MŰEGYETEM 1782

BME

Matematikai és Számítástudományok Doktori Iskola

The stable matching problem and its generalizations: an algorithmic and game theoretical approach

PhD Thesis

Péter Biró

supervisor: Tamás Fleiner

Budapest, September 2007

Declaration

Ezen értekezés bírálatai és a védésről készült jegyzőkönyv a későbbiekben a Budapesti Műszaki és Gazdaságtudományi Egyetem Természettudományi Karának Dékáni Hivatalában elérhető.

Alulírott, Biró Péter kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részletet, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a források megadásával megjelöltem.

Budapest, 2007. szeptember 13.

....
Biró Péter

Abstract

We say that a market situation is stable in a general sense, if there is no set of agents such that all of them are interested in creating a new cooperation (after breaking their other eventual cooperations). As a special question, Gale and Shapley [48] introduced and studied the problem of stable marriages. Here a matching, that corresponds to a set of marriages, is stable, if there exists no man and woman, who would both prefer to marry each other (after leaving their eventual partners). Gale and Shapley described a natural algorithm that finds a stable matching for the marriage, so when the graph, that models the possible partnerships, is bipartite.

The stable matching problem and its generalizations have been extensively studied in combinatorial optimization and game theory. The main reason is that these models are useful to describe economic and social situations. Moreover, as real applications, centralized matching programs have been established in several areas. In this thesis, beside describing the basic models and studying some special problems, we also present some important applications.

The stable matching problem for bipartite graphs is often studied in the context of stable marriages. Actually, whenever we use the marriages as an example for the above problem, we must have at least three assumptions: payment (dower) is not allowed, only men and women can marry each other, and everybody can have at most one partner. The most important generalizations of the stable matching models can be obtained by relaxing these conditions.

If we allow transfers between the agents in the stable marriage problem, then we get a problem, that corresponds to the assignment game, defined by Shapley and Shubik [102]. Stable matching problems with or without sidepayments are equivalent to the problem of finding a core element of the corresponding so-called TU or NTU-games, (i.e. cooperative games with or without transferable utility,) respectively. If the size of the basic coalitions may be more than 2, then instead of graphs, we can model these problems with hypergraphs.

In Chapter 1, we show that the nonemptyness of the core in bipartite matching games is a consequence of a game theoretical lemma of Scarf [97] and a theorem of Lovász [70] on normal hypergraphs. Besides, we describe some other interesting connections between graph theory and game theory. We note for example that the result of Shapley and Shubik [102] is an easy consequence of a theorem of Egerváry [39] about maximum weight matching in bipartite graphs.

If we study one-sided markets instead of two-sided ones, then we get the stable roommates problem and we model it with nonbipartite graphs. As Gale and Shapley [48] showed, a stable matching may not always exist. Irving [56] constructed the first polynomial algorithm that finds a stable matching if there exists one in the given

instance. Later, Tan [104] showed that a so-called stable half-matching always exists for the roommates problems.

In Chapter 2, we describe two special studies about the stable roommates problem. The first one [21] is about the dynamics of the stable marriage and roommates problems. In this work, we used the algorithm of Roth and Vande Vate [93] in the bipartite case, and a similar algorithm of Tan and Hsueh [105] in the one-sided case to model the dynamics of the matching market. We analyze the properties of the solutions obtained by these incremental algorithms. In the second paper [5], we study the complexity of the problem of "almost stable matchings", that is to find a matching for a roommates problem with the fewest number of blocking pairs.

If an agent can be involved in several partnerships at the same time (according to a quota), then we get the stable b-matching problem. This problem is also solvable by the Gale-Shapley algorithm, in fact, the original goal of their paper was to study this so-called college admission problem. Their algorithm is used currently for many two-sided markets. Moreover, Roth [83] discovered that the very same method had already been implemented in 1952 in the National Intern Matching Program [77].

In Chapter 3, we study the higher education admission process in Hungary. Here, the model is a bit more complicated, since students may have equal scores, so ties must be handled. The presented results have been published in [18]. In this chapter we also introduce and study the so-called stable allocation problem, that is the stable matching problem with bounds on the vertices and capacities on the edges. Here, we generalize the inductive algorithm of Baïou and Balinski [14] for nonbipartite graphs.

Finally, we study the problem of exchange of indivisible goods. Shapley and Scarf [103] introduced the houseswapping game as a basic NTU-game. The TU-version is called permutation game. Tijs *et at.* [106] showed that these games also have a nonempty core.

In Chapter 4, we focus on the problem, in which the lengths of the possible trading cycles are bounded. Note that if only pairwise exchanges are allowed, then we get again the stable matching problem (with or without sidepayments). For larger bounds, most of the problems become NP-hard. Complexity questions about the so-called stable exchange problems, that are related to the above NTU-games, are studied on the basis of paper [19]. We present a result published in [20] about the inapproximability of a special stable exchange problem. Finally, we consider the maximum weight directed cycle packing problem, that is related to the problem of finding a core element in a permutation game with restrictions. The presented results come from [22].

We note, that the problem of finding an "optimal" exchange with length restrictions is relevant also because it has a very special application: in some already established kidney exchange programs exactly the two- and three-way exchanges are allowed (see [78] for example).

Contents

Acknowledgements

I would like to thank Tamás Fleiner for his supervision. I contacted him in 2001, because the description of his subject for master's thesis on "the stable matching problem and their applications" was really attractive, since I was interested both in combinatorial optimization and in economic science. This subject was a perfect choice and Tamás has been a fantastic supervisor. During 6 years, he has been a constant source of advice and encouragement. Beside guiding me in my research and reading carefully my write-ups, he also supported me to find my special interests and to get involved in collaborations with several coauthors. I acknowledge him for all of his help and patience.

I am grateful to András Recski, who introduced me the beauty of graph theory in 1997, and as the head of the Department of Computer Science and Information Theory, he offered me the privilege to become a teaching assistant there in the following year. I would like to express my gratitude to the Department for every support and for providing such a friendly environment to work in.

I would like to thank Tamás Solymosi for introducing me to cooperative game theory, for supervising my master's thesis in economics and for the useful discussions.

I thank my coauthors: David J. Abraham, Katarína Cechlárová, David F. Manlove and Romeo Rizzi. Special thanks to Katarína for inviting me twice to Kosice, our collaborations were really exciting and fruitful. I am grateful to David F. Manlove for involving me in my first joint work, for his advice and for his precise corrections.

I would like to thank Elena Inarra for inviting me to Bilbao in 2006, and for the enjoyable discussion we had there.

Finally, I acknowledge my referees, Tamás Király and Tamás Solymosi for their useful comments on the first draft of this thesis. Any shortage or mistake in the final version is entirely my responsibility.

I would like to thank my parents and my wife, Márti for their love and support.

Chapter 1

Core of cooperative games

Introduction

Here, we survey the basic literature about the core of the cooperative NTU- and TU-games. Our final goal is to describe and to study the stable matching problem with the presented general game theoretical notions.

1.1 NTU-games

We recall the definition of *n-person games with nontransferable utility* (NTU-game for short) that can also be referred to as *games without side payments*.

Definition 1.1.1. *An* NTU-game *is given by a pair (N, V), where $N = \{1, 2, \ldots, n\}$ is the set of players and V is a mapping of a set of feasible utility vectors, a subset $V(S)$ of $\mathbb{R}^S$ to each coalition of players, $S \subseteq N$, such that $V(\emptyset) = \emptyset$, and for all $S \subseteq N$, $S \neq \emptyset$:*

a) $V(S)$ is a closed subset of $\mathbb{R}^S$

b) $V(S)$ is comprehensive, i.e. if $u^S \in V(S)$ and $\tilde{u}^S \leq u^S$ then $\tilde{u}^S \in V(S)$

c) The set of vectors in $V(S)$ in which each player in S receives no less than the maximum that he can obtain by himself is a nonempty, bounded set.

One of the most important solution concept is the core.

Definition 1.1.2. *A utility vector $u^N \in V(N)$ is in the core of the game, if there exists no coalition $S \subseteq N$ with a feasible utility vector $\tilde{u}^S \in V(S)$ such that $u_i^N < \tilde{u}_i^S$ for every player $i \in S$. Such a coalition is called a blocking coalition.*

An NTU-game (N, V) is *superadditive* if $V(S) \times V(T) \subseteq V(S \cup T)$ for every pair of disjoint coalitions S and T. In what follows, we restrict our attention to superadditive games.

1.1.1 Partitioning game, redundant and essential coalitions

Partitioning games are special superadditive games. Given a set of *basic coalitions* $\mathcal{B} \subseteq 2^N$, that contain all singletons (i.e. every single player has the right not to cooperate with the others), a partitioning game $(N, V, \mathcal{B})$ is defined as follows: if $\Pi_{\mathcal{B}}(S)$ denotes the set of partitions of S into basic coalitions, then $V(S)$ can be generated as:

$$V(S) = \{u^S \in \mathbb{R}^S | \exists \pi = \{B_1, B_2, \dots, B_k\} \in \Pi_{\mathcal{B}}(S) : u^S \in V(B_1) \times V(B_2) \times \dots \times V(B_k)\}$$

This means that u^S is a feasible utility vector of S if there exist a partition π of S into basic coalitions, such that each utility vector $u^S|_{B_i}$ can be obtained as a feasible utility vector for the basic coalition B_i of π.

Given an NTU-game (N, V), let $U(S)$ be the set of *Pareto optimal* utility vectors of the coalition S, i.e. $u^S \in U(S)$ if there exists no $\tilde{u}^S \in V(S)$, where $u^S \neq \tilde{u}^S$ and $u^S \leq \tilde{u}^S$.

A utility vector $u^S \in V(S)$ is *separable* if there exist a proper partition π of S into subcoalitions $S_1, S_2, \dots, S_k$ such that $u^S|_{S_i}$ is in $V(S_i)$ for every $S_i \in \pi$. A utility vector that is non-separable, Pareto-optimal and in which each player receives no less than the maximum that he can obtain by himself is called an *efficient* vector. A coalition S is *essential* if $V(S)$ contain an efficient utility vector. In other words, a coalition S is essential, if its members can obtain an efficient utility vector that is not achievable independently by its subcoalitions. The set of essential coalitions is denoted by $\mathcal{E}(N, V)$.

We say that a coalition S is not *relevant* if for every utility vector $u^S \in V(S)$ there exists a proper subcoalition $T \subset S$ such that $u^S|_T$ is in $V(T)$. The set of relevant coalitions is denoted by $\mathcal{R}(N, V)$. The idea behind this notion is that if a non-relevant coalition S is blocking with a utility vector u^S, then some of its subcoalitions, T_1 must be also blocking with utility vector $u^{T_1} = u^S|_{T_1}$. Moreover, if T_1 is not relevant or u^{T_1} is separable, then we can find another coalition $T_2 \subset T_1$, such that $u^{T_2} = u^{T_1}|_{T_2} = u^S|_{T_2}$, an so on. By this, there must be a relevant coalition $T_i \subset S$, that is blocking with a non-separable vector $u^{T_i} = u^S|_{T_i}$. This observation implies the following Proposition:

Proposition 1.1.3. *A utility vector $u^N \in V(N)$ is in the core if and only if it is not blocked by any relevant coalition with an efficient utility vector.*

Obviously, if a coalition is not essential, then it cannot be relevant either. In a partitioning game, the set of essential coalitions must be a subset of the basic coalitions by definition.

Proposition 1.1.4. *For every partitioning game $(N, V, \mathcal{B})$, $\mathcal{R}(N, V, \mathcal{B}) \subseteq \mathcal{E}(N, V, \mathcal{B}) \subseteq \mathcal{B}$ holds.*

1.1.2 Preferences over the set of outcomes

Scarf observed in [97] that the previously introduced notions are purely ordinal in character: they are invariant under a continuous monotonic transformation of the utility function of any individual. Hence, without loss of generality, we may assume that $U^{\{i\}} = \{0\}$ for every singleton, and all the efficient utility vectors are nonnegative. Moreover, the discussion can be carried out on an abstract level with the outcomes for each individual represented by arbitrary ordered sets.

In order to obtain a particular non-separable vector $u^{S,k}$ in $U(S)$, the members of S have to perform a common activity, say $\mathfrak{a}^{S,k}$. We denote by $\mathcal{A}^S$ the set of activities that yield efficient utility vectors in $U(S)$. The preference of a player over the possible activities in which he can be involved in is determined by the utilities that he obtains by the activities. Formally, we have $\mathfrak{a}^{S,k} \leq_i \mathfrak{a}^{T,l} \iff u_i^{S,k} \leq u_i^{T,l}$ for two activities $\mathfrak{a}^{S,k}$ and $\mathfrak{a}^{T,l}$, where $i \in S$ and $i \in T$.

Considering an efficient utility vector u^N of the grandcoalition N, the non-separability implies that it corresponds to a common activity $\mathfrak{a}^N$ of the entire set of players. Otherwise, if u^N is separable, then u^N can be obtained as a direct sum of independent efficient utility vectors of essential subcoalitions that form a partition of the grandcoalition. This can be considered as a set of independent activities of the subcoalitions. An *outcome of the game*, denoted by X then can be regarded as a partition π of the players and a set of activities $\mathcal{A}^\pi$ performed independently by the coalitions in π, so $X = (\pi, \mathcal{A}^\pi)$. An outcome X is judged by a player i according the activity he is involved in, denoted by $\mathfrak{a}_i(X)$. An outcome is in the core of the game, or in other words, it is *stable* if there exist no blocking coalition S and an activity $\mathfrak{a}^{S,l}$ that is strictly preferred by all of its members, so $\mathfrak{a}^{S,l} >_i \mathfrak{a}_i(X)$ for all $i \in S$. (This is equivalent to the blocking condition $u_i^N < \tilde{u}_i^S$, if the outcome X corresponds to the utility vector u^N.)

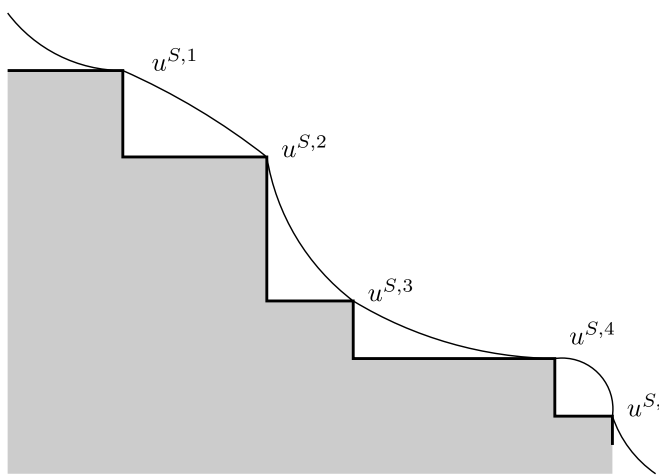


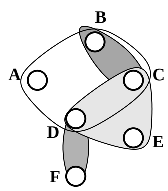
Figure 1.1: Approximation with finite number of efficient utility vectors.

An NTU-game is called *finitely generated* if for every essential coalition S, $U(S)$ contains a finite number of vectors. Here, the preference order of a player over the set of activities, in which he can be involved, can be represented by preference lists. As

Scarf observed in [97] and [96], a general NTU-game can be approximated by a finitely generated NTU-game (see an illustration in Figure 1.1), thus they have similar basic properties. Here, we will not discuss this question in details.

Suppose that in an NTU-game for every essential coalition S, $U(S)$ contains one single vector u^S. In this case, the outcome of the game is just a partition, since each essential coalition has one single activity to perform. So here, instead of activities, each player has a preference order over the essential coalitions in which he can be a member. These games are called *coalition formation games* (CFG for short), and an outcome that is in its core is called a *core-partition*.

Example 1.1.5. *In this example, we have 6 players: A, B, C, D, E and F, and 4 possible basic coalitions corresponding to activities, such that all members are interested to participate in the activity. The first activity, b (bridge) can be played by A, B, C and D, the second one, p (poker) can be played by C, D and E. Finally, B can play chess with C (c_1) and D can play chess with F (c_2).*



Activities	Participants	Players	Preference lists
b :	$\{A, B, C, D\}$	B	$[b, c_1]$
p :	$\{C, D, E\}$	C	$[p, b, c_1]$
c_1 :	$\{B, C\}$	D	$[b, p, c_2]$
c_2 :	$\{D, F\}$		

Here, $\{p, \{A\}, \{B\}, \{F\}\}$ is a core-partition, since b is not blocking because C prefers his present coalition p to b, similarly, c_1 is not blocking because C prefers playing poker with D and E to playing chess with B, and c_2 is not blocking because D also prefers playing poker to playing chess with F. One can easily check that $\{b, \{E\}, \{F\}\}$ is also a core-partition, but the partition $\{c_1, c_2, \{A\}, \{E\}\}$ is not stable, since p and b are blocking coalitions.

1.1.3 Fractional core by Scarf's lemma

First, we present Scarf's lemma [97] and we apply it to the original settings. By this, we introduce the notion of fractional core and we study some consequences. The following version of the Lemma is due to Aharoni and Fleiner [7], here $[n]$ denotes the integers $1, 2, \ldots, n$.

Theorem 1.1.6 (Scarf, 1967)**.** *Let n, m be positive integers, and b be a vector in $\mathbb{R}^n_+$. Also let $A = (a_{i,j}), C = (c_{i,j})$ be matrices of dimension $n \times (n+m)$, satisfying the following three properties: the first n columns of A form an $n \times n$ identity matrix (i.e. $a_{i,j} = \delta_{i,j}$ for $i, j \in [n]$), the set $\{x \in \mathbb{R}^{n+m}_+ : Ax = b\}$ is bounded, and $c_{i,i} < c_{i,k} < c_{i,j}$ for any $i \in [n], i \neq j \in [n]$ and $k \in [n+m] \setminus [n]$.*

Then there is a nonnegative vector x in $\mathbb{R}_+^{n+m}$ such that $Ax = b$ and the columns of C that correspond to supp(x) form a dominating set, that is, for any column $i \in [n+m]$ there is a row $k \in [n]$ of C such that $c_{k,i} \leq c_{k,j}$ for any $j \in$ supp(x).

Let the columns of A and C correspond to the efficient utility vectors (or equivalently to some activities) of the essential coalitions in a finitely generated NTU-game as follows. If the k-th columns of A and C correspond to the utility vector $u^{S,l}$, then let $a_{i,k}$ be 1 if $i \in S$ and 0 otherwise, (so the k-th column of A is the *membership vector* of coalition S). Furthermore, let $c_{i,k} = u_i^{S,l}$ if $i \in S$ and $c_{i,k} = M$ otherwise, where M is a sufficiently large number. We set $c_{i,i} = u_i^{\{i\}} = 0$ and $c_{i,j} = 2M$ if $i \neq j \leq n$. Finally, let $b = 1^N$. By applying Scarf's lemma for this setting, we obtain a solution x that we call a *fractional core element* of the game.

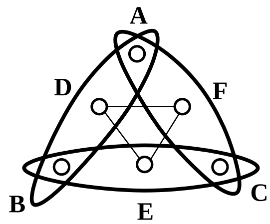
What is the meaning of a fractional core element? Let us suppose first, that a fractional core element x is integer, so $x_i \in \{0, 1\}$ for all i. In this case we show that x gives a utility vector u^N that is in the core of the game. Let u^N be the utility vector of N received by summing up those independent essential utility vectors for which $x(u^{S,k}) = 1$, then u^N is obviously in $V(N)$ by superadditivity. To prove that u^N must be in the core of the game, let $u^{S,k}$ be an essential utility vector, with $x(u^{S,k}) = 0$. By the statement of Scarf's lemma, there must be a player i and an essential utility vector $u^{T,l}$, such that $i \in T$, $x(u^{T,l}) = 1$ and $u_i^{S,k} \leq u_i^{T,l}$, so S cannot be a blocking coalition with the efficient utility vector $u^{S,k}$.

In other words, the $Ax = 1^N$ condition of the solution says that x gives a partition π of N and a set of activities $\mathcal{A}^\pi$ that are performed (i.e. $\mathfrak{a}^{S,k}$ is *performed*, so $\mathfrak{a}^{S,k} \in \mathcal{A}^\pi$ if $x(u^{S,k}) = 1$, thus S is a coalition in the partition π). Let $X = (\pi, \mathcal{A}^\pi)$ be the corresponding outcome, and let $\mathfrak{a}^{S,k}$ be a not performed activity, (i.e. $\mathfrak{a}^{S,k} \notin \mathcal{A}^\pi$). Then, by Scarf's lemma there must be a player i of S for which the performed activity $\mathfrak{a}_i(X)$, he is involved in is not worse than $\mathfrak{a}^{S,k}$, so $\mathfrak{a}^{S,k} \leq_i \mathfrak{a}_i(X)$, thus S cannot be a blocking coalition with the activity $\mathfrak{a}^{S,k}$.

In the non-integer case, we shall regard $x(u^{S,k})$ as the *intensity* the activity $\mathfrak{a}^{S,k}$ is performed with, by coalition S. The $Ax = 1^N$ condition means, that each player participates in activities with total intensity 1, including maybe the activity that this single player performs alone. The domination condition says, that for each activity, that is not performed with intensity 1, there exists a member of the coalition, who is not interested in increasing the intensity of this activity, since he is satisfied by some other preferred activities that fill his remaining capacity of intensity. Formally, if $x(u^{S,k}) < 1$, then there must be a player i in S such that $\sum_{a^{T,l} \geq_i a^{S,k}} x(u^{T,l}) = 1$.

In Example 1.1.5, $x(p) = \frac{1}{3}$, $x(b) = \frac{2}{3}$ is a fractional core element, since for each activity there is one player who is not interested in increasing the intensity of that activity. Below, we show another example, where the core of the game is empty, and which has exactly one fractional core element.

Example 1.1.7. *In this example we have 6 players, with 6 possible common activities with the following preferences:*



Activities	Participants	Players	Preference lists
p_1 :	$\{A, D, B\}$	A	$[p_1, p_3]$
p_2 :	$\{B, E, C\}$	B	$[p_2, p_1]$
p_3 :	$\{C, F, A\}$	C	$[p_3, p_2]$
c_1 :	$\{D, E\}$	D	$[p_1, c_1, c_2]$
c_2 :	$\{E, F\}$	E	$[p_2, c_2, c_3]$
c_3 :	$\{F, D\}$	F	$[p_3, c_3, c_1]$

It can be verified, that here the only fractional core element is $x(p_1) = x(p_2) = x(p_3) = \frac{1}{2}$ and $x(c_1) = x(c_2) = x(c_3) = \frac{1}{4}$. One can easily generalize this construction having $3n$ players, and intensity 2^{-n} of some activities in the unique fractional core element.

1.1.4 Breaking the ties in case of indifferences

In a finitely generated NTU-game it can happen that a player i is *indifferent* between some activities $\mathfrak{a}^{S,k}$ and $\mathfrak{a}^{T,l}$, since he gets the same utilities (so $\mathfrak{a}^{S,k} \sim_i \mathfrak{a}^{T,l}$ if $u_i^{S,k} = u_i^{T,l}$). In this case so-called *ties* occur in the preference lists. Here, we recall the three main core-, or stability-concepts:

- An outcome X is in the *weak core*, or it is *weakly stable*, if there exist no blocking coalition S with an activity $\mathfrak{a}^{S,k}$ such that every player $i \in S$ *strictly* prefers $\mathfrak{a}^{S,k}$ to $\mathfrak{a}_i(X)$, (so $\mathfrak{a}^{S,k} >_i \mathfrak{a}_i(X)$ for all $i \in S$).

- An outcome X is in the *strong core*, or it is *strongly stable*, if there exist no blocking coalition S with an activity $\mathfrak{a}^{S,k}$ such that every player $i \in S$ *strictly* prefers $\mathfrak{a}^{S,k}$ to $\mathfrak{a}_i(X)$ or is indifferent between them, and there exist at least one player j, who *strictly* prefers $\mathfrak{a}^{S,k}$ to $\mathfrak{a}_j(X)$, (so $\mathfrak{a}^{S,k} \geq_i \mathfrak{a}_i(X)$ for all $i \in S$ and $\mathfrak{a}^{S,k} >_j \mathfrak{a}_j(X)$ for some $j \in S$).

- An outcome X is in the *super core*, or it is *super stable*, if there exist no blocking coalition S with an activity $\mathfrak{a}^{S,k}$ such that every player $i \in S$ *strictly* prefers $\mathfrak{a}^{S,k}$ to $\mathfrak{a}_i(X)$ or is indifferent between them, (so $\mathfrak{a}^{S,k} \geq_i \mathfrak{a}_i(X)$ for all $i \in S$).

Note that the weak core is simply the core. A *derived game* for a given finitely generated NTU-game with indifferences is obtained by breaking all the ties in some way. If we consider a fractional core x, or an outcome X that is in the core for a derived game, then obviously x is a fractional core element, or X is an outcome in the core of the original game too. In fact, this technique is used also in the algorithm of Scarf, where the matrix C is perturbed at the beginning of the procedure in order to avoid indifferences, and thus the cycling caused by the degeneracy.[1]

1.1.5 Fractional stable matching in hypergraphs

For a finitely generated NTU-game, the problem of finding an outcome that is in the core, is equivalent to the *stable matching problem in a hypergraph* as defined by Aharoni and Fleiner [7]. Here, the vertices of the hypergraph correspond to the players, the edges correspond to the efficient vectors (or to activities), and the preference of a vertex over the edges it is incident with comes from the preference of the corresponding player over the activities he can be involved in. This is called a *hypergraphic preference system*. A matching corresponds to a set of common activities performed by some coalitions, that form a partition of the grandcoalition together with the singletons (i.e. vertices not covered by the matching). A matching is *stable* if there exist no blocking edge, i.e. an edge, that is not in the matching but strictly preferred to the edges in the matching by all its vertices. The corresponding set of activities gives a stable outcome, since there exist no blocking coalition with an activity that is strictly preferred by all of its members (in other words, the direct sum of the corresponding independent efficient utility vectors is a feasible utility vector for the grandcoalition which is in the core of the game). A hypergraph that represents the efficient utility vectors of a CFG is obviously simple.

The *fractional stable matching* for a stable matching problem in a hypergraph was defined by Aharoni and Fleiner [7] as follows. A function x assigning non-negative weights to edges of the hypergraph is called a *fractional matching* if $\sum_{v\in h} x(h) \leq 1$ for every vertex v. A fractional matching x is called *stable* if every edge e contains a vertex v such that $\sum_{v\in h, e\leq_v h} x(h) = 1$. The existence of a fractional stable matching can be verified by Scarf's lemma just like the existence of a fractional core element. Actually, these two notions are basically equivalent.

To show the equivalence formally, we consider on the one hand the polytope of intensity vectors $\{x|Ax = 1^N, x \geq 0\}$, where A is the membership-matrix of the efficient utility vectors (or the corresponding activities) of dimension $n \times (n+m)$ as defined by Scarf's lemma. On the other hand, the fractional matching polytope is $\{x|Bx \leq 1^N, x \geq 0\}$, where B is the vertex-edge incidence matrix of the hypergraph of dimension $n \times m$. Obviously, $A = (I_n|B)$, so the difference is only an $n \times n$ identity matrix, i.e. the membership-matrix of the singletons. So, there is a natural one-to-one

[1] We note that the nonemtyness of the core can depend dramatically on the way we break the ties. In fact, the problem of finding an outcome that is in the core for an NTU-game with indifferences can theoretically be more complex too. We will study some relevant cases considering matching games in Section 2.3.

correspondence between the elements of the two polytopes: if x^m is a fractional matching of dimension m, then let $\bar{x}^v = 1^N - Ax^m$ be a vector of dimension n, that gives the unfilled intensities of the players (or in other words, the intensities of the single activities). The direct sum of these two independent vectors, x is an intensity vector of dimension $n+m$, and vice versa. The stability condition is equivalent to the domination condition of Scarf's lemma.

Finally, we have to clarify the question of indifferences. According to the definition by Aharoni and Fleiner [7], in a hypergraphic preference system the preferences of the vertices over the edges are strict. If we allow *ties*, then the corresponding problem is called *stable matching problem in a hypergraph with ties.*[2]

Here, the stable matching problem obtained by breaking the ties in some way is also called a *derived problem.* If we consider a fractional stable matching for a derived problem, this fractional matching is obviously stable for the original setting too.

Aharoni and Fleiner showed that a fractional stable matching can be assumed to be an extremal point of the fractional matching polytope. This fact comes from a statement similar to the following Proposition:

Proposition 1.1.8. *If x is a fractional core element of a finitely generated NTU-game, and $x = \sum \alpha_i x^i$, where $\alpha_i > 0$ for all i, $\sum \alpha_i = 1$ and x^i satisfies the $Ax^i = 1^N$ and $x^i \geq 0$ conditions, then each x^i must be a fractional core element.*

The proof of this Proposition is obvious, since $supp(x^i) \subseteq supp(x)$, that implies the dominating property of the fractional core element.

Corollary 1.1.9. *For any finitely generated NTU-game, there exists a fractional core element that is an extremal point of the polytope $\{x|Ax = 1^N, x \geq 0\}$.*

Corollary 1.1.9 implies that if all the extremal points of the above polytope are integers, or in other words the polytope has the *integer property*, then the finitely generated NTU-game has a nonempty core.

1.1.6 Normality implies the nonemptyness of the core

The definition of a normal hypergraph is due to Lovász [70]. If H is a hypergraph and H' is obtained from H by deleting edges, then H' is called a *partial hypergraph* of H. The *chromatic index* $\chi_e(H)$ of a hypergraph H is the least number of colors sufficient to color the edges of H so that no two edges with the same color have a vertex in common. Note that the maximum degree, $\Delta(H)$ (that is, the maximum number of edges containing some one vertex) is a lower bound for the chromatic index. A hypergraph

[2]However, we note that in the literature the stable matching problem for graphs is sometimes defined by allowing ties in the preference lists (see e.g. the book of Roth ad Sotomayor [92]).

H is *normal* if every partial hypergraph H' of H satisfies $\chi_e(H') = \Delta(H')$. Obviously, the normality is preserved by adding or deleting parallel edges or loops. The following theorem of Lovász [70] gives an equivalent description of normal hypergraphs.

Theorem 1.1.10 (Lovász)**.** *The fractional matching polytope of a hypergraph H has the integer property if and only if H is normal.*

Suppose that for a finitely generated NTU-game the set of essential coalitions forms a normal hypergraph. The hypergraph of the corresponding stable matching problem must be also normal, since it is obtained by adding parallel edges and by removing the loops. By Theorem 1.1.10, the fractional matching polytope, $\{x|Bx \leq 1^N, x \geq 0\}$ has the integer property, and so has the polytope of intensity vectors, $\{x|Ax = 1^N, x \geq 0\}$ as it was discussed previously. This argument and Corollary 1.1.9 verify the following Lemma 1.1.11.

Lemma 1.1.11. *If, for a finitely generated NTU-game, the set of essential coalitions, $\mathcal{E}(N,V)$ forms a normal hypergraph, then the core of the game is nonempty.*

By Lemma 1.1.11 and Proposition 1.1.4 the following holds.

Theorem 1.1.12. *If the set of basic coalitions, $\mathcal{B}$ forms a normal hypergraph, then every finitely generated NTU-game $(N, V, \mathcal{B})$ has a nonempty core.*

Let $A^{\mathcal{B}}$ denote the membership-matrix of the set of basic coalitions $\mathcal{B}$. The fact that the integer property of the polytope $\{x|A^{\mathcal{B}}x = 1^N, x \geq 0\}$ implies the nonemptyness of every NTU-game $(N, V, \mathcal{B})$ was proved first by Vasin and Gurvich [108], and independently, by Kaneko and Wooders [63].

Later, Le Breton *et al.* [69], Kuipers [68] and Boros and Gurvich [28] observed independently that the integer property of the polytope $\{x|A^{\mathcal{B}}x = 1^N, x \geq 0\}$ is equivalent to the integer property of the matching polytope $\{x|A^{\mathcal{B}}x \leq 1^N, x \geq 0\}$, and to the normality of the corresponding hypergraph.

We note that the other direction of Theorem 1.1.12 is also true: Boros and Gurvich [28] proved that if the set of basic coalitions, $\mathcal{B}$ has the property that the polytope $\{x|A^{\mathcal{B}}x = 1^N, x \geq 0\}$ has a non-integer extremal point (thus the corresponding hypergraph is not normal), then there always exist a finitely generated NTU-game $(N, V, \mathcal{B})$ with an empty core.

1.2 TU-games

An *n-person game with transferable utility* (TU-game for short) is given by a pair (N, v), where N is the set of players, and v is a payoff function $v : 2^N \to \mathbb{R}$, such that $v(\emptyset) = 0$. Without loss of generality we may assume also that $v(\{i\}) = 0$ for all singletons, thus every efficient utility vector is nonnegative. The outcome of the game is a vector $u = u^N$ if it satisfies $u(N) = v(N)$ (also referred to as a *preimputation*), where the notation $u(S)$ means $u(S) = \sum_{i \in S} u_i^N$. The outcome u^N is in the core of the TU-game (N, v) if $u(S) \geq v(S)$ holds for every coalition $S \subseteq N$.

The definition of a TU-game can be easily derived from the definition of NTU-games, by setting $V(S) = \{u^S \in \mathbb{R}^S | \sum_{i \in S} u_i^S \leq v(S)\}$.

Superadditivity for a TU-game means $v(S) + v(T) \leq v(S \cup T)$ for every pair of disjoint coalitions S and T. Here again, we restrict our attention to superadditive games.

1.2.1 Partitioning game, relevant and essential coalitions

In a partitioning TU-game $(N, v, \mathcal{B})$ with a set of basic coalitions $\mathcal{B}$, $v(S)$ can be generated as:

$$v(S) = \max\{v(B_1) + v(B_2) + \cdots + v(B_k) | \pi = \{B_1, B_2, \ldots, B_k\} \in \Pi_{\mathcal{B}}(S)\}$$

To define the essential, relevant and balanced coalitions we introduce some compact formulas. Let $\mathcal{N} = 2^N \backslash \{\emptyset\}$ be the collection of nonempty coalitions. As it was defined in the previous section, A is the *membership-matrix* of dimension $|N| \times |\mathcal{N}|$ (i.e. $a_{i,S} = 1$ if $i \in S$ and 0 otherwise), and x is a column-vector of $\mathbb{R}_+^{\mathcal{N}}$ called *intensity vector*. Finally, let v be a row-vector of $\mathbb{R}^{\mathcal{N}}$ that corresponds to the payoff-function. Let $\max_{\mathcal{N}}(LP) = \max\{v \cdot x | Ax = 1^N\}$ and $\max_{\mathcal{N}}(IP) = \max\{v \cdot x | Ax = 1^N, x \in \{0,1\}^{\mathcal{N}}\}$ be the optimum of the linear and integer programs, respectively.

Similarly, if $\mathcal{C}$ is an arbitrary collection of nonempty coalitions, so $\mathcal{C} \subseteq \mathcal{N}$, then we define $\max_{\mathcal{C}}(LP)$ and $\max_{\mathcal{C}}(IP)$ to be the optimum of the restricted programs (i.e if $\cup_{C \in \mathcal{C}} = \bar{C}$, $x|_{\mathcal{C}} = x^{\mathcal{C}}$, $v|_{\mathcal{C}} = v^{\mathcal{C}}$, and $A^{\mathcal{C}}$ is the restricted membership-matrix of dimension $|\bar{C}| \times |\mathcal{C}|$, then $\max_{\mathcal{C}}(LP) = \max\{v^{\mathcal{C}} \cdot x^{\mathcal{C}} | A^{\mathcal{C}} x^{\mathcal{C}} = 1^{\bar{C}}\}$ and $\max_{\mathcal{C}}(IP) = \max\{v^{\mathcal{C}} \cdot x^{\mathcal{C}} | A^{\mathcal{C}} x^{\mathcal{C}} = 1^{\bar{C}}, x^{\mathcal{C}} \in \{0,1\}^{\mathcal{C}}\}$). Finally, let $\mathcal{P}(S) = \{T : T \subseteq S\}$ and $\mathcal{P}^*(S) = \{T : T \subset S\}$.

A coalition S is *essential* if $\max_{\mathcal{P}^*(S)}(IP) < v(S)$. The assumed superadditivity implies that $\max_{\mathcal{P}(S)}(IP) = v(S)$ for every coalition S. Let $\mathcal{E}(N, v)$ denote the set of essential coalitions again. In a partitioning game $(N, v, \mathcal{B})$, the payoff of the coalition S can be generated simply as $v(S) = \max_{\mathcal{B}}(IP)$, so $\mathcal{E}(N, v, \mathcal{B}) \subseteq \mathcal{B}$ is obvious. The following Proposition says that the optimum of each integer program remains the same, if we restrict the support of x to the set of essential coalitions.

Proposition 1.2.1. *Given a TU-game (N, v) and $\mathcal{E} = \mathcal{E}(N, v)$. For every coalition S $\max_{\mathcal{P}(\mathcal{S})}(IP) = \max_{\mathcal{E} \cap \mathcal{P}(\mathcal{S})}(IP)$ holds.*

A coalition S is *relevant*[3] if $\max_{\mathcal{P}^*(S)}(LP) < v(S)$. A coalition S is *balanced* if $\max_{\mathcal{P}(S)}(LP) = v(S)$. If $\mathcal{R}(N, v)$ denotes the set of relevant coalitions, then $\mathcal{R}(N, v) \subseteq \mathcal{E}(N, v)$ obviously. The meaning of the following Proposition is that the optimum of

[3]This notion was introduced by Gillies [51], he called these coalitions as *vital* coalitions.

each linear program remains the same, if we restrict the support of x to the set of relevant coalitions.

Proposition 1.2.2. *Given a TU-game* (N, v) *and* $\mathcal{R} = \mathcal{R}(N, v)$*. For every coalition* S $\max_{\mathcal{P}(\mathcal{S})}(LP) = \max_{\mathcal{R}\cap\mathcal{P}(\mathcal{S})}(LP)$ *holds.*

The above Proposition implies that if the core-condition $u(S) \geq v(S)$ is violated for some coalition S not in $\mathcal{R}(N, v)$, then it must also be violated for some relevant coalition $T \subset S$.

1.2.2 The core of balanced games

Recall that a TU-game (N, v) is *balanced* if the grandcoalition N is balanced. The following well-known theorem was proved independently by Bondareva [26], [27] and Shapley [101].

Theorem 1.2.3 (Bondareva-Shapley)**.** *A TU-game* (N, v) *has a nonempty core if and only if it is balanced.*

Here, we verify this theorem only in case v is superadditive, although, we note that the theorem is true for every TU-game.

Proof. We shall observe first that $\max_{\mathcal{N}}(LP) = \max\{v \cdot x | Ax \leq 1^N\}$ (so the equalities can be changed to inequalities, since all singletons are in $\mathcal{N}$ with nonnegative payoffs). By this we can define the dual of the above linear program: let $\min_{\mathcal{N}}(DLP) = \min\{y \cdot 1^N | yA \geq v, y \in \mathbb{R}_+^N\}$, where y is a row-vector, that may correspond to a payoff-function $y : N \to \mathbb{R}_+$. The inequalities $y(S) \geq v(S)$ are in fact the core-conditions, and the objective function is $\sum_{i\in N} y_i = y(N)$. If y^* is an optimal solution of the dual program, then by using the Duality Theorem we get

$$v(N) = \max_{\mathcal{N}}(IP) \leq \max_{\mathcal{N}}(LP) = \min_{\mathcal{N}}(DLP) = y^*(N) \tag{1.1}$$

Finally, if $u(N)$ is an outcome in the core, then on the one hand $u(N) = v(N)$, and on the other hand, u must satisfy the core-conditions, so $u(N) \geq y^*(N)$. That is possible if and only if $v(N) = \max_{\mathcal{N}}(LP)$, thus if the game is balanced. □

Moreover, Propositions 1.2.1 and 1.2.2 imply that $\max_{\mathcal{N}}(IP) = \max_{\mathcal{E}}(IP)$ and $\max_{\mathcal{N}}(IP) = \max_{\mathcal{E}}(IP) = \max_{\mathcal{R}}(LP)$, respectively. So, the core of a TU-game is nonempty if and only if $\max_{\mathcal{E}}(IP) = \max_{\mathcal{E}}(LP)$. For a partitioning TU-game $(N, v, \mathcal{B})$, the above equality holds if and only if $\max_{\mathcal{B}}(IP) = \max_{\mathcal{B}}(LP)$, since $\mathcal{E}(N, v, \mathcal{B}) \subseteq \mathcal{B}$.

We note, that the above primal and dual programs have a well-known equivalent meaning in the theory of hypergraphs. Let the hypergraph H be formed again by the essential coalitions of the TU-game, where the weight of an edge e in the hypergraphs, $w(e)$ is equal to the payoff of the corresponding coalition S in the TU-game, $v(S)$. Recall that $y : V(H) \to \mathbb{R}_+$ is a *cover* of the vertex set $V(H)$ of H, if $\sum_{v\in e} y(v) \geq w(e)$ for every edge e in H. The value of y is $\sum_{v\in V(H)} y(v)$. The following observation is obvious.

Remark 1.2.1. *The problems* $\max\{v \cdot x | Ax \leq 1^N, x \in \{0,1\}^{\mathcal{N}}\}$, $\max\{v \cdot x | Ax \leq 1^N\}$ *and* $\min\{y \cdot 1^N | yA \geq v, y \in \mathbb{R}^N_+\}$ *are equivalent to the maximum weight matching, maximum weight fractional matching and minimum value cover problems in the corresponding weighted hypergraph H, respectively. Thus a TU-game has nonempty core if and only if the maximum weight of a matching in H is equal to the maximum weight of a fractional matching in H.*

It is well-known, that the optimum of a linear program can be obtained as an extremal point of the corresponding polytope. Thus, if the membership-matrix of the set of basic coalitions, $A^{\mathcal{B}}$ is such that the extremal points of the polytope $\{x^{\mathcal{B}} | A^{\mathcal{B}} x^{\mathcal{B}} = 1^N, x^{\mathcal{B}} \geq 0\}$ are integers, then the game has a nonempty core for every payoff function v. By Theorem 1.1.10 this is possible if and only if the hypergraph formed by the basic coalitions is normal. This argument directly verifies Theorem 1.1.12 in case of TU-games.

Theorem 1.2.4. *If the set of basic coalitions $\mathcal{B}$ forms a normal hypergraph, then every TU-game $(N, v, \mathcal{B})$ has a nonempty core.*

1.2.3 Coalition formation game with sidepayments

Every TU-game can be equivalently described as a *coalition formation game with sidepayments*. We recall, that in a coalition formation game each essential coalition S has only one Pareto-optimal utility vector $u^S \in U(S)$. A sidepayment p^S between the members of S is a vector in $\mathbb{R}^S$, such that $\sum_{i \in S} p_i^S = 0$, (i.e. the members of S can reallocate their utilities between each other). Thus, an outcome of a CFG with sidepayments consists of a partition π of the grandcoalition, and a sidepayment vector p, that is the sum of the independent sidepayment-vectors p^S for all $S \in \pi$. An outcome (π, p) determine the utility of each player i, $u_i(\pi, p) = u_i^S + p_i$, where $i \in S \in \pi$. An outcome (π, p) is in the core (or it is *stable*), if there exists no blocking coalition T with some sidepayment p^T, such that $u_i^T + p_i^T > u_i(\pi, p)$ for each $i \in T$, (so if each member of T can be strictly better off by cooperating with each other and by reallocating the common utility in a suitable way).

The corresponding TU-game is defined with the payoff-function $v(S) = \sum_{i \in S} u^S$. Obviously, the set of utility vectors $u^{S,p} = u^S + p^S$, that is achievable for S by performing the common activity and by reallocating the common utilities with some sidepayments, contains every efficient vector of S. This is the reason of the fact that the core of a CFG with sidepayments is the same as the core of the corresponding TU-game.

1.3 Matching games

Matching games can be defined as partitioning NTU-games, where the cardinality of each basic coalition is at most 2. For simplicity, in this section we suppose that no player is indifferent between two efficient utility vectors, so their preferences between the common activities are strict. If a matching game is finitely generated, then the problem of finding an outcome that is in the core is equivalent to a stable matching problem in a graphic preference system, where the edges of the graph correspond to the efficient utility vectors (and to the common activities).

Considering the set of utility vectors, basically we have four cases. If each basic coalition (each couple), has only one efficient utility vector, then the representing graph is simple, and the problem of finding a core-solution in the obtained CFG can equivalently be called the *stable roommates problem* (or stable matching problem in a simple graph). If each couple has a finite number of efficient utility vectors, then these activities are represented by parallel edges in the graph, the obtained stable matching problem can also be referred to as a stable roommates problem with multiple activities. In case of a general matching game, as we mentioned before, the set of efficient utility vectors can be approximated with a finite number of vectors, thus this general game has properties similar to the stable matching problem. Finally, if the utility is transferable, then the problem can be called roommates game with transferable utilities (or with sidepayments). Figure 1.2 illustrates these four cases.

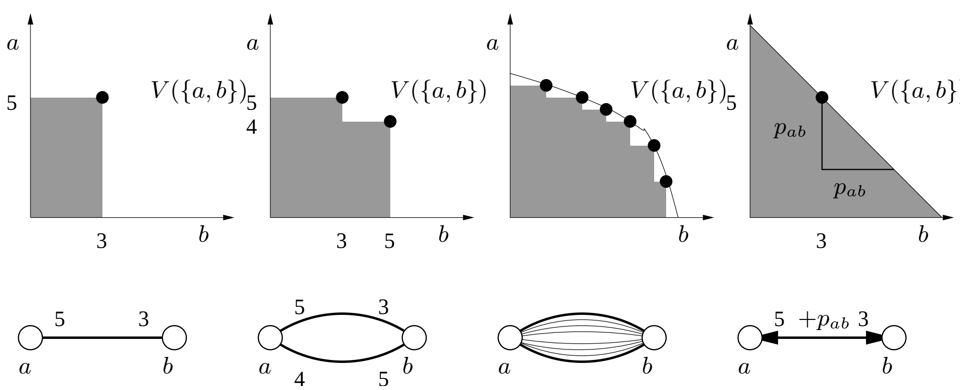


Figure 1.2: The set of efficient utility vectors and the corresponding edges.

We present two simple examples of three players considering both the stable roommates problem and the roommates problem with transferable utilities for the given utilities.

Example 1.3.1. *In the first example, the utility of the players from the possible common activities are the following:* $u_a^{\{a,b\}} = 6$, $u_b^{\{a,b\}} = 3$, $u_b^{\{b,c\}} = 4$, $u_c^{\{b,c\}} = 1$, $u_c^{\{c,a\}} = 2$, $u_a^{\{c,a\}} = 1$.

Here, there is no stable matching (the core of the CFG is empty), but there exists a matching $\{a, b\}$ with a sidepayment from a to b of a value between 2 and 3, that is stable (thus the core of the TU-game is nonempty).

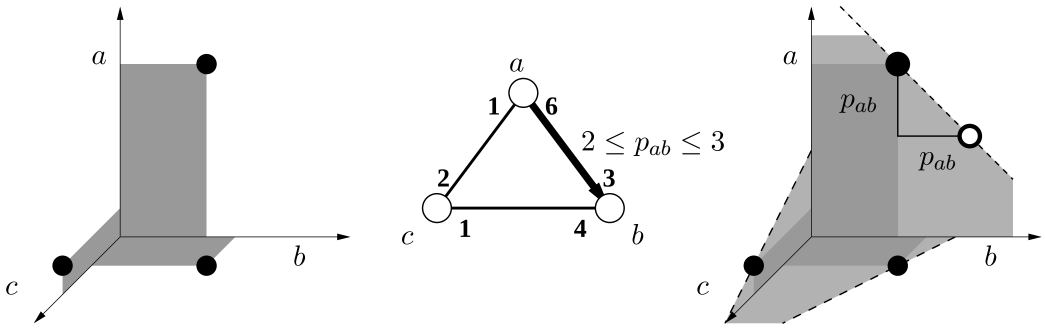


Figure 1.3: Example 1.3.1 illustrates a core-outcome in case of the TU-game.

Example 1.3.2. *The utility of the players from the possible common activities are the following:* $u_a^{\{a,b\}} = 2$, $u_b^{\{a,b\}} = 3$, $u_b^{\{b,c\}} = 2$, $u_c^{\{b,c\}} = 1$, $u_c^{\{c,a\}} = 2$, $u_a^{\{c,a\}} = 1$.

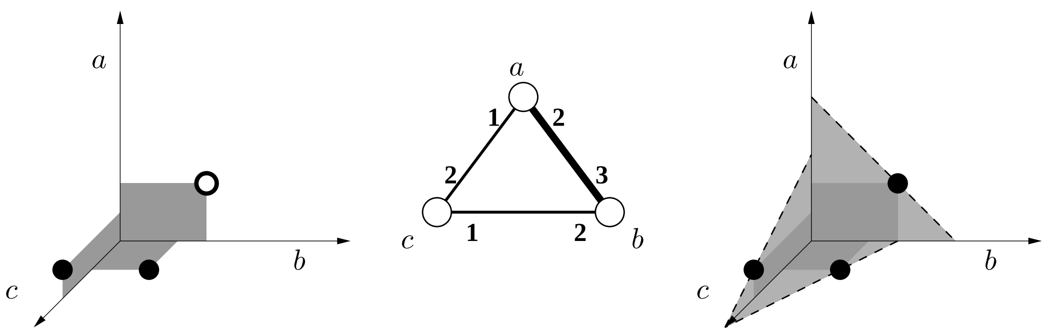


Figure 1.4: Example 1.3.2 illustrates a stable matching in case of NTU-game.

Here, the matching $\{a, b\}$ is stable (the core of the CFG is nonempty), but the roommates game with transferable utilities has an empty core. (The reason of this latter fact will be explained at the end of this subsection.)

Let us suppose the set of players N can be divided into two parts, M and W such that every two-member basic coalition contains one member from each side (so if $\{m, w\} \in \mathcal{B}$ then $m \in M$ and $w \in W$). In this case we get a *two-sided matching game* (in the general case the matching game is called *one-sided*).

1.3.1 Two-sided matching games

If a two-sided matching game is finitely generated, then the corresponding graphic representation of the stable matching problem is bipartite. For bipartite graphs, the following Proposition is well-known.

Proposition 1.3.3. *Every bipartite graph is normal.*

This Proposition and Theorem 1.1.12 imply the following result.

Theorem 1.3.4. *Every finitely generated two-sided matching game has a nonempty core.*

Theorem 1.3.4 was proved for every two-sided matching game, originally called central assignment game, by Kaneko [62]. For the corresponding CFG-s, called *stable marriage problems*, this result was proved by Gale and Shapley [48]. Finally, the same result for two-sided matching games with transferable utilities, defined as *assignment games*, was showed by Shapley and Shubik [102]. We note that by Remark 1.2.1 the latter result is an immediate consequence of the theorem by Egerváry [39], which states that the maximum weight of a matching and the minimum value of a cover are equal in every bipartite graph.

1.3.2 One-sided matching games

As it was shown by Examples 1.3 and 1.4, a one-sided matching game can have an empty core, even for the special CFG and TU-games. Here, we present that the half-integer property of the fractional matching polytope implies the existence of half-solutions in each case. The following observation is due to Balinski [15].

Theorem 1.3.5 (Balinski)**.** *The fractional matching polytope for every graph has only half-integer extremal points.*

This theorem of Balinski [15] and Scarf's lemma imply that in every matching game there exists a fractional core element x that has the half-integer property, i.e. $x_i \in \{0, \frac{1}{2}, 1\}$, called a *half-core element.*

Theorem 1.3.6. *If a matching game is finitely generated then it always has a half-core element.*

For CFG-s, the fact that for every stable matching problem there exists a stable half-matching was proved by Tan [104]. For matching games (or roommates games) with transferable utilities, a similar result was proved by Eriksson and Karlander [41]. In this case, by Remark 1.2.1 the nonemptyness of the core depends on whether the corresponding weighted matching and weighted fractional matching problems have the same optimum or not. This question is solvable in polynomial time by the algorithm of Edmonds [38]. This fact explains why the TU-game in Example 1.4 has an empty core: the maximum weight of a matching is at most 5, while there exist a half-matching with weight 5.5.

Chapter 2

Stable matching problems

Introduction

In this chapter, we start with a short overview on the stable matching problem. Then we present some results on the dynamics of stable matchings, published in [21]. Finally, we consider the stable matching problem with ties, where we collect a family of related problems and we study their complexity. A part of the results come from [5].

Some important results are collected in a table at the end of Section 2.3, and a part of them are included in another table at the end of Section 4.3 as well. To clarify the connections, we give a reference index [Ri] to each problem contained in these tables of results.

2.1 The stable marriage and roommates problem

2.1.1 Preliminaries

The stable matching problem on a graphic preference system (or simply the stable matching problem) was already defined in the previous section as a problem equivalent to finding an outcome in the core of a finitely generated NTU-game with basic coalitions of size at most two, called a matching game. Here, we recall the definition and introduce some further notions.

Let us model the *stable matching problem* with a graph G, where the agents are represented by vertices, and two vertices are linked by an edge if the agents are both acceptable to each other. For every vertex v, let $<_v$ be a linear order on the edges incident with v. That is, every agent has strict preferences on his possible partnerships.

We say that agent v prefers edge f to e (in other words f *dominates* e at v) if $e <_v f$ holds. A *matching* M is a set of edges with pairwise distinct vertices. If an edge $e = \{u, v\}$ belongs to M, then u and v are *matched* in M, so u and v are partners in the market. An agent is *single*, if his vertex is uncovered in M, i.e. it is not incident with a matching edge.

A matching M is called *stable* if every nonmatching edge, $e \notin M$ is dominated by some matching edge, $f \in M$. Alternatively, a stable matching can be defined as a matching without a *blocking edge*: an edge $e = \{u, v\}$ is blocking for a matching M if u is either unmatched or prefers edge e to the matching edge that covers u in M, and at the same time, v is either unmatched or prefers edge e to the matching edge that covers v in M. For a matching market, the stability means that no pair of agents can benefit by leaving their actual partners and establishing a new mutual partnership.

For simplicity, in this section we suppose that the graphs of the stable matching problems are simple, thus the corresponding NTU-games are coalition formation games. The stable matching problem was called *stable marriage* problem if the graph is bipartite (the two color classes represent men and women), and *stable roommates* problem if the graph is general [48]. According to the original definitions, it is assumed that each agent finds acceptable all his possible partners, so the graph and the preference lists are also complete. Here, we relax this assumption and study stable marriage and stable roommates problems with *incomplete lists*.

2.1.2 The stable marriage problem

The stable marriage problem was defined and studied by Gale and Shapley [48]. In their seminar paper they proved that stable matching always exists for the marriage problem, moreover their algorithm produces a solution that is optimal for each member in one of the sides. Beside presenting this algorithm, we show some other well-known results.

The original goal of Gale and Shapley was to study the college admission problem, that is a generalization of the stable marriage problem. Many years later, Roth discovered that the Gale–Shapley algorithm had been in use since 1952 by the National Intern Matching Program (now called the National Resident Matching Program, NRMP), for further details, see [83] and [92]. We study similar problems with capacities in Chapter 3, where we present the Hungarian higher education admission program as a special application.

Let the vertex set of a bipartite graph G be the disjoint union of M and W (i.e. men and women). We suppose that a man and a woman are linked by an edge if they are both acceptable to each other. Each man and each woman ranks his or her possible marriages (or equivalently his or her possible partners) in a strict order, represented by a preference list.

The deferred-acceptance algorithm

To start, let each man propose to his favorite woman. Each woman who receives more than one proposal rejects all but her favorite, who becomes her fiancé. In the further stages, each man who was rejected in the previous stage, proposes to his next choice according to his list, and each woman who was receiving proposals reject all but her most preferred among the group consisting of the new proposers together with her current fiancé. Obviously, men propose to less and less preferred partners, while women receive proposals from better and better fiancés, thus the algorithm stops in $O(m)$ time (where m denotes the number of edges in the graph), when no more proposal is left. The fiancés become husbands. The obtained set of marriages is obviously a matching.

To show stability, consider an edge $\{m, w\}$ not in the matching. If m made no proposal to w, then m has been engaged with a preferred woman in each stage, and also at the end of the procedure. So pair $\{m, w\}$ cannot block because of m. Otherwise, if m proposed to w during the algorithm, then w refused him because she had a better proposal at that time, and since her situation was improving during the algorithm, she must have a preferred husband at the end of the algorithm too. Again $\{m, w\}$ cannot block.

Theorem 2.1.1 (Gale–Shapley)**.** *For every stable marriage problem, there exists a stable matching. [R1]*

Moreover, it can be verified that each man gets his best possible partner (i.e. woman w is *possible* for man m if there exists a stable matching M, where $\{m, w\}$ is a matched pair) in the matching obtained by the deferred-acceptance algorithm, thus this matching is so-called *man-optimal*. Indirectly, let us suppose that m is the first such man who is rejected during the algorithm by a woman w who is possible for him, (i.e. there exist a stable matching M, where $\{m, w\} \in M$). At that moment, when m is rejected by w, w must have a preferred partner, say m'. Obviously, m' must be matched with a better partner than w in M, say w', since otherwise $\{m', w\}$ would block M. But, in this case, m' must have been rejected by w' (a possible partner of m') in a previous case, a contradiction.

Theorem 2.1.2 (Gale–Shapley)**.** *The stable matching obtained by the deferred - acceptance algorithm is man-optimal.*

Below we list some well-known results on stable marriages. All of them can be found in both the book of Gusfield and Irving [52] and also in the book of Roth and Sotomayor [92].

Let M and M' be two stable matchings for the same stable matching problem. If $e \in M \setminus M'$, then there must exist another edge $f \in M' \setminus M$ such that $e <_v f$ by the domination condition. This observation implies that these edges of $M \bigtriangledown M'$ form even cycles. The following three theorems are easy consequences of this argument for both the stable marriage and stable roommates problems.

Theorem 2.1.3. *For a given stable marriage problem, the same agents are matched in every stable matching.*

An agent is said to *prefer* a matching M to another matching M' if he prefers his partner in M to his partner in M'.

Theorem 2.1.4. *If $\{m, w\} \in M \setminus M'$ for some stable matchings M and M', then one of m and w prefers M to M', and the other prefers M' to M.*

Theorem 2.1.5. *If w is the best possible partner of m, then m is the worst possible partner of w.*

The following theorem presents the fact that in a two-sided matching market the agents of different sides have an opposite interest over the set of possible stable outcomes. Moreover, this statement preserves an interesting lattice structure of the set of stable marriages.

Theorem 2.1.6 (Conway)**.** *Let M and M' be two stable matchings for a stable marriage problem. If each man is given the better of his partners in M and M' (or each woman is given the worse of her partners), then the result is again a stable matching, denoted by $M \vee M'$. Similarly, if each man is given the better of his partners in M and M' (or each woman is given the less preferred of her partners), then the result is the same stable matching, denoted by $M \wedge M'$.*

Corollary 2.1.7. *For a stable marriage problem, the set of stable matchings forms a distributive lattice with the above meet and join operations.*

Detailed description of this lattice structure can be found in the book of Gusfield and Irving [52] and in the thesis of Fleiner [44].

2.1.3 The stable roommates problem

The stable matching problem for nonbipartite simple graphs was called the *stable roommates problem* by Gale and Shapley [48]. They showed the following example to prove that a stable matching may not exist.

Example 2.1.8.

Agents	*Preference-lists*
A :	$[B, C, D]$
B :	$[C, A, D]$
C :	$[A, B, D]$
D :	*arbitrary*

Let us imagine that these agents are tennis-players, each one is looking for a partner to play with for one hour a week. For example Andy would like to play mostly with Bill, then with Cliff and finally he prefers to play with Daniel the least. (In fact, everybody tries to avoid Daniel.) There is no stable solution. If a pair is formed from the first three players, say Andy plays with Bill, then the third one, Cliff must be matched with Daniel, but in this case Bill and Cliff block this matching.

The stable roommates problem had been a nontrivial open problem, until Irving [56] constructed the first polynomial time algorithm which determines whether a given instance of the stable roommates problem admits a stable matching, and if so, finds one [R2].

As we discussed in Section 1.3, for every matching game there exists a half-core, that corresponds to a stable half-matching for the stable roommates problem. So, if the agents can create half-time partnerships then a stable solution always exists in the sense that no pair of agents would simultaneously like to increase the intensity of their partnership.

Considering the above example, we suppose that Andy, Bill and Cliff agree to meet once a week and play half-time games in each formation. Thus, each of them play one hour in sum, only Daniel remains without any tennis-partner. Stability in this case means that no pair of tennis-players wants to play more time together with each other. For example Andy plays with Daniel no time at all, because Andy fills his one-hour by playing two half-hour games with better partners. Andy and Bill will not play more than a half-hour, because Bill fills the rest of his time (a half-hour) by playing with a better partner, Cliff.

Recall that *half-matching* hM consists of matching edges M and half-weighted edges H, so that $hM = H \cup M$ and each vertex is incident either with at most one matching edge or with at most two half-weighted edges. In a matching market an agent can have at most one partner or at most two half-partners. A half-matching hM is *stable* if for each edge e not in hM there exists a vertex v, where e is dominated either by one matching edge or by two half-weighted edges, and for every half-weighted edge h there exists a vertex v, where h is dominated by another half-weighted edge. So no pair of agents wants to improve their partnership simultaneously, because for each pair of agents who are not matched, one of them fills his capacities with better partnership(s). Otherwise, if a half-matching is unstable, then a *blocking edge* is an undominated edge.

The fact, that every half-weighted edge must be dominated by another half-weighted edge at one of its endvertices implies that the half-weighted edges form cycles, where the direction of the domination between two consecutive half-weighted edges is the same along the cycle. To illustrate this property in the figures, we orient each half-weighted edge to its endvertex, where it is dominated by the other half-weighted edge. It is obvious that an even-cycle can be replaced by matched pairs, but Tan [104] observed that if an odd-cycle C occurs in hM then C must belong to the H-part of any stable half-matching for the given graph, so no stable matching exists. He characterized the stable half-matching (originally, called *stable partition*) in the following way:

Theorem 2.1.9 (Tan)**.** *For any instance of the stable roommates problem there always exists a stable half-matching that consists of matched pairs and odd-cycles formed by half-weighted pairs. The set of agents can be partitioned into:*

a) unmatched (or single) agents,

b) cycle-agents and

c) matched agents.

Furthermore, the same agents remain unmatched and the same odd-cycles are formed in each stable half-matching of the given instance of the problem.

If for a half-matching $hM = H \cup M$ an edge $e = \{u, v\}$ is in M, then we say that the agents u and v are *partners*. If two agents can be partners in a stable half-matching we call them *stable partners*. If an edge $e = \{u, v\} \in H$ is in an odd-cycle, then u and v are *half-partners*. If u prefers v to his other half-partner, then v is the *successor* of u and u is the *predecessor* of v.

2.2 The dynamics of stable matchings

For the bipartite case, Knuth [67] asked whether it is possible to obtain a stable matching by starting from an arbitrary matching and successively satisfying blocking pairs. Roth and Vande Vate [93] gave a positive answer by a decentralized algorithm, in which pairs or single agents enter the market in a random order, and stability is achieved by a proposal–rejection process. Knuth's question for the bipartite case was also answered by Abeledo and Rothblum [4] by a common generalization of the Roth–Vande Vate and the Gale–Shapley algorithms. Later, Diamantoudi, Miyagawa and Xue [37] solved the same problem for the roommates case. They proved that one can always reach a stable matching, if one exists, from an arbitrary matching by successively satisfying blocking pairs. Recently, Inarra, Larrea and Molis [55] generalized this result for insolvable stable roommates problems by proving the same statement for the so called, P-stable matchings instead of stable matchings.

However, the original goal of Roth and Vande Vate was different, their algorithm can be used to model the dynamics of the two-sided matching market as well. In fact, they considered the situation when a new agent enters the market and the stability is restored by the natural proposal-rejection process. This mechanism also yields an algorithm to find a stable matching for a market by letting the agents enter the market in a random order. Independently, Tan and Hsueh [105] constructed an algorithm, that finds a stable half-matching for general graphs by using a similar incremental method. In the bipartite case, the Tan–Hsueh algorithm is equivalent to the Roth–Vande Vate algorithm. In the nonbipartite case infinite repetitions can occur, these are handled by the introduction of cycles. We shall call these two algorithms “incremental algorithms”.

Blum, Roth and Rothblum [24] described the properties of a dynamic two-sided matching market. They showed that their proposed algorithm is similar to the McVitie–Wilson’s version [75] of the original deferred-acceptance algorithm. So, the output of the process is predictable: if some men enter the market then each man either remains matched with the same partner (if it is possible) or gets a worse (but his best) stable partner for the new market. Blum and Rothblum [25] pointed out that these results imply that the lastcoming agent gets his best stable partner in the Roth–Vande Vate algorithm. Moreover, an agent can only benefit from entering the market later (we assume here that the others enter the market in the same order). Independently, Ma [72] observed on an example of Knuth, that if agents enter the market successively then the Roth–Vande Vate algorithm may not find all stable matchings in general. Cechlárová [30] strengthened Ma’s result by justifying that in a stable matching output by the incremental algorithm for a bipartite graph some agent gets his best stable partner. Here we give direct proofs for the above results in the bipartite case, and we generalize most of them to general graphs with the help of our Key Lemma.

Gale and Sotomayor [49] showed that if some man expands his preference-list then no other man is better off in the new men-optimal stable matching. This implies that the same statement is true if a number of men enter the market. Roth and Sotomayor [92] proved that if a man arrives and becomes matched, then certain women will be better off, and some man will be worse off under any stable matching for the new market than at any stable matching for the original market. We generalize this theorem by using an improved version of a result of Pittel and Irving [80] on the core configuration.

Our results also have an economic interpretation. Matching markets are well-known applications of the stable matching problem. A detailed description of two-sided markets can be found in the book of Roth and Sotomayor [92]. An important example is job matching. Blum, Roth and Rothblum [24] studied the dynamics of the two-sided matching market in this context by analyzing the formation of “vacancy chains”.

The dynamic formation of social and economic networks can be described by stable matching models as Jackson and Watts considered in [61]. They illustrated the occurring mechanisms with the Roth–Vande Vate algorithm in the bipartite case. We believe that the same model can be used in the nonbipartite case, where the connections between individuals might correspond to mutual “best friend” relationships. By similar reasons, Ericsson and Strimling used the same stable roommates model in [43] to analyse the mate searching processes for special preferences. Recently, the dynamics of firm mergers was also described as a one-sided stable matching market by Angelov [11].

2.2.1 The incremental algorithms

Suppose a matching market is in an equilibrium with a stable matching. A natural question is how the situation changes if a new player enters the game and the preferences over the former partnerships are unchanged. Let the newcomer make proposals according to his preference order. If no one accepts, then everybody has a better partner, so the former matching remains stable. If somebody accepts a proposal, then a new pair is formed along the proposal. The eventual left-alone partner has to leave the market and enter as a newcomer. Note that the same situation happens, when an agent leaves the market. If he was single, then the matching remains stable. Otherwise, if he was matched, then his partner has to leave the market and enter again as a newcomer.

The Roth–Vande Vate algorithm for the stable marriage problem

Suppose, that in the algorithm a bipartite graph G is built up step by stepby adding vertices to the graph in some order. In a *phase* of the algorithm we add a new agent and restore stability. To describe a phase, let us add a vertex v to $G-v$, where a stable matching M_v exists. Our task is to find a stable matching M for G.

If v is not incident to any blocking edge, then M_v remains stable for G, too. In this case we call the phase *inactive*.

A phase is *active* if the newcomer v is a member of some blocking pair, let $\{v, u\}$ be the best blocking pair for v. Let $v = a_0$ and $u = b_1$. If b_1 was unmatched in $M_v = M_{a_0}$, then $M_{a_0} \cup \{a_0, b_1\}$ is a stable matching for G. Otherwise, b_1 had a partner a_1 in M_{a_0}, whom he leaves after receiving a better proposal. In this case, the matching $M_{a_1} = M_{a_0} \setminus \{a_1, b_1\} \cup \{a_0, b_1\}$ is stable for $G - a_1$. So we have a similar situation as in the beginning: a_1 enters the market and makes proposals. Continuing the process, a *proposal-rejection sequence*, $S = (A|B) = a_0, b_1, a_1, \dots$ is constructed with the following properties:

1. $M_{a_k} = M_{a_{k-1}} \setminus \{a_k, b_k\} \cup \{a_{k-1}, b_k\}$ is a stable matching for $G - a_k$.
2. a_{k-1} is a better partner for b_k than a_k and
3. b_{k+1} is a worse partner for a_k than b_k.

Note that here, $a_0, a_1, \dots$ are from the same side, and $b_1, b_2, \dots$ are from the other one. Property 2 is true, since b_k accepted the proposal of a_{k-1} while he left a_k. To see 3, realize that pairs (a_k, b_k) and (b_{k+1}, a_{k+1}) are in $M_{a_{k-1}}$, so Property 2. and the assumption that a_k prefers b_{k+1} to b_k would imply that (a_k, b_{k+1}) is a blocking pair for $M_{a_{k-1}}$.

A proposal-rejection process is illustrated in Figure 2.1. In this and subsequent figures, a little arrow is directed from a dominated edge to a dominating one and thick lines correspond to matching and half-weighted edges of the current (half-)matching.

Observe that by this process, each $a_i \in A$ improves his situation and each $b_j \in B$ gets worse off. Consequently, the same agents cannot occur as new pairs. So a phase

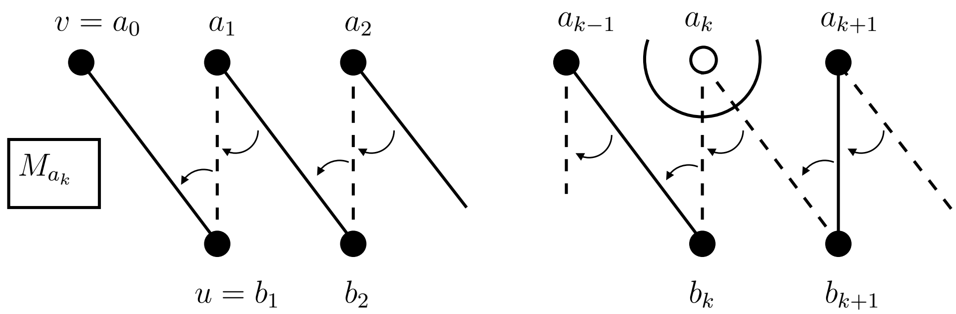


Figure 2.1: Proposal-rejection sequence in the Roth–Vande Vate algorithm.

terminates in $O(m)$ changes, when m denotes the number of the edges in the graph. A phase has two possible outcomes: either nobody accepts the proposals of some a_i (then the size of the matching remains the same) or the last b_j is unmatched, hence the size of the matching increases by one.

We illustrate with an example the mechanism of the incremental algorithm and we introduce briefly our results. The preferences of the agents on their possible partnerships in this two-sided market are the following:

Example 2.2.1.

$$\begin{array}{ll}
a_1 : [e_1, d_1, f_1] & b_1 : [f_3, d_2, n_1, e_1] \\
a_2 : [e_2, d_2, f_2] & b_2 : [f_2, d_1, e_3] \\
a_3 : [e_3, d_3, f_3] & b_3 : [f_1, d_3, e_2] \\
a_4 : [s] & b_4 : [s, n_2] \\
a_5 : [m_1, m_2] & b_5 : [m_2, n_3] \\
a_6 : [n_1, n_2, n_3, n_4] & b_6 : [n_4, m_1]
\end{array}$$

Let $d = \{d_1, d_2, d_3\}$, $e = \{e_1, e_2, e_3\}$, $f = \{f_1, f_2, f_3\}$. Suppose, that at the beginning a_6 is not present in the market. Partnerships $\{e, s, m_1\}$ form a stable matching in the market. (It is the best one for every agent a_i.)

When agent a_6 enters the market, four new possible partnerships are created. The best one for the newcomer is n_1, and this blocks the actual matching. Following the algorithm of Roth and Vande Vate let us satisfy this blocking edge: b_1 and a_6 form a new pair, and partnership e_1 terminates, so agent a_1 has to find a new partner as a newcomer. Continuing this process, the following edges will be satisfied and terminated in sequence: d_1, e_3; d_3, e_2; d_2, n_1. Afterwards, agent a_6 makes proposals again, that b_1 and b_4 refuse, because they prefer their partners to a_6. We will prove later, that if a new partnership is not blocking, then it cannot be present in any stable matching. In the last step of our example, a single agent b_5 accepts the proposal of a_1, and $\{d, s, m_1, n_3\}$ is a stable matching. This stable solution is the best possible for the newcomer a_6, since

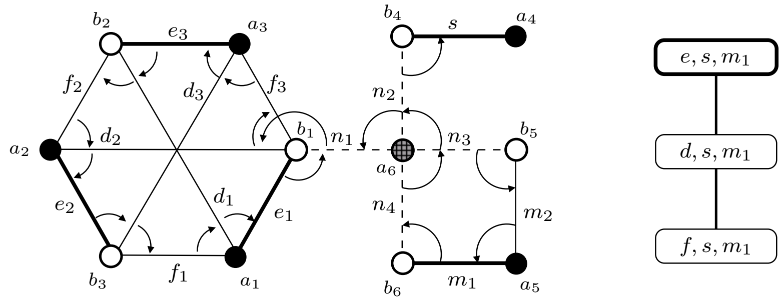


Figure 2.2: A stable matching and the lattice of the stable matchings for Example 2.2.1 before the arrival of a_6.

the better partnerships, that were refused by his possible partners cannot appear in any stable matching. This argument also shows that every agent that receives a partner by making a proposal during the process gets his best stable partner.

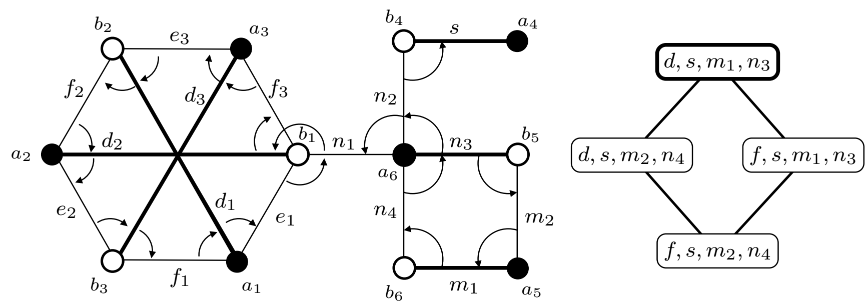


Figure 2.3: The obtained stable matching, and the lattice of the stable matchings for Example 2.2.1.

Note, that if we started with the stable matching $\{f, s, m_1\}$, then the process would stop in one step, since b_5 accepts first the proposal of a_6. The obtained stable matching $\{f, s, m_1, n_3\}$ yields the best stable partner to the newcomer a_6 again, but the other agents a_i do not necessarily get their best stable partners.

The Tan–Hsueh algorithm for the stable roommates problem

Tan and Hsueh [105] proposed an incremental algorithm to find a stable half-matching. In this more general setting, we use the terminology of the Roth–Vande Vate algorithm. The only difference is that G is not bipartite, so instead of a matching, we maintain a half-matching hM_v for $G - v$.

Hereafter, we suppose that the stable half-matchings have no even-cycles. As we mentioned before, an even-cycle can always be separated into matching pairs, moreover, as we will see later, the incremental algorithm does not create even-cycles. By Theorem 2.2.15 we know that for a fixed stable roommates problem, the same odd-cycles are present in each stable half-matching $hM^i = H \cup M^i$. So H is determined, only the M^i parts can differ for two stable half-matchings for a given graph. In fact, H can be considered as a disjoint union of half-weighted cycles, so whenever we modify a stable half-matching during the processes, we will only add or remove matching edges or half-weighted odd-cycles.

If nobody accepts the newcomer's proposal, then the phase is called *inactive* again and the current stable half-matching is unchanged.

If some agent u accepts the proposal of v then three cases are possible:

a) If u is unmatched in hM_v, then $hM = hM_v \cup \{v, u\}$ is a stable half-matching for G.

b) If u is a cycle-vertex in hM_v, so $u = c_0$ for some cycle $C = (c_0, c_1, \dots, c_{2k-1}, c_{2k})$, then $hM = hM_v \setminus C \cup \{v, u\} \cup \{c_1, c_2\} \cup, \dots, \cup \{c_{2k-1}, c_{2k}\}$ is a stable half-matching for G (i.e. we remove the half-weighted cycle C and we add some matching edges).

c) If u is matched with x in hM_v, then $hM_x = hM_v \setminus \{u, x\} \cup \{v, u\}$ is a stable half-matching for $G - x$.

The current phase ends in cases a) and b). Here, unlike in the bipartite case, it can happen that an agent, who made a proposal earlier can receive a proposal later during the same phase. So the proposal-rejection sequence might never end. One result of Tan and Hsueh [105] is that a repetition always occurs along an odd-cycle.

Theorem 2.2.2 (Tan–Hsueh). *If $S = (A|B) = a_0, b_1, a_1, \dots$ is a proposal-rejection sequence and $a_i = b_k$ $(i < k)$ is the first return, then this proposal-rejection sequence can be extended in such a way that it will return to a_k at b_{k+m+1}, and the following properties are true: $\{a_k, b_{k+1}, \dots, b_{k+m}, a_{k+m}\}$ are distinct vertices, and in the inverse order they form an odd-cycle C, and $hM = hM_{a_k} \setminus \{a_{k+1}, b_{k+1}\} \setminus \cdots \setminus \{a_{k+m}, b_{k+m}\} \cup C$ is a stable half-matching.*

The example in Figures 2.4 and 2.5 illustrate the Tan–Hsueh algorithm: Here, vertex v enters. The first vertex accepting v's proposal is u, and u's previous partner x is left alone. Figure 2.4 shows the stable half-matching hM_x for $G - x$. In the next step, x makes proposals. Figure 2.5 illustrates the termination of this phase by obtaining an odd cycle, namely the three-cycle containing vertex x and edges c_1 and c_2.

2.2.2 Getting the best stable partner by making proposals

Lemma 2.2.3 (Key Lemma). *If hM_v is a stable half-matching for $G - v$, and edge $\{v, u\}$ is not blocking hM_v, then v and u cannot be matched in a stable half-matching for G.*

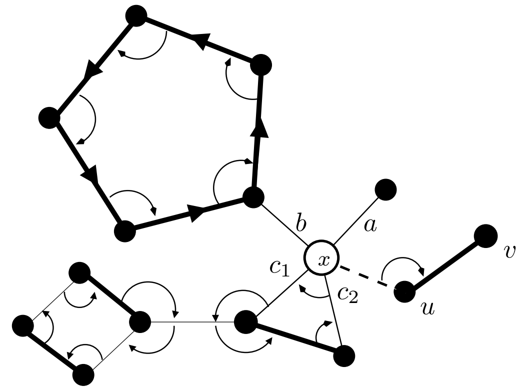


Figure 2.4: The Tan–Hsueh algorithm in an example.

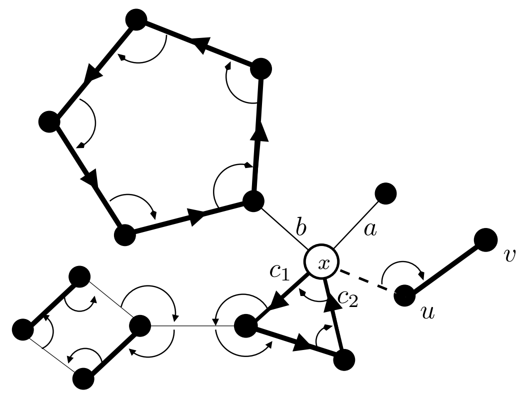


Figure 2.5: The obtained stable half-matching.

Proof. Let us suppose that $\{v, u\}$ is not blocking hM_v but there is a stable half-matching hM of G, where v and u are matched. Let $v = a_0$ and $u = b_1$. First we consider the case where none of hM and hM_v contains an odd-cycle. Then b_1 has a partner in M_v (say a_1), who is better than a_0. So $\{a_0, b_1\} <_{b_1} \{a_1, b_1\}$, where $\{a_0, b_1\} \in M \setminus M_v$. Since M cannot dominate $\{a_1, b_1\}$ at b_1, this edge must be dominated at a_1 by some edge $\{a_1, b_2\}$ of M. As $\{a_1, b_2\}$ is not in M_v, it must be dominated at b_2 by an edge $\{a_2, b_2\}$ of M_v, and so on. The alternating sequence $(a_0, b_1, a_1, b_2, \dots)$ has the following property: $\{a_{i-1}, b_i\} \in M \setminus M_v$ and $\{b_i, a_i\} \in M_v \setminus M$, furthermore the domination is also in sequence: $\{a_{i-1}, b_i\} <_{b_i} \{a_i, b_i\}$ and $\{a_i, b_i\} <_{a_i} \{a_i, b_{i+1}\}$ for every i. We call this sequence *alternating preference sequence*. Because a_0 is not covered by the stable matching M_v, the sequence can return neither to a_0, nor to any other vertex. Otherwise, the first such a repeated vertex would be covered by two matching edges, a contradiction. (This part of the proof already settles the bipartite case.)

The other case is, when hM_v or hM may contain odd-cycles. The properties of the alternating preference sequence remain the same, the difference is that the edges can be half-weighted edges as well. To avoid repetition, the idea is the following: when an edge $\{a_i, b_i\} \in hM_v$ is dominated at a_i in hM by two edges (so a_i is in a cycle in hM), then we choose b_{i+1} as the predecessor of a_i. Edge $\{a_i, b_{i+1}\}$ is still not in hM_v, so it must be dominated at b_{i+1}. But then the edge(s) that dominate(s) $\{a_i, b_{i+1}\}$ is (are) better than either of the edges that cover b_{i+1} in hM, so they are not in hM. This is why every new edge in this sequence will be alternately in $hM \setminus hM_v$ and $hM_v \setminus hM$.

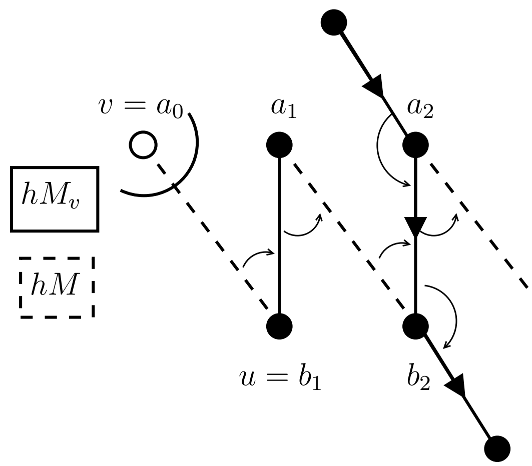


Figure 2.6: Alternating preference sequence with half-weighted edges.

As the number of agents is finite, the alternating preference sequence must return. Consider the first such repetition. If $a_k = a_i$ for some $k \neq i$ then $\{b_k, a_i\}$ and $\{b_i, a_i\}$ would be in the same odd-cycle in hM_v, but a_i would be the predecessor of both b_i and b_k by the inductive definition of the sequence, a contradiction. In the other case, assume that $a_k = b_i$ for some $k \neq i$. This means that $\{b_k, b_i\}$ and $\{b_i, a_i\}$ are in the same odd-cycle in hM_v. By definition, a_i is the predecessor of b_i, so b_k should be the successor of b_i, that would imply $\{a_i, b_i\} <_{b_i} \{b_k, b_i\}$. On the other hand, since $\{b_k, b_i\} \in hM_v \setminus hM$, this edge must be dominated at b_i in hM. By the inductive rules $\{a_{i-1}, b_i\} \in hM$, this implies $\{b_k, b_i\} <_{b_i} \{a_{i-1}, b_i\} <_{b_i} \{a_i, b_i\}$, a contradiction.

Similarly, if $b_k = b_i$ for some $k \neq i$ then $\{a_k, b_i\}$ and $\{a_i, b_i\}$ would be in the same odd-cycle in hM, but b_i would be the predecessor of both a_i and a_k by the inductive definition of the sequence, that is impossible. Finally, assume that $b_k = a_i$ for some $k \neq i$. This means that $\{a_k, a_i\}$ and $\{a_i, b_i\}$ are in the same odd-cycle in hM. By definition, b_i is the predecessor of a_i, so a_k should be the successor of a_i, that would imply $\{b_i, a_i\} <_{a_i} \{a_k, a_i\}$. On the other hand, since $\{a_k, a_i\} \in hM \setminus hM_v$, it must be dominated at a_i in hM_v. By the inductive rules $\{b_{i-1}, a_i\} \in hM_v$, this means $\{a_k, a_i\} <_{a_i} \{b_{i-1}, a_i\} <_{a_i} \{b_i, a_i\}$, a contradiction. $\square$

To generalize the results of Blum, Roth and Rothblum [24] we prove that the incremental algorithm assigns the newcomer to his best stable partner in the nonbipartite case as well.

Theorem 2.2.4. *Suppose that an agent v enters the market and stability is restored by a proposal-rejection process along the sequence $S = (A|B)$. Then each agent $a \in A$ ($b \in B$) , who became matched by making (accepting) a proposal gets his best (worst) stable partner in the obtained stable half-matching.*

Proof. If an agent a is matched in the output, and receives a partner by making a proposal, then later he cannot accept any proposal because then he would be a cycle-agent. The last time when agent a makes a proposal during the process he does not prefer his last partner only to some agents that refused him. Because of the Key Lemma, no one of these agents can be a partner of a in a stable solution, so obviously agent a received his best stable partner. Similarly, each matched agent $b \in B$ gets his worst stable partner by Theorem 2.1.5. □

Corollary 2.2.5. *If an agent enters the market last and becomes matched, then he gets his best stable partner.* □

If a phase is inactive in the incremental algorithm, then each stable half-matching of the extended graph is also a stable half-matching in the original. That is, if hM is a stable half-matching for G not covering some vertex x, then hM is a stable half-matching for $G - x$ too, because after deleting x from G no blocking edge can appear. So, by using the Key Lemma, we can confirm the following result:

Theorem 2.2.6. *Each matched agent, who gets a partner in the last active phase by making (accepting) a proposal, receives his best (worst) stable partner in the stable solution output by the incremental algorithm.* □

Remark 2.2.7. *The vertices that remained uncovered in the last active phase or entered later in an inactive phase, will still be uncovered at the end of the algorithm, just like they are in every stable matching. The vertices that form an odd-cycle in the last active phase will form an odd-cycle at the end of the algorithm, just like they do in every stable half-matching. Hence these agents also get their best stable partners in this sense.*

Corollary 2.2.8. *A stable matching, where none of the matched agent gets his best stable partner, cannot be the output of the incremental algorithm.* □

Let us remark that we did not prove that a stable matching where somebody gets his best stable partner or which contains an odd-cycle can be obtained by an incremental algorithm. Our result gives only a necessary condition not a sufficient one.

Blum, Roth and Rothblum [24] proved, that if a man m enters the market $G - m$ and another man m' was matched with w' in M_m, then m' and w' remain matched in the obtained stable matching M for the new market G if and only if they are stable partners for the new market. Otherwise m' and w' get those agents to whom they are matched in the men-optimal stable matching of G. (So m' receives his best stable partner, and w' receives her worst stable partner in this case.) Below, we generalize this statement for the nonbipartite case.

Theorem 2.2.9. *Suppose that w and u are matched in a stable half-matching hM_v for $G-v$. They remain matched in the stable half-matching hM, obtained by the proposal-rejection process after the arrival of v if and only if they are stable partners for G as well. Otherwise, if they are not involved in a cycle, then one of them gets a better partner than he had in hM_v but receives his worst stable partner, the other one becomes single or gets a worse partner than he had in hM_v but receives his best stable partner in hM .*

Proof. If w and u are not involved in the proposal-rejection process, then they remain matched. Otherwise, if $S = (A|B)$ is the proposal-rejection sequence, then one of them, say w is in A and u must be in B. As they are not involved in a cycle, u improves his situation and w gets worse off during the process, and finally (by Theorem 2.2.6) u gets his worst stable partner (better than w) and w gets his best stable partner (worse than u), so u and w cannot be stable partners in the output. □

2.2.3 Improving the situation by accepting proposals

Our next goal is to generalize Theorem 2.26. of Roth and Sotomayor in [92]. First we give its proof implied by the results presented above.

Theorem 2.2.10 (Roth–Sotomayor)**.** *Suppose a woman w is added to the market $G-w$. Let M^W be the woman-optimal stable matching for the new market, G and let M^M_w be the man-optimal stable matching for $G-w$. If w is not single in M^W, then there exists a nonempty subset of men, S, such that each man in S is better off, and each woman in S' is worse off under any stable matching for the new market than at any stable matching for the original market, when S' denotes the partners of men in S under matching M^M_w.*

Proof. After adding w to the market during the proposal-rejection process starting from M^M_w each man who gets a partner by accepting a proposal gets his worst possible partner at the end of the process by Theorem 2.2.4. So these men get the same partners as in M^W. But their new partners are strictly better than their original partners in M^M_w, who were actually their best stable partners for $G-w$. Similarly, each woman who gets a new partner during the process by making a proposal gets her best stable partner for G, so these women get the same partners as in M^W. But their new partners are strictly worse than their original partners in M^M_w, that were actually their worse stable partners for $G-w$. □

Pittel and Irving [80] considered the following situation. A new agent v enters the market, and a perfect stable matching (i.e. a stable matching where no agent is single) is achieved in such a way that the proposal-rejection sequence is as short as possible. They called this special half-matching with the associated alternating sequence a *core configuration relative to v*. Pittel and Irving [80] proved the following interesting property.

Theorem 2.2.11 (Irving–Pittel)**.** *If hM_v is a core configuration relative to v, then the associated proposal-rejection sequence $v = a_0, b_1, a_1, \ldots, a_{k-1}, b_k$ consists of $2k$ distinct persons, it is uniquely defined, and for every $i = 1 \ldots k-1$*

1. *b_i is the worst stable partner of a_i for $G - v$;*

2. *a_i is the best stable partner of b_i for $G - v$.*

We generalize Theorem 2.2.11 by extending the notion of core configuration. A stable half-matching hM_v is a *core configuration* relative to v if after adding v to the graph, the associated proposal-rejection sequence $S(hM_v)$ is as short as possible, by assuming that in case of cycling the sequence is restricted till b_k, where $a_i = b_k$ is the first return.

Theorem 2.2.12. *If hM_v is a core configuration relative to v, then the associated proposal-rejection sequence $a_0(= v), b_1, a_1, \ldots, a_{k-1}, b_k(, a_k)$ consists of distinct persons, it is uniquely defined. For every agent in the sequence, who is matched for G, the following properties are true:*

a) b_i is the worst stable partner of a_i for $G - v$ and b_{i+1} is the best stable partner of a_i for G;

b) a_i is the best stable partner of b_i for $G - v$ and a_{i-1} is the worst stable partner of b_i for G.

Proof. We shall construct a core configuration. Suppose that hM^0 is an arbitrary stable half-matching for G. Let a new agent u enter the market in such a way that u is acceptable only for v and u is the most preferred partner for v. Let us denote the proposal-rejection sequence by $S(hM^0)$ and the output stable half-matching for $G+u$ by hM^0_{+u}. Obviously, u and v are partners in any stable half-matching hM'_{+u} for $G+u$, moreover, hM'_{+u} is a stable half-matching for $G+u$ if and only if $hM'_v = hM'_{+u} \setminus \{u, v\}$ is a stable half-matching for $G - v$. So, by deleting $\{u, v\}$ from hM^0_{+u} we get a stable half-matching, say hM_v for $G - v$. We prove that hM_v is a core configuration relative to v. (We denote the associated proposal-rejection sequence by $S(hM_v)$ and the output stable half-matching for G by hM.)

To prove that $S(hM_v)$ is as short as possible, we show that each agent that is involved in $S(hM_v)$ must be involved in any other proposal-rejection sequence as well, and each agent occurs exactly once in $S(hM_v)$ (unless a new odd-cycle is created, when $a_i = b_k$ occurs twice.)

First, we prove that if $x \in S(hM_v)$ then $x \in S(hM'_v)$ for any stable half-matching hM'_v for $G - v$. We consider the cases according to the status of x (unmatched, cycle-agent or matched) in the stable half-matchings for $G - v$ and G.

1-2. No agent can be unmatched for $G - v$ and a cycle-agent for G at the same time, similarly, no agent can be a cycle-agent for $G - v$ and unmatched for G.

3-4. If an agent is unmatched/cycle-agent for $G-v$ and remains unmatched/cycle-agent

for G then he cannot be involved in any proposal-rejection sequence.
5. If x is matched for $G - v$ and becomes unmatched for G then $x = a_k$, so x is the last agent in $S(hM_v)$ (nobody accepts his proposal) and obviously x must be the last agent in any other $S(hM'_v)$ as well.
6. If x is unmatched for $G - v$ and becomes matched for G then $x = b_k$, so x is the last agent in $S(hM_v)$ (he accepts the last proposal) and obviously x must be the last agent in any other $S(hM'_v)$ as well.
7. If x is a cycle-agent for $G - v$ and becomes matched for G then $x = b_k$, so x is the last agent in $S(hM_v)$ (he accepts the last proposal). We prove that for any stable half-matching hM'_v, x is the last agent in $S(hM'_v)$ as well. Let $C = (c_0, c_1, \dots c_{2k})$ be the cycle that is eliminated when v enters the market. We suppose indirectly that two different cycle-agents $x = c_0$ and c_i accept the last proposals, made by y and y' in $S(hM_v)$ and $S(hM'_v)$ respectively. Obviously, the agent who made the final proposal is better than the predecessor of the cycle-agent who accepts it, (so $y >_{c_0} c_{2k}$ and $y' >_{c_i} c_{i-1}$). From Theorem 2.2.4, we also know that c_0 and c_i get their worst stable partners in hM and hM' respectively. This is a contradiction, because if i is even then c_i would be matched with c_{i-1} in hM and if i is odd then c_0 would be matched with c_{2k} in hM'.
8. If x is matched for $G - v$ and became a cycle-agent for G then x must occur in any proposal-rejection sequence until the first return, since Tan and Hsueh [105] proved that no new agent occurs in the sequence after the first return.
9. Finally, we consider the case where x is matched for $G - v$ and for G as well. Let us denote x's partners by y^0, y_v and y in hM^0, hM_v and hM, respectively.

a) If $y <_x y_v$, then x must receive y during $S(hM_v)$ by making a proposal, so by Theorem 2.2.4, y is the best stable partner of x for G. Thus, $y^0 \le y$ implies $y^0 <_x y_v$. It means that x must receive y_v during $S(hM^0)$ by accepting a proposal, so y_v is the worst stable partner of x for $G - v$. It follows that x gets a worse partner under any stable half-matching for G than at any stable half-matching for $G - v$, so x must be involved in any proposal-rejection sequence.

b) Similarly, if $y >_x y_v$, then x must receive y during $S(hM_v)$ by accepting a proposal, so by Theorem 2.2.4, y is the worst stable partner of x for G. Thus, $y^0 \ge y$ implies $y^0 >_x y_v$, it means that x must receive y_v during $S(hM^0)$ by making a proposal, so y_v is the best stable partner of x for $G - v$. It follows that x gets a better partner under any stable half-matching for G than at any stable half-matching for $G - v$, so x must be involved in any proposal-rejection sequence.

c) If $y = y_v$, then x cannot be involved in $S(hM_v)$.

Now, we prove that each agent occurs exactly once in $S(hM_v)$. Let us consider the above sequence with an extra stopping rule: if a_j looks for a new partner let us choose the best one among those who either form a blocking pair with a_j or some b_i with $i < j$

such that b_i prefers a_j to a_i (and not to his actual partner a_{i-1}). Assume that the first repetition (according to the extra stopping rule) would occur at b_{j+1}.

Case 1. If $b_i = b_{j+1}$ for some $i < j$ then let hM_{a_j} be the actual stable half-matching for $G - a_j$. We construct a new stable partition for $G - v$: $hM'_v = hM_{a_j} \cup \{a_j, b_i\} \setminus \{\{a_{p-1}, b_p\}, 1 \leq p \leq i\} \cup \{\{a_p, b_p\}, 1 \leq p \leq i-1\}$. It is stable, because by comparing with hM_v only agents $\{a_q, i \leq q \leq j\}$ get worse partners, but the extra stopping rule preserves that no edge $\{\{a_q, b_p\}, 1 \leq p < i \leq q \leq j\}$ can block hM'_v (and obviously no other edge).

In hM'_v every agent $\{b_q, i \leq q \leq j\}$ gets a better partner than in hM_v, and every agent $\{a_q, i \leq q \leq j\}$ gets a worse partner than in hM_v. If some of these agents is matched for $G - v$ and G as well, then it is a contradiction, because in hM_v they are matched with their best/worst stable partners, respectively.

The last case is that all of these agents are matched for $G - v$ and become a cycle-agent for G. These agents are in the same cycle (say $(c_0, c_1, \ldots, c_{2k})$) in hM^0 as well. So, when $S(hM^0)$ ends at c_0 by eliminating this cycle, each of these agents becomes matched in hM_v to either with his successor or with his predecessor (so $\{c_{2i-1}, c_{2i}\} \in hM_v$ for all $1 \leq i \leq k$). We show that a_{i-1} must also be a cycle-agent for G. Otherwise a_{i-1} must receive a worse partner than b_i in hM, and for b_i his predecessor is also worse than a_{i-1} (that is why b_i accepted the proposal of a_{i-1}). So a_{i-1} and b_i would block hM. By continuing this argument, for some $p < i$, a_p must be c_0, (the cycle-agent in hM_v that accepted the last proposal in $S(hM^0)$). But then b_{p+1} must be the predecessor of c_0: c_{2k}. Otherwise, if for some $1 \leq r < 2k$, $c_r = b_{p+1}$ then $c_{2k} <_{c_0} c_r$ (since c_{2k} is matched with c_{2k-1} in hM_v, so c_{2k} would accept the proposal of c_0) and $c_{r-1} <_{c_r} c_0$ (since c_r accepted the proposal of c_0), so c_0 and c_r would form a blocking pair in hM. Similarly, we can prove that the sequence goes along this odd-cycle, so for each d $(0 < d < j - p)$, $a_{p+d} = c_{2(k-d)+1}$ and $b_{p+d} = c_{2(k-d)}$. Finally, $b_i = b_{j+1}$ cannot be the predecessor of a_j in hM, a contradiction.

Case 2. If the first repetition is such that $a_i = b_{j+1}$, then the extra stopping rule was not used. This proves that a new odd-cycle can be created, so $hM = hM_{a_j} \setminus \{\{a_q, b_{q+1}\}, i \leq q \leq j\} \cup (a_i, a_j, b_j, a_{j-1}, \ldots, a_{i+1}, b_{i+1})$ is the output stable half-matching for G.

□

Theorem 2.2.12 implies the following generalization of Theorem 2.2.10.

Theorem 2.2.13. *Suppose that a new agent is added to the market. There may exist some agents that are better off, and some other agents that are worse off under any stable half-matching for the new market than at any stable half-matching for the original market. We can efficiently find all of these agents.* □

The arrival order determines the benefits

If we suppose that a centralized matching program uses the incremental algorithm, or if we model the dynamics of the matching market by the natural proposal-rejection process, then the obtained solutions are determined by the arrival order of the agents. We discuss here, as a consequence of the theorems from the last subsections, how the benefits of agents depend on the arrival orders.

Blum and Rothblum [25] realized that an agent can only benefit by arriving later to the market in the Roth-Vande Vate algorithm. By a similar argument, we can generalize this result for the nonbipartite case.

Lemma 2.2.14. *Assume that u is a matched agent for both $G-v$ and G, and let hM_v and hM'_v be two half-matchings for $G-v$ such that u gets at least as good partner in hM_v as in hM'_v (denoted by $hM_v \geq_u hM'_v$). Let hM and hM' be the outputs received by the proposal-rejection process after the arrival of v, respectively. Then u gets an at least as good partner in hM as in hM' (so $hM \geq_u hM'$).*

Proof. Indirectly, assume that u gets a better partner in hM' than in hM, so $hM <_u hM'$. This implies $hM_v >_u hM$ or $hM'_v <_u hM'$. In the first case u gets a worse partner by the proposal-rejection process, so by Theorem 2.2.9, u gets his best stable partner in hM, a contradiction. Similarly, in the second case u gets a better partner by the proposal-rejection process, so by Theorem 2.2.9, u gets his worst stable partner in hM', a contradiction. □

Lemma 2.2.15. *Let hM_v and hM'_v be two half-matchings for $G-v$ such that u is in the same situation in hM_v as in hM'_v, so u gets the same partner or u is unmatched or a cycle-agent. Let hM and hM' be the outputs received by the proposal-rejection process after the arrival of v, respectively. Then u is in the same situation in hM' as in hM, so u gets the same partner if he is matched for G.*

Proof. If u is matched for $G-v$, then Theorem 2.2.9 preserves the above property. If u is unmatched or a cycle agent for $G-v$, then the statement is an easy consequence of the points 6. and 7. from the proof of Theorem 2.2.12, respectively. □

Theorem 2.2.16. *In the incremental algorithm let two arrival orders σ and σ' differ only in one agent v in such a way that v arrives later in σ. Let hM and hM' be the outputs of the algorithm realized with the orders σ and σ', respectively. If v is a matched agent, then he gets an at least as good partner in hM as in hM'.*

Proof. Consider the market at the moment when v arrives according to σ. Now, the same agents are present in the market according to both arrival orders. Since v is the lastcomer according to σ, Theorem 2.2.6 implies that after the proposal-rejection processes v cannot be better off in the stable solution according to σ'. Afterwards, during the incremental algorithm, the same agents enter the market in each phase, so by the Lemmas 2.2.14 and 2.2.15 it follows, that v cannot be better off in the stable solutions according to σ' anymore. □

Let us define two relations, B^* and W^* between the agents in the following way: we denote by uB^*v if agent v occurs as an agent that makes a proposal in $S(hM_u)$ (the proposal-rejection sequence according to the core configuration relative to u). As an easy consequence of Theorem 2.2.12, uB^*v implies that v gets his best stable partner if u enters the market last, moreover the same theorem says, that v gets an even better partner if u does not enter the market at all. So v can only benefit if u is out of the market, or u arrives as late as possible. That is why we may regard u as a *nightmare-agent* of v.

Similarly, we denote by uW^*v if agent v occurs as an agent that receives a proposal in $S(hM_u)$. Here, uW^*v implies that v gets his worst stable partner if u enters the market last, but v gets an even worse partner if u is not present in the market. So v can only benefit if u is in the market, and he arrives as soon as possible. We call u as a *dream-agent* of v in this case.

Obviously, neither B^* nor W^* is symmetric. Moreover, the following Lemma proves, that uB^*v implies that vB^*u cannot be true, so the relation B^* is antisymmetric.

Lemma 2.2.17. *If uB^*v, then $S(hM_v)$ is the restriction of $S(hM_u)$.*

Proof. Directly follows from Theorem 2.2.12. □

Corollary 2.2.18. *The relation B^* is transitive, so uB^*v and vB^*w imply uB^*w. Moreover, uB^*v and vW^*w imply uW^*w.*

To show, that the relation B^* is not a linear order, one can easily find an example, where uB^*w and vB^*w, but there is no B^* relation between u and v. We believe that there should be some further relevant questions to consider about these relations.

2.2.4 The increasing side gets worse off

Finally, we give alternative proofs for certain results for two-sided matching markets. Lemma 2.2.19 is a straightforward consequence of Theorem 2 in [49].

Lemma 2.2.19. *If a man enters the market then no man can have a better partner in the new men-optimal stable matching than in the former men-optimal stable matching.*

Proof. Let m be the man that enters the market last. We shall prove that if a man m' gets w' in the men-optimal stable matching M^M, then m' cannot have a worse partner in the men-optimal stable matching M^M_m for $G-m$. If m is unmatched in M^M, then M^M is also stable for $G-m$. If $\{m,w\} \in M^M$, then $M^M \setminus \{m,w\}$ is stable for $G-\{m,w\}$. After w reenters the market, during the proposal-rejection process m' either remains matched with w' or receives a proposal from a better woman for him. □

Theorem 2.2.20. *If some men enter the market one after another then at the end of the proposal-rejection process they all get their best stable partners in the output stable matching.*

Proof. Suppose that a man m' is matched with his best stable partner w' before a new man, m enters. If m' remains matched with w' in the new obtained matching, then by Lemma 2.2.19, w' is still his best stable partner. If m' gets a new partner during the phase, then he must receive her by making a proposal, so Theorem 2.2.6 proves that m' gets his best stable partner again. □

The following theorem of Blum, Roth and Rothblum [24] can be proved the same way by using Theorem 2.2.9.

Theorem 2.2.21. *If some men enter the market then any other man m either remains matched with his original partner w if w is still a stable partner for m or m receives his best stable partner in the output.* □

If the arrival order is such that women enter the market first and men follow after that, then the output will be the same as the output of the deferred-acceptance algorithm with men proposing by Gale and Shapley [48]. Theorem 2.2.20 is an alternative proof for the statement that the obtained stable matching is optimal for the men.

2.3 Stable matching with ties, complexity results

In this section, we collect some results on the stable matching problem with ties. First, we study the problem of finding a stable matching according the three main stability concepts. Then, we consider the problem of finding a matching for which the number of blocking pairs is minimal.

We recall, the three main stability concepts for the stable matching problem with ties.

- A matching M is *weakly stable* if there exist no blocking edge $e = \{a_i, a_j\}$ where both agents strictly prefer the other to his partner in M.
- A matching is *strongly stable* if there exist no blocking edge $e = \{a_i, a_j\}$ such that one of the agents, say a_i strictly prefers a_j to his partner in M and a_j either strictly prefers a_i to his partner in M or is indifferent between them.
- A matching M is *super stable* if there exist no blocking edge $e = \{a_i, a_j\}$ where both agents either strictly prefers the other to his partner in M or are indifferent between them.

We note that weak stability is regarded as stability in general. If for a given instance I of the stable matching problem with ties, we break the ties, then the obtained instance I' of stable matching problem is called the *derived problem.* Obviously, a matching that is stable for some derived problem is also stable for the original one.

While, the above definitions are also correct in case the graph has parallel edges, here we restrict our attention to simple graphs, and call the nonbipartite case as stable roommates problem with ties, and the bipartite case as stable marriage problem with ties.

2.3.1 Finding a stable matching

Considering the stable roommates problem with ties, the problem of finding a weakly stable matching, denoted by WEAKLY SRT, is NP-complete. This was proved first by Ronn [82] for complete graphs. Later, Irving and Manlove [59] verified the same result by a different proof for the more general case, when the lists can be incomplete. Moreover, this result holds even if each tie is of length 2, i.e. there is at most one tie per list, and the vertices that have a tie in their list are not connected.

Theorem 2.3.1 (Ronn; Irving-Manlove)**.** *The problem of finding a weakly stable matching in an instance of the stable roommates problem is NP-complete [R3].*

On the other hand, Irving [57] constructed two polynomial algorithms which determine whether a given instance of the stable marriage problem with ties admits a {strongly, super-} stable matching, and if so find one. He proved these results for complete graphs, later Manlove [73] verified similar results for incomplete lists as well. The same questions are tractable also for stable roommates problems with ties. This was verified by Irving and Manlove [59] in case of super-stability and by Scott [99] in case of strong-stability.

For a stable matching problem it is a natural question to find such a stable matching that has maximal cardinality. If the preferences are strict then this problem is not interesting, since every stable matching has the same size by Theorem 2.2.15. Considering weak stability in case of ties Manlove *et al.* [74] proved the following theorem for bipartite graphs.

Theorem 2.3.2 (Manlove *et al.*, 2002)**.** *The decision problem related to finding the maximum size of weakly stable matchings for a given instance of the stable marriage problem with ties, denoted by* MAX WEAKLY SMT*, is NP-complete, even if ties occur only in one side of the bipartite graph. [R4].*

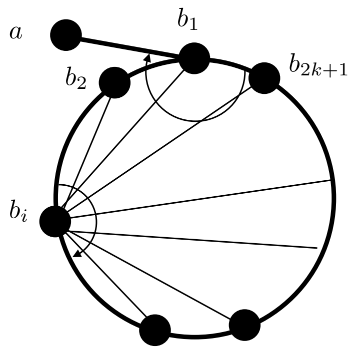
The problems of finding a maximum size strongly and super-stable matchings are solvable: Manlove [73] proved that for a given instance of the stable marriage problem, the same agents are matched in each strongly stable matching, and similarly, the same agents are matched in each super stable-matching. Similar results were proved in the roommates case by Irving and Manlove [59] for super-stable matchings and by Scott [99] for strongly stable matchings.

2.3.2 "Almost stable" matchings

Here we present some results of Abraham, Biró and Manlove, published in [5]. For a stable roommates problem, where no stable matching exists, it is a natural question to find a matching that admits the fewest number of blocking pairs: it can be regarded as a matching that is "as stable as possible".

Let us show first an example that is used as a gadget in the proofs of the presented theorems.

Example 2.3.3. *We have given $2k+2$ agents: $\{b_1, b_2, \ldots, b_{2k+1}\} = B$ and a. The preference of the agents are the following:*



$$
\begin{array}{rl}
a: & [b_1, \text{ arbitrary }] \\
b_1: & [a, b_2, b_3, \ldots, b_{2k}, b_{2k+1}] \\
b_2: & [b_3, b_4, \ldots, b_{2k+1}, b_1, a] \\
\ldots & \ldots \\
b_i: & [b_{i+1}, b_{i+2}, \ldots, b_{2k+1}, b_1, \ldots, b_{i-1}, a] \\
\ldots & \ldots \\
b_{2k+1}: & [b_1, b_2, \ldots, b_{2k}, a]
\end{array}
$$

This graph can be considered as a *"bomb"*, where edge $\{a, b_1\}$ is called *priming*. One can easily check that if $\{a, b_1\}$ is part of a matching M, then M can be stable if the remaining agents are matched as follows: $\{b_2, b_{k+2}\}, \{b_3, b_{k+3}\}, \ldots, \{b_i, b_{k+i}\} \ldots \{b_{k+1}, b_{2k+1}\} \in M$. On the other hand, if $\{a, b_1\} \notin M$, then the number of blocking pairs is at least k, so the bomb is exploding. To prove this, let b_i be an agent, who has no partner from B in M. If $\{b_j, b_k\} \in M$, then obviously either $\{b_i, b_j\}$ or $\{b_i, b_k\}$ is a blocking pair. If another agent b_l has no partner from B either, then the pair $\{b_i, b_l\}$ is blocking, so b_i must belong to at least k blocking pairs.

Define MIN-MM (respectively EXACT-MM) to be the problem of deciding, given a graph G and integer K, whether G admits a maximal matching of size at most (respectively exactly) K. Both problems are NP-complete even for cubic graph. This was proved in [53] for MIN-MM and in [5] for EXACT-MM.

Theorem 2.3.4. *The problem of finding a matching in a given an instance of stable roommates problem for which the number of blocking edges is minimal is not approximable within $n^{\frac{1}{2}-\varepsilon}$, for any $\varepsilon > 0$, unless $P = NP$. The result holds even for complete preference lists. [R5]*

Sketch of the proof. The proof is based on a gap-introducing reduction starting from EXACT-MM for a cubic graph G of size p with a positive integer K. We create an instance I of the stable roommates problem of size n, where $n >> p$ and p is fixed, so p can be considered as a constant relative to n. The instance I contains $\Theta(\sqrt{n})$ bombs of size $\Theta(\sqrt{n})$. The construction of I ensures that on the one hand, if every priming is part of a matching M' in I, so no bomb is exploded, then the minimum number of blocking pairs is bounded by a constant relative to n. On the other hand, if any of the priming is not part of M', so some bomb is exploded, then the minimum number of blocking pairs is at least $\Theta(\sqrt{n})$. In the proof it is verified that G has a maximal matching M of size K if and only if no bomb is exploded in I. Thus, an algorithm that could approximate the minimum number of blocking pairs in polynomial time within a factor less than $\Theta(\sqrt{n})$ would be able to decide EXACT-MM for cubic graphs as well.

Theorem 2.3.5. *The problem of finding a matching in a given an instance of stable roommates problem with ties for which the number of blocking edges is minimal is not approximable within $n^{1-\varepsilon}$, for any $\varepsilon > 0$, unless $P = NP$. The result holds even for complete preference lists. [R6]*

Sketch of the proof. The idea is similar to the above proof. The only difference here is that I contains a constant number of bombs of size $\Theta(n)$. By this it follows that the existence of an approximation algorithm with a factor less than $\Theta(n)$ would imply $P = NP$.

We note finally, that for bipartite graphs similar inapproximability results were proved recently by Manlove (personal communication), in case the matching is restricted to be a complete matching. [R7]

In the following table we collect the above results. Here, P denotes that the problem is polynomial time solvable, NPc denotes that the (related) problem is NP-complete, (NPh) denotes that the NP-hardness of the problem is obvious from the presented results.

The problem is to find a matching M,	where M	bipartite graph (strict)	bipartite graph with ties	arbitrary graph (strict)	arbitrary graph with ties
s.t. M is stable	(arb.)	Yes [R1]	(Yes)	P [R2]	NPc [R3]
	max	(P)	NPc [R4]	(P)	(NPh)
s.t. M has min no. of blocking pairs	(arb.)	(=0)	(=0)	NPc [R5]	NPc [R6]
	max	NPc [R7]	(NPh)	(NPh)	(NPh)

Chapter 3

Stable allocation problems

Introduction

In this chapter we study stable matching problems with vertex-bounds and edge-capacities. To introduce these notions, we recall Scarf's lemma. Then we show that if all the bounds and capacities are integers, then the so-called integral stable allocation problem for graphs can be reduced to the stable roommates problem by a sequence of constructions. We describe a strongly polynomial algorithm created by Baïou and Balinski for two-sided matching markets, and we study its generalization in the one-sided case. Finally, as a practical application of stable b-matchings, we describe the higher education admission program in Hungary on the basis of paper [18].

3.1 Stable allocation problem by Scarf's lemma

3.1.1 Fractional b-core element

In what follows, we introduce the notion of fractional b-core element as a solution of Scarf's lemma with the original settings. Let the same matrices A and C of dimension $n \times (n+m)$ correspond to the set of effective utility vectors (or activities) in a given finitely generated NTU-game as it was described in Section 1.1.3. The only modification is that now b is an arbitrary vector of $\mathbb{R}^n_+$ (instead of 1^N). Let $x \in \mathbb{R}^{n+m}_+$ be referred as a *fractional b-core element* if x is a solution of Scarf-lemma for the above setting.

Here, $b(i)$ is an upper bound for the total intensity of which player i is capable to perform activities, since $\sum_{i\in S} x(u^{S,l}) = b(i)$. The domination condition of the lemma says that for every activity $\mathfrak{a}$ there exists some player i, who is not interested in increasing the intensity of $\mathfrak{a}$, because his remaining intensity $b(i)$ is filled with better activities, so if $u^{T,l}$ corresponds to activity $\mathfrak{a}$, then $\sum_{u_i^{S,k} \geq u_i^{T,l}} x(u^{S,l}) = b(i)$.

In fact, to produce a fractional core element (in other words, a fractional 1^N-core element) by the algorithm of Scarf, we should perturbate not just matrix C (in case of indifferences at the beginning), but also the vector 1^N, to avoid the degeneracy. The standard nondegeneracy assumption provides that all variables associated with the n columns of a feasible basis for the equations $A\tilde{x} = \tilde{b} = 1^N + \varepsilon^N$ are strictly positive. Thus, the perturbation uniquely determines the steps of Scarf algorithm. By rounding the final fractional $\tilde{b}$-core element $\tilde{x}$, a fractional core element x is found.

The following simple Lemma says that the fractional b-core element has the *scaling property.*

Lemma 3.1.1. *Given a finitely generated NTU-game, and a positive constant λ. Suppose that $b' = \lambda \cdot b$, then x is a fractional b-core element if and only if $x' = \lambda \cdot x$ is a fractional b'-core element.*

An easy consequence is that by Theorem 1.3.5, for every finitely generated matching game, there always exists an integer 2^N-core element. That is actually a half-core element for the original setting.

3.1.2 Fractional b-core element with capacities: stable allocation

Assume now, that the intensities of the activities in the finitely generated NTU-game are bounded by capacities. Formally, for each common activity $\mathfrak{a}$ and for the corresponding utility vector $u^{S,l}$, there may exist a nonnegative capacity $c(u^{S,l})$ for which $x(u^{S,l}) \leq c(u^{S,l})$ is required.

The stable allocation problem can be defined for hypergraphs as follow. A hypergraph H is given and for each vertex v a strict preference order over the edges incident with v (this corresponds again to the preferences of the players over the activities, in which they can be involved). Suppose, that nonnegative *bounds* on the vertices $b : V(H) \to \mathbb{R}_+$, and nonnegative *capacities* on the edges $c : E(H) \to \mathbb{R}_+$ are fixed. A nonnegative function x on the edges, is an *allocation* if $x(e) \leq c(e)$ for every edge e and $\sum_{v \in h} x(h) \leq b(v)$ for every vertex v. An allocation is *stable* if every *unsaturated* edge e (i.e. $x(e) < c(e)$) contains a vertex v such that $\sum_{v \in h, e \leq_v h} x(h) = b(v)$. In this case we say that e is *dominated at* v.

The stable allocation problem was introduced by Baïou and Balinski [14] for bipartite graphs. The integer version, (i.e. if the allocation x is required to be integer on every edge for integer bounds and capacities) was called the *stable schedule problem* by Alkan and Gale [10], however they considered a more general model, the case of so-called *substitutable* preferences. Here, instead of stable schedule problem, we call the integral version of the stable allocation problem simply as the *integral stable allocation problem.*

The following theorem was presented in [17] as a joint result with Tamás Fleiner.

Theorem 3.1.2. *Every stable allocation problem for hypergraphs is solvable.*

Proof. Let $V(H) = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices, and let $E(H) = \{e_1, e_2, \ldots, e_m\}$ be the set of edges in a given hypergraph H. We define the extended membership-matrix A, and the extended preference-matrix C of size $(n+m)\times(n+2m)$ as follows.

The left part of A is an identity matrix of size $(n+m)\times(n+m)$, (i.e. $a_{i,j} = \delta_{i,j}$ for $i, j \in [n+m]$). At the bottom of the right side there is another identity matrix of size $m \times m$, so $a_{n+i,n+m+j} = \delta_{i,j}$ for $i, j \in [m]$. Finally, at the top of the right side we have the vertex-edge incidence matrix of H (i.e. $a_{i,n+m+j} = 1$ if $v_i \in e_j$ and 0 otherwise for $i \in [n]$ and $j \in [m]$).

The top-right part of C correspond to the preference of the vertices (that is the preference of the players over the activities). We require the following two conditions:

- $c_{i,n+m+j} < c_{i,n+m+k}$ whenever $v_i \in e_j \cap e_k$ and $e_j <_{v_i} e_k$;
- $c_{i,n+m+j} < c_{i,n+m+k}$ whenever $v_i \in e_j \setminus e_k$.

Furthermore, suppose that $c_{n+i,n+m+i} < c_{n+i,n+m+j}$ for every $i \neq j \in [m]$ in the bottom-right part of C. Finally, let the left part of C be such that it satisfies the conditions of Scarf's lemma.

Finally, the constant vector, $b \in R_+^{n+m}$ is given by the bounds and capacities, let $b_i = b(v_i)$ for $i \in [n]$ and $b_{n+j} = c(e_j)$ for $j \in [m]$.

We shall prove that the fractional core element x, obtained by Scarf's lemma, gives a stable allocation, x^e by simply taking the last m coordinates of x. Here, x_j^e is equal to $x^e(e_j)$ that is the weight of the edge e_j (or equivalently, this is the intensity, the corresponding activity is performed with). If $\bar{x}^v$ and $\bar{x}^e$ are the vectors obtained by taking the $[1, \ldots, n]$ and $[n+1, \ldots, n+m]$ coordinates of x, then these vectors correspond the remaining weights of the vertices and edges (or the remaining intensities of the players and the activities), respectively.

Obviously, x^e is an allocation by $Ax = b$, since the first n equations preserve the $\sum_{v\in h} x^e(h) \leq b(v)$ condition for every vertex v, and the last m equations preserve $x^e(e) \leq c(e)$ for every edge e.

To prove stability, let us consider an unsaturated edge e_j and let us suppose that the corresponding row by the lemma is k. First we show, that $i \in [n]$. From $Ax = b$, obviously $\bar{x}^e(e_k)+x^e(e_k) = c(e_k)$ for every edge e_k. Since $x^e(e_j) < c(e_j)$, then $\bar{x}^e(e_j) > 0$, thus the assumptions on C imply that $i \neq n+j$, for other $i \in [n+m] \setminus [n]$ the contradiction is trivial. If $i \in [n]$, then e is dominated at v_i, since $\bar{x}^v(v_i) = 0$ by the assumptions on C, and the $Ax = b$ condition for the i-th row with the statement of the lemma imply $\sum_{v_i\in h, e_j \leq_{v_i} h} x^e(h) = b(v_i)$. □

The following theorem is an easy consequence of Theorem 3.1.2. In the next subsection we give other evidences that verify this theorem.

Theorem 3.1.3. *For every integral stable allocation problem in a graph there exists a half-integer stable allocation. If the graph is bipartite, then every integral stable allocation problem is solvable.*

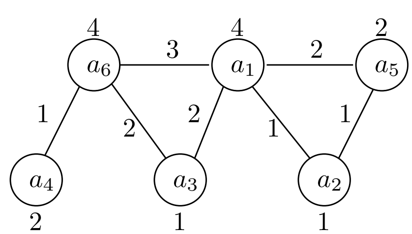
Proof. Suppose that we have a stable allocation x that has some weights that are not half-integers. We create another stable allocation x' with half-integer weights. If $x(e)$ is not integer, then let v be the vertex, where e is dominated. Since $b(v)$ is integer, there must be another edge f, that is incident with v and has non-integer weight. Moreover, f cannot be dominated at v. By this argument, it can be verified that the edges with non-integer weights form vertex-disjoint cycles, and in a such cycle, the fractional parts of the weights are ε and $1-\varepsilon$ alternately. If a cycle is odd, then ε must be $\frac{1}{2}$. If a cycle is even, then ε can be modified to be 0 (or 1), the obtained allocation x' remains stable and has half-integer weights.

If the graph is bipartite, thus has no odd cycle, then x' has only integer weights, so it is a integral stable allocation . □

We note that the scaling property holds also for the stable allocation problem. By this, if every bound and capacity is even, then the existence of an integral stable allocation is straightforward by Theorem 3.1.3.

In the next example, we illustrate a integral stable allocation problem in a graph for which a half-integer stable allocation is obtained by the algorithm of Scarf.

Example 3.1.4. *Here, 6 agents and 7 possible pairwise activities are given. The bounds of the agents are:* $b(a_1) = 4$, $b(a_2) = 1$, $b(a_3) = 1$, $b(a_4) = 2$, $b(a_5) = 2$ *and* $b(a_6) = 4$. *The capacities of the possible activities are:* $c(\{a_1, a_2\}) = 1$, $c(\{a_1, a_3\}) = 2$, $c(\{a_1, a_5\}) = 2$, $c(\{a_1, a_6\}) = 3$, $c(\{a_2, a_5\}) = 1$, $c(\{a_3, a_6\}) = 2$ *and* $c(\{a_4, a_6\}) = 1$. *The preference-lists of the agents are the following:*



$$\begin{array}{ll} a_1: & [a_6, a_3, a_2, a_5] \\ a_2: & [a_5, a_1] \\ a_3: & [a_6, a_1] \\ a_4: & [a_6] \\ a_5: & [a_1, a_2] \\ a_6: & [a_4, a_3, a_1] \end{array}$$

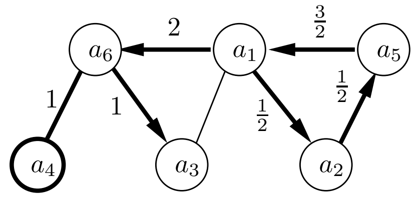
The extended membership-matrix A is the following:

$$A = \left[\begin{array}{cccccc|ccccccc|ccccccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&1\\
-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-\\
0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1
\end{array}\right]$$

The matrix C that represents the preference of the players can be generated as follows:

$$C = \left[\begin{array}{cccccc|ccccccc|ccccccc}
0&24&23&22&21&20&19&18&17&16&15&14&13&3&4&2&5&8&7&6\\
25&0&23&22&21&20&19&18&17&16&15&14&13&4&11&10&9&5&7&6\\
25&24&0&22&21&20&19&18&17&16&15&14&13&12&4&10&9&8&5&6\\
25&24&23&0&21&20&19&18&17&16&15&14&13&12&11&10&9&8&7&5\\
25&24&23&22&0&20&19&18&17&16&15&14&13&12&11&5&9&4&7&6\\
25&24&23&22&21&0&19&18&17&16&15&14&13&12&11&10&3&8&4&5\\
-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-&-\\
25&24&23&22&21&20&0&18&17&16&15&14&13&1&11&10&9&8&7&6\\
25&24&23&22&21&20&19&0&17&16&15&14&13&12&1&10&9&8&7&6\\
25&24&23&22&21&20&19&18&0&16&15&14&13&12&11&1&9&8&7&6\\
25&24&23&22&21&20&19&18&17&0&15&14&13&12&11&10&1&8&7&6\\
25&24&23&22&21&20&19&18&17&16&0&14&13&12&11&10&9&1&7&6\\
25&24&23&22&21&20&19&18&17&16&15&0&13&12&11&10&9&8&1&6\\
25&24&23&22&21&20&19&18&17&16&15&14&0&12&11&10&9&8&7&1
\end{array}\right]$$

In the algorithm, we used the following perturbation on b: we set $\tilde{b}_i = b_i + \varepsilon_i = b_i + 1/p_{101-i}$, where p_i is the i-th prime number. By rounding the output $\tilde{x}$ of the Scarf algorithm, we get the following fractional b-core element x, and stable allocation x^e:



$x = [\bar{x}^v|\bar{x}^e|x^e]$, where $x^e = [\frac{1}{2}, 0, \frac{3}{2}, 2, \frac{1}{2}, 1, 1]$

$\bar{x}^v = [0, 0, 0, 1, 0, 0]$ and $\bar{x}^e = [\frac{1}{2}, 2, \frac{1}{2}, 1, \frac{1}{2}, 1, 0]$

In the above figure, we oriented each unsaturated edge to its endvertex, where it is dominated.

3.2 Integral stable allocation problem for graphs

3.2.1 Special cases

If, for a integral stable allocation problem $c(e) = 1$ for every edge e, then this special case is called a *stable b-matching problem* (see Fleiner [46]). If the graph may contain parallel edges, then this problem was referred to as a *stable multiple activities problem* by Cechlárová and Fleiner [31]. In case of simple graphs, this problem was called a *stable fixtures problem* by Irving and Scott [60]. If $b(v) = 1$ for every vertex, then the stable matching problem is obtained, that is called stable roommates problem for simple graphs.

Furthermore, if the given graph is simple and bipartite, then the stable b-matching problem can be called *many-to-many stable matching problem*. If $b(v) = 1$ for every vertex of one of the sides, then the problem can be referred to as *many-to-one stable matching*, *college admission* or *hospitals/residents* problem. Finally, if $b(v) = 1$ for every vertex, then we get the stable marriage problem.

3.2.2 Reduction by constructions

Here, we present a result of Cechlárová and Fleiner [31], and we extend it by one trivial primary step. This argument says that every integral stable allocation problem on a graph can be reduced graph constructions to a stable roommates problem. However, this reduction is not polynomial.

1st step: creating parallel edges in order to omit the capacities

Given a integral stable allocation problem for a graph G^0. We reduce the problem to a stable b-matching problem for a graph G^1 as follows.

Let G^1 has the same vertex-set as G^0, with the same bounds, so $V(G^1) = V(G^0)$, and if v' corresponds to v, then $b(v') = b(v)$. If $\{u, v\} = e \in E(G^0)$ and $c(e) = l$, then let $e'_1, e'_2, \ldots, e'_l \in E(G^1)$, where $e'_i = \{u, v\}$ and $c(e'_i) = 1$ for every index $i = 1, \ldots l$. Preferences are the following: if $e <_v f$ in G^0, then let $e'_i <_{v'} f'_j$ in G^1 for each pair of indices i, j. Moreover, let $e'_i >_{v'} e'_j$ whenever $i < j$.

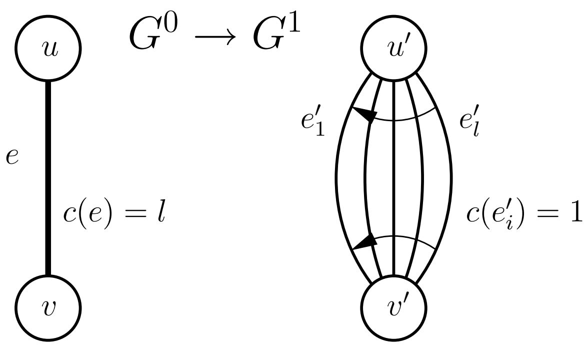


Figure 3.1: Creating parallel edges

Suppose that x^0 is a stable allocation for G^0, we shall construct a stable b-matching x^1 for G^1. If $x^0(e) = k$, then let $x^1(e'_i) = 1$ for every $i = 1, \ldots, k$. Obviously, x^1 is a b-matching. If an edge $e'_i \in E(G^1)$ is not in the b-matching, so if $x^1(e'_i) = 0$, and e is dominated at v in G^0, then $\sum_{v \in h, e \leq_v h} x^0(h) = b(v) = b(v') = \sum_{v' \in h, e'_i \leq_{v'} h} x^1(h)$, so e'_i is dominated at v'.

In the other direction, suppose that x^1 is a stable b-matching for G^1. Let $x^0(e) = \sum_{i=1}^{l} x(e'_i)$, x^0 is obviously a integral stable allocation for G^0.

We note, that here, $x^1(e'_j) = 1$ implies $x^1(e'_i) = 1$ for every $i < j$, since otherwise e'_i cannot be dominated. This observation yields to a one-to-one correspondence between the integral stable allocation s of G^0 and the stable b-matchings of G^1.

2nd step: introducing 6-cycles on the edges

Given a stable b-matching problem for a graph G^1. We reduce the problem to another stable b-matching problem for a graph G^2, where G^2 is simple and satisfies the *many-to-one property* (i.e. $\{u, v\} \in E(G^2) \Rightarrow b(u) = 1$ or $b(v) = 1$).

If $v \in V(G^1)$ then let $v' \in V(G^2)$ with the same bounds, $b(v') = b(v)$. If $\{u, v\} = e \in E(G^1)$, then let $u'_{e,1}$, $u'_{e,2}$, $u'_{e,3}$, $v'_{e,1}$, $v'_{e,2}$, $v'_{e,3} \in V(G^2)$ with unit bounds, and $\{u', u'_{e,1}\}$, $\{v', v'_{e,1}\}$, $\{u'_{e,1}, u'_{e,2}\}$, $\{u'_{e,2}, v'_{e,3}\}$, $\{v'_{e,3}, v'_{e,1}\}$,$\{v'_{e,1}, v'_{e,2}\}$, $\{v'_{e,2}, u'_{e,3}\}$, $\{u'_{e,3}, u'_{e,1}\} \in E(G^2)$ (with unit capacities), with the following preferences:

$$\begin{array}{llllll} u'_{e,1}: & [u'_{e,2}, u', u'_{e,3}] & u'_{e,2}: & [v'_{e,3}, u'_{e,1}] & u'_{e,3}: & [u'_{e,1}, v'_{e,2}] \\ v'_{e,1}: & [v'_{e,2}, v', v'_{e,3}] & v'_{e,2}: & [u'_{e,3}, v'_{e,1}] & v'_{e,3}: & [v'_{e,1}, u'_{e,2}] \end{array}$$

Suppose that x^1 corresponds to a stable b-matching M^1 for G^1 (i.e. $e \in M^1 \Leftrightarrow x^1(e) = 1$). We can construct another stable b-matching, M^2 for G^2

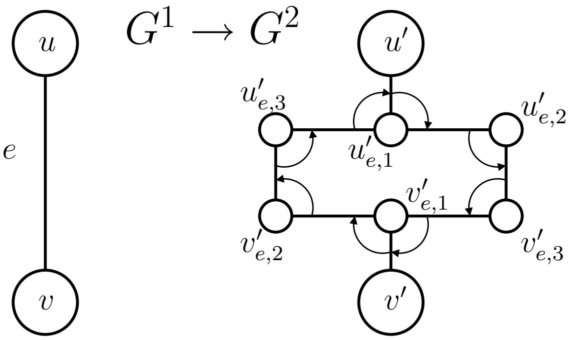


Figure 3.2: Introducing 6-cycles

as follows:
If $\{u, v\} = e \in E(G^1)$, then
- $e \in M^1 \Rightarrow \{u', u'_{e,1}\}, \{u'_{e,2}, v'_{e,3}\}, \{v'_{e,2}, u'_{e,3}\}, \{v', v'_{e,1}\} \in M^2$
- $e \notin M^1$ and e is dominated at $v \Rightarrow \{u'_{e,1}, u'_{e,2}\}, \{v'_{e,3}, v'_{e,1}\}, \{v'_{e,2}, u'_{e,3}\} \in M^2$

It can be verified that M^2 is a stable b-matching for G^2 (see [31]).

In the other direction, suppose that x^2 corresponds to a stable b-matching M^2 for G^2. Let us create M^1 is the following way:
- $\{u', u'_{e,1}\}, \{u'_{e,2}, v'_{e,3}\}, \{v'_{e,2}, u'_{e,3}\}, \{v', v'_{e,1}\} \in M^2 \Rightarrow e \in M^1$

It can be verified that M^1 is a stable b-matching for G^1 (see [31]).

3rd step: b-expansion of the vertices in order to omit the bounds

Given a stable b-matching problem for a graph G^2, where G^2 is simple and satisfies the many-to-one property. We reduce the problem to a stable roommates problem for a graph G^3 as follows.

If $u \in V(G^2)$ and $b(u) = 1$, then let $u' \in V(G^3)$ with $b(u') = 1$. If $v \in V(G^2)$ and $b(v) = k$, then let $v'_1, v'_2, \ldots, v'_k \in V(G^3)$ with $b(v'_i) = 1$. If $\{u, v\} \in E(G^2)$ and $b(u) = 1$, $b(v) = k$, then let $\{u', v'_1\}, \{u', v'_2\}, \ldots, \{u', v'_k\} \in E(G^3)$. Considering the preferences: if $\{u, v\} <_u \{u, w\}$ in G^2, then let $\{u', v'_i\} <_{u'} \{u', w'_j\}$ for every pair of indices i, j, and let $\{u', v'_j\} <_{u'} \{u', v'_i\}$ for $i < j$ in G^3.

Suppose that M^2 is a stable b-matching for G^2. We can construct a stable matching M^3 for G^3 as follows: If $\{u, v\} \in M^2$ for $b(u) = 1$ and $b(v) = k$, where $|\{e : e \in M^2, e \geq_v \{u, v\}| = j$ then let $\{u', v'_j\} \in M^3$.
It can be verified that M^3 is a stable matching for G^3 (see [31], or [52] p.38 and [92] p. 131-132 in the bipartite case).

In the other direction, suppose that M^3 is a stable matching for G^3. Let $\{u, v\} \in M^2$ whenever $\{u', v'_i\} \in M^3$ for some i. It can be verified that M^2 is a stable b-matching

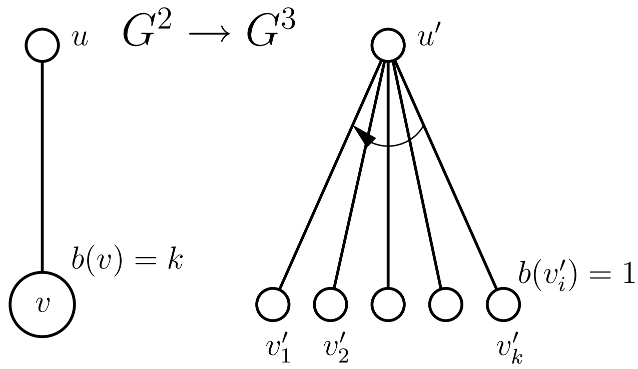


Figure 3.3: b-expansion of the vertices

for G^2 (see [31], or [52] p.38 and [92] p. 131-132 in the bipartite case).

3.2.3 Generalizations of the basic algorithms

By the above argument it is not surprising that every algorithm that solves the stable roommates or the stable marriage problem can be generalized to solve the integral stable allocation problem as well.

In the bipartite case, the generalized version of the deferred-acceptance algorithm of Gale and Shapley was applied for the many-to-one stable matching problem (or college admission problem) in their paper [48]. Its straightforward generalization was presented by Baïou and Balinski [14] for the integral stable allocation problem. We note that Gale–Shapley type algorithms were created to solve stable b-matching problems for more general preferences, or choice functions by Kelso and Crawford [66], Roth [84], [85], Blair [23], Fleiner [44], [46], [45], Alkan [9], Eguchi, Fujishige and Tamura [40], and for integral stable allocation problems by Alkan and Gale [10], Fujishige and Tamura [47].

Considering the nonbipartite case, the generalized version of Irving's algorithm [56] was used to solve the stable fixtures problem in [60] and for the stable multiple activities problem in [31] and [35].

However, we must remark, that the details of the generalization can be crucial considering the running time of the algorithm. The main result of Cechlárová and Viera [35] is that their implementation preserves an $O(m)$ running time also for their general settings.

On the other hand, the straightforward generalization of the Gale–Shapley algorithm becomes slow for the integral stable allocation problem. Baïou and Balinski [14] showed by an example with 4 vertices and 4 edges, that the number of rounds in the algorithm can be equal to the sum of the bounds.

In the next section we present the algorithm of Baïou and Balinski [14], that provide a stable allocation for every stable allocation problem for bipartite graphs in strongly polynomial time. (Thus, its running time does not depend on the bounds and capacities, only on the number of vertices and edges.) Then we generalize their inductive algorithm for integral stable allocation problems in the nonbipartite case, however we show by an example that this algorithm does not remain polynomial.

Actually, these inductive algorithms are also a kind of generalizations of the Roth-Vande Vate and the Tan–Hsueh algorithms, respectively. Thus, they can be used to model dynamic market situations. Moreover, the obtained stable allocations by these algorithms may have special properties, similar to the properties of the obtained stable matchings and half-matchings described in Section 2.2.

3.2.4 The inductive algorithm for bipartite graphs

Let us suppose that we have given a stable allocation problem for a (not necessarily bipartite) graph G with vertex-bounds b and edge-capacities c. Suppose that x is a stable allocation.

Recall, that an edge is *saturated* if $x(e) = c(e)$, and a vertex v is saturated if $x(v) := \sum_{v\in e} x(e) = b(v)$. Finally, edge e is dominated at vertex v if $\sum_{v\in e, e\leq_v h} x(h) = b(v)$. (Note, that an allocation is stable, if every non-saturated edge is dominated at some of its endvertices.)

Furthermore, let us define $l(v)$ to be the edge e, that is dominated at v and $x(e) > 0$. Let $f(v)$ be the edge e, that is dominated only at v, e is not saturated, and there exist no other edge f that satisfies the same property with $e <_v f$. (Note that $l(v)$ or $f(v)$ may not exist, but they are unique if they exist.)

Lemma 3.2.1. *Given a stable allocation problem for a graph G with vertex-bounds b and edge-capacities c, and given a stable allocation x.*

a) Suppose that $\{u, v\} = l(v)$ and $\varepsilon \leq x(\{u, v\})$.

$$\textit{Let } b'(w) = \begin{cases} b(w) - \varepsilon & \textit{if } w = u, v \\ b(w) & \textit{otherwise.} \end{cases} \quad \textit{Then } x'(e) = \begin{cases} x(e) - \varepsilon & \textit{if } e = \{u, v\} \\ x(e) & \textit{otherwise} \end{cases}$$

is a stable allocation for the modified bounds b'.

b) Suppose that $\{v, w\} = l(v)$ and $\varepsilon \leq c(\{v, w\}) - x(\{v, w\})$.

$$\textit{Let } b'(u) = \begin{cases} b(u) + \varepsilon & \textit{if } u = v, w \\ b(u) & \textit{otherwise.} \end{cases} \quad \textit{Then } x'(e) = \begin{cases} x(e) + \varepsilon & \textit{if } e = \{v, w\} \\ x(e) & \textit{otherwise} \end{cases}$$

is a stable allocation for the modified bounds b'.

Proof. a) Obvious, since $\{u, v\}$ remains dominated at v, and every other edge that was dominated at u or v remains dominated there.

b) Obvious, since $\{v, w\}$ remains dominated at v, and every other edge that was dominated at v or w remains dominated there.

□

Augmenting procedures

Suppose that our goal is to increment the bound of a vertex v from $b(v)$ to $b'(v) = b(v)+\varepsilon$, by simultaneously modifying x to x' in such a way that x' is a stable allocation for the new settings.

If $f(v)$ does not exist, then obviously $x' = x$ remains stable. Otherwise, we will construct, a so-call *augmenting path*, to conduct the improvement of $x(v)$. To create such a path, let $a_0 = v$, and let the $2i$-th and the $(2i + 1)$-th element of the sequence be

- b_i if $\{a_{i-1}, b_i\} = f(a_{i-1})$ and
- a_i if $\{b_i, a_i\} = l(b_i)$, respectively

for every index i from $i = 1$ while such vertex exists and no repetition occurs.

Depending on the termination of the above augmenting path, under the assumption that G is bipartite graph, we have 4 cases:

I1) $f(a_k)$ does not exist: the augmenting path is $a_0, b_1, a_1, \ldots, b_k, a_k$.

Let $\varepsilon_l := \min_{1\leq j\leq k}\{x(\{b_j, a_j\})\}$, $\varepsilon_f := \min_{1\leq j\leq k}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon = \min\{\varepsilon_l, \varepsilon_f\}$ and $b'(v) = b(v) + \varepsilon$. Finally, let

$$x'(e) = \begin{cases} x(e) + \varepsilon & \text{if } e = \{a_{j-1}, b_j\} \text{ for } 1 \leq j \leq k, \\ x(e) - \varepsilon & \text{if } e = \{b_j, a_j\} \text{ for } 1 \leq j \leq k \text{ and} \\ x(e) & \text{otherwise.} \end{cases}$$

Then x' is a stable allocation for b' by Lemma 3.2.1 and by the definition of $f(v)$.

I2) $l(b_k)$ does not exist: the augmenting path is $a_0, b_1, a_1, \ldots, a_{k-1}, b_k$.

Let $\varepsilon_l := \min_{1\leq j\leq k}\{x(\{b_j, a_j\})\}$, $\varepsilon_f := \min_{1\leq j\leq k}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon = \min\{\varepsilon_l, \varepsilon_f, b(b_k) - x(b_k)\}$ and $b'(v) = b(v) + \varepsilon$. Finally, let

$$x'(e) = \begin{cases} x(e) + \varepsilon & \text{if } e = \{a_{j-1}, b_j\} \text{ for } 1 \leq j \leq k, \\ x(e) - \varepsilon & \text{if } e = \{b_j, a_j\} \text{ for } 1 \leq j \leq k-1 \text{ and} \\ x(e) & \text{otherwise.} \end{cases}$$

Then x' is a stable allocation for b' by Lemma 3.2.1 and by the definition of $l(v)$.

C1) $a_k = a_i$ for some $k > i$: the augmenting path is $a_0, b_1, a_1, \ldots, b_i, a_i, \ldots, b_k, a_k = a_i$.

Let $\varepsilon_l := \min_{i \le j \le k}\{x(\{b_j, a_j\})\}$, $\varepsilon_f := \min_{i \le j \le k}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon = \min\{\varepsilon_l, \varepsilon_f\}$ and let

$$x'(e) = \begin{cases} x(e) + \varepsilon & \text{if } e = \{a_{j-1}, b_j\} \text{ for } i \le j \le k, \\ x(e) - \varepsilon & \text{if } e = \{b_j, a_j\} \text{ for } i \le j \le k \text{ and} \\ x(e) & \text{otherwise.} \end{cases}$$

Then x' is a stable allocation for b by Lemma 3.2.1.

C2) $b_k = b_i$ for some $k > i$: the augmenting path is $a_0, b_1, a_1, \ldots, b_i, a_i, \ldots, a_{k-1}, b_k = b_i$.

Let $\varepsilon_l := \min_{i \le j \le k-1}\{x(\{b_j, a_j\})\}$, $\varepsilon_f := \min_{i \le j \le k}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon = \min\{\varepsilon_l, \varepsilon_f\}$ and let

$$x'(e) = \begin{cases} x(e) + \varepsilon & \text{if } e = \{a_{j-1}, b_j\} \text{ for } i \le j \le k, \\ x(e) - \varepsilon & \text{if } e = \{b_j, a_j\} \text{ for } i \le j \le k-1 \text{ and} \\ x(e) & \text{otherwise.} \end{cases}$$

Then x' is a stable allocation for b by Lemma 3.2.1.

The algorithm of Baïou and Balinski

A stable allocation problem for a bipartite graph G is given with vertex set $V(G) = A \cup B$, vertex-bounds b and edge-capacities c. Let n be the number of vertices and let m be the number of edges in G. Here, we present the so-called *inductive* algorithm of Baïou and Balinski [14], that solves this problem by using at most $O(n + m)$ times the above augmenting procedures.

At the beginning of the inductive algorithm, we set the bounds to be $b^0(u) = 0$ for every $u \in A$ and $b^0(v) = b(v)$ for every $v \in B$. Here $x^0(e) = 0$ for every $e \in E(G)$ is a trivial stable allocation. The idea of the algorithm is to successively increment the bounds of the vertices in A, considering the vertices in order of their indices, by simultaneously modifying the stable allocation for the problem with the actual bounds, until reaching the vertex-bounds b for every vertex.

Suppose that after i augmenting steps a stable allocation x^i for vertex-bounds b^i is given, where $b^i(a_j) = b(a_j)$ for every $j < k$, $b^i(a_k) < b(a_k)$ and $b^i(a_l) = 0$ for every $l > k$. We construct the augmenting path, and we execute the formed augmenting procedure, by setting $\bar{\varepsilon} = \min\{\varepsilon, b(a_k) - b^i(a_k)\}$ in cases I1) and I2).

Analysis of the running time

For the actual stable allocation problem in the i-th step, with bounds b^i, and stable allocation x^i, we define the following partition of $E(G)$:

$T_1(i)$: $\{e|x^i(e) = 0$ and e is dominated only at some $u \in A\}$,

$T_2(i)$: $\{e|0 < x^i(e) < c(e)$ and e is dominated only at some $u \in A\}$,

$T_3(i)$: $\{e|x^i(e) = c(e)\}$,

$T_4(i)$: $\{e|0 < x^i(e) < c(e)$ and e is dominated at some $v \in B\}$,

$T_5(i)$: $\{e|x^i(e) = 0$ and e is dominated at some $v \in B\}$.

Obviously, $T_1(0) = E(G)$. It can be easily verifyied that for two indices $1 \leq p \leq q \leq 5$ it is impossible an edge e belongs to both $T_q(i)$ and $T_p(i+1)$, no matter which augmenting procedure is executed in the i-th step. Furthermore, if a vertex $v \in B$ is unsaturated before the i-th step, then it can become saturated after the i-th step, but never conversely.

Moreover, after an augmenting step, one of the following four cases holds:

- a new edge e becomes saturated, so $e \in T_3(i+1) \setminus T_3(i)$,
- a new edge e becomes 0-weighted, so $e \in T_5(i+1) \setminus T_5(i)$,
- a new vertex $v \in B$ becomes saturated, so $b(v) = x^{i+1}(v) > x^i(v)$,
- a new bound is reached, so $b(a_k) = b^{i+1}(a_k) > b^i(a_k)$.

Thus, the number of augmenting steps in the Baïou–Balinski algorithm is at most $2m + n$.

3.2.5 The inductive algorithm for general graphs

Additional augmenting procedures

Assume that x is a stable allocation for a graph G with vertex-bounds b and edge-capacities c. If G is not necessarily bipartite, then the augmenting path can have two more possible endings, thus these two cases yield two new possible augmenting procedures:

I3) $a_k = b_i$ for some $k > i$: the augmenting path is $a_0, b_1, a_1, \ldots, b_i, a_i, \ldots, b_k, a_k = b_i$.

Let $\varepsilon_l := \min_{1\le j\le i-1}\{x(\{b_j, a_j\})\}$, $\varepsilon_f := \min_{1\le j\le i}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon_{hl} := \min_{i\le j\le k}\{x(\{b_j, a_j\})\}$, $\varepsilon_{hf} := \min_{i+1\le j\le k}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon = \min\{\varepsilon_l, \varepsilon_f, 2\cdot\varepsilon_{hl}, 2\cdot\varepsilon_{hf}\}$ and let

$$x'(e) = \begin{cases} x(e)+\varepsilon & \text{if } e=\{a_{j-1},b_j\} \text{ for } 1\le j\le i,\\ x(e)-\varepsilon & \text{if } e=\{b_j,a_j\} \text{ for } 1\le j\le i-1,\\ x(e)+\frac{\varepsilon}{2} & \text{if } e=\{a_{j-1},b_j\} \text{ for } i+1\le j\le k,\\ x(e)-\frac{\varepsilon}{2} & \text{if } e=\{b_j,a_j\} \text{ for } i\le j\le k \text{ and}\\ x(e) & \text{otherwise.}\end{cases}$$

Then x' is a stable allocation for b by Lemma 3.2.1.

I4) $b_k = a_i$ for some $k > i$: the augmenting path is $a_0, b_1, a_1, \ldots, b_i, a_i, \ldots, a_{k-1}, b_k = a_i$.

Let $\varepsilon_l := \min_{1\le j\le i}\{x(\{b_j, a_j\})\}$, $\varepsilon_f := \min_{1\le j\le i}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon_{hl} := \min_{i+1\le j\le k-1}\{x(\{b_j, a_j\})\}$, $\varepsilon_{hf} := \min_{i+1\le j\le k}\{c(\{a_{j-1}, b_j\}) - x(\{a_{j-1}, b_j\})\}$, $\varepsilon = \min\{\varepsilon_l, \varepsilon_f, 2\cdot\varepsilon_{hl}, 2\cdot\varepsilon_{hf}\}$ and let

$$x'(e) = \begin{cases} x(e)+\varepsilon & \text{if } e=\{a_{j-1},b_j\} \text{ for } 1\le j\le i,\\ x(e)-\varepsilon & \text{if } e=\{b_j,a_j\} \text{ for } 1\le j\le i,\\ x(e)+\frac{\varepsilon}{2} & \text{if } e=\{a_{j-1},b_j\} \text{ for } i+1\le j\le k,\\ x(e)-\frac{\varepsilon}{2} & \text{if } e=\{b_j,a_j\} \text{ for } i+1\le j\le k-1 \text{ and}\\ x(e) & \text{otherwise.}\end{cases}$$

Then x' is a stable allocation for b by Lemma 3.2.1.

We refer hereby to the augmenting procedures I1), I2), I3) and I4) as *improving procedures*, and to C1), C2) as *cycle-swapping procedures*.

The general inductive algorithm

Fix a integral stable allocation problem for a graph G with vertex-bounds b and edge-capacities c. The idea of the inductive algorithm in this general case is the same: We start with a trivial problem with $b^0(v) = 0$ for every $v \in V(G)$, and a integral stable allocation $x^0(e) = 0$ for every edge $e \in E(G)$. Then we successively execute augmenting procedures considering the vertices in order of their indices, until reaching the vertex-bounds b for every vertex.

To prove that this algorithm terminates in finitely many steps, we shall show first that each stable allocation x^i has only half-integer weights, moreover, the non-integer edges form disjoint odd cycles. This property trivially holds for x^0. We suppose that it is true for x^i, and we show that this remains true for x^{i+1}.

Suppose that v is incident with two non-integer edges e and e', such that $e <_v e'$. In this case e is obviously dominated only at v, so $e = l(v) = f(v)$. This observation implies that if a vertex v is the first vertex along the actual augmenting path that belongs to an odd cycle of non-integer weights, then case I3) or I4) occurs. This is because the first repetition is at v, moreover each edge in the augmenting path has integer weight until v, and has non-integer weights after v. Thus the improvement, ε must be integer, and x^{i+1} must have the required property.

We shall verify that cases C1) and C2) cannot occur consecutively infinite many times. Indeed, we prove that within $2m$ steps, at least one improving procedure must be executed in the inductive algorithm.

Lemma 3.2.2. *Assume that a stable allocation problem for a graph G is given with a stable allocation x. Let the number of edges in G be $|E(G)| = m$. If we invoke the increment procedure continuously for one vertex $a_0 \in V(G)$, then after at most $2m$ rounds an improving procedure will be executed.*

Proof. Indirectly suppose, that after each call a cycle-swapping procedure is executed. We build a bipartite subgraph G' of G with vertex set $A \cup B$ in the following way. Let the first augmenting path be $P^1 = (A^1|B^1)$. If $u = a_{j,1} \in A^1$ then let $u \in A$ and if $v = b_{j,1} \in B^1$ then $v \in B$. Moreover, we create a directed subgraph D of G' by directing each occuring vertex in the sequence to the consecutive vertex. So, if $next(u) = b_{j+1,1}$, then $(u, next(u)) \in A(D)$. Similarly, if $next(v) = a_{j,1}$, then $(v, next(v)) \in A(D)$. Note, that $\{u, next(u)\} = f(u)$ before the first round and $\{u, next(u)\} = l(u)$ after the first round. Similarly $\{v, next(v)\} = l(v)$ before the first round and $\{v, next(v)\} = f(v)$ after the first round. Let us denote the vertex, where the first repetition occurs in P^1 by r^1. Obviously there exist a directed path to r^1 from each vertex of P^1 in D. Moreover, r^1 must be part of the second augmenting path, P^2, since until r^1 the augmenting path P^2 is the same as P^1.

Similarly, let the t-th augmenting path be $P^t = (A^t|B^t)$. If $u = a_{j,t} \in A^t$ then let $u \in A$ and if $v = b_{j,t} \in B^t$ then $v \in B$. If $v \in V(D)$ then we remove (v, v') from $A(D)$ and after we add $(v, next(v))$ to $A(D)$, where $next(v)$ is defined as above. Let r^t denote the vertex, where the repetition occurs in P^t. Obviously, there exists a directed path to r^t from each vertex of P^t in D, thus from r^{t-1}, as well. This implies, by induction, that there exists a directed path to r^t from each vertex of $V(D)$ in D.

To show that the above definition of G' is correct, we have to verify that $u \in A^t \cap B$ (and similarly $v \in B^t \cap A$) is not possible. Suppose otherwise that u is the first vertex such that $u = a_{j,t} \in B$. Obviously, r^{t-1} is already part of P^t before u, since until r^{t-1}, the augmenting path P^t is the same as P^{t-1}. We can assume by induction, that there exists a directed path from u to r^{t-1} in D. Moreover, this directed path actually is the same as the augmenting path from u by the definition of $next(v)$, where all the vertices are in the opposite side as before (i.e. $next(u) \in B^t \cap A$, $next(next(u)) \in A^t \cap B$ and so on). So, a repetition must occur, either $b_{k,t} = a_{i,t}$ or $a_{k,t} = b_{i,t}$ thus case I3) or I4) holds, a contradiction.

Finally, we define $T_1'(t)$, $T_2'(t)$, $T_3'(t)$, $T_4'(t)$ and $T_5'(t)$ on G' in the very same way as $T_1(i)$, $T_2(i)$, $T_3(i)$, $T_4(i)$ and $T_5(i)$ were defined before. Similarly, it can be verified that for two indices $1 \leq p \leq q \leq 5$ it is impossible an edge e belongs to both $T_q'(t)$ and $T_p'(t+1)$, if one of the cycle-swapping procedure is executed in the t-th step. Moreover, one of the following two cases surely happen after a cycle-swapping procedure:

- a new edge e of G' becomes saturated, so $e \in T_3'(t+1) \setminus T_3'(t)$,
- a new edge e of G' becomes 0-weighted, so $e \in T_5'(t+1) \setminus T_5'(t)$,

So, we could execute at most $2m$ cycle-swapping procedures consequently. □

A corollary of Lemma 3.2.2 is that the number of steps of the inductive algorithm is $O(n+m)\sum_{v\in V(G)} b(v)$.

An example for bad case running time

Below, we create a integral stable allocation problem for a graph G, for which the number of steps in the inductive algorithm is not polynomial in $n = |V(G)|$.

Example 3.2.3. *Let F_i denotes the i-th Fibonacci number (i.e. $F_1 = F_2 = 1$, and $F_i = F_{i-2} + F_{i-1}$ for $i \geq 3$). A graph G of n vertices is given: $a_1, a_2, \ldots, a_n$ with the following edges, preferences, bounds and capacities:*

$$\begin{array}{rl} a_1: & [a_3, a_2] \\ a_2: & [a_4, a_3, a_1] \\ a_i: & [a_{i+2}, a_{i+1}, a_{i-1}, a_{i-2}] \\ & \text{for every } i = 3, \ldots, n-2 \\ a_{n-1}: & [a_n, a_{n-2}, a_{n-3}] \\ a_n: & [a_{n-1}, a_{n-2}] \end{array} \qquad \begin{array}{rl} b(a_i) = F_i & \text{for every } i = 1, \ldots, n \\ c(\{a_i, a_{i+1}\}) = F_i & \text{for every } i = 1, \ldots, n-1 \\ c(\{a_i, a_{i+2}\}) = F_i & \text{for every } i = 1, \ldots, n-2 \end{array}$$

It can be verified that the inductive algorithm executes F_i improving operations on a_i, namely I1) and I2) alternately, each with improvement 1. Thus the number of steps in the algorithm is $\sum_{i=1}^{n} F_i$, that is an exponential function of n

However, we have just shown that the general inductive algorithm is not polynomial, we note that the scaling property ensures that this algorithm can be modified with standard techniques to become polynomial for the integral stable allocation problem. But it remains an open question whether there exists a strongly polynomial algorithm for the integral stable allocation or for the more general stable allocation problem on nonbipartite graphs.

Properties of the obtained solutions

Finally, we remark that solutions for the stable allocation problem obtained by the incremental algorithm have similar properties to the stable matchings and half-matchings for the stable roommates problem obtained by the inductive algorithm, since the Baïou–Balinski algorithm is a kind of generalization of the Roth-Vande Vate algorithm, and the general inductive algorithm is a generalization of the Tan–Hsueh algorithm.

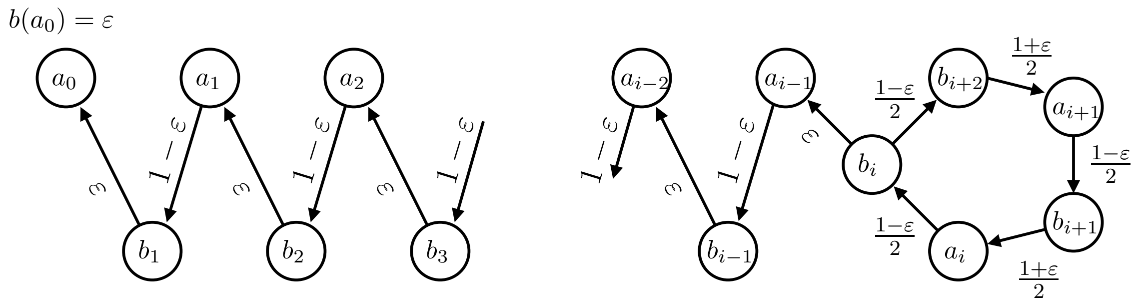


Figure 3.4: A core configuration as an ε^v-core element.

For example, let us run the inductive algorithm for a stable matching problem. If we consider an improving procedure executed for a vertex v, then the augmenting path is actually the proposal-rejection sequence, associated to a core-configuration relative to v. To prove this by the uniqueness of the above sequence, it is enough to see that an ε-improvement for every $0 \leq \varepsilon \leq 1$ creates a stable allocation for the problem with bounds $b(v) = \varepsilon$ and $b(u) = 1$ for every $u \neq v$, that is a special fractional b-core element, called ε^v*-core element*, illustrated in Figure 3.4.

3.3 An application: College admission in Hungary

Since 1985 the admission procedure of higher education institutions is organized by a centralized matching program in Hungary. We present the implemented algorithm, which is similar to the college-proposing version of the Gale–Shapley algorithm. The difference is that here ties must be handled, since applicants may have equal scores. The presented score-limit algorithm finds a solution that satisfies a special stability condition. We describe the applicant-proposing version of the above algorithm and we prove that the obtained solutions by these algorithms are the maximal and the minimal stable score-limits, respectively.

Preliminaries

As we already noted in Section 2.1., the college admission problem was introduced and studied by Gale and Shapley [48]. Later Roth [83] described the history of the

National Intern Matching Program, that have used the same type of algorithm since 1952. Further literature about the two-sided matching markets can be found in the book of Roth and Sotomayor [92].

Recently, the student admission problem came again into prominence (detailed description about several applications can be found in the paper of Abdulkadiroğlu and Sönmez [3]). New centralized matching programs have been implemented for public schools in Boston, and for high schools in New York (see [1] and [2]).

However, there are some studies about existing college admissions programs as well (see the papers [81] and [16] about the programs in Spain and in Turkey, respectively), the description of many other important implementations are not available in the literature.

In Hungary, the admission procedure of higher education institutions is organized by a centralized matching program. The Ministry of Education founded the Admission to Higher Education National Institute (OFI) in 1985 in order to create, operate and develop the admission system of the higher education. The number of applicants is around 150000 in each year, about 100000 of them are admitted, and the fees are payed by the state for approximately 60% of the students (exact statistics in Hungarian are available at [76]).

First, we note that instead of colleges, in Hungary the universities have faculties, where the education is organized in different fields of studies quite independently. So here, students apply for fields of studies of particular faculties. For simplicity, these fields are referred as colleges later in order to keep the original terminology of Gale and Shapley.

At the beginning of the procedure, students give their ranking lists that correspond to their preferences over the fields they apply for. There is no limit for the length of the list, however applicants are charged after each item. The students receive scores at each field they applied for according to their final notes at the high school, and entrance exams. Note, that the score of a student can differ at two fields. These scores are integer numbers, currently limited to 144. Universities can admit a limited number of students to each of their fields, these *quotas* are determined by the Ministry of Education.[1]

The administration is organized by a state-owned center. After collecting the applicants' rankings and their scores, a centralized program computes the score-limits of the fields. An applicant is admitted by the first place on his list, where he is above the score-limit.

Here, we present the currently used basic algorithm that yields a kind of stable assignment. This algorithm is very similar to the Gale–Shapley [48] algorithm, in fact, if the score of the applicants are different at each place then this algorithm is equivalent to the college-proposing algorithm of Gale and Shapley. This explains why it is not suprising that similar statements can be proved for the score-limit algoritms. Here, we show that the score-limits at each field is maximal for the college-proposing version,

[1] We describe some further specialities and requirements in the last subsection, that are not included in the presented basic model.

and minimal for the applicant-proposing version in the set of the stable score-limits.

3.3.1 The definition of stable score-limit

Let $A = \{a_1, a_2, \dots, a_n\}$ be the set of applicants and C be the set of colleges, where q_u denotes the quota of college c_u. Let the ranking of the applicant a_i be given by a preference list P^i, where $c_v >_i c_u$ denotes if c_v preceeds c_u in the list, i.e. the applicant a_i prefers the college c_v to c_u. Let s^i_u be a_i's final score at the college c_u.

The score-limit l is a nonnegative integer mapping $l : C \to \mathbb{N}$. An applicant a_i is admitted by a college c_u, if he achieves the limit at college c_u, and that is the first such place in his list, i.e. $s^i_u \geq l(c_u)$, and $s^i_v < l(c_v)$ for every college $c_v >_i c_u$. If the score-limit l implies that a college c_u admits applicant a_i, then we set the boolean variable $x^i_u(l) = 1$, and 0 otherwise. Let $x_u(l) = \sum_i x^i_u(l)$ be the number of applicants allocated to c_u. A score-limit l is *feasible* if $x_u(l) \leq q_u$ for every college.

Let $l^{u,t}$ be defined as follows: $l^{u,t}(u) = l(u) - t$ and $l^{u,t}(v) = l(v)$ for every $v \neq u$. That is, we decrease the score limit by t at college c_u, by leaving the other limits unchanged. We say that a score-limit l is *stable* if l is feasible but for each college c_u, $l^{u,1}$ is not feasible. This stability condition means that no college can decrease its limit without violating its quota (assuming that the others do not change their limits). We note that if no ties occur (i.e. two applicants have different scores at each college), then this stability condition is equivalent to the original one by Gale and Shapley.

3.3.2 Score-limit algorithms and optimality

First we present the currently used algorithm and verify its correctness, then we describe its applicant-proposing version. Finally, we prove that these algorithms produce the maximal and the minimal stable score-limits, respectively.

The score-limit algorithms

Both score-limit algorithms are very similar to the two versions of the original Gale–Shapley algorithm. The only difference is that now, the colleges cannot select exactly as many best applicants as their quotas are, since the applicants may have equal scores. Here, instead each college sets its score-limits always to be the smallest one, such that its quota is not exceeded. If the scores of the applicants are distinct at each college then these algorithms are equivalent to the original ones.

College-proposing algorithm:
In the first stage of the algorithm, let us set the score-limit at each college independently to be the smallest value such that the number of admitted applicants does not exceed its quota by considering all its applications. Let us denote this limit by l_1. Obviously,

there can be some applicants, who are admitted by several places. These applicants keep their best offer, and reject all the less preferred ones, moreover they cancel also their less preferred applications.

In the further stages, the colleges check whether their score-limits can be further decreased, since some of their applications may have been cancelled in the previous stage, hence they look for new students to fill up the empty places. So each college sets its score-limit independently to be the least possible, considering their actual applications. If an applicant is admitted by some new, better place, then he accepts the best offer in suspense, and rejects or cancels his other, worse applications.

Formally, let l_k be the score-limit after the k-th stage. In the next stage, at every college c_u, the largest integer t_u is chosen, such that $x_u(l_k^{u,t_u}) \leq q_u$. That is, by decreasing its score-limit by t_u, the number of admitted applicants by c_u does not exceed its quota, supposing that all other score-limits remained the same. For every college let $l_{k+1}(c_u) := l_k^{u,t_u}(c_u)$ be the new score-limit. If some limits are decreased simultaneously, then some applicants can be admitted by more than one place, so $x_u(l_{k+1}) \leq x_u(l_k^{u,t_u})$. Obviously, the new score-limit remains feasible.

Finally, if no college can decrease its limit, then the algorithm stops. The stability of the final score-limit is obvious by definition.

Example 3.3.1. *In this example we consider only 3 colleges, c_{cs}, c_e and c_m (i.e. college of computer science, economics and maths, respectively) and the effect caused by two applicants, a_i and a_j. We suppose that all the other applicants have only one place in their lists. The preferences of a_i and a_j are the following: $P^i = c_e, c_{cs}, c_m, \ldots$ and $P^j = c_{cs}, c_m, c_e, \ldots$. Their scores are: $s^i_{cs} = 112$, $s^i_e = 100$, $s^i_m = 117$, $s^j_{cs} = 110$, $s^j_e = 103$ and $s^j_m = 105$. Let the quotas be $q_{cs} = 500$, $q_e = 500$ and $q_m = 100$. We suppose that the number of applicants having*

- *at least* 110 points at c_{cs} is 510,
- *more than* 110 points at c_{cs} is 483,
- *at least* 100 points at c_e is 501,
- *more than* 100 points at c_e is 460,
- *at least* 105 points at c_m is 101,
- *more than* 105 points at c_m is 87.

In the first stage of the college-proposing algorithm the score-limits are $l_1(c_{cs}) = 111$, $l_1(c_e) = 101$ and $l_1(c_m) = 106$. At these limits a_i is admitted to the college of computer science and to the college of maths too, while a_j is admitted to the college of economics only. Since a_i prefers the computer science, he rejects the latter offer (and he cancels also his other less preferred applications.) Now, in the second stage, the score-limit can be decreased by one at the college of maths, because the number of currect applications having at least 105 points is exactly 100. In this way, a_j becomes admitted to this college, and since he prefers maths to economics, he rejects his offer there. In the third stage, the score-limit can be decreased by one at the college of economics. After this

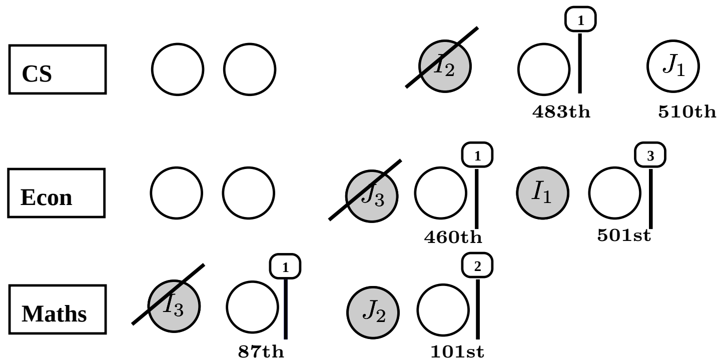


Figure 3.5: The score-limits in the college-proposing algorithm

change a_i is admitted to the college of economics, that is his most preferred place, so he cancels all his other applications. In the final stage no score-limit can be decreased, so the algorithm stops.

Applicant-proposing algorithm:
Let each applicant propose to his first choise in his list. If a college receives more applications than its quota, then let this score-limit be the smallest value such that the number of temporary accepted applicants does not exceed its quota. We set the other limits to be 0.

Let the score-limit after the k-th stage be l_k. If an applicant has been rejected in the k-th stage, then let him apply for the next place in his list, say c_u where he achieves the actual score-limit $l_k(c_u)$, (if there remained such a place in his list). Some colleges may receive new proposals, so if the number of admitted applicants exceeds their quota at a college, they set a new, higher score-limit l_{k+1}. At the same time, they reject all those applicants that do not achieve this new limit.

The algorithm stops if there is no new application. The final score-limit is obviously feasible. It is also stable, because after a limit is increased for the last time, then the rejected applicants get worse and worse offers during the algorithm. So if the limit were decreased by one at the final solution in this place, then these applicants would accept the offer, and the quota would have been exceeded.

Theorem 3.3.2. *Both the score-limit l_C, obtained by the college-proposing algorithm and the score-limit l_A, obtained by the applicant-proposing algorithm are stable.*

Below, we give a simple example to show that not only some applicants can be admitted by preferred places in l_A than in l_C, but the number of admitted applicants can also be larger in l_A.

Example 3.3.3. *We consider only two places c_{cs} and c_e with two applicants a_i and a_j. We suppose that all the other applicants have only one single place in their lists. The* *preference-lists of* a_i *and* a_j *are* $P^i = c_e, c_{cs}, \dots$ *and* $P^j = c_{cs}, c_e, \dots,$ *and their scores are:* $s^i_{cs} = 112$, $s^i_e = 100$, $s^j_{cs} = 110$ *and* $s^j_e = 103$. *Both places have quotas 500. We suppose that the number of applicants having*

- *at least* 110 points at c_{cs} is 501,
- *more than* 110 points at c_{cs} is 487,
- *at least* 100 points at c_e is 501,
- *more than* 100 points at c_e is 460.

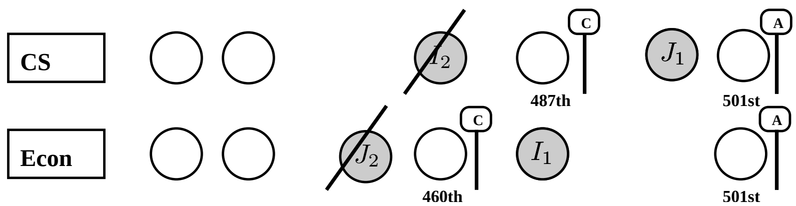


Figure 3.6: The final score-limits of the college-proposing and the applicants-proposing algorithm

Here, both algorithms stop after one stage. The final score-limit obtained by the college-proposing algorithm is $l_C(c_{cs}) = 111$ and $l_C(c_e) = 101$. The number of admitted applicants are $x_{cs}(l_C) = 487$ and $x_e(l_C) = 460$, respectively. While, the final score-limit obtained by the applicant-proposing algorithm is $l_A(c_{cs}) = 110$ and $l_A(c_e) = 100$. Moreover, the number of admitted applicants are 500 at both places. This extreme example shows that the difference between the solutions can be relevant.

The optimality

We say that a score-limit l is *better* than l_* for the applicants if $l \leq l_*$, (i.e. $l(c_u) \leq l_*(c_u)$ for every college c_u). In this case every applicant is admitted by the same or by a preferred place at score-limit l than at l_*.

Theorem 3.3.4. *l_C is the worst possible and l_A is the best possible stable score-limit for the applicants, i.e. for any stable score-limit l, $l_A \leq l \leq l_C$ holds.*

Proof. Both proofs are based on indirect arguments, that are similar to the original one of Gale and Shapley's.

Suppose first, that there exists a stable score-limit l_* and a college c_u such that $l_*(c_u) > l_C(c_u)$. During the college-proposing algorithm there must be two consecutive stages with score-limits l_k and l_{k+1}, such that $l_* \leq l_k$ and $l_*(c_u) > l_{k+1}(c_u)$ for some college c_u. Obviously, $l_k^{u,t_u}(c_u) = l_{k+1}(c_u)$ by definition and $x_u(l_k^{u,t_u}) \leq q_u < x_u(l_*^{u,1})$ by the stability of l_*. So, on the one hand, there must be an applicant, say a_i who is admitted by c_u at $l_*^{u,1}$ but not admitted by c_u at l_k^{u,t_u}. On the other hand, the indirect assumption $l_k^{u,t_u}(c_u) = l_{k+1}(c_u) \leq l_*(c_u) - 1 = l_*^{u,1}(c_u)$ implies that a_i must be admitted by another, preferred place than c_u at l_k^{u,t_u} (since a_i has at least $l_k^{u,t_u}(c_u)$ score there), and obviously also at l_k. That is impossible if $l_* \leq l_k$, a contradiction.

To prove the other direction, we suppose that there exist a stable score-limit l_* and a place c_u such that $l_*(c_u) < l_A(c_u)$. During the applicant-proposing algorithm there must be two consecutive stages with score-limits l_k and l_{k+1}, such that $l_* \geq l_k$ and $l_*(c_u) < l_{k+1}(c_u)$ for some college c_u. At this moment, the reason of the incrementation is that more than q_u students are applying for c_u with at least l_* score. This implies that one of these students, say a_i is not admitted by c_u at l_* (however he has at least $l_*(c_u)$ score there). So, by the stability of l_*, he must be admitted by a preferred place, say c_v at l_*. Consequently, a_i must have been rejected by c_v in a previous stage of the algorithm, that is possible only if $l_*(c_v) < l_k(c_v)$, a contradiction. □

3.3.3 Further notes

There are many further rules required by the law. Some of them are considered in the present algorithm, some are handled manually afterwards.

At each place there is a minimum score that is generally equal to 60% of the maximum score (that is 144 points usually). If an applicant does not have the minimum score at a place, then this application is simply deleted.

In Hungary, some studies are completely financed by the state, some are partly financed by the students. At most of the places there are two different quotas for both kind of studies. The applicants have to indicate also in their rankings which kind of study they apply for at some field.[2] These are considered in the algorithm as distinct places with distinct quotas. However, there are some requirements on their score-limits: the difference between the score-limits of the state-financed and the privately-financed studies at the same place can not be more than 10%. This rule is tracted by the current algorithm.

Another speciality is that certain pairs of fields can be chosen simultaneously, and some others must come in pairs. These cases are solved manually after the first run of the program, and might cause overflowings.

An actual problem of the program is that the scoring system is not fine enough, that is why huge ties are likely to emerge. As a consequence, the difference between the quota and the number of admitted applicants can be large. Moreover, in an extreme case, if the number of applicants having maximum score is greater than the quota of that place, no student can be admitted. So, it is a good news, that recently a finer scoring system has been proposed in the actual law that will increase the maximum score from 144 to 480.

[2]An applicant may rank first a state-financed study in economics at a university in Budapest, then secondly another state-financed study in economics at another university in Pécs, and thirdly a privately-financed study in economics at the first university in Budapest. So the fees are included in the preferences of the applicants in this way.

In our opinion, to change the direction of the algorithm would also be reasonable. Not just because some applicants could be admitted by preferred places, but also because the number of admitted applicants could increase too. We think that the effect of such a change would be more significant than the effect of a similar change in the National Resident Matching Program (see the study of Roth and Peranson [86] about this).

Chapter 4

Exchange of indivisible goods

Introduction

In this chapter we study the question of exchanging indivisible goods. First we give a game theoretical overview on the basic results and we describe some applications that motivate our studies. In Section 4.2, we consider the optimal exchange problem with restrictions, based on the paper [22]. Finally, we study the problem of stable exchanges in Section 4.3, where we present some results published in [20] and [19].

4.1 Preliminaries, applications

4.1.1 The core of exchange games

Assume that a simple digraph $D = (V, A)$ is given, where V is the set of agents. Suppose that each agent has exactly one piece of indivisible goods, and $(i, j) \in A$ if the good of agent i is suitable for agent j. An *exchange* is a permutation π of V such that, for each $i \in V$, $i \neq \pi(i)$ implies $(i, \pi(i)) \in A$. We denote by $C^\pi(i)$ the cycle of π containing i. If $C^\pi(i)$ has length at least 2, then the agent is said to be *covered*.

The houseswapping game

Shapley and Scarf [103] described the exchange problem of indivisible goods as a partitioning NTU-game, referred also as *houseswapping game*. Here, the set of common activities for a coalition S corresponds to the set of permutations of S. The preferences of the agents over the possible permutations are derived from the preferences over the goods that are suitable for them. As in an exchange π each agent i receives the good of his *predecessor*, $\pi^{-1}(i)$, agent i prefers an exchange π to another exchange σ, if he prefers $\pi^{-1}(i)$ to $\sigma^{-1}(i)$. Thus, an exchange π is in the core of the game, or it is *stable*, if there is no blocking coalition B and permutation σ of B, such that each agent $i \in B$ prefers σ to π.

Shapley and Scarf proved that each such market has a nonempty core. Moreover, they showed that a core solution can always be found by the Top Trading Cycle (TTC) algorithm proposed originally by Gale.

The permutation game

The *permutation game* is the houseswapping game with payments. Alternatively, this is a TU-game, where the utility of each agent i having j's good is given as a weight, $w(a)$ of the corresponding arc $a = (i, j)$. The payoff of a coalition S is the maximum utility of a permutation π of S, that is the sum of the utilities, so $v(S) = \max\{\sum_{a\in\pi} w(a) | \pi$ is a permutation of $S\}$.

The nonemptyness of the core of permutation games was proved first by Tijs *et al.* [106], they showed that these games are always balanced.

Note that an exchange is equivalent to a vertex-disjoint packing of directed cycles, thus the payoff of a coalition is the maximum weight of a directed cycle packing. Moreover, by Remark 1.2.1, it is obvious that the balancedeness condition is equivalent with the statement that the maximum weight of a directed cycle packing is equal to the maximum weight of a fractional packing of directed cycles. (Formally, if $\mathcal{C}$ is the set of directed cycles in D, then $x^c : \mathcal{C} \to \mathbb{R}_+$ is a fractional packing if $\sum_{v\in C_i} x(C_i) \leq 1$ for every $v \in V(D)$.)

It is well-known, that every maximum weight directed cycle packing problem can be reduced to a maximum weight perfect matching problem in a bipartite graph through the following construction. Define bipartite graph G_{dp} by dublicating the vertices of D, so $v_d, v_p \in V(G_{dp})$ if $v \in V(D)$. Let $\{u_d, v_p\} \in E(G_{dp})$ if $(u, v) \in A(D)$ with weight $w(\{u_d, v_p\}) = w(u, v)$ and let $\{u_d, u_p\} \in E(G_{dp})$ with weight 0 for every $u \in V(D)$. If $\mathcal{C}^* \subseteq \mathcal{C}$ is a set of vertex-disjoint directed cycles of maximum weight, then we create a perfect matching M of maximum weight as follows: let $\{u_d, v_p\} \in M$ if $(u, v) = a \in C_i \in \mathcal{C}^*$, and $\{u_d, u_p\} \in M$ if u is not covered by $\mathcal{C}^*$. And vica versa.

Furthermore, a fractional packing of directed cycles x^c corresponds to a fractional perfect matching x^m (i.e. a fractional matching that covers each vertex with total weight 1). Formally, for every $(u, v) = a \in A(D)$, let $x^m(\{u_d, v_p\}) = \sum_{a\in C_i} x^c(C_i)$. Since the fractional perfect matching polytope has the integer property for every bipartite graph, the integer and the relaxed problems have the same optimum, that implies the balancedeness of the game.[1]

[1] We note, that the problem can be reduced to the minimum-cost circulation problem as well: We direct the edges of G_{dp} from u_p to v_d if $u = v$ and from v_d to u_p otherwise. We set $f(a) = 0$ lower and $g(a) = 1$ upper bounds for each arc and use the negative of the weights in G_{dp}. Here, a minimum-cost 0-1 circulation corresponds to a maximum weight directed cycle packing. Furthermore, the dual solution of this problem corresponds to an outcome in the core. (For details, see [98], page 289.)

The problem of finding a maximum weight perfect matching in a bipartite graph can be solved in $O(n(m + nlogn))$ time, where n is the number of vertices and m is the number of edges in G_{dp} (see [98], page 288). So, a core-solution of the permutation game can be found by a polynomial-time algorithm.

Finally, let us remark, that although a core-solution provides an exchange with maximum utility, the assumption that the utilities are transferable is quite unrealistic in several barter exchange markets. Moreover, transfers may not be allowed at all, like in the case of living organ exchanges. Therefore, to achieve a maximum utility, or so-called *optimal* exchange in the market the establishment of a centralized clearinghouse is needed. Furthermore, a maximum utility solution always exists even if the core of the game is empty. This is the case when the lengths of the trading cycles are restricted.

Exchanges with restricted lengths

Let us suppose that the size of the basic coalitions are restricted, so $\mathcal{B} = \{B : |B| \leq l\}$. Thus here, the outcome of the game is an *l-way exchange* that contains no cycle with length more than l. Obviously, an l-way exchange is equivalent to a *vertex-disjoint packing of directed cycles with length at most l*.

If $l = 2$, so only pairwise exchanges are allowed, then the problem becomes a matching problem in an undirected graph G with the same vertex set. Here, an edge links two vertices if a pairwise exchange is possible between the corresponding pairs. So $\{u, v\} \in E(G)$ if both (u, v) and $(v, u) \in A(D)$. Thus, the houseswapping game is equivalent to the stable roommates problem, and the permutation game is equivalent to the stable roommates problem with transferable utilities, where the utility of a possible pair is $w(\{u, v\}) = w(u, v) + w(v, u)$, respectively. As we have seen in Section 1.3. these games may have an empty core, but the problem of finding a core-solution, if one exists is solvable in polynomial time in both cases.

For $l \geq 3$ the problem of exchanging indivisible goods, becomes theoretically hard for the NTU and TU-games as well. We study these problems in details in Section 4.2 and 4.3.

4.1.2 An application: Kidney exchange problem

The trade of goods without transfer (barter exchange) is not usual in markets nowadays. However, in some countries the exchange of tenancies [42] or residences [109] is only allowed in this way. Abraham *et al.* [6] also mention several such exchange programs as Peerfix (DVDs), Read It Swap It (books), Intervac (holiday houses) or the National odd shoe exchange. The most serious recent application of the studied model is certainly the kidney exchange problem. Here, to find an optimal solution by a centralized program is not just possible, but also required.

Currently, living donation is the most effective treatment for kidney failure. But patients needing transplants may have donors who cannot donate them because of immunological incompatibility. So these incompatible patient-donor pairs may want to exchange kidneys with other pairs. Kidney exchange programs have already been established in several countries [71], [65], [78]. The most important question, what the goal of the program is. As a first priority, most of the current models want to maximize the number of patients who receive a suitable kidney in the exchange (see [90], [89], [91], [94], [95]) by considering only the suitability of the kidneys. Some more sophisticated models ([100], [22], [6]) consider the difference between suitable kidneys and try to find a solution where the sum of benefits is maximal. A third concept, introduced in [88] and developed in [32], [33], [20] requires first the stability of the solution under various criteria.

Formally, if we model this problem by a digraph $D(V, A)$, then the set of nodes V represent incompatible-donor pairs and an arc (u, v) means that the kidney of u's donor is suitable for $v's$ patient. In terms of the above concepts, the first problem is actually the problem of finding a maximum size directed cycle packing in this digraph. The second is a maximum weight directed cycle packing problem (that is obviously a generalization of the first problem). We remark that such a maximum weight solution is actually a core-solution of the corresponding permutation game. Finally, the third concept is actually the problem of finding a core-solution in the corresponding houseswapping game.

In these models, the difficulties of the corresponding problems are due to the fact that the length of the cycles in the exchanges is bounded. The reason is that all surgeries along a cycle have to be done simultaneously. Most programs allow only pairwise exchanges. Sometimes 3-way exchanges are also possible, like it is possible currently in the New England Program [78], and it may also be allowed in the national program of the USA (as it is declared to be a goal of the system in the future in the Proposal for National Paired Donation Program [107]).

In Section 4.2, we consider the maximum size and maximum weight exchange problems (also called optimal exchange problem) with length restrictions. In Section 4.3 as a generalization of the houseswapping game, we define and study the so-called stable exchange problems.

4.2 Optimal exchange with restrictions

Here, we prove that the problem of finding a 3-way exchange that covers the maximum number of vertices is APX-hard. We give a $2+\varepsilon$ approximation for the maximum weight 3-way exchange problem, and finally we present an exact algorithm for the weighted version.

Suppose that a directed graph $D(V, A)$ is given. Let MAXCOVER l-WAY EXCHANGE denote the problem of finding an l-way exchange that covers the maximum number of vertices. Furthermore, if $w : A(D) \longrightarrow \mathbb{R}_+$ is a weight-function on the edges, then let MAXWEIGHT l-WAY EXCHANGE denote the problem of finding a maximum weight l-way exchange.

4.2.1 The APX-hardness of MAXCOVER l-WAY EXCHANGE

In this subsection, we show that MAXCOVER l-WAY EXCHANGE is APX-hard for any constant $l \geq 3$.

The problem we reduce from and the background of APX-hardness

Let $G = (V, E)$ be a simple and undirected graph. A *triangle* of G is any induced subgraph of G having precisely 3 edges and 3 vertices. A family of triangles $T_1, \dots, T_k$ of G is called a *vertex-packing of triangles* if $T_1, \dots, T_k$ are vertex-disjoint. The *size* of this packing is k. The problem of finding a maximum size vertex-packing of triangles in a given graph G, called NODE-DISJOINT TRIANGLE PACKING (NTP), is known [50] to be NP-hard.

In particular, NTP is known to be NP-hard also in the planar case. Moreover, NTP is known to be APX-hard [64] even for graphs with maximum degree 4 [13]. The above negative results also hold in the case when we restrict the problem to 3-partite graphs, that is, when the vertex set V of G is partitioned into three disjoint color classes $V = A \cup B \cup C$ and no edge of G has its two endvertices in the same color class. Notice that when G has this special structure, every triangle of G must have precisely one vertex in each one of the three color classes.

Given a maximization problem P, let $opt_P(I)$ denote the optimal solution value for some instance I of P and, for a solution S of I, let $val_P(I, S)$ denote the associated value. Given a constant $\varepsilon \in (0, 1)$, a $\frac{1}{1-\varepsilon}$-*approximation algorithm* for P is an algorithm that, applied to any instance I of P, runs in time polynomial in the size of I and produces a solution whose value is at least $(1 - \varepsilon) \cdot opt_P(I)$. If such an algorithm exists, P belongs to APX. Moreover, P is said to be APX-*hard* if the existence of a $\frac{1}{1-\varepsilon}$-approximation algorithm for P for *any* $\varepsilon \in (0, 1)$ would imply the existence of a $\frac{1}{1-\delta}$-approximation algorithm for any $\delta \in (0, 1)$ for *all* problems in APX. To show that P is APX-hard, it sufficies to show a special type of polynomial time reduction from some problem Q already known to be APX-hard to P. An *L-reduction* [13] from Q to P consists of a pair of polynomial-time computable functions (f, g) such that, for two fixed constants α and β: (a) f maps input instances of Q into input instances of P; (b) given a Q-instance I, the corresponding P-instance $f(I)$, and any feasible solution S for $f(I)$, $g(I, S)$ is a feasible solution for the Q-instance I; (c) $|opt_P(f(I))| \leq \alpha |opt_Q(I)|$ for all I, and (d) $|opt_Q(I) - val_Q(I, g(I, S))| \leq \beta |opt_P(f(I)) - val_P(f(I), S)|$ for each I and for every feasible solution S for $f(I)$. From this definition it follows that the relative errors are linearly related, i.e.

$$\frac{|opt_Q(I) - val_Q(I, g(I,S))|}{opt_Q(I)} \leq \alpha\, \beta\, \frac{|opt_P(f(I)) - val_P(f(I),S)|}{opt_P(f(I))}.$$

Hence, if both Q and P are maximization problems, the existence of a $\frac{1}{1-\varepsilon}$-approximation algorithm for P implies the existence of a $\frac{1}{1-\alpha\beta\varepsilon}$-approximation algorithm for Q.

The inapproximability of MAXCOVER 3-WAY EXCHANGE

We first illustrate the bare reduction idea for the case $l = 3$ and then we give a construction to generalize the APX-hardness result to any constant l. Let $G = (V, E)$ be a 3-partite graph received in input and comprising an instance of NTP. Let A, B and C be the 3 color classes in which V is partitioned. We construct a digraph $D = (V, A)$ by simply orienting each edge e of G as follows: if e has one endpoint in A and the other in B, then orient e as to go from A to B; if e has one endpoint in B and the other in C, then orient e as to go from B to C; if e has one endpoint in A and the other in C, then orient e as to go from C to A. To summarize, $V(D) := V(G) = A \cup B \cup C$, and

$$\begin{aligned} A(D) \;:=\; & \{(u,v) \mid uv \in E(G), u \in A, v \in B\} \,\cup \\ & \{(u,v) \mid uv \in E(G), u \in B, v \in C\} \,\cup \\ & \{(u,v) \mid uv \in E(G), u \in C, v \in A\}\,. \end{aligned}$$

Clearly, the digraph D can be constructed in polynomial time starting from the graph G. Moreover, the following lemma says that the above is an objective function preserving reduction (a primary case of L reduction), whence the claimed APX-hardness result follows.

Lemma 4.2.1. *The graph G admits a packing of vertex-disjoint triangles covering t vertices if and only if the digraph D admits a 3-way exchange covering t vertices.*

Proof. Three vertices $u, v, z \in V$ induce a triangle in G if and only if they induce a directed cycle with length 3 in D. As a consequence, a packing of triangles in G can be regarded as a 3-way exchange covering precisely the same set of vertices. In the other direction, notice that D contains no digons (anti-parallel arcs). Thus, any 3-way exchange contains only cycles with length 3 and can hence be regarded as a packing of triangles in G covering precisely the same set of vertices. □

The inapproximability of MAXCOVER L-WAY EXCHANGE

The trick to generalize the above reduction to a generic $l \geq 3$ is as follows. Once D has been obtained, then obtain a second digraph D' from D as follows. For each arc

$a = (u, v)$ of D with $u \in C$ and $v \in A$, add the vertices $w_{a,1}, \ldots, w_{a,l-3}$ and replace the arc $a = (u, v)$ with the arcs $(u, w_{a,1})$, $(w_{a,l-3}, v)$ and $(w_{a,i}, w_{a,i+1})$ for $i = 1, \ldots, l-4$. To summarize, $V(D') := V(D) \cup \{w_{a,i} \mid a = (u, v) \in A(D), u \in A, v \in C, i = 1, \ldots, l-3\}$, and

$$\begin{aligned} A(D') \;:=\; & \{(u, v) \mid uv \in E(G), u \in A, v \in B\} \cup \\ & \{(u, v) \mid uv \in E(G), u \in B, v \in C\} \cup \\ & \{(u, w_{a,1}), (w_{a,l-3}, v) \mid uv \in E(G), u \in C, v \in A\} \cup \\ & \{(w_{a,i}, w_{a,i+1}) \mid uv \in E(G), u \in C, v \in A, i = 1, \ldots, l-3\}\,. \end{aligned}$$

Clearly, the digraph D' can be constructed in polynomial time starting from the graph G. Moreover, the following lemma says that the above is an L-reduction, whence the problem is APX-hard for any constant l.

Lemma 4.2.2. *The graph G admits a packing of vertex-disjoint triangles covering t vertices if and only if the digraph D' admits an l-way exchange covering $tl/3$ vertices.*

Proof. Notice that D' contains no directed cycle of length less then l. Moreover, three vertices $a \in A$, $b \in B$ and $c \in C$ induce a triangle in G if and only if the arc $f = (c, a)$ belongs to D and the vertices $a, b, c, w_{f,1}, \ldots, w_{f,l-4}$ induce a directed cycle in D'. As a consequence, a packing of triangles in G can be regarded as an l-way exchange in D covering precisely $l/3$ as many vertices. In the other direction, since D' contains no directed cycle of length less then l, then any l-way exchange contains only cycles with length l and can hence be regarded as a packing of triangles in G covering precisely $3/l$ as many vertices. □

Lemmas 4.2.1 and 4.2.2 imply the following theorem.

Theorem 4.2.3. MAXCOVER l-WAY EXCHANGE *is APX-hard for any integer $l \geq 3$.*

We remark, that MAXWEIGHT l-WAY EXCHANGE is also APX-hard, since MAXCOVER l-WAY EXCHANGE is its particular problem for unit weights.

4.2.2 Approximation of the MAXWEIGHT l-WAY EXCHANGE

The MAXWEIGHT l-WAY EXCHANGE problem can be reduced to the maximum weight matching problem in a hypergraph. Let $H(V, E)$ be defined on the same vertex set as D. A hyperedge e_X corresponds to a set of vertices $X \subset V(D)$ if $|X| \leq l$ and there exist a directed cycle on X that covers every vertex of X. Let the weight $w_H(e_X)$ be equal to the weight of a maximum weight such cycle that can be formed on X. Obviously, there is a one-to-one correspondence between the l-way exchanges in D and the matchings in H. Moreover, the weights of the maximum-weight solutions are equal, so the following problem is a generalization.

Given a hypergraph $H = (V, E)$ where every hyperedge has size at most l and a nonnegative weight $w_H(e) \in \mathbb{R}_+$ associated to each hyperedge e. Define MAXWEIGHT

l-SET PACKING to be the problem of finding a maximum weight matching M of H. First we describe two simple l-approximations by greedy algorithms.

Greedy algorithm

The *greedy algorithm* constructs a matching M_{gr} in the following way. We start with the original edge set $E_0 := E(H)$ and $M_{gr}^0 = \emptyset$. In the i-th iteration we add a maximum weight edge e_i to the current matching $M_{gr}^i := M_{gr}^{i-1} \cup e_i$ and delete e_i together with the set of edges that intersect e_i, denoted by $N^i(e_i)$, from the current edge set, so $E_i := E_{i-1} \setminus N^i(e_i) \setminus e_i$. We repeat until all edges are removed.

This algorithm is an l-approximation. To prove this, let denote a maximum weight matching by M_{opt}. We partition M_{opt} into subsets M^i, where M^i denotes the edges of M_{opt} that were removed in the i-th iteration of the greedy algorithm. Here, M^i is either equal to e_i or it is an independent set of edges from $N^i(e_i)$, so $|M^i| \leq l$. Since each edge in $N^i(e_i)$ has weight less or equal to $w_H(e_i)$, then $w_H(M^i) \leq l \cdot w_H(e_i)$ holds for each index i. By summarizing these inequalities, we get $w_H(M_{opt}) \leq l \cdot w_H(M_{gr})$.

Mean-greedy algorithm

The *mean-greedy algorithm* is a modification of the above algorithm as we look for a maximum *mean-weights* edge, that is the original weight divided by the size of the edge. So let $mw_H(e) := w_H(e)/|e|$ for each $e \in E(H)$, and let the final matching be M_{mgr}.

To prove the approx-factor, we use the similarly defined partition of a maximum weight solution M_{opt} into subsets M^i as for the mean-greedy algorithm. By using a similar notation here, $|M^i| \leq |e_i|$. Since each edge in $N^i(e_i)$ has mean-weight less or equal to $mw_H(e_i)$, then $mw_H(M^i) \leq |e_i| \cdot mw_H(e_i) = w_H(e_i)$ holds for each index i, which implies $w_H(M^i) \leq l \cdot w_H(e_i)$. By summarizing these inequalities, we get $w_H(M_{opt}) \leq l \cdot w_H(M_{mgr})$.

We remark, that in a kidney exchange program, the greedy algorithm is likely to choose 3-way cycles first, which are not optimal, since two-way exchanges are more safe. That is why the mean-greedy algorithm seems to be a better heuristic for kidney exchange programs.

Local search

The MAXWEIGHT l-SET PACKING problem can further be reduced to the problem of MAXWEIGHT INDEPENDENT SET IN $(l+1)$-CLAW FREE GRAPHS, by working on the *intersection graph* of the hypergraph. In this simple graph, $L(H)$ the vertices are the edges of H, and two vertices are adjacent if the corresponding edges intersect. The fact that H contains only edges with size at most l implies that $L(H)$ is an $(l+1)$-claw free graph. We define a weight-function on the vertices of $L(H)$ in a natural way: let $w_L(v_X) = w_H(X)$ if X is an edge in H and v_X is the corresponding vertex in $L(H)$.

Obviously, a maximum weight independent set in $L(H)$ corresponds to a maximum weight matching in H (that corresponds to a maximum weight exchange in D). Our goal is to approximate this more general problem.

Fix an independent set I in $L(G)$. A natural idea to improve the actual solution is the *t-local search* (see [12]). This means that we attempt to add an independent set X to I with cardinality at most t and remove the subset of I that is in $N(X)$, so that the total weight increases. If no such t-local improvement exists then the solution is a *t-local optimum*.

Note, that if we compare two disjoint independent sets, say I and I_{opt}, then these sets can be viewed as the two sides of a bipartite subgraph of $L(H)$. Moreover, each vertex in this subgraph has degree at most l, by the $(l+1)$-claw freeness of $L(H)$. That is why the conditions of the following theorem can describe the relation of a t-local optimum I and a global optimum I_{opt}.

Theorem 4.2.4 (Arkin, Hassin, 1998)**.** *For any given k and t and every instance $G = (A, B, E)$ satisfying the following three conditions:*

- *$|N(a)| \le k$ for each $a \in A$;*
- *$|N(b)| \le k$ for each $b \in B$;*
- *any subset $X \subseteq A$ of at most t vertices satisfies $w(X) \le w(N(X))$;*

we have

$$\frac{w(A)}{w(B)} \le k - 1 + \frac{1}{t}\,.$$

The above theorem obviously implies that a t-local optimum approximates the global optimum with the same factor for any t. By this, it can be proved that an algorithm, based on the t-local search method, also approximates the MAXWEIGHT l-SET PACKING problem with factor $l - 1 + \varepsilon$ for any $\varepsilon > 0$.

Augmenting path search

Here we show, that the same approximating ratio can be reached by using only a particular t-local search. The *t-augmenting path search* is a special t-local search, where the new set X is chosen along an alternating path in $L(H)$. Formally, let $X = x_1, x_2, \dots, x_s$, where $s \le t$ and there is a subset Y of the actual solution I, such that $|Y| = s - 1$ and $X = x_1, y_1, x_2, y_2, \dots x_{s-1}, y_{s-1}, x_s$ is a path in the intersection graph, $L(H)$.

Theorem 4.2.5. *For any given k and t and every instance $G = (A, B, E)$ satisfying the following three conditions:*

- *$|N(a)| \le k$ for each $a \in A$;*
- *$|N(b)| \le k$ for each $b \in B$;*

- *any subset $X \subseteq A$ of at most t vertices, where X is a set of consecutive vertices in an alternating path of G, satisfies $w(X) \le w(N(X))$;*

we have

$$\frac{w(A)}{w(B)} \le k - 1 + \frac{2}{t}\,.$$

This theorem implies an alternative proof for the existence of an approximation algorithm – that use “only” augmenting path searches – with factor $l - 1 + \varepsilon$ for any $\varepsilon > 0$.

We use the following well-known lemma in the proof of Theorem 4.2.5:

Lemma 4.2.6. *If $G(A, B, E)$ is a k-regular bipartite graph, then the set of edges, $E(G)$ can be partitioned into k perfect matchings.*

Proof. **(of Theorem 4.2.5)** We complete G into a k-regular graph G' by adding dummy vertices with zero weight and some edges. By Lemma 4.2.6 we can partition the set of edges into k perfect matchings. Two of these perfect matchings form a 2-factor, that is a perfect covering by a set of disjoint alternating cycles. Le us fix this 2-factor $\mathcal{C}$.

The main idea of the proof is that the cycles of $\mathcal{C}$ with length exceeding $2t$ can be cut into alternating paths of length at most $2t$ such that the total weight of the endvertices of these paths in B is at most $\frac{2}{t}w(B)$. This is done as follows: Consider a cycle $C_i = (X_i|Y_i) = (x_i^1, y_i^1, x_i^2, \ldots, x_i^c, y_i^c)$ from $\mathcal{C}$, with $|X_i| = |Y_i| = c > t$. We show that we can always find a set of vertices $R_i \subset Y_i$ with cardinality $\lceil \frac{c}{t} \rceil$, that satisfies the following properties:

$$\text{If } r_i^j = y_i^p \text{ and } r_i^{j+1} = y_i^{p+s} \text{ are in } R_i \text{ then } s \le t \pmod{c} \tag{4.1}$$

$$\frac{w(R_i)}{|R_i|} \le \frac{w(Y_i)}{|Y_i|} \tag{4.2}$$

(Here, property (4.1) ensures that between two consecutive vertices in R_i the distance is at most t. Property (4.2) says that the mean-weight of R_i is less or equal than the mean-weight of Y_i.)

The proof of the existence is easy: first we choose a set with the given cardinality that satisfies the (4.1) property, then we rotate this set along the cycle (we increase all of the indexes one by one) and we select the set with minimum total weight.

Using property (4.2) and $c > t$, we get

$$w(R_i) \le \frac{|R_i|}{|Y_i|} w(Y_i) = \frac{\lceil \frac{c}{t} \rceil}{c} w(Y_i) < \frac{2}{t} w(Y_i)\,.$$

If we consider all of the cycles $C_i \in \mathcal{C}$ with length more than $2t$, then for $R = \cup R_i$ we have

$$w(R) = \sum_{i:|X_i|>t} w(R_i) < \sum_{i:|X_i|>t} \frac{2}{t} w(Y_i) \leq \frac{2}{t} w(B). \tag{4.3}$$

Now, we create a partition of A the following way: we restrict the graph of $\mathcal{C}$ from G' to G, then we delete the vertices of R. The graph we get consists of disjoint paths and cycles - the cycles have length at most $2t$ whereas the paths contain at most t vertices in A. Form partition $\mathcal{A}$ from sets of vertices $X \subseteq A$ that are in the same component in the above graph. Obviously, for each $X \in \mathcal{A}$ the third condition of the Theorem 4.2.5 applies. Notice that a vertex $b \in B$ belongs to at most k sets of the form $N(X)$ with $X \in \mathcal{A}$. Moreover, if $b \in B \setminus R$, then b belongs to at most $k-1$ sets of the form $N(X)$ with $X \in \mathcal{A}$ since either $|N(b)| \leq k-1$ or b has two neighbors in the same class of the partition $\mathcal{A}$. Therefore, by summing up these inequalities and using (4.3), (since only the vertices of R can be counted k-times, from each of their k neighbors,) we get:

$$\begin{aligned} w(A) &= \sum_{X\in\mathcal{A}} w(X) \leq \sum_{X\in\mathcal{A}} w(N(X)) \\ &\leq (k-1)w(B \setminus R) + k\, w(R) \leq (k-1)w(B) + w(R) \\ &< \left(k - 1 + \frac{2}{t}\right) w(B)\,. \end{aligned}$$

The proof is complete. □

Combined methods, heuristics

Chandra and Halldórsson [36] showed a $2(l+1)/3$-approximation algorithm for the MAXWEIGHT INDEPENDENT SET problem in $(l+1)$-claw free graphs by combining the above presented two methods. Their algorithm starts with an independent set obtained by the greedy algorithm. Then the solution I is improved by a special local search, where the new, added set X is chosen from the neighbors of a vertex from I, in such a way that the ratio of the weights of X and the removed set, $N(X) \cap I$ is always maximal.

We remark, that this algorithm beats the two previously described local search methods for $l \geq 5$. It is open, whether a better approximation can be achieved by starting with the greedy (or with the mean-greedy) algorithm, and then always making the best t-local improvement (or t-augmenting path improvement). Such an algorithm might be useful as a heuristic for the original problem.

Approximability of MAXCOVER l-WAY EXCHANGE

For the approximability of MAXCOVER l-WAY EXCHANGE, a general result of [54] leads to a polynomial time $(l/2+\varepsilon)$-approximation algorithm (for any $\varepsilon > 0$) for any fixed l. In the special case of $l = 3$, this gives a $(3/2+\varepsilon)$-approximation algorithm (for any fixed $\varepsilon > 0$). An approximation algorithm improving this ratio could be directly translated into an algorithm for NTP with a better approximation guarantee than the one known so far.

4.2.3 Exact algorithm for MAXWEIGHT 3-WAY EXCHANGE

The MAXCOVER 3-WAY EXCHANGE problem is currently solved in the New England Program by integer programming methods as it is described in [91] and [95]. Recently, Abraham *et al.* [6] implemented such a special IP-heuristic for the MAXWEIGHT 3-WAY EXCHANGE problem that is capable of handling the data of the USA (for approximately 10.000 couples) according to their simulations. In spite of these promising results, it is still an interesting question to construct an exact algorithm for this special NP-hard problem. Below, we present some ideas about this question that might be useful to improve heuristic algorithms.

Fix a MAXWEIGHT 3-WAY EXCHANGE problem in a digraph D, and an optimal solution π^* that contains some 3-way exchanges $\{C_1, C_2, \dots, C_l\}$. Suppose that somebody tells us a list of arcs: one arc from each 3-way exchange, so let $Y = \{a_1, a_2, \dots a_l\}$, where $a_i \in C_i$. We show that by this information we can efficiently find a maximum weight set of 3-way exchanges.

We transform the MAXWEIGHT 3-WAY EXCHANGE problem to a maximum weight matching problem in an undirected graph G_Y in the following way. We denote by $V(Y)$ the set of vertices in D that are covered by Y. Let $y_{i,j} \in V(G_Y)$ if $(v_i, v_j) \in Y$, otherwise let $x_i \in V(G_Y)$ if $v_i \in V(D) \setminus V(Y)$. Let $\{x_i, x_j\} \in E(G_Y)$ if both (v_i, v_j) and $(v_j, v_i) \in A(D)$, and let $\{x_k, y_{i,j}\} \in E(G_Y)$ if both (v_k, v_i) and $(v_j, v_k) \in A(D)$. Considering the weights, $w'(\{x_i, x_j\}) := w(v_i, v_j) + w(v_j, v_i)$, and $w'(\{x_k, y_{i,j}\}) := w(v_k, v_i) + w(v_i, v_j) + w(v_j, v_k)$.

Obviously, a matching M in G_Y corresponds to a set of 3-way exchanges π in D, in such a way that $\{x_i, x_j\} \in M$ if and only if $\pi(v_i) = v_j$ and $\pi(v_j) = v_i$ is a 2-way exchange in D, furthermore $\{x_k, y_{i,j}\} \in M$ if and only if $\pi(v_k) = (v_i)$, $\pi(v_i) = (v_j)$ and $\pi(v_j) = (v_k)$ is a 3-way exchange in D. By this, the weight of a matching M is equal to the weight of the corresponding set of exchanges π. So, if π^* is an optimal solution in D, and Y is a set of arcs as it is described above, then by solving a maximum weight matching problem in G_Y we can find a matching that corresponds either to π^* or to an other maximum weight set of 3-way exchanges in D.

An exact algorithm can be constructed from the above idea: first we guess a set of arc $Y_i \subset A(D)$ and then we find a maximum weight matching in G_{Y_i}, that corresponds to a feasible solution in D. The question is how can we reduce the number of guesses.

As a trivial approach, we can try each of the set of arcs, that produce 2^m rounds, where m is the number of arcs in D. But obviously, we should try only the independent subsets (i.e. that do not admit two incident arcs).

One idea is to reduce the set of arcs to a subset T, such that by deleting T from D, the obtained subgraph does not contain any cycle with length 3. Since if $\{C_1, C_2, \dots, C_l\}$ are the 3-way exchanges of an optimal solution π^*, then we can always find a suitable subset $Y_T = \{a_1, a_2, \dots, a_l\}$ from T, as required. If $t = |T|$, then the number of guesses is only 2^t. We note, that $t \leq \frac{m}{2}$ always holds, since every directed graph can become acyclic by removing at most half of its arcs. Thus this approach gives an $O(2^{\frac{m}{2}})$ time exact algorithm.

Another idea, is to consider a vertex set S, such that $D \setminus S$ (the digraph, obtained by removing the vertices of S from D) does not contain any cycle with length 3. For a vertex v let $\Delta^+(v)$ be the set of arcs with tail from v, and $\Delta^-(v)$ the set of arcs with head to v. We denote by $\delta^+(v) := |\Delta^+(v)|$ and $\delta^-(v) := |\Delta^-(v)|$, furthermore, let $o(v) := min\{\delta^+, \delta^-\}$ and $s = |S|$. If we choose either $\Delta^+(v)$ or $\Delta^-(v)$ for each vertex $v \in S$, (practically to one with less cardinality), that we denote by $T^*(v)$, then the union of these sets of arcs: $T^* := \cup_{i=1}^{s} T^*(v_i)$ is a suitable candidate for T, as defined above. Moreover, if we guess an independent set from this T^*, then obviously it will contain at most one arc from each $T^*(v)$. This implies that the number of possible guesses is at most $\prod_{i=1}^{s}(o(v_i) + 1)$. Since D is a simple digraph, this gives an n^s upper bound for the number of rounds.

Finally, we remark, that the latter approach can be useful, because the number of incompatible pairs, that are in fact ABO-compatible, is relatively small in the current pools (these pairs are in the "short side" of the market, see [91] for details). Obviously, without these pairs no 3-way exchange is possible.

4.3 Stable exchange problems

As a generalization of the houseswapping game, we study the question of stable exchanges, where the length of the cycles in the exchange may be restricted, and the size of the blocking coalitions can also be bounded, independently. We consider the problem of finding weakly, strongly and super-stable exchange in case ties are allowed in the preference lists. We recall the definition of $\mathcal{L}$-preferences introduced in [32]. There, the preferences of the agents are lexicographic, the agents primarily care about the goods that they receive, secondarily they want to minimize the length of their trading cycle. Beside finding a stable solution, another natural goal is to maximize the number of agents involved in the exchange. Here, we describe well-known complexity results from a general point of view, and we present an inapproximability result published in [20] and two theorems on the NP-hardness of some basic 3-way stable 3-way exchange problems.

4.3.1 Stable exchange with ties

The stable exchange problem, as we defined it in the Preliminaries, is equivalent to the problem of finding an outcome in the core of the corresponding houseswapping game. In this study we consider the case of strict preferences and also the case of preferences with *ties*. Strict preference means linear ordering. If some goods are tied in a preference list of agent i, then agent i is indifferent between them. In the *stable exchange problem* (SE) we suppose that the preferences are strict. The stable exchange problem with ties is denoted by SE+T.

Here, we recall again the three main stability concepts introduced in Subsection 1.1.4. for general NTU-games, and specified in Section 2.3 for stable matchings.

- An exchange is *weakly stable* if there exists no blocking coalition B and permutation σ of B, such that each agent $i \in B$ *strictly* prefers σ to π.
- An exchange is *strongly stable* if there exists no blocking coalition B and permutation σ of B, such that one of the agents from B strictly prefers σ to π, and each other agent $i \in B$ either strictly prefers σ to π or is indifferent between them.
- An exchange is *super-stable* if there exists no blocking coalition B and permutation σ of B, such that σ is not equal to π on B and each agent $i \in B$ either strictly prefers σ to π or is indifferent between them.

The following propositions are obvious.

Proposition 4.3.1. *Given an instance of SE+T.*

4.3.1/a) If an exchange π is strongly stable, then it is also weakly stable.

4.3.1/b) If an exchange π is super-stable, then it is also strongly stable.

Given an instance I of SE+T. If an instance I' of SE is obtainable from I by breaking the ties, then I' is a *derived* instance from I.

Proposition 4.3.2. *Let I be an instance of SE+T.*

4.3.2/a) An exchange π is weakly stable if it is weakly stable in at least one *instance I' of SE that can be derived from I.*

4.3.2/b) An exchange π is super-stable if it is super-stable in every *instance I' of SE that can be derived from I.*

4.3.2 Stable exchange under $\mathcal{L}$-preferences

Under $\mathcal{L}$-*preferences*, an agent i prefers a permutation π to another permutation σ if he either prefers $\pi^{-1}(i)$ to $\sigma^{-1}(i)$ or is indifferent between them, but the length of $C^{\pi}(i)$ is smaller than the length of $C^{\sigma}(i)$. This notion was introduced by Cechlárová *et al.* in [32]. They called the NTU-game related to the problem of SE under $\mathcal{L}$-preferences the *kidney exchange game*. (To distinguish $\mathcal{L}$-preferences and the original ones, the latter will be referred to as *normal preferences* hereafter.)

We remark, that a similar ordering, the $\mathcal{B}$-preference was defined earlier by Cechlárová and Romero-Medina in [34]. There, an agent i prefers a coalition C to another coalition D, if either he prefers the best member of C to the best member of D, or he is indifferent between them, but the size of C is smaller than the size of D. The following proposition is obvious.

Proposition 4.3.3. *Given an instance of SE+T.*

4.3.3/a) If an exchange π is (weakly) stable under $\mathcal{L}$-preferences, then it is also (weakly) stable under normal preferences.

4.3.3/b) If an exchange π is super-stable under normal preferences, then it is also super-stable under $\mathcal{L}$-preferences.

4.3.3 Restrictions on the lengths

Considering an *l-way exchange problem*, the size of the blocking coalitions can also be restricted. We say that an exchange is *b-way stable* if there exist no blocking coalition of size at most b. Obviously, the most relevant problems are the ones with $l = 2, 3$. If $b = l$ then a stable exchange is actually a core-solution of the corresponding NTU-game, as it was described in the Preliminaries. The following observations can be easily verified.

Proposition 4.3.4. *Suppose that π is a b-way stable l-way exchange, then*

4.3.4/a) it is also a $(b-1)$-way stable l-way exchange, and

4.3.4/b) it is also a b-way stable $(l+1)$-way exchange.

A 2-way exchange is called *pairwise exchange*, that is actually a *matching* of the agents.

Proposition 4.3.5. *Given an instance of SE+T.*

4.3.5/a) If a pairwise *exchange π is (weakly) stable under normal preferences, then it is also (weakly) stable under $\mathcal{L}$-preferences.*

4.3.5/b) If a pairwise *exchange π is strongly stable under normal preferences, then it is also strongly stable under $\mathcal{L}$-preferences.*

The weakest stability condition is the 2-way stability (or in other words, the *pairwise stability*), where no pair of agents can block a stable solution.

Proposition 4.3.6. *Given an instance of SE+T.*

4.3.6/a) If an exchange π is pairwise *super-stable under $\mathfrak{L}$-preferences, then it is also* pairwise *super-stable under normal preferences.*

4.3.6/b) If an exchange π is pairwise *strongly stable under $\mathfrak{L}$-preferences, then it is also* pairwise *strongly stable under normal preferences.*

Corollary 4.3.7. *Given an instance I of SE+T. A pairwise exchange π is pairwise {weakly, strongly, super-} stable under $\mathfrak{L}$-preferences if and only if π is* pairwise *{weakly, strongly, super-} stable under normal preferences, respectively.*

If we consider strict preferences, then some further statements can be verified.

Proposition 4.3.8. *Given an instance of SE.*

4.3.8/a) If an exchange π is strongly stable, then π is also super-stable.

4.3.8/b) If an exchange π is strongly stable under $\mathfrak{L}$-preferences, then π is also super-stable under $\mathfrak{L}$-preferences.

Proposition 4.3.9. *Given an instance of SE.*

4.3.9/a) If a pairwise *exchange π is strongly stable under $\mathfrak{L}$-preferences, then π is also weakly stable under $\mathfrak{L}$-preferences.*

4.3.9/b) If an exchange π is pairwise *strongly stable under $\mathfrak{L}$-preferences, then π is also* pairwise *weakly stable under $\mathfrak{L}$-preferences.*

Corollary 4.3.10. *In an instance I of SE the same* pairwise *exchanges are {weakly, strongly, super-} stable under $\mathfrak{L}$-preferences, and {strongly, super-} stable under normal preferences. Moreover, the same exchanges are pairwise {weakly, strongly, super-} stable under $\mathfrak{L}$-preferences, and weakly stable under normal preferences. Thus, in case of pairwise stable pairwise exchanges, these stability concepts are equivalent if the preferences are strict.*

4.3.4 Maximum size stable exchange problems

Beside finding a stable exchange, we may want to find such a stable solution, where the number of covered agents is maximal. We denote the problem of finding such a maximal solution for a stable exchange problem by

- MAXCOVER-SE in the basic case, (i.e. normal preferences, no ties, no restrictions)

- MAXCOVER-{W,SU,ST}-SE+T for {weakly, strongly, super-} stable exchanges with ties,
- MAXCOVER-$\mathcal{L}$SE under $\mathcal{L}$-preferences, and
- MAXCOVER-S_bE_l for b-way stable l-way exchanges.

4.3.5 Complexity results

We present some complexity results on exchange problems, matching problems and 3-way exchange problems using the above definitions. Some results from Section 2.3 are recalled, and together with the results presented below, collected in a table at the end of this section, with reference indices [Ri].

Exchange problems

Shapley and Scarf showed in [103], that there always exists a (weakly) stable exchange in an instance I of SE+T [R8]. Roth and Postlewaite [87] proved that the exchange obtained by the TTC algorithm is super-stable for instances of SE. Moreover, this is the only possible super-stable solution. We remark that this uniqueness holds also for strongly stable exchanges by Proposition 4.3.1/a, but obviously, not for the weakly stable exchanges. Thus here, the MAXCOVER-SE problem is nontrivial. In the case of $\mathcal{L}$-preferences, Biró and Cechlárová [20] proved recently the following theorem.

Theorem 4.3.11. MAXCOVER-$\mathcal{L}$SE *is not approximable within* $n^{1-\varepsilon}$ *for any* $\varepsilon > 0$ *unless* $P = NP$.

Sketch of the proof. The proof is based on a gap-introducing reduction starting from MIN-MM for a cubic graph G of size p with a positive integer K. We create a MAXCOVER-$\mathcal{L}$SE problem I of size n, that contains one directed cycle C of size $\Theta(n)$. Here, $n >> p$ and p is fixed, so p can be considered as a constant relative to n. If C is part of an exchange π, then the number of covered vertices is $\Theta(n)$, whilst if C does not belong to π then the number of covered vertices is just a constant relative to n. In the proof it is verified that G admits a maximal matching of size at most K if and only if C is part of some stable exchange in I. Thus, an algorithm that could approximate MAXCOVER-$\mathcal{L}$SE in polynomial time within a factor less than $\Theta(n)$ would be able to decide MIN-MM for cubic graphs as well.

Matching problems

As we already explained in Section 4.1, the pairwise stable pairwise exchange problem is equivalent to the stable roommates problem. Here, we note that the results, presented in Section 2.3 can be equivalently described by the above definitions as follows. The algorithm of Irving [56] finds a pairwise stable pairwise exchange if there exists one for the given stable exchange problem [R2]. Theorem 2.3.1 states that the decision problem of finding a (weakly) pairwise stable pairwise exchange in an instance of SE+T is NP-complete [R3]. By Theorem 2.3.2, the decision problem related to MAXCOVER-S$_2$E$_2$+T is NP-complete, even for bipartite graphs [R4].

Recently, Irving [58] showed that the decision problem of finding a *cycle stable matching* in an instance of the *cycle stable roommates problem* (SCR) is NP-complete. Moreover, the length of each possible blocking cycle is at most 3 in his construction, so he proved the following:

Theorem 4.3.12 (Irving)**.** *The problem of finding a stable pairwise exchange in an instance of SE is NP-complete [R9]. The same result holds for* 3-way *stable pairwise exchanges [R10].*

Exchange with restricted lengths

Assume now, that the 2-way and 3-way exchanges are also allowed. The following theorems were presented in [19].

Theorem 4.3.13. *The 3-way stable 3-way exchange problem in an instance of SE is NP-complete [R11].*

Proof. Clearly, the problem is in NP. To show NP-hardness, we give a transformation from WEAKLY SRT. This problem is NP-complete [59], even if there is no edge between those vertices that have ties in their preference lists. Let $G = (V, E)$ be the undirected graph in the instance of WEAKLY SRT. Without loss of generality, assume that $\{v_1, v_2, \ldots, v_t\} \subseteq V(G)$ are the vertices that have no tie in their lists, and $\{v_{t+1}, v_{t+2}, \ldots, v_n\} \subseteq V(G)$ are the vertices that have preferences with ties. We denote by T_i^k the k-th tie in v_i's preference list.

We construct an instance I of SE as follows: let $U' \cup U'' \cup X \cup Y$ be the set of vertices of the digraph D. Here $U' = \{u_1, u_2, \ldots, u_t\}$ and $U'' = \{u_{t+1}, u_{t+2}, \ldots, u_n\}$. Let X correspond to the edges without ties, so $x_{i,j} \in X$ if $\{v_i, v_j\} \in E(G)$ for some $i < j$ and v_i is not tied in the preference of v_j. Let Y correspond to the ties in the lists, so $y_i^k \in Y$ if T_i^k exists. We define $A(G)$ in the following way: if $\{v_i, v_j\} \in E(G)$ for some $i < j$ such that v_i is not tied in the preference of v_j, then let (v_j, v_i), $(v_i, x_{i,j})$ and $(x_{i,j}, v_j)$ be in $A(D)$. In the other case, if $v_i \in T_j^k$, then let (v_j, v_i), (v_i, y_j^k) and (y_j^k, v_j) be in $A(D)$. Let the preference list of each $u_i \in U' \cup U''$ remain similar to v_i's list with some natural modifications: namely, we write

- u_j instead of v_j if $i < j$,
- $x_{i,j}$ instead of v_j if $j < i$ and v_j is not tied in v_i's list, and
- y_i^k instead of tie T_i^k.

Furthermore, let each y_i^k's preference list be arbitrary. (We note that each $x_{i,j}$ has only one incoming arc.)

There is no directed cycle with length 2 in D, so only 3-way cycles are possible in the exchange. Moreover, each 3-way cycle in D corresponds to an edge in G. Thus, there is a one-to-one correspondence between the matchings in G and the 3-way exchanges in D. So let M be a matching in G, we construct a 3-way exchange π in the following way: if $\{v_i, v_j\} \in M$ for some $i < j$, where v_i is not tied in the preference of v_j, then let $(v_j, v_i, x_{i,j}) \in \pi$, otherwise if $\{v_i, v_j\} \in M$ for some $v_i \in T_j^k$, then let $(v_j, v_i, y_j^k) \in \pi$.

Finally, we shall prove that a matching M in G is weakly stable if and only if the corresponding 3-way exchange π is 3-way stable in I. To verify this, it is enough to see that an edge $\{v_i, v_j\} \notin M$ is blocking for M if and only if the corresponding 3-way cycle is blocking for π, and this is obvious. $\square$

As a natural generalization of the SM problem, Knuth [67] defined the *three-sided stable matching problem*. In this special coalition formation game the possible coalitions are (m, w, c) triples (i.e. families) from $M \times W \times C$ (i.e. men, women, cats), where $|M| = |W| = |C| = n$, and everybody prefers being in a family to remaining single. Alkan [8] showed that a stable solution may not exist if the preferences of the agents can be arbitrary over the pairs from the other two sides. Moreover, Ng and Hirschberg [79] proved that the problem of determining whether a stable solution exists is NP-complete.

Boros *et al.* [29] showed that the core of this coalition formation game can be nonempty even if the preferences of the agents are *lexicographically cyclic* (i.e. men primarily care about women, women primarily care about cats, and cats primarily care about men). They raised the same question for *purely cyclic preferences*, in which the men only care about women, so a man is indifferent between two families if he has the same wife in both of them, (same conditions for women and cats). This problem is equivalent to the 3-way stable 3-way exchange problem for *three-sided cyclic digraphs*, (i.e. $V(D) = M \cup W \cup C$ and every arc $(i, j) \in A(D)$ is from either $W \times M$ or $C \times W$ or $M \times C$), in that special case where the sizes of the three sides are the same, and the digraph contains all possible edges. Here, we prove that without the two latter restrictions, the general problem of finding a maximum size 3-way stable 3-way exchange for three-sided cyclic digraphs is NP-hard.

Theorem 4.3.14. *The decision problem related to* MAXCOVER-S_3E_3 *is NP-complete, even for three-sided cyclic digraphs [R12].*

Proof. Clearly, the problem is in NP. To show NP-hardness, we give a transformation from MAX WEAKLY SRT. This problem is NP-complete [74], even if ties occur only in one side of the bipartite graph.

The construction is similar to the above. If $V(G) = A \cup B$, then the three sides of D are $U' \sim A$, $U'' \sim B$ and $X \cup Y$. We can find a weakly stable matching M with $|M| \geq l$ if and only if for the corresponding 3-way stable 3-way exchange π the number of covered vertices is at least $3l$. $\square$

4.3.6 Summary, open questions

We summarize the presented complexity results in the following table for (weakly) stable exchanges under normal preferences. (Here again, P denotes that the problem is polynomial time solvable, NPc denotes that the (related) problem is NP-complete, (NPh) denotes that the NP-hardness of the problem is obvious from the presented results.) Finally, ??? means that we think that these unsolved problems are relevant, the reasons are explained below.

	$l =$	2-way exchange		3-way exchange		exchange	
$b =$		(strict)	ties	(strict)	ties	(strict)	ties
2-way stable	existence	P [R2]	NPc [R3]	???		(Yes)	(Yes)
	MAXCOVER	P	NPc [R4]			???	
3-way stable	existence	NPc [R10]	(NPh)	NPc [R11]	(NPh)	(Yes)	(Yes)
	MAXCOVER	(NPh)	(NPh)	NPc [R12]	(NPh)		
(cycle) stable	existence	NPc [R9]	(NPh)	(NPh)	(NPh)	(Yes)	Yes [R8]
	MAXCOVER	(NPh)	(NPh)	(NPh)	(NPh)	???	

In the current applications of kidney exchange, some programs allow three-way exchanges, and pairwise stability may become a natural expectation. This is why the problem of finding a pairwise stable 3-way exchange in an instance of SE is important. However, it is easy to construct an example to show that such a stable solution may not exist, but the complexity of this problem is unclear.

Considering the exchanges without restriction on the cycle-lengths, the TTC algorithm always provides a stable exchange, that is also a pairwise stable exchange in an instance of SE. Here, MAXCOVER-SE and MAXCOVER-S$_2$E are two important problems to solve.

Moreover, the problems of {weakly, strongly, super-} stable exchange with ties, under normal and $\mathcal{L}$-preferences yield various interesting open questions.

Bibliography

[1] Atila Abdulkadiroğlu, Alvin E. Parag A. Pathak, Roth, and Tayfun Sönmez. The Boston public school match. *American Economic Review, Papers and Proceedings*, 95(2):368–371, 2005.

[2] Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. The New York City high school match. *American Economic Review, Papers and Proceedings*, 95(2):364–367, 2005.

[3] Atila Abdulkadiroğlu and Tayfun Sönmez. School Choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

[4] Hernán Abeledo and Uriel G. Rothblum. Paths to marriage stability. *Discrete Appl. Math.*, 63(1):1–12, 1995.

[5] David J. Abraham, Péter Biró, and David F. Manlove. "Almost stable" matchings in the roommates problem. In *Approximation and online algorithms*, volume 3879 of *Lecture Notes in Comput. Sci.*, pages 1–14. Springer, Berlin, 2006.

[6] David J. Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter-exchange markets: Enabling nationwide kidney exchanges. *To appear in ACM-EC 2007: the Eighth ACM Conference on Electronic Commerce*, 2007.

[7] Ron Aharoni and Tamás Fleiner. On a lemma of Scarf. *J. Combin. Theory Ser. B*, 87(1):72–80, 2003. Dedicated to Crispin St. J. A. Nash-Williams.

[8] Ahmet Alkan. Nonexistence of stable threesome matchings. *Math. Social Sci.*, 16(2):207–209, 1988.

[9] Ahmet Alkan. A class of multipartner matching markets with a strong lattice structure. *Econom. Theory*, 19(4):737–746, 2002.

[10] Ahmet Alkan and David Gale. Stable schedule matching under revealed preference. *J. Econom. Theory*, 112(2):289–306, 2003.

[11] Nikolay Angelov. Modelling firm mergers as a roommate problem. *(preprint)*.

[12] Esther M. Arkin and Refael Hassin. On local search for weighted packing problems. *Math. Oper. Res.*, 23:640–648, 1998.

[13] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation*. Springer-Verlag, Berlin, 1999. Combinatorial optimization problems and their approximability properties, With 1 CD-ROM (Windows and UNIX).

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

[15] M. L. Balinski. Integer programming: Methods, uses, computation. *Management Sci.*, 12:253–313, 1965.

[16] Michel Balinski and Tayfun Sönmez. A tale of two mechanisms: student placement. *J. Econom. Theory*, 84(1):73–94, 1999.

[17] Péter Biró. *Stable b-matchings on graphs*. 2003. Master's thesis, Budapest University of Technology and Economics, (in Hungarian).

[18] Péter Biró. Higher education admission in Hungary by a "score-limit algorithm". *The 18th International Conference on Game Theory at Stony Brook University*, 2007.

[19] Péter Biró. Stable exchange of indivisible goods with restrictions. *Proceedings of the 5th Japanese-Hungarian Symposium*, pages 97–105, 2007.

[20] Péter Biró and Katarína Cechlárová. Inapproximability of the kidney exchange problem. *Inf. Process. Lett.*, 101(5):199–202, 2007.

[21] Péter Biró, Katarína Cechlárová, and Tamás Fleiner. The dynamics of stable matchings and half-matchings for the stable marriage and roommates problem. *Internat. J. Game Theory*, in print.

[22] Péter Biró and Romeo Rizzi. Maximum weight directed cycle packing problem in optimal kidney exchange programs. *(working paper)*, 2007.

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

[24] Yosef Blum, Alvin E. Roth, and Uriel G. Rothblum. Vacancy chains and equilibration in senior-level labor markets. *J. Econom. Theory*, 76(2):362–411, 1997.

[25] Yosef Blum and Uriel G. Rothblum. "Timing is everything" and marital bliss. *J. Econom. Theory*, 103(2):429–443, 2002.

[26] O. N. Bondareva. Kernel theory in n-person games. *Vestnik Leningrad. Univ.*, 17(13):141–142, 1962.

[27] O. N. Bondareva. Some applications of the methods of linear programming to the theory of cooperative games. *Problemy Kibernet. No.*, 10:119–139, 1963.

[28] Endre Boros and Vladimir Gurvich. Perfect graphs are kernel solvable. *Discrete Math.*, 159(1-3):35–55, 1996.

[29] Endre Boros, Vladimir Gurvich, Steven Jaslar, and Daniel Krasner. Stable matchings in three-sided systems with cyclic preferences. *Discrete Math.*, 289(1-3):1–10, 2004.

[30] Katarína Cechlárová. Randomised matching mechanism revisited. *(preprint)*, 2002.

[31] Katarína Cechlárová and Tamás Fleiner. On a generalization of the stable roommates problem. *ACM Trans. Algorithms*, 1(1):143–156, 2005.

[32] Katarína Cechlárová, Tamás Fleiner, and David F. Manlove. The kidney exchange game. *Proc. SOR'05*, Eds. L. Zadik-Stirn, S. Drobne:77–83, 2005.

[33] Katarína Cechlárová and Vladimir Lacko. The kidney exchange game: How hard is to find a donor? *IM Preprint*, 4/2006, 2006.

[34] Katarína Cechlárová and Antonio Romero-Medina. Stability in coalition formation games. *Internat. J. Game Theory*, 29(4):487–494, 2000.

[35] Katarína Cechlárová and Viera Valova. The stable multiple activities problem. *(preprint)*, 2005.

[36] Barun Chandra and Magnús M. Halldórsson. Greedy local improvement and weighted set packing approximation. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 1999)*, pages 169–176, New York, 1999. ACM.

[37] Effrosyni Diamantoudi, Eiichi Miyagawa, and Licun Xue. Random paths to stability in the roommate problem. *Games Econom. Behav.*, 48(1):18–28, 2004.

[38] Jack Edmonds. Maximum matching and a polyhedron with $0,1$-vertices. *J. Res. Nat. Bur. Standards Sect. B*, 69B:125–130, 1965.

[39] J. Egerváry. Matrixok kombinatorius tulajdonságairól. *Matematikai és Fizikai Lapok*, 38:16–28, 1931.

[40] Akinobu Eguchi, Satoru Fujishige, and Akihisa Tamura. A generalized Gale-Shapley algorithm for a discrete-concave stable-marriage model. In *Algorithms and computation*, volume 2906 of *Lecture Notes in Comput. Sci.*, pages 495–504. Springer, Berlin, 2003.

[41] Kimmo Eriksson and Johan Karlander. Stable outcomes of the roommate game with transferable utility. *Internat. J. Game Theory*, 29(4):555–569, 2000.

[42] Kimmo Eriksson and Jonas Sjöstrand. On two theorems of Quinzii and rent controlled housing allocation in Sweden. *To appear in International Game Theory Review*, 2007.

[43] Kimmo Eriksson and Pontus Strimling. How unstable are matchings from the decentralized mate search? *(preprint)*.

[44] Tamás Fleiner. *Stable and crossing structures*. Technische Universiteit Eindhoven, Eindhoven, 2000. Dissertation, Technische Universiteit Eindhoven, Eindhoven, 2000.

[45] Tamás Fleiner. A matroid generalization of the stable matching polytope. In *Integer programming and combinatorial optimization (Utrecht, 2001)*, volume 2081 of *Lecture Notes in Comput. Sci.*, pages 105–114. Springer, Berlin, 2001.

[46] Tamás Fleiner. On the stable b-matching polytope. *Math. Social Sci.*, 46(2):149–158, 2003.

[47] Satoru Fujishige and Akihisa Tamura. A general two-sided matching market with discrete concave utility functions. *Discrete Appl. Math.*, 154(6):950–970, 2006.

[48] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *Amer. Math. Monthly*, 69(1):9–15, 1962.

[49] David Gale and Marilda Sotomayor. Some remarks on the stable matching problem. *Discrete Appl. Math.*, 11(3):223–232, 1985.

[50] Michael R. Garey and David S. Johnson. *Computers and intractability*. W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[51] Donald B. Gillies. Solutions to general non-zero-sum games. In *Contributions to the theory of games, Vol. IV*, Annals of Mathematics Studies, no. 40, pages 47–85. Princeton University Press, Princeton, N.J., 1959.

[52] Dan Gusfield and Robert W. Irving. *The stable marriage problem: structure and algorithms*. Foundations of Computing Series. MIT Press, Cambridge, MA, 1989.

[53] J. D. Horton and K. Kilakos. Minimum edge dominating sets. *SIAM J. Discrete Math.*, 6(3):375–387, 1993.

[54] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discrete Math.*, 2(1):68–72, 1989.

[55] Elena Inarra, Concepcion Larrea, and Elena Molis. Random path to p-stablility in the roommate problem. *Internat. J. Game Theory*, forthcoming.

[56] Robert W. Irving. An efficient algorithm for the "stable roommates" problem. *J. Algorithms*, 6(4):577–595, 1985.

[57] Robert W. Irving. Stable marriage and indifference. *Discrete Appl. Math.*, 48(3):261–272, 1994. Combinatorial Optimization Conference (CO89) (Leeds, 1989).

[58] Robert W. Irving. The cycle roommates problem: a hard case of kidney exchange. *Inf. Process. Lett.*, 103:1–7, 2007.

[59] Robert W. Irving and David F. Manlove. The stable roommates problem with ties. *J. Algorithms*, 43(1):85–105, 2002.

[60] Robert W. Irving and Sandy Scott. The stable fixtures problem: a many-to-many extension of stable roommates. *(submitted)*.

[61] Matthew O. Jackson and Alison Watts. The evolution of social and economic networks. *J. Econom. Theory*, 106(2):265–295, 2002.

[62] Mamoru Kaneko. The central assignment game and the assignment markets. *J. Math. Econom.*, 10(2-3):205–232, 1982.

[63] Mamoru Kaneko and Myrna Holtz Wooders. Cores of partitioning games. *Math. Social Sci.*, 3(4):313–327, 1982.

[64] Viggo Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Inform. Process. Lett.*, 37(1):27–35, 1991.

[65] K. M. Keizer, M. de Klerk, B. J. J. M. Haase-Kromwijk, and W. Weimar. The Dutch algorithm for allocation in living donor kidney exchange. *Transplantation Proceedings*, 37:589–591, 2005.

[66] Alexander S. Kelso and Vincent P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50:1483–1504, 1982.

[67] Donald E. Knuth. *Mariages stables et leurs relations avec d'autres problèmes combinatoires*. Les Presses de l'Université de Montréal, Montreal, Que., 1976. Introduction à l'analyse mathématique des algorithmes, Collection de la Chaire Aisenstadt.

[68] Jeroen Kuipers. *Combinatorial methods in cooperative game theory*. Maastricht University, 1994. PhD Thesis.

[69] M. Le Breton, G. Owen, and S. Weber. Strongly balanced cooperative games. *Internat. J. Game Theory*, 20(4):419–427, 1992.

[70] L. Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Math.*, 2(3):253–267, 1972.

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

[72] Jinpeng Ma. On randomised matching mechanism. *Econ. Theory*, 8(2):377–381, 1996.

[73] David F. Manlove. Stable marriage with ties and unacceptable partners. *Technical Report no. TR-0999-29 of the Computing Science Department of Glasgow University*, 1999.

[74] David F. Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theoret. Comput. Sci.*, 276(1-2):261–279, 2002.

[75] D. G. McVitie and L. B. Wilson. Stable marriage assignment for unequal sets. *BIT*, 10:295–309, 1970.

[76] Natinal Higher Education Information Centre. *http://www.felvi.hu*.

[77] National Resident Matching Program. *http://www.nrmp.org*.

[78] New England Program for Kidney Exchange. *http://www.nepke.org*.

[79] Cheng Ng and Daniel S. Hirschberg. Three-dimensional stable matching problems. *SIAM J. Discrete Math.*, 4(2):245–252, 1991.

[80] Boris G. Pittel and Robert W. Irving. An upper bound for the solvability probability of a random stable roommates instance. *Random Structures Algorithms*, 5(3):465–486, 1994.

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

[82] Eytan Ronn. NP-complete stable matching problems. *J. Algorithms*, 11(2):285–304, 1990.

[83] Alvin E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 6(4):991–1016, 1984.

[84] Alvin E. Roth. Stability and polarization of interests in job matching. *Econometrica*, 52(6):47–57, 1984.

[85] Alvin E. Roth. Conflict and coincidence of interest in job matching: some new results and open questions. *Math. Oper. Res.*, 10(3):379–389, 1985.

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

[87] Alvin E. Roth and Andrew Postlewaite. Weak versus strong domination in a market with indivisible goods. *J. Math. Econom.*, 4(2):131–137, 1977.

[88] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Kidney exchange. *J. Econom. Theory*, 119:457–488, 2004.

[89] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. A kidney exchange clearinghouse in New England. *American Economic Review, Papers and Proceedings*, 95(2):376–380, 2005.

[90] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Pairwise kidney exchange. *J. Econom. Theory*, 125(2):151–188, 2005.

[91] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Coincidence of wants in markets with compatibility based preferences. *American Economic Review*, (forthcoming).

[92] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-sided matching*, volume 18 of *Econometric Society Monographs*. Cambridge University Press, Cambridge, 1990. A study in game-theoretic modeling and analysis, With a foreword by Robert Aumann.

[93] Alvin E. Roth and John H. Vande Vate. Random paths to stability in two-sided matching. *Econometrica*, 58(6):1475–1480, 1990.

[94] S. L. Saidman, Alvin E. Roth, Tayfun Sönmez, M. Utku Ünver, and S. L. Delmonico. Increasing the opportunity of live kidney donation by matching for two and three way exchanges. *Transplantation*, 81(5):773–782, 2006.

[95] S. L. Saidman, Alvin E. Roth, Tayfun Sönmez, M. Utku Ünver, and S. L. Delmonico. Utilizing list exchange and undirected good samaritan donation through chain paired kidney exchanges. *American Journal of Transplantation*, 6(11):2694–2705, 2006.

[96] Herbert Scarf. *The computation of economic equilibria.* Yale University Press, New Haven, Conn., 1973. With the collaboration of Terje Hansen, Cowles Foundation Monograph, No. 24.

[97] Herbert E. Scarf. The core of an N person game. *Econometrica*, 35:50–69, 1967.

[98] Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics.* Springer-Verlag, Berlin, 2003. Paths, flows, matchings, Chapters 1–38.

[99] Sandy Scott. *A study of stable marriage problems with ties.* 2005. PhD Thesis, University of Glasgow.

[100] S. L. Segev, S. E. Gentry, D. S. Warren, B. Reeb, and R. A. Montgomery. Kidney paired donation and optimizing the use of live donor organs. *J. Am. Med. Assoc.*, 293:1883–1890, 2005.

[101] L. S. Shapley. On balanced sets and cores. *Naval. Res. Logist. Quart.*, 14:453–460, 1967.

[102] L. S. Shapley and M. Shubik. The assignment game. I. The core. *Internat. J. Game Theory*, 1(2):111–130, 1972.

[103] Lloyd Shapley and Herbert Scarf. On cores and indivisibility. *J. Math. Econom.*, 1(1):23–37, 1974.

[104] Jimmy J. M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *J. Algorithms*, 12(1):154–178, 1991.

[105] Jimmy J. M. Tan and Yuang Cheh Hsueh. A generalization of the stable matching problem. *Discrete Appl. Math.*, 59(1):87–102, 1995.

[106] S. H. Tijs, T. Parthasarathy, J. A. M. Potters, and V. Rajendra Prasad. Permutation games: Another class of totally balanced games. *OR Spectrum*, 6:119–123, 1984.

[107] United Network for Organ Sharing. *http://www.unos.org.*

[108] A. A. Vasin and V. A. Gurvič. Reconcilable sets of coalitions. In *Questions in applied mathematics (Russian)*, pages 20–30. Sibirsk. Ènerget. Inst., Akad. Nauk SSSR Sibirsk. Otdel., Irkutsk, 1977.

[109] Y Yuan. Residence exchange wanted: A stable residence exchange problem. *European Journal of Operational Research*, 90:331–341, 1996.