saturated with better edges so $g(f_{v_2}) = 0$ and $g(e_{v_2}) > 0$. Again, that means that $v_2^{out}v_3^{in}$ can only be dominated at v_3^{in} and continuing using this argument we obtain that $v_{k-1}^{out}v_k^{in}$ has to be dominated at v_k^{in}. But since P is a blocking walk, there is an arc uv_k with $f(uv_k) > 0$ that v_k prefers less than $v_{k-1}v_k$ and $f(a_{k-1}) < c(a_{k-1})$. But that cannot happen if v_k is dominated at v_k^{in}, contradiction.

For the other direction suppose that f is a stable flow in D. Let S be the set of those vertices $v \in V$, for which there is a path $P = (v_1, a_1, ..., v)$ such that the path is unsaturated by f and there is an arc v_1u with $f(v_1u) > 0$ that v_1 prefers less than a_1. Now define g as $g(u^{out}v^{in}) = f(uv)$,

$$g(e_v) = \begin{cases} q - \sum_{u \in V}, & \text{if } v \in S \\ 0 & \text{otherwise} \end{cases} \text{ and } g(f_v) = \begin{cases} q - \sum_{u \in V}, & \text{if } v \notin S \\ 0 & \text{otherwise} \end{cases}.$$

It is easy to see that all vertices are saturated by g in G_D and that the capacity constraints are not violated. Let us suppose that there is a blocking edge. First assume it is of the form e_v or f_v. But these edges are the worst ones for one of their endpoints, so since every vertex is saturated they cannot block. Now assume the blocking edge is $u^{out}v^{in}$. If $v \in S$, then then $g(f_v) = 0$ and there is a path that is unsaturated, ends at v and is not dominated at its startpoint. So if there would be an edge wv that v prefers less with $f(wv) = g(w^{out}v^{in}) > 0$, then P would be a blocking path to f, contradiction. Lastly, if $v \notin S$ then $g(e_v) = 0$ and $g(f_v) > 0$ and the path $P = (u, uv, v)$ is unsaturated and not dominated at u, so it is dominated at v, meaning that $u^{out}v^{in}$ is dominated at v^{in}, contradiction. So g is a stable allocation. □

This lemma leads to the following theorem:

Theorem 6.5. *[11] In every (D, s, t, c) network with preferences there exists a stable flow that can be found in polynomial time. Also, if the capacities are integral, then there is a stable integral flow.*

Another remarkable consequence of this is that just like in the stable marriage problem, the same set of agents are matched in every stable matching, so all of them has the same size, the same is true for stable flows too, that is, all stable flows have the same size, which can be proved by the above reduction and a use of the Rural Hospital theorem.

Theorem 6.6. *Let (D, s, t, c) be a network with preferences and f_1,f_2 be two different stable flows. Then the two flows have same size, moreover $f_1(e) = f_2(e)$ for all arcs leaving s and all arcs entering t.*

6.2 The Stable Multicommodity Flow problem

In this section we generalize the concept of stable flows even further, allowing multiple commodities to be forwarded through the network. It has been introduced by Király and Pap [18]. In this model, the agents can sell different types of products between each other and have preferences that can depend on the type of the product too. Also, importantly, we allow the carriers of the commodities to have preferences over the product they transport, so in this case even the edges will have preferences too. Most results of this section are new results from my working paper [7].

Now we formalize the problem precisely.

In the stable multicommodity flow problem, we are given a directed graph $D = (V, E)$ with sources $s_1, .., s_n$ and sinks $t_1, .., t_n$, one for each of the n commodities. Each vertex $v \in V$ has orderings $>^j_v$, $j = 1, ..., n$ on the edges incident to v. Furthermore, each $e \in E$ has an ordering $>_e$ on the set of commodities. Each edge has $c(e)$ and $c^j(e)$ capacities too, where $c(e)$ is the upper bound on the sum of the flow on e and $c^j(e)$-s are the upper bounds on the commodity flows $f^j(e)$. Let $\delta^j(v)$ denote the outflow and $\rho^j(v)$ the inflow of a vertex v with respect to f^j. So a flow $f = (f^1, .., f^n)$ is called *feasible*, if $\delta^j(v) = \rho^j(v)$ for each $j = 1, .., n$ and $v \in V \setminus \{s_j, t_j\}$, $f^j(e) \leq c^j(e)$ for each $e \in E$ and $j = 1, .., n$ and $\sum_{j=1}^n f^j(e) \leq c(e)$ for all $e \in E$.

Now we can introduce the notion of stability in multicommodity flows.

Definition 6.7. A walk $W = (v_1, a_1, ..., v_k)$ *blocks with respect to commodity* j, if the following four conditions hold:

1. $f^j(a_i) < c^j(a_i)$, $i = 1, .., k$,

2. $v_1 = s^j$ or there exists an $u \in V$ such that $f^j(v_1u) > 0$ and $v_1u <^j_{v_1} v_1v_2$, so v_1 likes v_2 more to send commodity j to,

3. $v_k = t^j$ or there exists a $w \in V$, such that $f^j(wv_k) > 0$ and $wv_k <^j_{v_k} v_{k-1}v_k$,

4. if $\sum_{j=1}^{n} f^j(a_i) = c(a_i)$, then there exists some $j' \neq j$, such that $f^{j'}(a_i) > 0$ and $j' <_{a_i} j$, so if an arc is saturated, then there is some commodity flow on it, that the arc is willing to trade in order to send more of commodity j.

Definition 6.8. A multicommodity flow is called *stable*, if it is feasible, and there is no blocking walk with respect to any commodity.

SMF
Input: An instance I of the stable multicommodity flow problem.
Output: A stable fractional multicommodity flow f.

ISMF
Input: An instance I of the stable multicommodity flow problem.
Question: Is there an integral stable multicommodity flow?

The variants of these two problems where the number of commodities are a fixed constant k will be denoted as k-SMF and k-ISMF respectively.

First we give a simpler proof of the existence of a stable multicommodity flow, first proven by Király and Pap [18], that also shows that SMF is in PPAD:

Theorem 6.9. *[7] The SMF problem can be reduced to SCARF in polynomial-time, so in every multicommodity flow instance there always exist a stable fractional solution.*

Proof. We will construct a matrix (see Figure 1) and apply Scarf's Lemma, then prove that all the dominating solutions correspond to stable multicommodity flows. Then, since a dominating solution always exists, we are done. Let Q be the following matrix: It has $n|E| + 2n|V|$ columns and $(n+1)|E| + 2n|V|$ rows. The columns are indexed by e_i^j, $i = 1, \dots, |E|$, $j = 1, \dots, n$ and these correspond to the $f^j(e_i)$ values in the solution. The remaining $2n|V|$ columns are indexed by $v_i^{j,in}$, $i = 1, \dots, |V|$, $j = 1, \dots, n$ and $v_i^{j,out}$, $i = 1, \dots, |V|$,$j = 1, \dots, n$ and we will need them to maintain that $\delta^j(v) = \rho^j(v)$ for every v, j. The rows are the following:

1. First we add one row for each $e_i \in E$. There are n 1-s in each row, such that $Q_{e_i, e_i^j} = 1$ for $j = 1, \dots, n$, and the other entries are 0. The constraints on the right side are the $c(e_i)$ capacities and the ranking of the columns are the same as the $<_{e_i}$ ranking of the edges.

2. Then we add $n|E|$ rows for each e_i^j that have only one 1-s, such that $Q_{e_i^j, e_i^j} = 1$ and 0 otherwise. Obviously these rows do not need orderings. The constraint on the right side for these rows are the $c^j(e_i)$ capacities.

3. Then we have a row $v_i^{j,in}$ for each $v_i \in V$ and $j = 1, \dots, n$. In the first $n|E|$ columns $Q_{v_i^{j,in}, e^{j'}} = 1$ if and only if $e = uv_i$ for some $u \in V$ and $j = j'$ and 0 otherwise. Furthermore, $Q_{v_i^{j,in}, v_i^{j,in}} = 1$ and $Q_{v_i^{j,in}, v_i^{j,out}} = 1$ and the other entries are 0. The right side has constraints $q(v_i)$, where

	E^1	E^2		E^n	$V^{1,in}$	$V^{1,out}$		$V^{n,in}$	$V^{n,out}$		
E	I	I		I							$c(E)$
E^1	I	0		0		0					$c^1(E)$
										$\leq$	
E^n	0	0		I							$c^n(E)$
$V^{1,in}$	A^{in}	0		0	I	I		0	0		$q(V)$
$V^{1,out}$	A^{out}	0		0	I	I		0	0		$q(V)$
										$\leq$	
$V^{n,in}$	0	0		A^{in}	0	0		I	I		$q(V)$
$V^{n,out}$	0	0		A^{out}	0	0		I	I		$q(V)$

Figure 1: The matrix used in the reduction. A^{in} and A^{out} denote the in and out incidence matrices of D respectively.

$q(v_i)$ is large enough such that at least one of $x_{v_i^{j,in}}$ and $x_{v_i^{j,out}}$ has to be positive in order to achieve equality. (For example summing $c(e)$ on the adjacent edges to v and adding 1 is enough). The ranking on the columns are the following: The worst entry in row $v_i^{j,in}$ is $v_i^{j,out}$, the best is $v_i^{j,in}$ and the others are ranked according to $>^j_{v_i}$.

4. Finally we have a row $v_i^{j,out}$ for each $v_i \in V$ and $j = 1, \ldots, n$. Here $Q_{v_i^{j,out}, e^{j'}} = 1$ if and only if $e = v_i u$ for some $u \in V$ and $j = j'$, 0 otherwise, and $Q_{v_i^{j,out}, v_i^{j,out}} = 1$, $Q_{v_i^{j,out}, v_i^{j,in}} = 1$ same as above. The constraints will be the same $q(v_i)$-s too. The ranking on the columns are the following: The worst entry in row $v_i^{j,out}$ is $v_i^{j,in}$, the best is $v_i^{j,out}$ and the others are ranked according to $>^j_{v_i}$.

Now, we have a polyhedron $\{Qx \leq q, x \geq 0\}$, which satisfies the conditions of Scarf's lemma, so we know that there is some solution x^* that dominates every column. We have to show that $f = (f^1, \ldots, f^n)$, $f^j(e) = x^*_{e^j}$ is a feasible flow and that it is stable.

Lemma 6.10. *f is a feasible flow, that is $\delta^j(v) = \rho^j(v)$ for each $j = 1, \ldots, n$ and $v \in V \setminus \{s_j, t_j\}$, $f^j(e) \leq c^j(e)$ for each $e \in E$ and $j = 1, \ldots, n$ and $\sum_{j=1}^n f^j(e) \leq c(e)$ for all $e \in E$.*

Proof. First of all, because of the first $(n+1)|E|$ rows, the conditions $f^j(e) \leq c^j(e)$ and $\sum_{j=1}^n f^j(e) \leq c(e)$ are trivially satisfied. To show that the inflows and outflows are equal everywhere we need to observe the other rows. Let us take any commodity $j \in \{1, \ldots, n\}$ and any vertex $v_i \in V \setminus \{s^j, t^j\}$. We know that since x^* is a dominating it dominates both of columns $v_i^{j,in}$ and $v_i^{j,out}$. Because of the construction of the matrix they can only be dominated at row $v_i^{j,in}$ or $v_i^{j,out}$. We have two cases:

a) If they are dominated at separate rows, then by definition, both row binds (the constraint is satisfied with equality). That means $\delta^j(v_i) = \rho^j(v_i) = q(v_i) - x^*_{v_i^{j,out}} - x^*_{v_i^{j,in}}$, so we are done in this case.

b) If they are both dominated at, say row $v_i^{j,in}$, then since column $v_i^{j,in}$ is the best for this row, it can only be dominated, if every other x coordinate is zero, where there is a 1 in row $v_i^{j,in}$. So $x^*_{v_i^{j,in}} = q(v_i)$. Then, because of $x^* \geq 0$, and $Q_{v_i^{j,out}} x^* \leq q(v_i)$, and $Q_{v_i^{j,out}, v_i^{j,in}} = 1$ we can see that every other entry is zero in row $v_i^{j,out}$ too. So $\delta^j(v_i) = \rho^j(v_i) = 0$. The case when the columns are dominated in row $v_i^{j,out}$ is analogous. □

Lemma 6.11. *f is stable.*

Proof. Let us suppose f is not stable and there exists a blocking walk $W = (v_1, a_1, \ldots, v_k)$ with respect to some commodity j. Since x^* is a dominating solution, it dominates every column. First observe, that the a_i^j-s cannot be dominated at row a_i^j, since that would mean that they are saturated by f^j, which contradicts the fact that W is a blocking walk. They also cannot be dominated at the a_i rows, since that would mean that $\sum_{l=1}^n f^l(a_i) = c(a_i)$ and for every $j' \neq j$ for which $f^{j'}(a_i) > 0$, a_i prefers $a_i^{j'}$ to a_i^j, so there is no $j' \neq j$ such that $f^{j'}(a_i) > 0$ and $j' <_{a_i} j$, contradiction.

Now let us look at where a_1^j is dominated. If $v_1 \neq s^j$ and it is dominated at row $v_1^{j,out}$, then that means that every entry that $v_1^{j,out}$ prefers less has value 0 in x^*. So there is no $v_1 u \in E$, such that $f^j(v_1 u) > 0$ and $v_1 u <^j_{v_1} v_1 v_2$, contradiction. If $v_1 = s^j$, then there is no $v_1^{j,out}$ row, so a_1^j obviously cannot be dominated there.

So the only possible case left is that it is dominated at row $v_2^{j,in}$. So by the domination we can see that $x^*_{v_2^{j,out}} = 0$, since it is the worst column for this row, so it is worse than a_1^j too. This also means that $x^*_{v_2^{j,in}} > 0$, since one of them has to be positive, because the corresponding row binds. So if we take a look at a_2^j, it still cannot be dominated at row a_2 or a_2^j, and neither at row $v_2^{j,out}$, since that would mean that every entry that the row prefers less is zero in x^*, but we have just seen that $x^*_{v_2^{j,in}} > 0$ and since it is the worst column it is also worse than a_2^j. So a_2^j can only dominated at row $v_3^{j,in}$. Continuing using the same argument we can conclude that a_{k-1}^j has to be dominated at row $v_k^{j,in}$. If $v_k = t^j$, then there is no such row, contradiction. If not, then this would mean that there is no $uv_k \in E$, such that $f^j(uv_k) > 0$ and $uv_k <^j_{v_k} v_{k-1}v_k$, because of the domination, so we get a contradiction in this last case too. □

So now we can conclude that the multicommodity flow given by x^* is a stable multicommodity flow indeed, and as a consequence of Scarf's lemma, one always exists. □

Next we prove a useful lemma, that shows that every instance of the stable multicommodity flow problem can be reduced to a much simpler one in polynomial-time, such that the solution of the two instances are in a bijection, which preserves integrality.

Lemma 6.12. *[?] For every $k \in \mathbb{N}$, we can reduce any instance of k-SMF or k-ISMF in polynomial time to another instance of k-SMF or k-ISMF where every commodity has the same terminal and source, the $>_v^i$ preferences are the same for all i for any $v \in V$, there are no commodity specific capacities and $\rho(v) + \delta(v) \leq 3$ for all non s, t vertices.*

Proof. First we use the reduction used by *Cseh* [9] to get a version that satisfies the first three conditions. Now let v be any non s,t vertex and let $e_1 >_v e_2 \cdots >_v e_k$ be its incoming edges and $f_1 >_v f_2 \cdots >_v f_l$ be its outgoing edges (after the reduction). Then we substitute each vertex v with a gadget G_v. G_v will consist of vertices $v'_1, \ldots, v'_k$ and $v''_1, \ldots, v''_l$. There is a $v'_i v'_{i-1}$ and a $v''_i v_{i+1}$ arc for all i and a $v'_1 v''_1$ arc. These have large enough capacity, such that they cannot be saturated (for example $\sum_{i=1}^{k} c(e_i)+1$). These edges can forward any commodity and have arbitrary preferences on them. Then for every $e_i = uv$, $i = 1, \ldots, k$ we make an arc $e'_i = u''_j v'_i$ that has the same capacity and preferences on the commodities as e_i, where j is the rank of e_i in u's preference list. Similarly for each $f_j = vw$ we make an arc $f'_j = v''_j w'_i$, where i is the rank of f_j in w's ranking. Furthermore, let the rankings of the new vertices be $e'_i >_{v'_i} v'_{i+1} v'_i$ and $f'_j >_{v''_j} v''_j v''_{j+1}$.

It is clear that a flow f can be extended to a flow f' in the new graph and vice versa any flow f' in the new graph can be reduced to a flow in the original graph by contracting the vertices in the gadgets, since any flow can only enter a gadget on a copy of an original edge and leave on a copy of another original edge. It is also clear that this preserves integrality.

Now let us suppose f was stable but there is a blocking walk $W' = (v_1, a_1, \ldots, v_k)$ with respect to commodity j to f'. That means that all a_i-s are either unsaturated or there is a worse (for them) commodity flowing through them, $v_1 = s$ or there is an edge $v_1 u$ such that $f'^j(v_1 u) > 0$, $v_1 u <_{v_1} v_1 v_2$ and $v_k = t$ or there is an edge $w v_k$ with $f'^j(v_k) > 0$ and $w v_k <_{v_k} v_{k-1} v_k$. But this means that the corresponding walk W in the original graph will be a blocking walk too, since if any v'_i copy of v in a gadget has incoming flow from the worse edge than $a_1 = e'_i$, then v has incoming flow from a worse edge than e_i too, since only paths that come from a worse edge can use $v'_{i+1} v'_i$, and analogously for outgoing edges. So W blocks with commodity j, contradiction. The other direction is similar, since if each edge of a walk is unsaturated/or there is a worse commodity on them, then in the new graph this is also true because the $v'_i v'_{i-1}$, $v'_1 v''_1$ and $v''_i v''_{i+1}$ edges cannot be saturated. Also, if there was a flow coming from a worse edge for v than some e, then this flow uses a $v'_i v'_{i-1}$ edge to v_i (for some i, such that e is v's i-th choice) which is worse for v'_i than e'_i and if there was a flow leaving on a worse edge f for v then this flow leaves v''_j on a $v''_j v''_{j+1}$ edge which is worse for v''_i, than f'_j so we get a blocking walk again, contradiction ∎

Remark 6.13. Because the SMF problem is PPAD-hard for unit capacities, we can assume that the $q(v)$ values we need to reduce SMF to SCARF are bounded above by the number of edges. That means, making $q(v)$ copies of objects in our reduction can be done in polynomial-time.

Using that SMF is in PPAD and it is PPAD-hard, as proven by Király and Pap [18], the following theorem is immediate.

Theorem 6.14. *The SMF problem is PPAD-complete.*

Now we strengthen this result by showing that the problem remains PPAD-complete for very restricted instances and only 3 commodities. Recall from theorem 4.21 that FRACTIONAL HYPERGRAPH MATCHING is PPAD-complete even if each hyperedge has size at most 3, and each vertex has degree at most 3, each capacity is 1 and the hypergraph is polynomial-time 3 edge-colorable.

Theorem 6.15. *[7] 3-SMF is PPAD-complete even if all capacities are 0 or 1, all commodities have the same source and terminal and the subgraphs corresponding to the edges that can forward a commodity form disjoint st paths.*

Proof. Since we have proven that FRACTIONAL HYPERGRAPH MATCHING is PPAD-complete even if the hypergraph is 3 edge-colorable, we only need to reduce hypergraphs with this property. But before we make the reduction, we color the edges of the hypergraph with 3 colors as mentioned before. The trick is that now we will only need 3 commodities, one for each color.
Let us describe the reduction itself. Let our (3-colorable) hypergraph be $(V, \mathcal{E})$, $V = \{v_1, \dots, v_n\}$. Now the new graph for the 3-SMF instance will have the following vertices: for each $v_i \in V$ we have two vertices v_i' and v_i''. Furthermore we make a source s and a sink t. The edges will be the following: For each i we add an edge $e_i = v_i'v_i''$ with capacity 1, that can only forward the commodities corresponding to the colors of the edges adjacent to v_i (so other commodity capacities are 0). The ranking of e_i on the relevant commodities is the same as the ranking of v_i on the adjacent hyperedges. Then, for each hyperedge $e = \{v_j, v_k, v_l\}$ (we can suppose the hypergraph is 3 uniform by padding smaller hyperedges with dummy vertices) we add sv_j', $v_j''v_k'$, $v_k''v_l'$ and $v_l''t$ edges that can only carry the commodity corresponding to the color of e.

Now observe that because of the properties of the coloring, for each commodity, the edges that it can travel on form internally disjoint st paths. So each commodity i can only flow on a path corresponding to a hyperedge of color i, and for each hyperedge of color i we can send a flow of commodity i on the corresponding path. So the fractional flows correspond to fractional matchings and vice versa. Furthermore, a blocking hyperedge would mean that the corresponding path is blocking, and a blocking st path would mean that the corresponding hyperedge is blocking. Since a blocking walk can only start from s and end in t in this graph, we are done. ☐

Now we turn our attention to the integral case and prove that deciding if there exists an integral stable multicommodity flow is hard even if we have only three commodities. First, we mention that the containment of ISMF in NP is already known:

Lemma 6.16 (Király and Pap [18]). *ISMF is in NP.*

Proof. It suffices to prove that it can be verified if a given flow is stable or not. Suppose we are given a multicommodity flow f. For each pair $u, v \in V$ and commodity $j \in [n]$, we make a directed graph $D_{u,v;j}$, where only edges with $f^j(e) < c^j(e)$ are included, and saturated edges are only included, if there is a commodity j', that is worse for e and $f^{j'}(e) > 0$. Also, if $u \neq s^j$, we delete the worst edge uv' for u with $f^j(uv') > 0$ and every edge worse that uv'. Similarly, if $v \neq t^j$, we delete the worst edge $u'v$ with $f^j(u'v) > 0$ and every worse edge for v.

It is easy to see that if $D_{u,v;j}$ contains an uv path, than it will block. Similarly, if f is not stable, then there is a blocking walk $W = \{v_1, e_1, .., v_k\}$, so when $u = v_1$ and $v = v_k$, we will find a blocking walk.

This shows that deciding the stability of a given flow f can be done in polynomial-time, even for fractional flows.

□

Recall the following theorem (theorem 5.6) we proved in the previous section. Now we state it in more detail, but the proof is exactly the same.

Theorem 6.17 (Biró et al.). *(2,2)-HRC is NP-complete, even if the following conditions hold:*
I) Each hospital has capacity one,
II) Each hospital appears at most once in every couple's preference list,
III) There are no single residents,
IV) Most hospitals receive applications only from men or women (we will call these man; and woman-Hospitals respectively) and the hospitals for which this is not true can be grouped to triangles (h^i_1, h^i_2, h^i_3) with couples c^i_1, c^i_2, c^i_3 such that they only receive application from c^i_1, c^i_2, c^i_3 and the preference lists are the following:
$c^i_1 : (*, *) > (h^i_1, h^i_2)$
$c^i_2 : (*, *) > (h^i_2, h^i_3)$
$c^i_3 : (*, *) > (h^i_3, h^i_1)$
$h^i_1 : c^i_3 > c^i_1$
$h^i_2 : c^i_1 > c^i_2$
$h^i_3 : c^i_2 > c^i_3$,
where $*$-s denote man or woman-hospitals. Furthermore each man; or woman-hospital only receives at most one application from these triangles.

We will need the details of the construction, so we describe it again here. The reduction is from the problem problem (2,2)-E3-SAT, that is the problem of deciding if a given Boolean formula φ in CNF over a set of variables V is satisfiable, where φ has the following properties: (i) each clause contains exactly 3 literals and (ii) for each $v_i \in V$, each of literals v_i and $\overline{v_i}$ appears exactly twice in φ. Let $V = \{v_1, \ldots, v_n\}$ be the set of variables and $C = \{c_1, \ldots, c_m\}$ be the

set of clauses. Then the couples of the HRC insctance will be $A \cup B$, where $A = \{(a_i^r, b_i^r) : r = 1, 2, i = 1, \ldots, n\}$ and $B = \{(c_j^s, d_j^s) : s = 1, 2, 3, j = 1, \ldots, m\}$. The hospitals will be $H \cup T$, where $H = \{h_i^r : r = 1, \ldots, 6, i = 1, \ldots, n\}$, $T = \{t_j^r : r = 1, \ldots, 6, j = 1, \ldots, m\}$. Define $h(d_j^s)$ to be h_i^{2r+1} if the r-th appearance of v_i is at position s at c_j, and h_i^{2r+2} if the r-th appearance of $\overline{v_i}$ is at position s at c_j. Similarly let $d(h_i^{2r+1})$ be d_j^s if the r-th occurrence of literal v_i is at the s-th position of c_j and $d(h_i^{2r+2})$ be d_j^s if the r-th occurrence of literal $\overline{v_i}$ is at the s-th position of c_j. So the possible applications, with the order of the preferences are the following:
$(a_i^1, b_i^1) : (h_i^1, h_i^3) > (h_i^2, h_i^4)$
$(a_i^2, b_i^2) : (h_i^2, h_i^5) > (h_i^1, h_i^6)$
$(c_j^1, d_j^1) : (t_j^4, h(d_j^1)) > (t_j^3, t_j^1)$
$(c_j^2, d_j^2) : (t_j^5, h(d_j^2)) > (t_j^1, t_j^2)$
$(c_j^3, d_j^3) : (t_j^6, h(d_j^3)) > (t_j^2, t_j^3)$.

Theorem 6.18. *[7] 3-ISMF is NP-complete even if all capacities are 0 or 1, all commodities have the same source and terminal and the subgraphs corresponding to the edges that can forward a commodity form disjoint st paths.*

Proof. The problem is in NP by Lemma 6.16, so we only have to prove NP-hardness.

We will prove that the instances we can get in the reduction of theorem 5.6 can be further reduced to an instance of a FRACTIONAL HYPERGRAPH MATCHING problem, where the hypergraph is 3 edge-colorable. Then we are done, because we can use the same reduction as in the proof of Theorem 6.15, since it preserves integrality too and its two-way, so stable matchings correspond to stable flows and vice versa.

Let our hypergraph be $\mathcal{H} = (V, \mathcal{E})$, where $V = A \cup B \cup H \cup T$, and the hyperedges are the possible couple-hospital triples. Since in the HRC instance, each couple only applies to a given hospital in one of their choices, the rankings of the hospitals on the residents can be extended to a ranking on the couples (since from every couple only one of them applies to any given hospital). This also means that any hospital and couple vertex uniquely defines a hyperedge in $\mathcal{E}$ if one exists. So the preferences of a hospital h on the couples can be directly extended to the preference of vertex h on the adjacent hyperedges. The couples' preferences are easily extended too.

Now it is easy to observe that integral stable matchings of the two instances correspond to each other, since a blocking (c, h, h') coalition would mean a blocking hyperedge and vice versa. So again, it only remains to prove that we can color the hyperedges of $\mathcal{H}$ by three colors.

First, we color the edges that contain couples of the form (c_j^s, d_j^s). We claim that we can color these in such a way that any such edge containing a hospital of the form h_i^3, h_i^4, h_i^5 or h_i^6 will be green or red. But this can be seen from the fact that these couples can be partitioned to groups of three, whose hyperedges only intersect other edges outside (and only ones containing (a_i^r, b_i^r) couples) in the h_i^3, h_i^4, h_i^5 and h_i^6 vertices. Also in any group, a hospital of this form can only be in one hyperedge, since we know it is in at most two, and one of them contains an (a_i^r, b_i^r) pair. So we can just color two of the three hyperedges of the form $((c_j^s, d_j^s), t_j^{s+3}, h(d_j^s))$ green and the third red and we can still color the remaining $((c_j^1, d_j^1), t_j^3, t_j^1)$, $((c_j^2, d_j^2), t_j^1, t_j^2)$ and $((c_j^3, d_j^3), t_j^2, t_j^3)$ edges properly.

Now, the remaining edges form disjoint quartets, apart from the $h_i^{\geq 3}$ vertices. But we now know that on every h_i^3, h_i^4, h_i^5 and h_i^6 there is only green and red hyperedges. It is straightforward to verify that in every case we can color the $((a_i^1, b_i^1), h_i^1, h_i^3)$, $((a_i^2, b_i^2), h_i^2, h_i^5)$, $((a_i^1, b_i^1), h_i^2, h_i^4)$ and $((a_i^2, b_i^2), h_i^1, h_i^6)$ hyperedges properly with 3 colors. For example when h_i^3 and h_i^4 are red and h_i^5 and h_i^6 are green the following coloring will be appropriate: $((a_i^1, b_i^1), h_i^1, h_i^3)$ and $((a_i^2, b_i^2), h_i^2, h_i^5)$ yellow, $((a_i^1, b_i^1), h_i^2, h_i^4)$ green and $((a_i^2, b_i^2), h_i^1, h_i^6)$ red. □

If the number of commodities are only two, then if we assume that the two sources and terminals are the same, and the edges corresponding to one commodity form disjoint *st* paths, then the problem always admits a stable integral multicommodity flow, and it can be found in polynomial time.

Lemma 6.19. *[7] 2-ISMF and 2-SMF can be solved in polynomial time by Algorithm 6 if the two sources and terminals are the same, and the edges that can forward a given commodity form disjoint st paths.*

Proof. Since in a network like this every blocking walk has to be an *st* path, we only have to check the stability regarding these. But if an *st* path would block with respect to commodity 1, then the path is unsaturated by commodity 1, so there was a step when we decreased a flow, but in the last such step, an edge that prefers commodity 2 became saturated, so the path cannot block. A path also cannot block with commodity 2, because otherwise we would have sent more flow of commodity 2 in the steps corresponding to that path. □

Algorithm 6

Saturate all *st* paths corresponding to commodity 1 with commodity 1.
Let P_1,..,P_k be the *st* paths that can forward commodity 2.
for $i = 1, ..., k$ **do**
  while P_i is not saturated with commodity 2 and all saturated edges $e \in P_i$ that forward a positive value of commodity 1 prefer commodity 2 **do**
    increase the flow of commodity 2 on P_i by 1,
    decrease the flow of commodity 1 on the commodity 1 *st* paths that became oversaturated, by 1.
  end while
end for

An immediate consequence is the following:

Theorem 6.20. *Every instance of the FRACTIONAL HYPERGRAPH MATCHING where the hypergraph is 2 edge-colorable always admits an integral stable matching and it can be found in polynomial time.*

Remark 6.21. The algorithm of Ishizuka and Kamiyama also works for the 2 edge-colorable case, since then all degrees are at most 2, but only if each capacity is 1. Furthermore, their algorithm only returns a half integer solution and it can happen that it returns a fractional solution even if there is an integer one. It is easy to check that for example in an even cycle, where everyone prefers its left neighbor, there exists a stable matching, but the algorithm of Ishizuka and Kamiyama returns the fractional matching with all edge weights set to 0.5.

If we do not require the edges which can forward a given commodity to form disjoint st paths, then we can further reduce the number of commodities needed for NP-completeness to two.

Theorem 6.22. *[?] 2-ISMF is NP-complete even if all capacities are 0 or 1.*

Proof. We will prove this by reducing the version of (2,2)-HRC described in theorem 6.17.

Let $\mathcal{I}$ be an instance of HRC. We call a couple c *ordinary* if c^m only applies to man-hospitals and c^w only applies to woman-hospitals.

We construct an instance $\mathcal{I}'$ of 2-ISMF the following way: First of all let us make a source vertex s that will be the source of every commodity, and a sink vertex t that will be the sink of every commodity. Then, for each couple c we assign vertices $v(c^m)$ and $v(c^w)$ and an edge $e(c) = v(c^m)v(c^w)$ with $c(e(c)) = c^1(e(c)) = c^2(e(c)) = 1$. If c is ordinary, then let $e(c)$'s ranking on the commodities be $1 >_{e(c)} 2$ (for non-ordinary couples we will clarify this later). Next for every man-hospital h, we assign a vertex $v(h)$ and an edge $e(h) = sv(h)$ with $c(e(h)) = c^1(e(h)) = c^2(e(h)) = 1$. Then, since h can appear at most twice on the preference lists of couples, we make one or two edges from $v(h)$ to the $v(c^m)$ vertices of these couples. If c is an ordinary couple, then the edge $v(h)v(c^m)$ can only carry commodity 1 (so other capacities are 0) if h is in c's first choice, and commodity 2 if it's in c's second choice. If c is not ordinary, then the edge $v(h)v(c^m)$ can forward only one commodity too, that will be clarified later in the description of the gadgets. Now if the two edges leaving $v(h)$ can carry only different commodities, then there is no need to rank the outgoing edges, but we rank the commodities on $e(h)$ in such a way that the commodity that can travel to the better couple for h is first and the other is second. If the two edges can only carry the same commodity, then we do not need rankings on $e(h)$, since only one commodity will be able to travel on it, but we rank the two outgoing edges the same way as h ranks the two couples they are pointing to. Similarly, we make vertices $v(h)$, and edges $v(c^w)v(h)$ and $v(h)t$ for all woman-hospitals. The rankings, incoming edges and capacities are made symmetrically.

For a triumvirate (c_1, c_2, c_3) consisting of non-ordinary couples we make the following gadgets (see Figure 2):

We make four vertices u_{12}, v_{12}, u_3, v_3 and edges su_{12}, $u_{12}v(c_1^m)$, $u_{12}v(c_2^m)$, $v(c_1^w)u_3$, $v(c_2^w)v_3$ and $v_{12}t$ each with capacities $c(e) = c^1(e) = 1$, $c^2(e) = 0$. We add further edges $sv(c_3^m)$, $v(c_3^w)u_3$, v_3t with capacities $c(e) = c^2(e) = 1$, $c^1(e) = 0$. Finally we add an u_3v_{12} edge with $c(u_3v_{12}) = c^1(u_3v_{12}) = c^2(u_3v_{12}) = 1$,

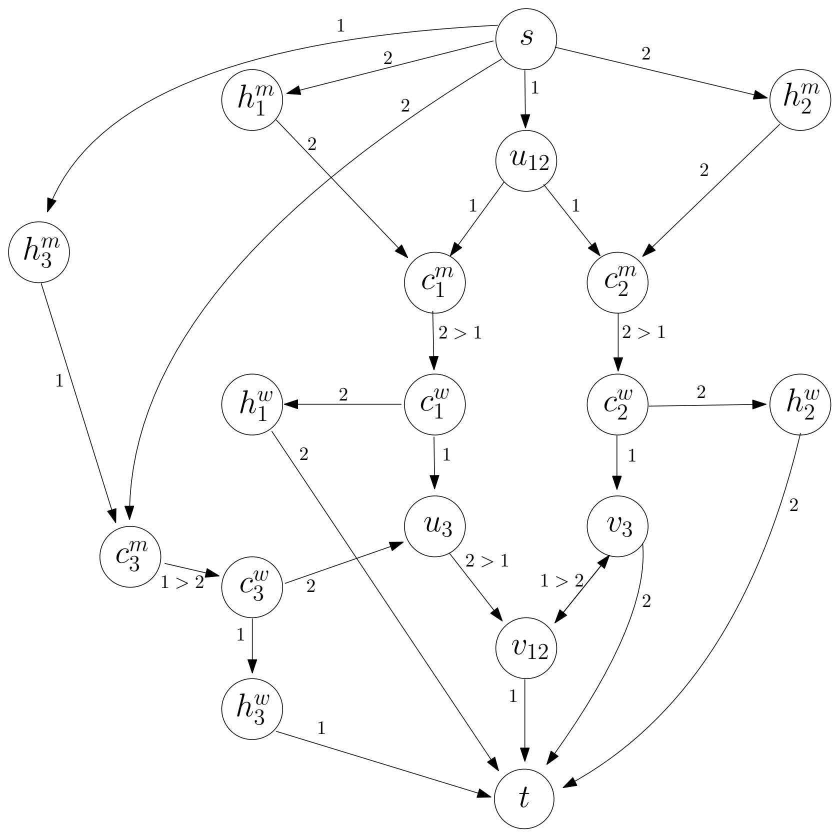

Figure 2: The gadget used in the reduction of Theorem 6.22. The h_i^m and h_i^w hospitals are c_i's first choice hospitals. Label 1 on an edge means $c^1(e) = 1$, $c^2(e) = 0$, similarly for 2 and $1 > 2$, $2 > 1$ labels represent the edge's ranking when both commodity can travel on it.

with ranking $2 > 1$ and a $v_{12}v_3$ edge, which can forward commodity 1 from v_3 to v_{12} and commodity 2 from v_{12} to v_3 (but only one unit of flow can go through it), with ranking $1 > 2$. We can implement this two-way edge with a subgadget G' the following way: make two more vertices v'_{12} and v'_3 and add directed edges $v_3v'_3$ and $v'_{12}v_{12}$ with unit capacity that can only carry commodity 1 $v_{12}v'_3$ and $v'_{12}v_3$ with unit capacity that can only carry commodity 2 and an edge $v'_3v'_{12}$ with $c(v'_3v'_{12}) = c^1(v'_3v'_{12}) = c^2(v'_3v'_{12}) = 1$ and preference $1 > 2$. It is easy to check that this subgadget only lets commodity 1 and commodity 2 flow in the direction we want it, and since both must pass through edge $v'_3v'_{12}$, the total flow on v_3v_{12} is at most one. From these new vertices only u_{12} and v_{12} need rankings, since only they have two incoming/outgoing edges that can forward the same commodity. Let $u_{12}v(c_1^m) >_{u_{12}} u_{12}v(c_2^m)$ and $u_3v_{12} >_{v_{12}} v_3v_{12}$ be their preferences.

Later, we will refer to all these edges that we added as gadget edges. Now we can describe the non-ordinary couple-hospital edges' preferences: if c_j's first choice is (h_j, h'_j), then the edges $v(h_j)v(c_j^m)$ and $v(c_j^w)v(h'_j)$ can only carry commodity 2 if $j = 1, 2$ and commodity 1, if $j = 3$. Furthermore let the preferences on the $v(c_j^m)v(c_j^w)$ edges be $2 > 1$, if $j = 1, 2$ and $1 > 2$, if $j = 3$.

The flows and matchings will correspond to each other the following way: if an integer flow of commodity 1 or 2 flows through an ordinary couple's edge $e(c)$, then in the corresponding matching c is matched to their first or second choice respectively. If no flow uses $e(c)$, then c remains unassigned. If the integer flow uses edges $v(h)v(c^m)$ and $v(c^w)v(h')$ then c^m is assigned to h and c^w is assigned to h'. If $c = c_j$ $j = 1, 2$ is part of a triumvirate, then if commodity 2 flows through $v(c_j^m)v(c_j^w)$ (and then through corresponding hospitals) then c_j is matched to its first choice and if commodity 1 flows through $v(c_j^m)v(c_j^w)$ (and then through the gadget to t) then c_j is matched to its second choice (h_j, h_{j+1}). Similarly for c_3 with the difference that commodity 1 means it is matched to its first choice and commodity 2 (which can flow through the gadget) that it is matched to its second choice.

Conversely, if an ordinary couple c is matched to hospitals (h, h'), then we send a flow of commodity 1 or 2 (depending on whether its c's first or second choice) through $s \to v(h) \to v(c^m) \to v(c^w) \to v(h') \to t$. If a non-ordinary couple c_j is matched to its first choice, then we do the same, if it is matched to its second choice, then we send a flow of commodity 1 if $j = 1, 2$ and commodity 2, if $j = 3$ through the gadget. (For clarity, the path for c_1 is $s \to u_{12} \to v(c_1^m) \to v(c_1^w) \to u_3 \to v_{12} \to t$, and for c_2 it is $s \to u_{12} \to v(c_2^m) \to v(c_2^w) \to u_3 \to v_{12} \to t$ and finally for c_3 it is $s \to v(c_3^m) \to v(c_3^w) \to u_3 \to v_{12} \to v_3 \to t$).

Lemma 6.23. *If M is a feasible matching in $\mathcal{I}$, then the constructed flow f is feasible in $\mathcal{I}'$ and vice versa.*

Proof. Let M be a feasible matching. Then, every couple is assigned to at most one hospital and every hospital has at most one resident. So on each path that goes through an ordinary couple's edge, there is at most one unit of flow and similarly through each $sv(h)$ and $v(h)t$ edge for each man or woman-hospital. Other edges not in gadgets are only forwarding flow between these types of edges, so they aren't oversaturated either.

To see the flow is feasible on the gadget edges too, observe that its feasibility on the $v(c_j^m)v(c_j^w)$ edges follow from the same argument, and on the other edges it could only fail if we send more than one unit of flow through su_{12}, u_3v_{12} or $v_{12}v_3$, but either would mean that two of the corresponding c_1, c_2, c_3 couples are assigned to h_1, h_2, h_3, so at least one hospital exceeds its quota, contradiction.

Now, if f is feasible, then on each couple and ordinary hospital edge there is at most one unit of flow, so the capacity constraints of couples and man; or woman-hospitals are satisfied. And since only one unit of flow can run on each of the edges su_{12}, u_3v_{12} or $v_{12}v_3$, the capacity of the non-ordinary hospitals are satisfied too. ☐

Lemma 6.24. *M is stable if and only if f is stable.*

Proof. Suppose M is stable, but f is not, so there exists a blocking path W with commodity l ($l = 1$ or 2). It can only be a path, since the set of edges on which a given commodity can travel form an acyclic graph.

Case a) W does not go through a gadget edge.

Case a1) W is an $s \to t$ path.

In this case W goes through an $sv(h^m)$ edge, an $e(c)$ edge and a $v(h^w)t$ edge too. All of them are either unsaturated or a worse commodity goes through them, which means that the couple is either unemployed or it is at its second choice, while (h^m, h^w) is its first choice, h^m is either empty, or the man at h^m is worse for h^m and similarly for h^w. But either of the options means that (c, h^m, h^w) is a blocking coalition, contradiction.

Case a2) W is not an $s \to t$ path.

This means that it starts at a vertex $v(h^m)$ or ends at a vertex $v(h^w)$ or both. It cannot start or end at any other vertex, since there is no other vertex from where there are two outgoing or two incoming edges that could carry the same commodity (except in the gadget). If W starts at $v(h^m)$, then there has to be a flow of the same commodity on the other edge from $v(h^m)$, that $v(h^m)$ prefers less. But this means that h^m has a resident it prefers less than c^m. Similarly if W ends at $v(h^w)$ then the resident at h^w is worse for h^w than c^w. So we get that (c, h^m, h^w) is a blocking coalition in this case too, contradiction again.

Case b) W goes through a gadget edge.

By the construction of the gadgets we basically have only 3 choices for (W, l): The first one is $W = s \to u_{12} \to v(c_1^m) \to v(c_1^w) \to u_3 \to v_{12} \to t$ and $l = 1$, which would mean that c_1 is unassigned (because $v(c_1^m)v(c_1^w)$ has to be unsaturated, since $2 > 1$ for them), and commodity 2 does not flow through the gadget, since then $(W, 1)$ would not be blocking on u_3v_{12}. So c_3 is not assigned to (h_3, h_1), therefore (c_1, h_1, h_2) blocks, contradiction. Here, since $u_{12}v(c_1^m) >_{u_{12}} u_{12}v(c_2^m)$ and $u_3v_{12} >_{v_{12}} v_3v_{12}$, W could start at u_{12} and end at v_{12}, but this would mean the same (here we know in addition that c_2 must be at (h_2, h_3), but that does not change the blocking of (c_1, h_1, h_2), since c_1 is better than c_2 at h_2).

The second choice is $W = s \to v(c_3^m) \to v(c_3^w) \to u_3 \to v_{12} \to v_3 \to t$, $l = 2$. This means that c_3 is unassigned and commodity 1 does not flow through the gadget on v_3v_{12} (since commodity 2 could not block there), so c_2 is not assigned to (h_2, h_3) and therefore (c_3, h_2, h_3) blocks.

The third choice is $W = s \to u_{12} \to v(c_2^m) \to v(c_2^w) \to u_3 \to v_{12} \to t$, $l = 1$. This means c_1 is not at (h_1, h_2) (since then su_{12} would be saturated) and c_2 is unassigned, so (c_1, h_1, h_2) blocks, contradiction again.

Now let us suppose we have a stable flow f, but there is a coalition (c, h, h') blocking M.

Case a) c is an ordinary couple.

Then $h = h^m$ and $h' = h^w$ for some man; and woman-hospitals respectively. Since c prefers these to its current assignment, either there is no flow through $e(c)$ or it is of commodity 2. The hospitals would hire the residents in c too, which means by the definition of the rankings, that either there is no flow through $sv(h^m)$ and $v(h^w)t$ or another type of flow goes through them that the corresponding edges prefer less, or if the same commodity goes through them, then $v(h^i)$ would prefer to send or receive the commodity on the other edge. Either would mean that there is a blocking path through these vertices (with commodity 1 if it was c's fist choice and commodity 2 otherwise), contradiction.

Case b) c is not an ordinary couple.

If h and h' are man; and woman-hospitals respectively, then using the same reasoning and that since c cannot be at its first choice, the edge $v(c_j^m)v(c_j^w)$ is either empty or has the commodity which it prefers less, we can construct a blocking walk with commodity 2 ($j = 1, 2$) or commodity 1 ($j = 3$) through $v(h)$, $e(c)$ and $v(h')$ again.

If $(c, h, h') = (c_1, h_1, h_2)$, then c_1 is unassigned and c_3 cannot be at (h_3, h_1), so there is no flow of commodity 2 through the gadget. There may be a flow through $s \to u_{12} \to v(c_2^m) \to v(c_2^w) \to u_3 \to v_{12} \to t$, but since $u_{12}v(c_1^m) >_{u_{12}} u_{12}v(c_2^m)$ and $u_3v_{12} >_{v_{12}} v_3v_{12}$, either $W = s \to u_{12} \to v(c_1^m) \to v(c_1^w) \to u_3 \to v_{12} \to t$ or $W = u_{12} \to v(c_1^m) \to v(c_1^w) \to u_3 \to v_{12}$ is a blocking walk with commodity 1.

If $(c, h, h') = (c_2, h_2, h_3)$, then c_2 is unassigned and c_1 cannot be at (h_1, h_2), so there can only be a flow of commodity 2 on the gadget, but either way $W = s \to u_{12} \to v(c_2^m) \to v(c_2^w) \to u_3 \to v_{12} \to t$ blocks with commodity 1.

Finally if $(c, h, h') = (c_3, h_3, h_1)$, then c_3 is unassigned and c_2 isn't at (h_1, h_2), therefore $W = s \to v(c_3^m) \to v(c_3^w) \to u_3 \to v_{12} \to v_3 \to t$ blocks with commodity 2. □

So we have proven that integral stable matching in $\mathcal{I}$ and integral stable flows in $\mathcal{I}'$ correspond to each other, which proves that 2-ISMF is NP-complete indeed. □

7 Back to the Hospital-Resident-Couple problem

7.1 Finding near-feasible solutions

Although as we have shown, stable matching need not exist in instances of the Hospital-Resident problem, where couples are allowed, and even deciding their existence in a given instance is NP-hard. in this section we will show that at least we can always find so called near feasible solutions, although not necessarily in polynomial time.

To be able to always find stable matchings, we relax the capacity constraints of the hospitals, such that each k_h capacity can be violated by at most 2. As we will see, this is enough to guarantee the existence of a stable solution. Since the quotas can be violated now, we call such solutions *near-feasible* stable solutions.

The rest of the section summarizes the results of Nguyen and Vohra [27].

First of all we extend the rankings of the hospitals on the residents to a strict ranking over all the possible coalitions. For a hospital h, if two coalitions contain h, then h ranks these two coalitions according to the worst resident in the coalition that is applying to h (this is only relevant when both members of a couple c apply to h). In the case when this resident is the same, h breaks the ties according to the strict preference of the resident/couple that is contained in both coalitions.

Then, we make a matrix Q whose columns are the possible (d, h) and (c, h, h') coalitions and the rows are the elements of $\mathcal{H} \cup \mathcal{D} \cup \mathcal{C}$. There is a 1 in the corresponding entry of the matrix Q, if the agent/hospital corresponding to the row is a part of the coalition corresponding to the column and a 0 otherwise. The only exception is when the row is some hospital h, and the column is a coalition of the form (c, h, h). In that case the corresponding entry in Q is 2. The bounding vectors are 1 in the residents and couples' rows and k_h in the hospital's rows.

Equivalently, we take the matrix describing the following inequalities:

$$\sum_{d \in D} x_{(d,h)} + \sum_{c \in C} \sum_{h' \neq h} (x_{(c,h,h')} + x_{(c,h',h)}) + \sum_{c \in C} 2x_{(c,h,h)} \leq k_h \qquad \forall h \in \mathcal{H}$$

$$\sum_{h \in \mathcal{H}} x_{(d,h)} \leq 1 \qquad \forall d \in D$$

$$\sum_{h,h' \in \mathcal{H}} x_{(c,h,h')} \leq 1 \qquad \forall c \in C$$

Call this system $\{Qx \leq q\}$. Since we have extended the rankings of the hospitals on the residents to a strict ranking on the coalitions, we now have a strict ordering on the columns for each row of the matrix, therefore it satisfies the conditions of Scarf's lemma, meaning that there always exists a dominating extreme point solution $x \geq 0$.

Definition 7.1. For convenience, we say that h *weakly prefers* resident r_1 to r_2 if $r_1 >_h r_2$ or $r_1 = r_2$.

Lemma 7.2. *Let x be a dominating solution of $\{x : \ Qx \leq q, \ x \geq 0\}$ and suppose x is integral. Then, x is a stable matching.*

Proof. Suppose there is a blocking coalition to x.

If it is of the form (d, h), then d is unassigned or at a worse hospital in x and also, h is unsaturated or has a worse resident r, than d. This means that column (d, h) is not dominated, contradiction.

If the blocking coalition is of the form (c, h, h') with $h \neq h'$, then c is unassigned or at a worse choice. Similarly, h is either unsaturated or has a resident r such that h weakly prefers c^w to r, and for h' it is also the case. By the definition of the rankings, any of the possibilities mean that the column (c, h, h') is not dominated at any row, contradiction.

If the coalition is of the form (c, h, h), then again c is unassigned or at a worse place and h has 2 or more free places; or h has only 1 free space and there is a resident r such that h weakly prefers a member of c to r; or h is full and there are two residents r_1, r_2 at h such that h weakly prefers the worse member of c to both r_1 and r_2; or if the better member of c is already at h, then it is enough that there is a resident r_1 such that the worse member of the couple is preferred to r_1 by h. Again, by the definitions of the rankings, all of the possibilities lead to the fact that column (c, h, h) is not dominated. ☐

Now we state a crucial lemma that guarantees that rounding a fractional dominating solution in a certain way results in a near-feasible stable matching.

Lemma 7.3. *Let $\overline{x}$ be a (fractional) dominating extreme point of the system $\{Qx \leq q,\ x \geq 0\}$ and let x be an integer vector that satisfies the following properties:*

1. *If $\overline{x_i} = 0$ for a coalition i, then $x_i = 0$,*
2. *If $\sum_h \overline{x}_{(d,h)} = 1$ for a single doctor d, then $\sum_h x_{(d,h)} = 1$ and similarly for couples if $\sum_{h,h'} \overline{x}_{(c,h,h')} = 1$, then $\sum_{h,h'} x_{(c,h,h')} = 1$.*

Then, if we let $k'_h = Q_h x$, if $Q_h \overline{x} = k_h$ and $k'_h = \max\{Q_h x, k_h\}$ otherwise, then x corresponds to a stable matching with respect to the new k'_h capacities.

Proof. Suppose that there is a blocking coalition (d, h) to x. If (d, h) was dominated at row d in $\overline{x}$, then each h' with $\overline{x}_{(d,h')} > 0$ is at least as good as h for d, so by condition 1 and 2 d is at a better hospital than h, contradiction. If it was dominated at h, then h was saturated in $\overline{x}$, so by the definition of k'_h, it is also saturated in x. And since h weakly preferred everyone with a positive $\overline{x}$ value to d, by condition 1, this still holds for x, therefore h is saturated with better residents, contradiction. One can check similarly that any coalition involving a couple also cannot block, so by our previous lemma, since x is integral, it is a stable matching.

Also, by the definition of the k'_h capacities, it is trivial that x is a feasible matching with respect to those quotas. ☐

Now we describe the algorithm of Nguyen and Vohra [27] for finding a near feasible stable solution. Its main idea is just to find a fractional dominating solution with Scarf's algorithm and then round it to an integral solution in a way such that the capacities do not change a lot. First, we add an aggregate capacity row

$$\sum_h \Big(\sum_{d \in D} x_{(d,h)} + \sum_{c \in C} \sum_{h' \neq h} (x_{(c,h,h')} + x_{(c,h',h)}) + \sum_{c \in C} 2x_{(c,h,h)}\Big) \leq \sum_h k_h$$

to $\{Qx \leq q\}$.

Theorem 7.4. *(Nguyen and Vohra) [27] Let I be an instance of the* HRC. *Then, there always exists k'_h capacities for each $h \in \mathcal{H}$ satisfying that $|k_h - k'_h| \leq 2$ $\forall h \in \mathcal{H}$ and $\sum_h k_h \leq \sum_h k'_h \leq \sum_h k_h + 4$, such that there is a stable integral matching M with respect to the k'_h capacities and these k'_h capacities and the matching M can be found by the* IR *algorithm.*

Algorithm 7 IR algorithm

Step 0: Let $x := \overline{x}$ be a dominating extreme point of $\{Qx \leq q,\ x \geq 0\}$ found by Scarf's algorithm

Step 1:

if x is integral **then**

STOP

else if There is a hospital row Q_h with $Q_h(\lceil x \rceil - \lfloor x \rfloor) \leq 3$ **then**

Eliminate it and go to Step 2.

else if There are at most 2 non-binding couple or single doctor rows such that each of them contains a fractional variable **then**

Eliminate the aggregate capacity constraint and go to Step 2.

end if

Step 2: Find an extreme point z^* maximizing $\sum_i z_i$ such that:

- $z_i = x_i$ if $x_i = 0$ or 1
- if $Q_d x = 1$, then $Q_d z = 1$, if $Q_c x = 1$, then $Q_c z = 1$ and if $Q_h x = k_h$ then $Q_h z = k_h$
- z satisfies all remaining capacity constraints

$x := z^*$ and go to Step 1.

Proof. First we prove that if the algorithm terminates, then it returns a near feasible stable solution.

Suppose the algorithm terminated with an integral solution x^*. Define the capacities k'_h as in Lemma 7.3. Notice that a hospital's capacity only changes if we eliminate its row: it can only increase if we eliminate its row, and it could only decrease if its row was binding originally, but then it is always binding in the solution until its row is eliminated. Observe that we only eliminate rows satisfying $Q_h(\lceil x \rceil - \lfloor x \rfloor) \leq 3$, and for $Q_h x$ to change through the rounding, x has to have corresponding fractional components, meaning that if $Q_h x$ is an integer value strictly between $Q_h \lfloor x \rfloor$ and $Q_h \lceil x \rceil$, then $Q_h x^*$ can only change by at most 2 and otherwise $Q_h x < k_h$. Hence, we only change the capacity if it increases and it can only increase by less than 3, so it will be at most $k_h + 2$ in the end. This proves that $|k_h - k'_h| \leq 2$ for all $h \in \mathcal{H}$.

Since in each iteration we find an extreme point z^* that maximizes $\sum_i z_i$, $\sum_h k_h$ cannot decrease. Also, it can only increase if we delete the aggregate capacity constraint. But that can only happen, if there are at most 2 doctor or couple rows such that they can be rounded up, so $\sum_h k_h$ can only increase by at most 4.

To show that x^* is stable, we observe that the conditions of lemma 7.3 are satisfied, therefore by lemma 7.3, x^* is stable.

It only remains to prove that the algorithm terminates, which means that if the solution x is not yet integral, then at least one of the elimination conditions have to be satisfied.

Suppose the contrary. If all Q_h rows and the aggregate capacity row have been eliminated, then the remaining matrix is TU, because in each column there is at most one nonzero element, which is a 1. So all extreme points of it are integer valued, which would mean that x is integer, contradiction.

Otherwise, suppose there are some remaining Q_h rows.

We will use the following well known lemma:

Lemma 7.5. *(Lau et. al.[22]) Let x^* be an everywhere strictly positive extreme point of $\{Qx \leq q,\ x \geq 0\}$. Then the number of variables equals the maximum number of linearly independent binding rows (where $Q_i x^* = q_i$) of Q.*

Let us make a matrix Q' by deleting the columns i of Q whose x_i components are already integral. Furthermore let $q' = q - Q''x''$, where Q'' consists of the columns whose x_i components are integer and x'' is the restriction of x to those components. For the rest it obviously still holds that $Q'x' \leq q'$. Furthermore, if x was an extreme point of $\{Qx \leq q,\ x \geq 0\}$, then x' is an extreme point of $\{Q'x \leq q',\ x \geq 0\}$, which is everywhere strictly positive. This results in the following Lemma:

Lemma 7.6. *Let x be an extreme point of $\{Qx \leq q,\ x \geq 0\}$. Then, the number of non-integral components of x, which is the number of columns of Q', is equal to the maximum number of linearly independent rows of Q'.*

Denote by k the number of components of x'. We give each component of x' 1 token. Then, we redistribute the tokens among the rows of Q' the following way.

If the corresponding component is (d, h), then we give $\frac{3}{4}$ token to d and $\frac{1}{4}$ token to h.

If the corresponding component is (c, h, h'), then we credit c with $\frac{1}{2}$ token and h, h' with $\frac{1}{4}$ token each.

Because for each remaining hospital row $Q_h(\lceil x \rceil - \lfloor x \rfloor) \geq 4$, since none of them can be eliminated, each hospital gets at least $4 \cdot \frac{1}{4} = 1$ tokens. Similarly, each binding couple or single doctor row has to get tokens from at least 2 rows, since each component of x' is fractional. Therefore, they also get at least $2 \cdot \frac{1}{2} = 1$ or $2 \cdot \frac{3}{4} = \frac{3}{2}$ tokens respectively. So every binding row apart from the aggregate capacity row obtains at least one token.

Case a) The aggregate capacity row has been eliminated.

In this case, all of the tokens are distributed among the hospital, doctor and couple rows. As we have seen, each such binding row must get at least one token and since x' is an extreme point, there are at least k of them. That can only happen if there are exactly k and each get exactly 1 token.

This means that no single doctor row binds, since then it would get more than one token, as we have seen.

Since every component of x' is strictly positive, it follows that there cannot be columns corresponding to single doctors and all such rows that have a nonzero element must bind. So we can throw out these all zero rows and get a matrix where every row binds.

Therefore, for each remaining row it holds that either

$$\sum_{c \in C} \sum_{h' \neq h} (x_{(c,h,h')} + x_{(c,h',h)}) + \sum_{c \in C} 2x_{(c,h,h)} = k_h$$

or

$$\sum_{h,h' \in \mathcal{H}} x_{(c,h,h')} = 1$$

But then each type of the rows has to sum up to the same, contradicting the fact that the rows must be linearly independent.

Case b): The aggregate capacity row hasn't been eliminated.

Then, we know that at least $k-1$ tokens are distributed among the binding constraints that are not the aggregate capacity row, since there are at least $k-1$ of them (since the aggregate capacity row could bind too). But, since the aggregate capacity row cannot be eliminated, there are at least 3 doctor or couple rows that do not bind, but contain a fractional variable. Therefore, these non-binding additional rows must get at least $\frac{3}{2}$ tokens combined, meaning the number of tokens are more than k, contradiction.

□

7.2 Hardness of f-HRC

In this section we show that even the fractional version of the HRC problem is hard. To be more precise, we show that it is PPAD-complete.

This has important implications, since as we have just seen, there are algorithms for finding near feasible stable matchings that start with finding a fractional stable matching first, but now that we know that fractional solutions are also hard to find, such methods usually will not be efficient.

This section also contains new results, that appeared in my working paper [7].

We also show the hardness of the fractional version of the 3-dimensional stable matching problem, or stable family problem. This problem is simply the restriction of the stable hypergraph matching problem to 3-uniform, 3-partite hypergraph. Then, the three classes of the vertices are usually interpreted as the men, women and dogs respectively, hence its name. The hardness of the integral version, that is deciding whether a stable matching always exists in such an instance was one of the 12 open problems proposed by Knuth [20] and has been answered to be indeed NP-hard by Ng and Hirschberg [26]. The fractional version however has not been studied yet. Since it is a special version of the stable hypergraph matching problem, as we have seen, Scarf's Lemma implies that there is always a fractional solution.

Lemma 7.7. *The SMF is polynomial-time reducible to f-SFP.*

Proof. First consider the matrix we used to reduce SMF to SCARF as shown in Figure 1. First of all we can assume, that each commodity has the same source and sink as proven in [9], so for each commodity we have $v^{i,in}$ and $v^{i,out}$ rows for the same set of vertices. For reasons clarified later, instead of one $v^{i,in}$ and one $v^{i,out}$ column, we make $q(v)$ columns of both of them. We will call these $v_j^{i,out}$, $j = 1,..,q(v)$. The orderings of the rows will be almost the same, we just have to rank the $q(v)$ columns of $v^{i,out}$ and $v^{i,in}$ among each other, but we can rank these arbitrarily, for example let $v_1^{i,out}$ be the best and $v_{q(v)}^{i,out}$ be the worst. Then we add two new rows for each $v_j^{i,in}$ and $v_j^{i,out}$, $j = 1,..,q(v)$ such that there is a 1 in the corresponding column and 0 everywhere else with bounding vector all 1. (so we make two identity matrices). Moreover we add $s^{i,out}$ and $t^{i,in}$ rows (with 1 in the columns corresponding to the appropriate edges and 0 elsewhere), with large enough bounding vectors, such that there cannot be any solution where these rows bind, so no column can be dominated here. It is straightforward to verify that an original dominating solution x can be extended to x' such that it remains a dominating solution and vice versa. In one direction, we only have to fill up the $v_j^{i,out}$ components starting from the best and always filling up the next in the ranking. Then the e^j columns are obviously still dominated. If a $v_j^{i,in}$ row is fully filled up, then it is dominated at the newly added identity matrix. If there is a $v_j^{i,in}$ row that is not fully filled up, then $v^{i,in}$ had to be dominated at row $v^{i,out}$ (otherwise $x_{v^{i,in}}$ would be $q(v)$, so every component would be filled up). Since this row still has to bind and the $v_j^{i,in}$-s are its worst columns, a partially filled up (there can only be at most one) and every empty (by our rule of filling up they have to be worse) columns are dominated here, so all of them are dominated somewhere. The reasoning with the $v_j^{i,out}$-s is symmetric. In the other direction we just have to sum up the $v_j^{i,out}$ and $v_j^{i,in}$ components. If all $v_j^{i,in}$-s are dominated in the identity matrix, then their sum is $q(v)$, so $v^{i,in}$ is dominated at row $v^{i,in}$. If there is one that is dominated at row $v^{i,in}$ or $v^{i,out}$, then similarly $v^{i,in}$ is dominated here too.

Now we make a couple of observations about this matrix:

1. We can swap the rows of the matrix, such that it has an identity matrix at its bottom.

2. The remaining matrix can be seen as the incidence matrix of a 3-uniform 3-partite hypergraph, because each column has exactly three 1-s. If it is of type e^i, $e = uv$, then the 1-s are at row e, row $v^{i,in}$ and row $u^{i,out}$. If it is of type $v_j^{i,in/out}$, then there is a 1 at $v^{i,in}$-s row, $v^{i,out}$'s row, and one of the newly added $v_j^{i,in/out}$ rows, that is not moved to the bottom identity matrix. Furthermore, we can see that if we partition the rows to $E \cup \{$ *newly added rows* $\}$, $v^{i,in}$ rows and $v^{i,out}$ rows, then each column (hyperedge) contains exactly one vertex from each class. It is important to note that here the $s^{i,out}$ and $t^{i,in}$ rows are counted among the $v^{i,in}$ and $v^{i,out}$ rows respectively, not among the newly added rows.

3. Since the stable multicommodity flow problem is still PPAD-hard if all edge capacities are 0 or 1, and each added $v_j^{i,in/out}$ row has upper bound 1, we

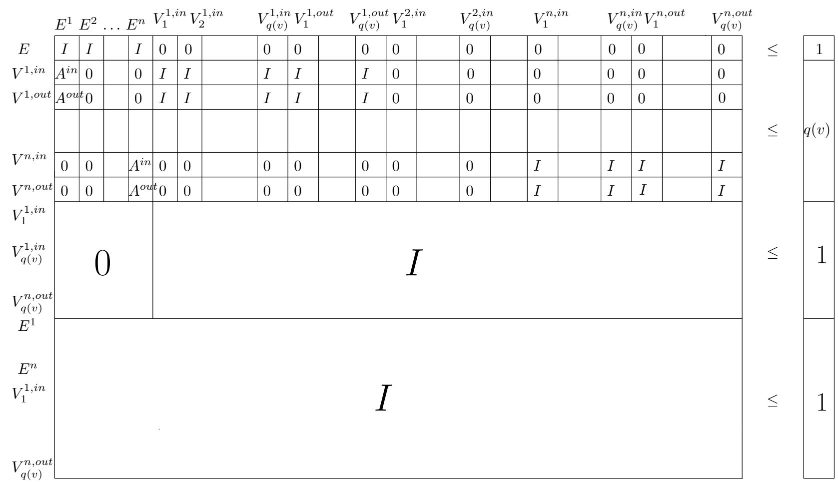

Figure 3: The new matrix for the reductions of lemma 7.7 and 7.9

can assume that the bottom identity matrix has the all 1 bounding vector, so in our hypergraph all hyperedges have unit capacity, therefore we have a standard FRACTIONAL HYPERGRAPH MATCHING problem.

Let the preferences of the vertices on the hyperedges be the same as the corresponding row's preferences on the nonzero columns. What remains to be shown is, that the stable matchings of the 3-partite hypergraph correspond to the stable solutions of the SCARF instance.

Let us suppose that we have a stable fractional matching, but the corresponding vector is not a dominating one. Then there has to be a column that is not dominated at any row. But this means, that each of the three rows that could dominate are either nonbinding, or there are less preferred columns with nonzero value. So the edge corresponding to the column is a blocking edge, contradiction.

However, we are still not completely done, since in a f-SFP instance we need that every vertex has capacity 1 too. We will construct another 3-partite hypergraph from the previous one in the following way: Let the three classes be X, U and V with $X = E \cup \{$ *newly added rows*$\}$. Take each vertex v of the hypergraph that has capacity $q(v) > 1$ (they can only be from U or V) and make $q(v)$ copies of that vertex, that will all have capacity 1. (We can assume that $q(v)$ is integer). Then for each (x, u, v) hyperedge we make (x, u_i, v_j) hyperedges for all $i = 1, .., q(u)$, $j = 1, ..., q(v)$. Let the u_i vertices' ranking of the hyperedges be the same as the u vertices' ranking with the extension that for a given (x, u, v) hyperedge, $(x, u_i, v_1) >_{u_i} (x, u_i, v_2)$.... Similarly for the v_j-s. The x vertices' preferences are extended in such a way that if $(x, u, v) >_x (x, u', v')$, then $(x, u_i, v_j) >_x (x, u'_k, v'_l)$ for all i, j, k, l. Furthermore, we rank the (u_i, v_j) pairs in an arbitrary way. For example, $(x, u_1, v_1) >_x (x, u_1, v_2) >_x (x, u_2, v_1) >_x (x, u_1, v_3)$....

Now let us suppose we have a stable fractional matching in this new graph. We obtain a matching for the original hypergraph by summing on the hyperedges corresponding to the same original one. Since each of those contain the same x vertex with capacity 1, this new matching will be feasible too. Now let us suppose that there is a blocking hyperedge (x, u, v). So x is unsaturated or prefers (u, v) to some (u', v'), where he is matched with some positive value. This means that x is unsaturated or there is an (u'_k, v'_l) where x is matched with positive value, that it prefers less than any (u_i, v_j). Similarly u is unsaturated or there is a v', such that $(x', u, v') <_u (x, u, v)$, and (x', u, v') has positive value in the matching. In the first case, there has to be an i, such that u_i is unsaturated, and in the second case if $(x, u, v) >_u (x', u, v')$, then $(x, u_i, v_j) >_{u_i} (x', u_i, v'_k)$ for any i, j, k, so there is at least one (i, k) pair that is both worse and has positive value. Using the same argument for v, we get that there is a pair i, j, such that (x, u_i, v_j) is a blocking edge, contradiction.

Since this hypergraph is 3-partite and had all one edge and vertex capacities, it corresponds to a stable family instance, so we reduced the problem to f-SFP. □

Remark 7.8. We have to be careful, because making $q(v)$ copies of vertices is not necessarily polynomial in the input if the $q(v)$-s can be arbitrarily large, but from Lemma 6.12 and the fact that PPAD-completeness holds for 0/1 capacities we we can assume that each $q(v)$ is at most 4.

Lemma 7.9. *The SMF is polynomially reducible to f-HRC.*

Proof. First we reduce it to the 3-partite stable hypergraph problem with vertex capacities as before (so here we do not make $q(v)$ copies of the vertices in U and V). Then we observe that, since we can assume that the aggregate capacities of the stable multicommodity flow were all 0 or 1, and the newly added $v_j^{i,in/out}$ rows have capacity 1 too, all vertices in the $X = E \cup \{$ *newly added rows*$\}$ class of our hypergraph have capacity 1. Now let the elements of the X class be the couples, and the hospitals will consists of the union of the other two classes' vertices. The ranking of the couples on the pairs of hospitals will be the same as the corresponding vertex's ranking on the hyperedges containing it. (The other hospital pairs will be considered unacceptable to the couple). Now observe that any two vertices with one from X uniquely determines a hyperedge, if there is any, because e $(e = uv)$ and $u^{i,out}$ uniquely determines the last vertex, that is $v^{i,in}$, and vice versa, so the ranking on the hyperedges containing a vertex v gives a unique ranking on the vertices of X adjacent to v. So the hospital's ranking on the couples will be the following: $c > c'$ for h, if $(c, h, *) >_h (c', h, *)$ in the hypergraph, where $(c, h, *)$ denotes the unique hyperedge containing c and h. Furthermore let the hospital's capacities be the $q(v)$ capacities of the vertices.

Now let us suppose we have a stable matching in the f-HRC instance. Assume that there is a blocking hyperedge (x, u, v) in the 3-partite stable hypergraph problem. Then x prefers (x, u, v) to some (x, u', v'), where it is matched with some positive value meaning $(u, v) >_x (u', v')$ or it is unsaturated. u is either unsaturated, or there is a hyperedge (x', u, v'), that it prefers less (here v could be v', but $x \neq x'$ because of our observation). But $(x, u, v) >_u (x', u, v')$ means that in the hospital's ranking, u prefers (the corresponding member of) x more than x'. Similarly either v is unsaturated, there is an x' at v, that it prefers less than x. So (x, u, v) would form a blocking coalition, contradiction. □

As a consequence, we have the following theorems:

Theorem 7.10. *The f-HRC problem is PPAD-complete, even if each hospital only receives applications from men or women.*

Theorem 7.11. *The f-SFP problem is PPAD-complete.*

Conclusions

In this thesis we gave a short introduction to some of the most important generalizations of the stable matching problem. Of course, with the countless new results that have been discovered since the introduction of the field, a complete review of all the aspect of the problem would be impossible.

Here, we focused on a complexity theoretic viewpoint and analyzed which kind of generalizations make the problem hard, and which are still efficiently solvable. We have seen many from both cases: on one side the clever algorithms of Irving and Fleiner for the stable roommates and stable flow problems respectively, while on the other side the stable hypergraph matching problem and stable multicommodity flow problem that are NP-hard even with very severe restrictions and they even stay hard if we only want to find a fractional solution. Also, interestingly, just allowing ties or couples make the problem intractable.

Whilst most of the problems we introduced here have been thoroughly studied, the number of open questions are still uncountable. One such intriguing open question is whether there is an efficient algorithm for the stable hypergraph matching problem, if the underlying hypergraph is Totally Unimodular (which means its incidence matrix is TU). Then, by Scarf's Lemma, we know a stable solution always exists. Or a similar question is whether this holds for normal hypergraphs and unit capacities. Interestingly, solving the latter problem would also answer a long standing open question about graph kernels, namely whether it is possible to find a kernel in a superoriented perfect graph efficiently.

References

[1] Ron Aharoni and Ron Holzman. "Fractional Kernels in Digraphs". In: *J. Comb. Theory Ser. B* 73.1 (May 1998), pp. 1–6. ISSN: 0095-8956. DOI: 10.1006/jctb.1997.1731. URL: https://doi.org/10.1006/jctb.1997.1731.

[2] Mourad Baıou and Michel Balinski. "The stable allocation (or ordinal transportation) problem". In: *Mathematics of Operations Research* 27.3 (2002), pp. 485–503.

[3] Piotr Berman, Marek Karpinski, and Alexander Scott. *Approximation hardness of short symmetric instances of MAX-3SAT*. Tech. rep. 2004.

[4] Péter Biró, David F Manlove, and Iain McBride. "The hospitals/residents problem with couples: Complexity and integer programming models". In: *International Symposium on Experimental Algorithms*. Springer. 2014, pp. 10–21.

[5] Katarına Cechlárová and Tamás Fleiner. "On a generalization of the stable roommates problem". In: *ACM Transactions on Algorithms (TALG)* 1.1 (2005), pp. 143–156.

[6] Jiehua Chen, Piotr Skowron, and Manuel Sorge. "Matchings under preferences: Strength of stability and tradeoffs". In: *ACM Transactions on Economics and Computation* 9.4 (2021), pp. 1–55.

[7] Gergely Csáji. "On the complexity of Stable Hypergraph Matching, Stable Multicommodity Flow and related problems". In: (2021).

[8] Ágnes Cseh, Robert W Irving, and David F Manlove. "The stable roommates problem with short lists". In: *Theory of Computing Systems* 63.1 (2019), pp. 128–149.

[9] Ágnes Cseh and Jannik Mattuschke. "New and Simple Algorithms for Stable Flow Problems". In: *Algorithmica* 81 (Jan. 2019). DOI: 10.1007/s00453-018-00544-7.

[10] Brian C Dean and Siddharth Munshi. "Faster algorithms for stable allocation problems". In: *Algorithmica* 58.1 (2010), pp. 59–81.

[11] Tamás Fleiner. "On stable matchings and flows". In: *International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer. 2010, pp. 51–62.

[12] David Gale and Lloyd S Shapley. "College admissions and the stability of marriage". In: *The American Mathematical Monthly* 69.1 (1962), pp. 9–15.

[13] Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*. MIT press, 1989.

[14] Joseph Douglas Horton and Kyriakos Kilakos. "Minimum edge dominating sets". In: *SIAM Journal on Discrete Mathematics* 6.3 (1993), pp. 375–387.

[15] Robert W Irving. "An efficient algorithm for the "stable roommates" problem". In: *Journal of Algorithms* 6.4 (1985), pp. 577–595.

[16] Takashi Ishizuka and Naoyuki Kamiyama. "On the Complexity of Stable Fractional Hypergraph Matching". In: *ISAAC*. 2018.

[17] Shiva Kintali et al. "Reducibility among fractional stability problems". In: *SIAM Journal on Computing* 42.6 (2013), pp. 2063–2113.

[18] Tamás Király and Julia Pap. "Stable Multicommodity Flows". In: *Algorithms* 6 (Mar. 2013), pp. 161–168. DOI: 10.3390/a6010161.

[19] Zoltán Király. "Better and simpler approximation algorithms for the stable marriage problem". In: *Algorithmica* 60.1 (2011), pp. 3–20.

[20] DE Knuth. "Mariages stables et leurs relations avec d'autres problemes combinatoires, Les Presses de l'Universite de Montreal (1976), English translation by M. Goldstein: Stable Marriage and its Relation to other Combinatorial Problems". In: *CRM Proceedings and Lecture Notes*. Vol. 10. 1997.

[21] Chi-Kit Lam and C Gregory Plaxton. "A (1+ 1/e)-approximation algorithm for maximum stable matching with one-sided ties and incomplete lists". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2823–2840.

[22] Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*. Vol. 46. Cambridge University Press, 2011.

[23] David Manlove. *Algorithmics of matching under preferences*. Vol. 2. World Scientific, 2013.

[24] David F Manlove et al. "Hard variants of stable marriage". In: *Theoretical Computer Science* 276.1-2 (2002), pp. 261–279.

[25] Eric McDermid. "A 3/2-approximation algorithm for general stable marriage". In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2009, pp. 689–700.

[26] Cheng Ng and Daniel S. Hirschberg. "Three-Dimensional Stabl Matching Problems". In: *SIAM Journal on Discrete Mathematics* 4.2 (1991), pp. 245–252. DOI: 10.1137/0404023. eprint: https://doi.org/10.1137/0404023. URL: https://doi.org/10.1137/0404023.

[27] Thành Nguyen and Rakesh Vohra. "Near-Feasible Stable Matchings with Couples". In: *American Economic Review* 108.11 (Nov. 2018), pp. 3154–69. DOI: 10.1257/aer.20141188. URL: https://www.aeaweb.org/articles?id=10.1257/aer.20141188.

[28] Christos H Papadimitriou. "On the complexity of the parity argument and other inefficient proofs of existence". In: *Journal of Computer and system Sciences* 48.3 (1994), pp. 498–532.

[29] Adèle Pass-Lanneau, Ayumi Igarashi, and Frédéric Meunier. "Perfect graphs with polynomially computable kernels". In: *Discrete Applied Mathematics* 272 (2020), pp. 69–74.

[30] Eytan Ronn. "On the complexity of stable matchings with and without ties". PhD thesis. Yale University, 1986.

[31] Alvin E Roth, Uriel G Rothblum, and John H Vande Vate. "Stable matchings, optimal assignments, and linear programming". In: *Mathematics of operations research* 18.4 (1993), pp. 803–828.

[32] Uriel G Rothblum. "Characterization of stable matchings as extreme points of a polytope". In: *Mathematical Programming* 54.1 (1992), pp. 57–67.

[33] Herbert E Scarf. “The core of an N person game”. In: *Econometrica: Journal of the Econometric Society* (1967), pp. 50–69.

[34] Jimmy JM Tan. “A maximum stable matching for the roommates problem”. In: *BIT Numerical Mathematics* 30.4 (1990), pp. 631–640.

[35] Jimmy JM Tan. “A necessary and sufficient condition for the existence of a complete stable matching”. In: *Journal of Algorithms* 12.1 (1991), pp. 154–178.

[36] John H Vande Vate. “Linear programming brings marital bliss”. In: *Operations Research Letters* 8.3 (1989), pp. 147–153.