

The Complexity of Matching Games: A Survey

Márton Benedek

Péter Biró

*Institute of Economics, KRTK,
Tóth Kálmán u. 4., Budapest, Hungary
Corvinus University of Budapest
Fővám tér 8., Budapest, Hungary*

BENEDEK.MARTON@KRTK.HU

BIRO.PETER@KRTK.HU

Matthew Johnson

Daniël Paulusma

Xin Ye

*Department of Computer Science, Durham University,
Upper Mountjoy Campus, Stockton Road, Durham, UK*

MATTHEW.JOHNSON2@DURHAM.AC.UK

DANIEL.PAULUSMA@DURHAM.AC.UK

XIN.YE@DURHAM.AC.UK

Abstract

Matching games naturally generalize assignment games, a well-known class of cooperative games. Interest in matching games has grown recently due to some breakthrough results and new applications. This state-of-the-art survey provides an overview of matching games and extensions, such as b -matching games and partitioned matching games; the latter originating from the emerging area of international kidney exchange. In this survey we focus on computational complexity aspects of various game-theoretical solution concepts, such as the core, nucleolus and Shapley value, when the input is restricted to a matching game or one of its variants.

1. Introduction

We consider a wide range of matching problems that involve monetary payments. In order to illustrate this, we start with an artificial but simple example where the payments come in the form of prize money. This example has its origins in a paper of Eriksson and Karlander (2001).

1.1 An Initial Example

Suppose a group N of tennis players play a doubles tennis tournament. Each pair of players estimates the expected prize money when they form a doubles partnership. Two players that are not matched to each other may leave their partner if together they can win more prize money. Eriksson and Karlander (2001) studied the question of whether the players can be *matched* in pairs in such a way that this situation does not occur. A wider question is whether the players can be matched together such that no *group* of players expects to be awarded more prize money by playing an event on their own. This situation describes a *matching game*; see Section 3 for a formal definition.



Figure 1: *Left*: a pool of patient-donor pairs, where an arc from a pair $i = (p, d)$ to a pair $j = (p', d')$ indicates that donor d is compatible with patient p' . *Right*: the corresponding compatibility graph.

1.2 A Second Example: International Kidney Exchange

Our next example is more elaborate and leads to the notion of a *partitioned matching game*, which we also formally define in Section 3. This example is related to an emerging application area, namely that of *international kidney exchange*. This application area formed the main motivation for us to write our survey and below we discuss it in detail.

Chronic kidney disease can lead to kidney failure, which can have a devastating impact on patients' lives. Currently, the most effective treatment for kidney failure is transplantation, which could be either from a deceased or living donor. The long-term outcomes of the latter type of donation are best. However, one major obstacle that prevents a living donor from donating a kidney to a family member or friend is medical incompatibility between patient and donor. Namely, with high probability the donor's kidney would be rejected by the patient's body.

To circumvent medical incompatibility, national Kidney Exchange Programmes (KEPs) have been established in many countries in the last 30 years (Biró et al., 2019a). In a KEP, a patient and their willing but incompatible donor are placed in a pool with other patient-donor pairs. Now, if donor d of patient p is compatible with patient p' and simultaneously, donor d' of p' is compatible with p , then a *pairwise exchange* can be made, that is, d and d' can be “swapped”. In this way, both p and p' can now receive treatment. Hence, the problem is to find a solution in a given pool of patient-donor pairs that helps as many patients as possible. Such a solution corresponds to a maximum matching in the *compatibility graph*, which is obtained by introducing a vertex for each patient-donor pair in the pool and adding an edge between two vertices (p, d) and (p', d') if and only if d is compatible with p' and d' is compatible with p ; see Figure 1 for an example.¹

1. Countries may also allow cycles, e.g. d_1 donates to p_2 , d_2 to p_3 and d_3 to p_1 ; or paths, e.g. an altruistic donor d^* donates to p_1 , d_1 to p_2 , d_2 to p_3 and d_3 to the deceased donor waiting list. More patients now receive treatment. However, in many countries altruistic donors are forbidden. Moreover, cycles must be kept small to prevent them from breaking. Therefore, in several countries only pairwise exchanges are allowed, such as in France and Hungary (Biró et al., 2019a). If the upper bound on the cycle length is larger than 2 or if altruistic donors are allowed, then we can no longer model the problem of maximizing the number of transplants as a matching problem; the problem also becomes NP-hard (Abraham et al., 2007).

KEPs operate in regular rounds, each time considering an updated pool, where some patient-donor pairs have left and new ones have come in. In each round, the primary goal of a KEP is to maximize the number of kidney transplants. However, KEPs may also take other factors into account, such as the quality of the transplants. Transplant qualities can be represented by assigning weights to the edges of the compatibility graph. The goal is then to find a matching of maximum weight in the compatibility graph.

As merging pools of national KEPs could lead to even more patients being treated, *international* KEPs are now starting to be formed. The problem now is not just to treat many patients, but to do so in such a way that each country believes they are being treated fairly. If some countries receive fewer kidneys than they could receive by forming a smaller international KEP or by running their own KEP again, then the international KEP may fall apart. Thus we must focus on the *stability* of international KEPs.

Klimentova et al. (2021) introduced a credit system to ensure stability of international KEPs. In their model, the number of kidney transplants for a country i is specified by a “fair” target number for i . The differences between the actual number of transplants for i and this target number is given as credits for i in the next round. Simulations using their model have also been performed, with different parameters, by Biró et al. (2020), and for international KEPs with pairwise exchanges only, but with a larger number of countries than the two previous studies, by Benedek et al. (2022). One of the central considerations in this model is to decide what target numbers are fair. Biró et al. (2019b) modelled an international KEP with pairwise exchanges only as a partitioned matching game, in which each country is a player. This makes it possible to use solution concepts from Cooperative Game Theory for finding fair target allocations. We will explain this in more detail in Section 3.

1.3 Overview of the Survey

In Section 2 we introduce some background information on Cooperative Game Theory and Graph Theory. This enables us to formally define the concept of a matching game, its special case of an assignment game and two closely related generalizations of matching games in Section 3, one of which is the notion of a partitioned matching game and the other is known as the b -matching game. In the same section, we also define the fractional matching game, a known relaxation of a matching game. In Section 4 we discuss the well-known network bargaining games, which have a close relationship with matching games.

In Sections 5–7 we survey known computational complexity results of the core, nucleolus and Shapley value, respectively, for assignment games, b -assignment games, matching games, b -matching games and partitioned matching games. Unless specified otherwise, running times of algorithms are expressed in the input size $n + m$ of the underlying (weighted) graph $G = (V, E)$ where $n = |V|$ and $m = |E|$. In Section 5 we explain in particular to what extent solutions for network bargaining games correspond to finding core allocations of matching games and their variants. In the same section we also present complexity results on *core stabilizers*. Stabilizers allow for making small modifications to the underlying graph, such that the adjusted game has a non-empty core.

Throughout our survey we list relevant open problems, and in Section 8 we conclude our survey by listing some more.

2. Background

In Section 2.1 we give some basic cooperative game theory terminology and notation; see also the textbooks of Owen (2013) and Chalkiadakis, Elkind, and Wooldridge (2011). In Section 2.2 we give the necessary graph theory terminology and notation; see also the textbooks of Diestel (2012) and Bondy and Murty (2008).

2.1 Cooperative Game Theory

A (*cooperative*) *game* is a pair (N, v) , where $N = \{1, \dots, |N|\}$ is a set of *players* (agents) and $v : 2^N \rightarrow \mathbb{R}_+$ is a *value function* (or *characteristic function*) with $v(\emptyset) = 0$; here, \mathbb{R}_+ denotes the set of non-negative real numbers. A *coalition* is a subset of N . For a coalition $S \subseteq N$, the value $v(S)$ describes the joint profit or costs if the players in S collaborate with each other. In line with the type of game we consider in this survey, we let v be a profit function in the remainder of the paper. Then, assuming that $v(N)$ is maximum over all partitions of N , the central problem is how to keep the *grand coalition* N stable by distributing the *total value* $v(N)$ amongst the players of N in a fair way.

An *allocation* (or *pre-imputation*) for a game (N, v) is a vector $x \in \mathbb{R}^N$ with $x(N) = v(N)$; here, for $S \subseteq N$, we write $x(S) = \sum_{i \in S} x_i$. A *solution concept* prescribes a set of allocations for every game. Each solution concept has its own fairness properties and computational complexities; the choice of a certain solution concept depends on context. In this survey we focus on computational complexity aspects of solution concepts.

The *core* was formally introduced by Gillies (1959) and is the best-known solution concept. It consists of all allocations $x \in \mathbb{R}^N$ with $x(S) \geq v(S)$ for every $S \subseteq N$. Core allocations (that is, allocations that belong to the core) ensure stability, as no group of players has any incentive to leave the grand coalition N . However, the core of a game might be empty. In Section 3 we illustrate by an example that this may happen even for matching games with only three players.

Should the core be empty we might seek for an allocation x that satisfies all core constraints “as much as possible” by solving the following linear program:

$$\begin{aligned} \text{(LC)} \quad \varepsilon_1 := \max \varepsilon \\ x(S) \geq v(S) + \varepsilon \quad \text{for all } S \in 2^N \setminus \{\emptyset, N\} \\ x(N) = v(N). \end{aligned}$$

The set of allocations x for which (x, ε_1) is an optimum solution of (LC) has been introduced by Maschler et al. (1979) as the *least core* of (N, v) . The least core belongs to the core if the core is non-empty, which is the case if and only if $\varepsilon_1 \geq 0$. Note that by excluding the sets \emptyset and N in the definition of (LC), the least core might be a proper subset of the core, should the core be non-empty.

In contrast to the core, the *Shapley value* $\phi(N, v)$ always exists. It is the unique singleton solution concept that satisfies some set of desirable fairness properties (Shapley, 1953). It assigns every player $i \in N$:

$$\phi_i(N, v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left(v(S \cup \{i\}) - v(S) \right).$$

The Shapley value might be negative but not for the games we consider. Moreover, the Shapley value might not be in the core even if the core is non-empty. This may happen even for matching games, as we will demonstrate in Section 5.1 with an example.

We now define another well-known solution concept, namely the nucleolus. In contrast to the Shapley value, the nucleolus is always in the core if the core is non-empty. The *excess* of a non-empty coalition $S \subsetneq N$ for an allocation x is given by $e(S, x) = x(S) - v(S)$. Ordering the $2^{|N|} - 2$ excesses non-decreasingly gives us the *excess vector* $e(x)$. The *nucleolus* $\eta(N, v)$ is the unique allocation (Schmeidler, 1969) that lexicographically maximizes $e(x)$ over the set of *imputations* $I(N, v)$, which consists of all allocations x with $x_i \geq v(\{i\})$ for $i \in N$; the nucleolus is only defined if $I(N, v)$ is non-empty. In Sections 3 and 5.1 we give some examples of matching games, for which we computed the nucleolus.

An alternative definition of the nucleolus is given by Maschler et al. (1979); solve for $r \geq 1$, the following sequence of linear programs until a unique solution, which gives us the nucleolus, remains:

$$\begin{aligned}
 (\text{LP}_r) \quad \varepsilon_r &:= \max \varepsilon \\
 x(S) &\geq v(S) + \varepsilon \quad \text{for all } S \in 2^N \setminus \{\mathcal{S}_0 \cup \dots \cup \mathcal{S}_{r-1}\} \\
 x(S) &= v(S) + \varepsilon_i \quad \text{for all } S \in \mathcal{S}_i \quad (0 \leq i \leq r-1) \\
 x &\in I(N, v),
 \end{aligned}$$

where $\varepsilon_0 = -\infty$, $\mathcal{S}_0 = \emptyset$ and for $i \geq 1$, \mathcal{S}_i consists of all coalitions S with $x(S) = v(S) + \varepsilon_i$ for every optimal solution (x, ε_i) of (LP_i) . From this definition and the definition of the least core, it follows that if the core is non-empty, then the nucleolus exists and even belongs to the least core.

Finally, we define the pre-kernel, introduced by Maschler et al. (1971). We let

$$s_{ij}(x) = \min\{e(S, x) \mid S \subseteq N \text{ with } i \in S, j \notin S\}.$$

The *pre-kernel* of a game (N, v) is the set of all allocations $x \in \mathbb{R}^N$ with $s_{ij}(x) = s_{ji}(x)$ for every pair of distinct players $i, j \in N$.

2.2 Graph Theory

Throughout this section, the graph G is an undirected graph with no self-loops and no multiple edges. Let V be the vertex set of G , and let E be the edge set of G . We denote an edge between two vertices i and j as ij (or ji). For a subset $S \subseteq V$, we let $G[S]$ denote the subgraph of G *induced* by S , that is, $G[S]$ is the graph obtained from G after deleting all vertices that are not in S (and their incident edges). We say that G is *bipartite* if there exist two disjoint vertex subsets A and B such that $V = A \cup B$ and every edge in E joins a vertex in A to a vertex in B .

A function f that maps some set X to \mathbb{R} is *positive* if $f(x) > 0$ for every $x \in X$. For an integer t , we write $f \equiv t$ if $f(x) = t$ for every $x \in X$, and we write $f \leq t$ if $f(x) \leq t$ for every $x \in X$.

Now, let $M \subseteq E$ be some subset of edges. For a positive edge weighting w of G , the *weight* of M is defined as $w(M) = \sum_{e \in M} w(e)$. For a positive vertex capacity function b ,

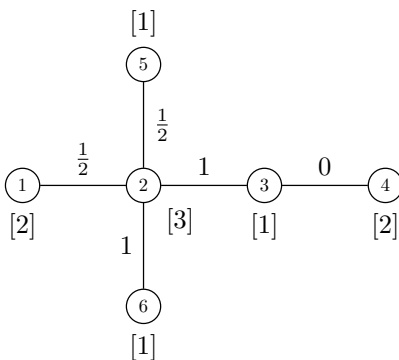


Figure 2: An example of a half- b -matching f in a graph G with $w \equiv 1$ and where b is given by $b(3) = b(5) = b(6) = 1$; $b(1) = b(4) = 2$; and $b(2) = 3$, as illustrated by the $[x]$ labels in the figure. Note that $f(12) = f(25) = \frac{1}{2}$; $f(23) = f(26) = 1$; $f(34) = 0$; and $w(f) = 3$.

we say that M is a b -matching of G if each $i \in V$ is incident to at most $b(i)$ edges in M . If $b \equiv 1$, then M is a *matching*.

A *fractional b -matching* of G is an edge mapping $f : E \rightarrow [0, 1]$ such that $\sum_{e:i \in e} f(e) \leq b(i)$ for each $i \in V$. The *weight* of a fractional b -matching f is defined as $w(f) = \sum_{e \in E} w(e)f(e)$. If $b \equiv 1$, then f is a *fractional matching*. Note that f is a b -matching of G if and only if $f(e) \in \{0, 1\}$ for every $e \in E$. We say that f is a *half- b -matching* of G if $f(e) \in \{0, \frac{1}{2}, 1\}$ for every $e \in E$. If $b \equiv 1$, then a half- b -matching of G is also called a *half-matching*. See Figure 2 for an example of a half- b -matching.

3. Matching Games and Generalizations

In this section, we define matching games, assignment games, b -matching games, b -assignment games and partitioned matching games. We also show how the two examples from Section 1 can be seen as matching games and partitioned matching games, respectively.

Definition 1. (Deng et al., 1999) A *matching game* on a graph $G = (V, E)$ with a positive edge weighting w is the game (N, v) where $N = V$ and for $S \subseteq N$, the value $v(S)$ is the maximum weight $w(M)$ over all matchings M of $G[S]$.

Matching games on bipartite graphs are commonly known as *assignment games* (Shapley & Shubik, 1971). In the example of Section 1.1, the tennis players form the vertices of a graph such that there exists an edge between two vertices i and j if and only if i and j can form a doubles pair together, with expected prize money $w(ij)$. A game on a graph G with edge weighting w is *uniform* (or *simple*) if $w \equiv 1$.

Already uniform matching games may have an empty core. For example, consider the uniform matching game (N, v) defined on a graph G that is a triangle with vertices h, i, j and edges hi, ij and jh ; see also Figure 3. The reason for core-emptiness for matching games is that every imputation x has a so-called blocking pair. That is, a pair of adjacent players $\{i, j\}$ in a matching game (N, v) is a *blocking pair* for an imputation x if $x_i + x_j < w(ij)$

holds. The *blocking value* of $\{i, j\}$ for x is defined as

$$e_x(ij)^+ = \max\{0, w(ij) - (x_i + x_j)\}.$$

If $\{i, j\}$ is not blocking x , then $e_x(ij)^+ = 0$; otherwise, $e_x(ij)^+$ expresses the extent to which $\{i, j\}$ blocks x . We let $B(x) = \sum_{ij \in E} e_x(ij)^+$ be the *total blocking value* for x . Deng et al. (1999) showed that an imputation x is in the core of a matching game (N, v) defined on a graph G with positive edge weighting w if and only if for every $ij \in E$,

$$x_i + x_j \geq w(ij),$$

so if and only if $B(x) = 0$ and x has no blocking pairs. This follows as if there is a blocking pair then x is not in the core, and, for the other direction, suppose that S is a blocking coalition. Let M be a maximum weight matching in $G[S]$. Then $x(S) < v(S) = w(M) = \sum_{e_i \in M} w(e_i)$, and so for at least one of the edges $ij \in M$, $x_i + x_j < w(ij)$ must hold; hence x not in the core. For a (larger) example of a matching game with a non-empty core, we refer to Example 2 in Section 5.1.

Definition 2. (Biró et al., 2018) A *b-matching game* on a graph $G = (V, E)$ with a positive vertex capacity function b and a positive edge weighting w is the game (N, v) where $N = V$ and for $S \subseteq N$, the value $v(S)$ is the maximum weight $w(M)$ over all *b-matchings* M of $G[S]$.

We refer to Figure 4 for an example. If $b \equiv 1$, then we have a matching game. If G is bipartite, then we speak of a *b-assignment game*. These games were introduced by Sotomayor (1992) as multiple partner assignment games. If $b \equiv t$ for some integer t , then we speak of a *t-matching game* or a *t-assignment game*. Note that 1-assignment games correspond to assignment games. A *b-matching game* is called a *multiple partners matching game* in the paper of Biró et al. (2018).

In the next definition, the sets N (of players) and V (of vertices) are different, as we now associate each player $i \in N$ with a distinct subset V_i of the vertex set V .

Definition 3. A *partitioned matching game* (Biró et al., 2019b) on a graph $G = (V, E)$ with a positive edge weighting w and partition $(V_1, \dots, V_{|N|})$ of V is the game (N, v) , where $N = \{1, \dots, |N|\}$, and for $S \subseteq N$, the value $v(S)$ is the maximum weight $w(M)$ over all matchings M of $G[\bigcup_{i \in S} V_i]$.

Partitioned matching games were introduced as *generalized matching games* by Biró et al. (2019b) and are also known as *international matching games* (Király & Mészáros-Karkus, 2019). We say that

$$c = \max\{|V_i| \mid 1 \leq i \leq |N|\}$$

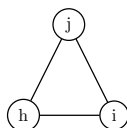


Figure 3: The graph K_3 , whose corresponding uniform matching game has an empty core.

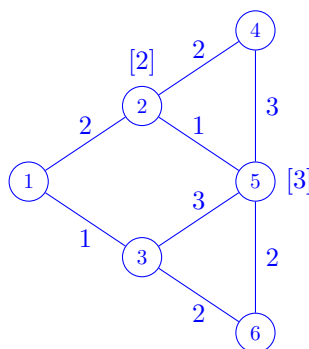


Figure 4: A b -matching game (N, v) with six players, where $b \equiv 1$ except $b(2) = 2$ and $b(5) = 3$, so $b^* = 3$. Note that $v(N) = 10$ (take $M = \{12, 35, 45, 56\}$).

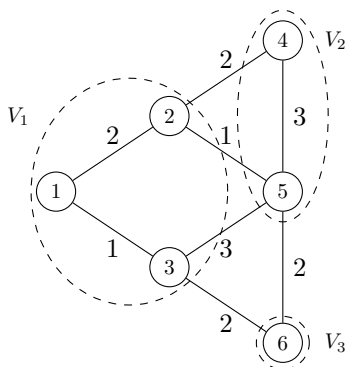


Figure 5: A partitioned matching game (N, v) of width $c = 3$ defined on a graph $G = (V, E)$ with an edge weighting w . Note that N consists of three players and that V is partitioned into $\{1, 2, 3\}, \{4, 5\}, \{6\}$, as indicated by the dotted circles. Also note that $v(N) = 7$.

is the *width* of (N, v) . Note that if $V_i = \{i\}$ for every $i \in N$, then we obtain a matching game (which has width 1). We refer to Figure 5 for an example of a partitioned matching game with width 3.

We can now model an international KEP with pairwise exchanges only as a partitioned matching game (N, v) that is defined on a compatibility graph $G = (V, E)$ where

- the set $N = \{1, \dots, |N|\}$ is a set of countries participating in the international KEP;
- the set V is partitioned into subsets $V_1, \dots, V_{|N|}$, where V_i consists of the patient-donor pairs of country i for every $i \in N$; and
- the set E may have an edge weighting w , where $w(uv)$ expresses the utility of a transplant exchange between patient-donor pairs u and v (if the aim is to maximize the number of transplants we can set $w \equiv 1$).

To compute the target numbers for each of the countries, Benedek et al. (2022) used two solution concepts, the benefit value and the contribution value, which can be computed in linear time,² and two solution concepts, the Shapley value and the nucleolus, which in

2. This follows immediately from their definitions, see Benedek et al. (2022).

general take exponential time to compute; in particular, they employed the state-of-the-art method (Benedek et al., 2021) for computing the nucleolus. From their simulations, the Shapley value yielded the best results. This is in line with the findings of Klimentova et al. (2021) and Biró et al. (2020), who considered international KEPs that allowed cycles of length 3, but for simulations involving fewer countries than in the study of Benedek et al. (2022) (see also Section 1.2).

Apart from international KEPs, partitioned matching games can also be used to model economic settings where multi-organizations own pools of clients (Gourvès et al., 2013). As we will discuss further in Section 5, Biró et al. (2019b) performed a theoretical study for partitioned matching games and showed how b -matching games and partitioned matching games are related to each other.

We finish this section by defining a known relaxation of matching games. A *fractional matching game* on a graph $G = (V, E)$ with a positive edge weighting w is the game (N, v) where $N = V$ and for $S \subseteq N$, the value $v(S)$ is the maximum weight $w(M)$ over all fractional matchings M of $G[S]$. We can define the notions of a *fractional b -assignment game* and *fractional b -matching game* analogously. These notions have been less studied than their non-fractional counterparts and are not the focus of our survey. Nevertheless, we will mention relevant complexity results for these games as well.

4. Network Bargaining Games

Some economic situations involve only preferences of players. These situations are beyond the scope of this survey. The economic setting that is described in this section includes payments. As we will explain in Section 5, this setting is closely related to the setting of b -matching games.

Let G be a graph with a positive edge weighting w and a positive vertex capacity function b . Previously we defined a b -matching game (N, v) on (G, b, w) , but the triple (G, b, w) is also known as a *network bargaining game*. Again, the vertices are players, but now an edge ij represents a possible contract between players i and j . The weight $w(ij)$ represents the value of the contract, while the *capacity* $b(i)$ is the maximum number of contracts player i can commit to. A b -matching M can now be viewed as a set of pairwise contracts between players. However, M might be *blocked* by two players i and j . To define the notion of a blocking pair in this context we need some further terminology.

If $ij \in M$, we distribute the value $w(ij)$ of the contract ij by defining *pay-offs* $p(i, j) \geq 0$ and $p(j, i) \geq 0$ to i and j , respectively, such that

$$p(i, j) + p(j, i) = w(ij).$$

If $ij \notin M$, then we set $p(i, j) = p(j, i) = 0$. The vector p with entries $p(i, j)$ is a *pay-off* and the pair (M, p) is a *solution* for (G, b, w) . The *total pay-off vector* p^t is defined by $p^t(i) = \sum_{j:ij \in E} p(i, j)$ for every $i \in N$. The *utility* $u_p(i)$ is equal to $\min\{p(i, j) \mid ij \in M\}$ if i is incident to $b(i)$ edges of M ; otherwise $u_p(i) = 0$. We can now define a *blocking pair* with respect to (M, p) as a pair $\{i, j\}$ with $ij \in E \setminus M$ such that

$$u_p(i) + u_p(j) < w(ij).$$

The goal is to decide if (G, b, w) has a (*pairwise*) *stable* solution, that is, a solution with no blocking pairs. This problem is called the STABLE FIXTURES WITH PAYMENTS (SFP) problem (Biró et al., 2018). Restrictions of SFP are called:

- MULTIPLE PARTNERS ASSIGNMENT (MPA) if G is bipartite (Sotomayor, 1992);
- STABLE ROOMMATES WITH PAYMENTS (SRP) if $b \equiv 1$ (Eriksson & Karlander, 2001);
- STABLE MARRIAGE WITH PAYMENTS (SMP) if G is bipartite, $b \equiv 1$ (Koopmans & Beckmann, 1957).

We denote instances of SRP and SMP by (G, w) instead of $(G, 1, w)$.

It is known that SFP (and thus MPA, SMP, SRP) is polynomial-time solvable (Biró et al., 2018; Farczadi et al., 2013).

The main reason for discussing network bargaining games in our survey is due to their insightful relationship with the core of b -matching games (see Section 5). However, we also note the well-known result of Kleinberg and Tardos (2008) who gave a polynomial-time algorithm for finding a *balanced* stable solution (M, p) for SRP. Afterwards, Bateni et al. (2010) proved that a solution (M, p) for an instance (G, w) of SRP is stable and balanced if and only if p^t belongs to the intersection of the core and the pre-kernel of the matching game (N, v) defined on (G, w) . Balanced stable solutions are beyond the scope of our survey; for the sake of completeness, we include their definition below.

Farczadi et al. (2013) extended the original definition of a balanced stable solution from SPR to SFP. Given a stable solution (M, p) for an instance (G, b, w) of SFP, we let

$$\alpha_p(i) = \max(0, \max\{w_{ij} - u_p(i) \mid ij \in E(G) \setminus M\})$$

be the *outside gain*, which is the maximum that player i can gain on top of its total payoff $p^t(i)$ from any neighbour j with $ij \notin M$. A stable solution (M, p) is *balanced* if for every $ij \in M$,

$$p(i, j) - \alpha_p(i) = p(j, i) - \alpha_p(j).$$

Intuitively, this means that the value w_{ij} is shared between i and j in a balanced way: the outside gain of player i is equal to the outside gain of player j . We discuss the paper of Farczadi et al. (2013) in more detail in the next section.

5. Complexity Aspects of the Core

In Section 5.1 we survey complexity results for the problems of core membership, core non-emptiness and finding core allocations for matching games and the variants of matching games defined in Section 3. In Section 5.2 we survey the known results for core stabilizers for matching games.

5.1 Testing Core Membership, Non-Emptiness and Finding Core Allocations

We consider three natural problems, which have also been studied for other games and with respect to other solution concepts. Each of these problems takes as input a game (N, v) .

	bipartite graphs	general graphs
$b \equiv 1$	SMP assignment game	SRP matching game
stable solution	<i>exists</i> ¹	<i>may not exist</i>
core	<i>always non-empty</i> ²	<i>may be empty</i>
any b	MPA b -assignment game	SFP b -matching game
stable solution	<i>exists</i> ³	<i>may not exist</i>
core	<i>always non-empty</i> ³	<i>may be empty</i>

Table 1: The four variants of network bargaining games and their counterpart matching games. Recall that SFP (and thus MPA, SMP, SRP) is polynomial-time solvable (Biró et al., 2018; Farczadi et al., 2013). The references for the statements on the existence of a stable solution and on the core are ¹Koopmans and Beckmann (1957), ²Shapley and Shubik (1971), ³Sotomayor (1992). See Table 2 for a summary of the complexity results for matching games.

- P1.** (core membership) decide if a given allocation x belongs to the core, or else find a coalition S with $x(S) < v(S)$;
- P2.** (core non-emptiness) decide if the core is non-empty; and
- P3.** (core allocation computation) find an allocation in the core (if it is non-empty).

We note that if P1 is polynomial-time solvable for some class of games \mathcal{G} , then by using the ellipsoid method (Grötschel et al., 1981; Khachiyan, 1979), it follows that P2 and P3 are also polynomial-time solvable for \mathcal{G} .

In the remainder we explain that the problems SMP, MPA, SRP and SFP are closely related to problems P1–P3 for the corresponding matching game variant. To help the reader keeping track of this we refer to Table 1 for an overview (below we will explain the results in this table as part of our discussion).

We start by considering the assignment games and b -assignment games. The problems SMP and MPA are closely related to problems P2 and P3 for assignment games and b -assignment games, respectively. Namely, Shapley and Shubik (1971) proved that every core allocation of an assignment game corresponds to a pay-off vector in a stable solution for SMP and vice versa. Moreover, Koopmans and Beckmann (1957) proved that SMP, and thus P2, only has yes-instances and that a stable solution, and thus a core allocation of every assignment game, can be found in polynomial time. Finally, Sotomayor (1992) extended all these results to MPA and b -assignment games; recall that p^t denotes the total pay-off vector.

Theorem 1 (Sotomayor, 1992) *Every instance (G, b, w) of MPA has a stable solution, which can be found in polynomial time. Moreover, if (M, p) is a stable solution for (G, b, w) , then M is a maximum weight b -matching of G and p^t is a core allocation of the b -assignment game defined on (G, b, w) .*

The converse of the second statement of Theorem 1 does not hold, as there may exist an allocation in the core which does not correspond to a stable solution (Sotomayor, 1992).

In contrast to the situation for assignment games, the core of a matching game defined on a non-bipartite graph may be empty; just recall the example from Section 3 where G is the triangle and $w \equiv 1$. Problem P1 is linear-time solvable for matching games, and thus also for assignment games. Recall from Section 3 that it suffices to check if for an allocation x , we have for every $ij \in E$,

$$x_i + x_j \geq w(ij).$$

As P1 is polynomial-time solvable, we immediately have that P2 and P3 are also polynomial-time solvable for matching games. In particular, the following connection with SRP is well known; note that for SRP, total pay-off vectors and pay-off vectors are in 1-1-correspondence (see also Example 1).

Theorem 2 (Eriksson & Karlander, 2001) *A solution (M, p) for an instance (G, w) of SRP is stable if and only if M is a maximum weight matching of G and p^t is a core allocation of the matching game defined on (G, w) .*

Deng et al. (1999) followed a more direct approach for solving P2 and P3 for matching games. Later, a faster algorithm for P2 and P3 was given by Biró et al. (2012). This algorithm runs in $O(nm + n^2 \log n)$ time and is based on an alternative characterization of the core of a matching game, which follows from a result of Balinski (1965).

Theorem 3 (Biró et al., 2012) *The core of a matching game (N, v) , on a graph G with positive edge weighting w , is non-empty if and only if the maximum weight of a matching in G equals the maximum weight of a half-matching in G .*

In order to use Theorem 3 algorithmically, Biró et al. (2012) used the duplication technique of Nemhauser and Trotter Jr. (1975), as illustrated in Figure 6. That is, for an input graph G with edge weighting w , first construct in polynomial-time the so-called *duplicate* G^d with edge weighting w^d , as follows. For every vertex $i \in V$, we introduce two vertices i' and i'' in G^d . For every edge $ij \in E$, we introduce the edges $i'j''$ and $i''j'$, each of weight $w^d(i'j'') = w^d(i''j') = \frac{1}{2}w(ij)$, in G^d . It is not difficult to see that the maximum weight of a half-matching in G equals the maximum weight of a matching in G^d . Hence, it remains to compare the latter weight with the maximum weight of a matching in G . One can compute both these values in $O(n^3)$ time by using Edmonds' algorithm (Edmonds, 1965).

We now give two examples. The goal of Example 1 is to illustrate the working of Theorems 2 and 3, whereas Example 2 gives another illustration of the power of Theorem 3.

Example 1. Consider the graph G and its duplicate G^d in Figure 6. The matching $M = \{12, 36, 45\}$ is a maximum weight matching of G with maximum weight 7. The maximum weight of a matching in G^d is also equal to 7. Hence, by Theorem 3, we find that the core is non-empty and thus the nucleolus $\eta = (\frac{1}{2}, \frac{3}{2}, \frac{3}{2}, 1, 2, \frac{1}{2})$ is a core allocation, as can be readily verified. Moreover, the pair (M, p) , where p is defined by $p(1, 2) = \frac{1}{2}$, $p(2, 1) = \frac{3}{2}$, $p(3, 6) = \frac{3}{2}$, $p(4, 5) = 1$, $p(5, 4) = 2$ and $p(6, 3) = \frac{1}{2}$ and else $p(i, j) = 0$, is a stable solution for the corresponding instance (G, w) of SRP, in line with Theorem 2. On the other hand, the Shapley value $\phi = (\frac{23}{30}, \frac{41}{30}, \frac{36}{30}, \frac{31}{30}, \frac{55}{30}, \frac{24}{30})$ is not a core allocation, as $v(\{4, 5\}) = 3 > \frac{86}{30} = \phi(\{4, 5\})$. \diamond

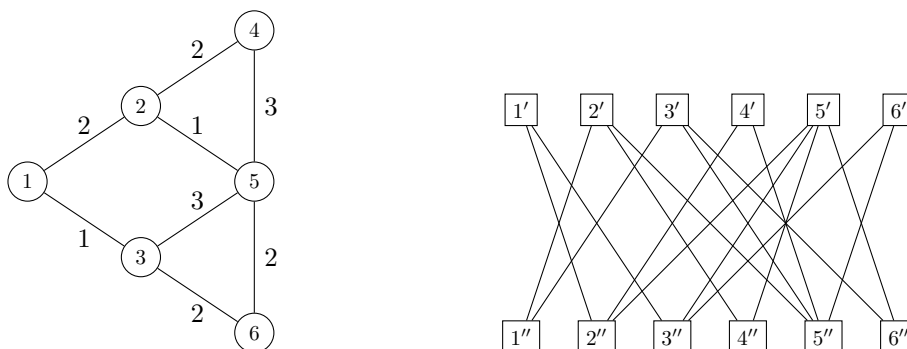


Figure 6: *Left*: an example of a matching game on a graph G . *Right*: the duplicate G^d of G ; note that, for instance, $w^d(3'5'') = \frac{3}{2}$. See also Examples 1 and 2.

Example 2. Let G' be the graph obtained from the graph G in Figure 6 after removing vertex 6. Unlike the matching game on G , the matching game on G' has an empty core. The reason is that the maximum weight ($5\frac{1}{2}$) of a matching in G'^d is larger than the maximum weight (5) of a matching in G' , and we can apply Theorem 3 again. \diamond

By Theorem 1, every b -assignment game is a yes-instance for P2 and P3 is polynomial-time solvable for b -assignment games. In contrast, Biró et al. (2018) proved that P1 is coNP-complete even for uniform 3-assignment games.

On the positive side, Biró et al. (2018) also showed that for $b \leq 2$, P1 (and thus P2 and P3) is polynomial-time solvable even for b -matching games. They did this by a reduction to the polynomial-time solvable *tramp steamer* problem. The latter problem is that of finding a cycle C in a graph G with maximum profit-to-cost ratio

$$\frac{p(C)}{w(C)},$$

where p and w represent profit and cost functions, respectively, that are defined on $E(G)$.

Before considering partitioned matching games, it remains to discuss P2 and P3 for b -matching games for which $b \not\leq 2$. Theorem 2 implies that besides the aforementioned direct algorithms, any polynomial-time algorithm for SRP can also be used for immediately solving P2 and P3 for the corresponding matching game. As such, the following theorem for SFP, shown independently by Biró et al. (2018) and Farczadi et al. (2013)³, is initially promising for b -matching games.

Theorem 4 (Biró et al., 2018; Farczadi et al., 2013) *SFP can be solved in polynomial time. Moreover, if (M, p) is a stable solution for an instance (G, b, w) of SFP, then M is a maximum weight b -matching of G and p^t is a core allocation of the b -matching game defined on (G, b, w) .*

3. The algorithm of Farczadi et al. (2013) even computes a balanced stable solution by mapping an instance of the problem to an instance of SRP where each vertex i has as many copies as its original capacity $b(i)$, and then applying the result of Kleinberg and Tardos (2008) on finding balanced stable solutions for SRP (see Section 4). The $O(n^2 m \log(n^2/m))$ -time algorithm of Biró et al. (2018) is faster, as it makes use of an analogue of Theorem 3 for SFP.

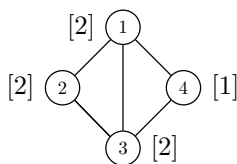


Figure 7: The graph G with $b(1) = b(2) = b(3) = 2$; $b(4) = 1$; and $w \equiv 1$ from Example 3.

	P1	P2	P3
assignment game	poly	yes ¹	poly
matching game	poly (trivial)	poly ²	poly
<i>b</i> -assignment game			
if $b \leq 2$	poly	yes	poly
if $b \leq 3$	coNPc ³	yes ⁴	poly ⁴
<i>b</i> -matching game			
if $b \leq 2$	poly ³	poly	poly
if $b \leq 3$	coNPc ³	coNP ^h ⁵	coNP ^h
partitioned matching game			
if $c \leq 2$	poly ⁵	poly ⁵	poly
if $c \leq 3$	coNPc ⁵	coNP ^h ⁵	coNP ^h

Table 2: Complexity dichotomies for the core (P1–P3), with the following short-hand notations, yes: all instances are yes-instances; poly: polynomial-time; coNPc: coNP-complete; and coNP^h: coNP-hard. The seven hardness results in the table hold even if $w \equiv 1$. The references are ¹Shapley and Shubik (1971), ²Deng et al. (1999), ³Biró et al. (2018), ⁴Sotomayor (1992), ⁵Biró et al. (2019b). The unreferenced results follow from the referenced results.

By Theorem 4 we can first solve SFP for a given instance (G, b, w) and if a stable solution is found, then we immediately obtain a core allocation for the b -matching game defined on (G, b, w) . However, Theorem 4 does not give us anything if (G, b, w) has no stable solutions. The following example given by Biró et al. (2018) shows that the corresponding b -matching game may still have a core allocation, even if $b \leq 2$. Farczadi et al. (2013) gave a different example for $b \leq 2$, where the core allocation even belongs to the pre-kernel.

Example 3. Consider the graph G with vertex capacity function b from Figure 7. We have $v(N) = 3$ in the corresponding uniform b -matching game; take $M = \{12, 23, 34\}$. Note that $(1, 1, 1, 0)$ is a core allocation. Biró et al. (2018) proved that an instance of SFP has a stable solution if and only if the maximum weight of a b -matching in the underlying graph equals the maximum weight of a half- b -matching. Now, the maximum weight of a half b -matching in G is $3\frac{1}{2}$; take $f(12) = f(23) = 1$ and $f(13) = f(14) = f(34) = \frac{1}{2}$, while the maximum weight of a b -matching is $v(N) = 3$. So no stable solution for (G, b, w) exists. \diamond

From Example 3 we find that Theorem 2 cannot be generalized to SFP. Indeed, some time later, Biró et al. (2019b) proved that P2 and thus P3 are in fact coNP-hard even for uniform b -matching games if $b \leq 3$.

All the results for P1–P3 are summarized in Table 2. This table also displays a dichotomy for partitioned matching games. This dichotomy is obtained after establishing a close

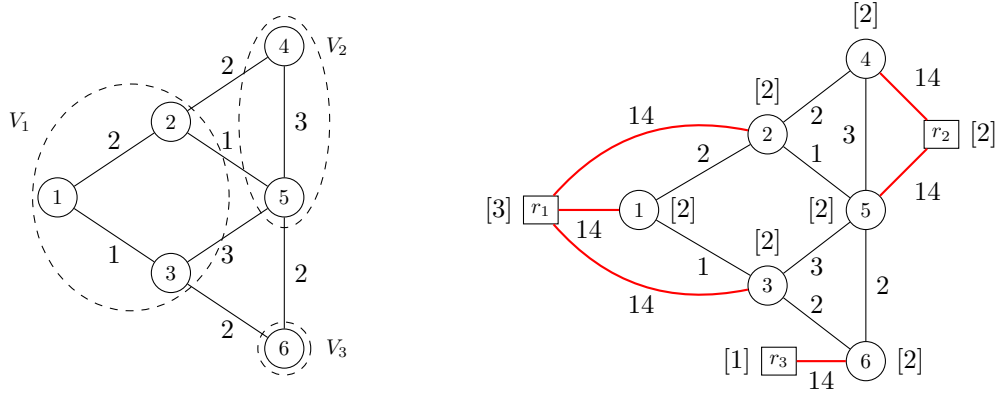


Figure 8: *Left*: the example from Figure 5: a partitioned matching game (N, v) with three players and with width $c = 3$. Recall that $v(N) = 7$. *Right*: the reduction to the b -matching game (\bar{N}, \bar{v}) . Note that $|\bar{N}| = 9$ and that for every $i \in \bar{N}$, $b_i \leq c$.

relationship between the core of partitioned matching game and the core of a b -assignment game, which we discuss below.

First, let (N, v) be a partitioned matching game of width c defined on a graph $G = (V, E)$ with a positive edge weighting w and with a partition $(V_1, \dots, V_{|N|})$ of V . We assume that $c \geq 2$, as otherwise we obtain a matching game. Construct a graph $\bar{G} = (\bar{N}, \bar{E})$ with a positive vertex capacity function b and a positive edge weighting \bar{w} as follows.

- Put the vertices of V into \bar{N} and the edges of E into \bar{E} .
- For each V_i , add a new vertex r_i to \bar{N} that is adjacent to all vertices of V_i and to no other vertices in \bar{G} .
- Let \bar{w} be the extension of w to \bar{E} by giving each new edge weight $2v(N)$.
- In \bar{N} , give every $u \in V$, capacity $b(u) = 2$ and every r_i capacity $b(r_i) = |V_i|$.

We denote the b -matching game defined on (\bar{G}, b, \bar{w}) by (\bar{N}, \bar{v}) . See Figure 8 for an example (based on Figure 5), and note that, by construction, we have $b \leq c$. Biró et al. (2019b) used the above construction to show the following theorem.⁴

Theorem 5 (Biró et al., 2019b) *P1–P3 can be reduced in polynomial time from partitioned matching games of width c to b -matching games with $b \leq c$.*

As a consequence, P1–P3 are polynomial-time solvable for partitioned matching games of width $c \leq 2$; see also Table 2.

Now, let (N, v) be a b -matching game defined on a graph $G = (V, E)$ with a positive vertex capacity function b and a positive edge weighting w . Construct a graph $\bar{G} = (\bar{V}, \bar{E})$ with a positive edge weighting \bar{w} and partition \mathcal{V} of \bar{V} by applying the aforementioned construction of Tutte (1954).

4. In the paper of Biró et al. (2019b), Theorems 5 and 6 are explicitly proven for P2. The fact that they hold for P1 and P3 as well is implicit in their proofs; see the work of Benedek et al. (2023) for the details.

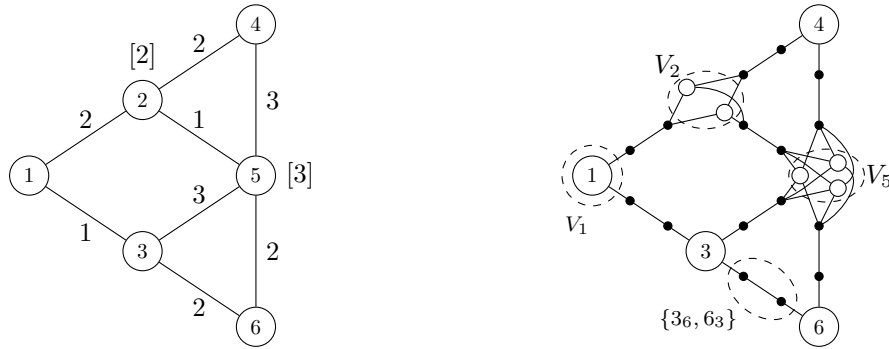


Figure 9: *Left*: the example from Figure 4: a b -matching game (N, v) with six players, where $b \equiv 1$ apart from $b(2) = 2$ and $b(5) = 3$, so $b^* = 3$. Recall that $v(N) = 10$ (take $M = \{12, 35, 45, 56\}$). *Right*: the reduction to the partitioned matching game (\bar{N}, \bar{v}) . Note that $|\bar{N}| = 14$ and $c = b^*$.

- Replace each vertex $i \in V$ with capacity $b(i)$ by a set V_i of $b(i)$ vertices $i^1, \dots, i^{b(i)}$.
- Replace each edge $ij \in E$ by a tree T_{ij} connecting the copies of i to the copies of j . The tree T_{ij} consists of a central edge with end-vertices i_j and j_i , where i_j is adjacent to all copies of i , and j_i is adjacent to all copies of j .
- Give every edge in T_{ij} weight $w(ij)$.
- Let \bar{V} consist of the sets V_i and the 2-vertex sets $\{i_j, j_i\}$.

We denote the partitioned matching game defined on (\bar{G}, w) with partition \mathcal{V} by (\bar{N}, \bar{v}) . See Figure 9 (based on Figure 4) for an example. We let b^* be the maximum $b(i)$ -value for the vertex capacity function b . Note that, by construction, we have that $c = b^*$ and that uniformity is preserved (that is, if the original b -matching game was uniform, then the constructed partitioned matching game will be uniform as well). Biró et al. (2019b) used the above construction to show the following theorem.

Theorem 6 (Biró et al., 2019b) *P1–P3 can be reduced in polynomial time from b -matching games to partitioned matching games of width $c = b^*$, preserving uniformity.*

As a consequence, P1 is coNP-complete and P2, P3 are coNP-hard, even for uniform partitioned matching games of width $c \leq 3$; see also Table 2.

5.2 Core Stabilizers

Apart from considering alternative solution concepts, other directions that try to overcome core emptiness have been explored as well. For example, we may ask for the minimum subsidy α such that the adjusted game (N, v') with $v'(N) = v(N) + \alpha$ and $v'(S) = v(S)$ for $S \subsetneq N$ has a non-empty core (Bachrach et al., 2018). Recall from Section 3 that the core of a matching game (N, v) defined on a graph $G = (V, E)$ with some positive edge weighing w is described by all imputations x with $x_i + x_j \geq w(ij)$ for every $ij \in E$ (Deng et al., 1999).

Hence, for matching pairs the minimum subsidy can be computed by solving the following linear program:

$$\begin{aligned}
 \min \alpha &= \sum_{i \in N} y_i \\
 \text{s.t. } \quad x_i + x_j + y_i + y_j &\geq w(ij) && \text{for all } ij \in E \\
 x(N) &= v(N) \\
 x_i &\geq 0 && \text{for all } i \in N \\
 y_i &\geq 0 && \text{for all } i \in N.
 \end{aligned}$$

Solving this linear program takes polynomial time by using the ellipsoid method as observed by Biró et al. (2012), who also gave a combinatorial algorithm⁵ that runs in $O(nm+n^2 \log n)$ time.

Alternatively, we may want to minimize the total blocking value. Biró et al. (2012) showed that the difference between the minimum total blocking value and the minimum subsidy can be arbitrarily large. To solve the problem of finding the minimum total blocking value, they formulated another linear program:

$$\begin{aligned}
 \text{(BV)} \quad \min \sum_{ij \in E} z_{ij} \\
 \text{s.t. } \quad x_i + x_j + z_{ij} &\geq w(ij) && \text{for all } ij \in E \\
 x(N) &= v(N) \\
 x_i &\geq 0 && \text{for all } i \in N \\
 z_{ij} &\geq 0 && \text{for all } ij \in E.
 \end{aligned}$$

Solving (BV) takes polynomial time, again using the ellipsoid method, which led Biró et al. (2012) to ask the following question:

Open Problem 1 *Is it possible to also give a polynomial-time combinatorial algorithm for computing the minimum total blocking value for a matching game with an empty core?*

On the negative side, Biró et al. (2012) proved that finding the minimum *number* of blocking pairs is NP-hard even for uniform matching games. However, Könemann et al. (2015) gave a polynomial-time algorithm for finding the minimum number of blocking pairs approximately for uniform matching games on sparse graphs up to a constant factor depending on the density of the graph.

The following example shows some shortcomings of the above concepts and illustrates that a more robust concept of a stabilizer is needed.

Example 4. Consider the uniform matching game (N, v) on the graph $K_2 + 2K_3$, the disjoint union of a graph consisting of two vertices joined by an edge and two triangles; see also Figure 10. Then, $v(N) = 3$ and (N, v) has an empty core. Now, the minimum subsidy, minimum total blocking value and minimum number of blocking pairs are all equal to 1: set $x \equiv \frac{1}{2}$ on the two triangles and $x \equiv 0$ on the edge. However, if we remove the isolated edge from G , then the core stays empty. \diamond

5. We use the notion of a combinatorial algorithm to indicate that it is an algorithm that does not rely on the ellipsoid method.

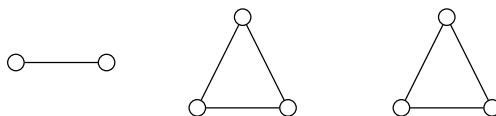


Figure 10: The graph $K_2 + 2K_3$ used in Examples 4 and 5.

We now describe a way to circumvent the issue in Example 4. Let π be some graph operation, such as the vertex deletion **vd**, edge deletion **ed**, vertex addition **va** or edge addition **ea**. Let (N, v) be a game defined on a graph G with a positive edge weighting w . Then a π -stabilizer for (N, v) is a sequence of operations of type π that transforms G into a graph G' , such that the resulting game has a non-empty core. A set of vertices or edges of G is a *blocking set* if its removal from G results in a game with a non-empty core.

Example 5. Let $\pi = \mathbf{ed}$ in Example 4. Then the sequence that deletes one edge of each of the two triangles of $K_2 + 2K_3$ is a π -stabilizer of minimum length for the uniform matching game defined on $K_2 + 2K_3$. Note that every pair of such edges is a blocking set. \diamond

Let π -STABILIZER be the problem of determining a π -stabilizer of minimum length. This type of problems falls under the classical framework of *graph modification*. For an in-depth survey that includes a wide range of aspects of stabilizers, we refer to the survey of Chandrasekaran (2017). Below we give the main complexity results including some new results.

The first result was given by Biró et al. (2014) (using the terminology from Section 4). They proved that **ed**-STABILIZER is NP-hard for matching games and asked if it is also NP-hard for uniform matching games. Afterwards, Bock et al. (2015) gave an affirmative answer to this question even for uniform matching games on *factor-critical* graphs (graphs that have a perfect matching after deleting an arbitrary vertex). In contrast, Chandrasekaran et al. (2019) proved that the variant that minimizes the total weight that might be added to the existing edges is polynomial-time solvable for uniform matching games on factor-critical graphs.

Ahmadian et al. (2018) and Ito et al. (2017) independently proved that **vd**-STABILIZER is polynomial-time solvable for uniform matching games. Ito et al. (2017) also proved that **ea**-STABILIZER and **va**-STABILIZER are polynomial-time solvable for uniform matching games. The same authors considered the variant where non-edges are given some positive weight and we must find a minimum weight set of non-edges whose addition to the graph results in a game with a non-empty core. They proved that this problem is NP-hard for arbitrary matching games. More recently, Koh and Sanità (2020) generalized the positive results of Ahmadian et al. (2018) and Ito et al. (2017) by giving the first polynomial-time algorithm for π -STABILIZER on arbitrary matching games by doing this for $\pi = \mathbf{vd}$. See Table 3 for a general overview.

Open Problem 2 *Is it possible to generalize the positive results from Table 3 to (uniform) b -matching games?*

We finish this section with the following two problems. First, the MINIMUM EDGE WEIGHT REDUCTION is to determine a blocking edge set of minimum weight. Biró et al. (2014) proved that MINIMUM EDGE WEIGHT REDUCTION is NP-hard for matching games (using

π	uniform weights	arbitrary weights
edge deletion (ed)	NP-hard ¹	NP-hard ²
vertex deletion (vd)	polynomial-time solvable ^{3,4}	polynomial-time solvable ⁵
edge addition (ea)	polynomial-time solvable ⁴	-
vertex addition (va)	polynomial-time solvable ⁴	-

Table 3: The complexity of π -STABILIZER for matching games. The cases $\pi = \mathbf{ea}$ and $\pi = \mathbf{va}$ are not well defined and have not been studied for (arbitrary) matching games. The references are ¹Bock et al. (2015), ²Biró et al. (2014), ³Ahmadian et al. (2018), ⁴Ito et al. (2017), ⁵Koh and Sanità (2020).

the terminology of Section 4) and asked the following question, which was also asked by Chandrasekaran (2017):

Open Problem 3 *Determine the computational complexity of MINIMUM EDGE WEIGHT REDUCTION for uniform matching games.*

Second, the MINIMUM VERTEX WEIGHT REDUCTION is the variant where every vertex i is given some positive weight $w(i)$ and we must find a blocking vertex set of minimum weight. Ahmadian et al. (2018) and Ito et al. (2017) both showed that this problem is NP-hard for uniform matching games even if there are only two distinct vertex weights. Ahmadian et al. (2018) also showed that the complementary variant that maximizes the weight of the remaining players is NP-hard.

6. Complexity Aspects of the Nucleolus

The sequence (LP_r) has length at most n (as each time the dimension of the feasible regions decreases). Nevertheless, computing the nucleolus takes exponential time in general due to the computation of the sets \mathcal{S}_i and the exponential number of inequalities in each linear program of the scheme of Maschler et al. (1979). As shown below, this situation changes for matching games. Note that the nucleolus exists even for b -matching games and partitioned matching games, as the set of imputations for these games is non-empty.

We first observe that the least core of matching games with a non-empty core (such as assignment games) has a compact description: for these games, we may restrict the set of constraints to only those for coalitions of size at most 2.⁶ This immediately leads to a polynomial-time algorithm for the nucleolus using the ellipsoid method and the scheme of Maschler et al. (1979). Solymosi and Raghavan (1994) gave a faster algorithm, running in $O(n^4)$ time, for computing the nucleolus of assignment games. By using again the duplication technique of Nemhauser and Trotter Jr. (1975), Biró et al. (2012) translated this algorithm into an $O(n^4)$ -time algorithm for matching games with a non-empty core.

Faigle et al. (1998) proved the following result for the nucleon of a matching game; the definition of the *nucleon* is obtained from the definition of the nucleolus by replacing (additive) excesses by multiplicative excesses $e'(S, x) = x(S)/v(S)$ in the definition of the nucleolus. In this result, the *size* of w is the number of bits in its binary representation.

6. If x is an allocation in this compact description, then $x(S) \geq \sum_{ij \in M} (x_i + x_j) \geq w(M) + \varepsilon = v(S) + \varepsilon$ for any coalition S (where M is a maximum weight matching of $G[S]$), and hence, the compact description describes the least core.

Theorem 7 (Faigle et al., 1998) *The nucleon can be computed in time polynomial in n and the size of w for matching games.*

Faigle et al. (1998) naturally asked if the same holds for the nucleolus. This led to a series of papers, which we discuss below.

Chen et al. (2012) proved that the nucleolus of a fractional matching game can be found in polynomial time. Kern and Paulusma (2003) gave a polynomial-time algorithm for uniform matching games; in fact this was shown for matching games with edge weights $w(ij)$ that can be expressed as the sum of positive vertex weights $w'(i) + w'(j)$ of the underlying graph G (Paulusma, 2001). Afterwards, Farczadi (2015) extended this result by relaxing the vertex weight condition. By combining techniques of Kern and Paulusma (2003) and Biró et al. (2012), Hardwick (2017) gave an $O(n^4)$ -time combinatorial algorithm for computing the nucleolus of a uniform matching game that does not rely on the ellipsoid method. Finally, Könemann et al. (2020) solved the open problem of Faigle et al. (1998).

Theorem 8 (Könemann et al., 2020) *The nucleolus can be computed in polynomial time for matching games.*

The proof of Theorem 8 is highly involved. For a least core allocation x , a matching M is x -tight if the incident vertices of M form a set S with $x(S) = v(S) + \varepsilon_1$ (so (x, ε) is an optimal solution for the least core, see also Section 2.1). A matching M is *universal* if it is x -tight for every least core allocation x . A least core allocation x is *universal* if the set of x -tight matchings is precisely the set of universal matchings; in particular the nucleolus is a universal allocation. To prove Theorem 8, Könemann et al. (2020) first characterized universal matchings using classical descriptions of matching polyhedra. They used this characterization to obtain a new description of the set of universal allocations. They showed that this description is of a highly symmetric nature with respect to the least core allocations. This symmetric nature enabled them to give a new description of the least core that is sufficiently compact for computing the nucleolus in polynomial time, by using the aforementioned scheme of Maschler et al. (1979) and the ellipsoid method.

As the proof of Theorem 8 relies on the ellipsoid method, Könemann et al. (2020) asked the following question.

Open Problem 4 *Is it possible to give a polynomial-time combinatorial algorithm for computing the nucleolus of a matching game?*

In contrast to the (least) core of assignment games, which is much better understood (see, for example, Solymosi & Raghavan, 1994; Vazirani, 2022b; Wako, 2006), this requires new structural insights into the least cores of matching games. Progress in this direction might also help in answering an old question of Faigle et al. (1998) who asked the same question as Könemann et al. (2020) but for the nucleon.

Open Problem 5 *Is it possible to give a polynomial-time combinatorial algorithm for computing the nucleon of a matching game?*

The aforementioned $O(n^4)$ -time algorithm of Hardwick (2017) for computing the nucleolus of uniform matching games is a promising start towards resolving Open Problems 4 and 5.

We now turn to b -matching games. Könemann et al. (2021) proved that computing the nucleolus is NP-hard for uniform b -assignment games with $b \leq 3$. The same authors proved that the nucleolus can be found in polynomial time for two subclasses of b -assignment games with $b \leq 2$, one of which is the natural case where $b \equiv 2$, and the other one is where $b \equiv 2$ on one bipartition class, but $b \equiv 2$ on only a subset of vertices of constant size in the other bipartition class of the underlying graph. These results complement a result of Bateni et al. (2010) who showed that the nucleolus of a b -assignment game can be found in polynomial-time if $b \equiv 1$ on one of the two bipartition classes of the underlying bipartite graph. Finally, Könemann and Toth (2020) gave a polynomial-time algorithm for computing the nucleolus of uniform b -matching games on graphs of bounded treewidth.

The following problems are still open.

Open Problem 6 *Determine the complexity of computing the nucleolus of*

- *b -matching games with $b \leq 2$;*
- *fractional b -assignment games;*
- *fractional b -matching games; and*
- *partitioned matching games.*

Open Problem 7 *Determine the complexity of computing the nucleon of b -matching games and partitioned matching games.*

With respect to Open Problem 7 we recall that the case $b \equiv 1$ is polynomial-time solvable by Theorem 7. We also note that the nucleon lies in the *multiplicative ε -core*, or equivalently, the *α -approximate core*, which relaxes the core constraints by replacing $x(S) \geq v(S)$ by $x(S) \geq \alpha v(S)$. Vazirani (2022a) proved that every matching game has a non-empty $\frac{2}{3}$ -approximate core and that $\alpha = \frac{2}{3}$ is best possible. This result was extended to b -matching games by Xiao et al. (2021). Polynomial-time algorithms for finding a $\frac{2}{3}$ -approximate core allocation were given in both papers, but these allocations might not be the nucleon.

7. Complexity Aspects of the Shapley Value

Only a few complexity results are known for computing the Shapley value of matching games. Aziz and de Keijzer (2014) proved the following:

Theorem 9 (Aziz & de Keijzer, 2014) *Computing the Shapley value is #P-complete for uniform matching games.*

Aziz and de Keijzer (2014) also proved polynomial-time solvability for uniform matching games on graphs G if G has maximum degree at most 2 or if G allows a modular decomposition into k cliques for some constant k . They also proved the same results for matching games on graphs G with a positive edge weighting w if G allows a modular decomposition into k independent sets (sets consisting of pairwise non-adjacent vertices); for example, if G is a complete k -partite graph. They posed the following problem.

Open Problem 8 *Determine the computational complexity of finding the Shapley value for fractional matching games.*

Moreover, Aziz and de Keijzer (2014) asked if there exists a (non-trivial) class of graphs that have at least one vertex of degree at least 3 for which the Shapley value of the corresponding matching game can be computed in polynomial time. This question was answered in the affirmative by Bousquet (2015) who proved that the Shapley value can be computed in polynomial time for uniform matching games defined on trees.

Bousquet (2015) asked the natural question whether his result for uniform matching games could be generalized from trees to graph classes of bounded treewidth. Greco et al. (2020) showed that this was indeed possible.

Finally, Kumabe and Maehara (2020a) considered b -matching games with an additional property. Namely, a game (N, v) is *convex* if for all $S, T \subseteq N$, it holds that $v(S) + v(T) \leq v(S \cup T) + v(S \cap T)$. It is well known that a convex game has a non-empty core that contains the Shapley value (Shapley, 1971). Recall from Section 3 that even uniform matching games with three players may have an empty core. Consequently, not all b -matching games are convex. Kumabe and Maehara (2020a) gave an $O(n \log n + m\alpha(n))$ -time algorithm (where α denotes the inverse Ackermann function) for computing the Shapley value for convex b -matching games.

8. Conclusions

In this survey we discussed computational complexity results on classical solution concepts for matching games, b -matching games and partitioned matching games. We also considered the fractional variant of these games whenever relevant. In this final section, we briefly mention two other concepts related to matching games. These concepts are well studied in the literature but beyond the immediate scope of our survey.

8.1 Hypergraph Matching Games

A *hypergraph* $H = (V, E)$ consists of a set of vertices V and a set of hyperedges E , which are non-empty subsets of V . If every edge in E has size 2, then H is a graph. A subset of edges $M \subseteq E$ is a *matching* in H if the edges of M are pairwise disjoint. For a subset $S \subseteq V$, we let $H[S]$ denote the hypergraph with vertex set S and edge set $\{e \in E \mid e \subseteq S\}$.

A *hypergraph matching game* on a hypergraph $H = (V, E)$ with a positive edge weighting w is the game (N, v) where $N = V$ and for $S \subseteq N$, the value $v(S)$ is the maximum weight $w(M)$ over all matchings M of $H[S]$. Hypergraph matching games were introduced by Kaneko and Wooders (1982) under the name of *partitioning games* and are also known as *packing games* (Deng et al., 1999) or *synergy coalition group representation* (Conitzer & Sandholm, 2006).

Conitzer and Sandholm (2006) proved that deciding core non-emptiness is NP-hard for hypergraph matching games. As even computing a value $v(S)$ is NP-hard for hypergraph matching games, Kumabe and Maehara (2020b) considered convex hypergraph matching games. They first proved that convex hypergraph matching games can be recognized in polynomial time. They then proved that the Shapley value of convex hypergraph matching games can be found in polynomial time.

8.2 NTU Matching Games

Cooperative games (N, v) are also known as *transferable utility games* (TU games), as it is possible to distribute the value $v(S)$ among the players of a coalition S in any possible way. *Non-transferable utility games* (NTU games) can be seen as a generalization of TU games and play a role in economic settings where goods are indivisible. That is, in an NTU game (N, v) , the value function v now also associates each non-empty set $S \subseteq N$ with a non-empty set $X(S)$ that consists of vectors $x \in \mathbb{R}^S$. Note that we obtain the corresponding TU game if for each $S \subseteq N$ we have that $X(S)$ consists of all possible positive vectors x such that $\sum_{i \in S} x_i = v(S)$. We refer to the work of Biró and Fleiner (2016) for a survey on results on NTU matching games and some natural open problems for these games and to a more recent paper of Király and Mészáros-Karkus (2019) for complexity results for NTU partitioned matching games, which they call international matching games.

Furthermore, there is a large body of related work on core stabilizers in this setting; for example on the problem of minimizing the number of blocking pairs (Abraham et al., 2005; Biró et al., 2010), the problem of maximizing the number of vertices such that these pairs are stable among themselves (in our terminology, the `vd-STABILIZER` problem) (Tan, 1990), and on the topic of manipulative attacks (Boehmer et al., 2021; Chen et al., 2021; Eiben et al., 2023).

Acknowledgments

Benedek acknowledges the support of the National Research, Development and Innovation Office of Hungary (OTKA Grant No. K138945). Biró acknowledges the support of the Hungarian Scientific Research Fund (OTKA, Grant No. K143858) and the Hungarian Academy of Sciences (Momentum Grant No. LP2021-2). Paulusma acknowledges the support of the Leverhulme Trust (Grant RF-2022-607). All authors thank three anonymous reviewers for their helpful comments on our paper.

References

- Abraham, D. J., Biró, P., & Manlove, D. F. (2005). “Almost stable” matchings in the roommates problem. *Proceedings WAOA 2005, Lecture Notes in Computer Science, 3879*, 1–14.
- Abraham, D. J., Blum, A., & Sandholm, T. (2007). Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. *Proceedings EC 2007*, 295–304.
- Ahmadian, S., Hosseinzadeh, H., & Sanità, L. (2018). Stabilizing network bargaining games by blocking players. *Mathematical Programming, 172*, 249–275.
- Aziz, H., & de Keijzer, B. (2014). Shapley meets Shapley. *Proceedings STACS 2014, LIPIcs, 25*, 99–111.
- Bachrach, Y., Elkind, E., Malizia, E., Meir, R., Pasechnik, D. V., Rosenschein, J. S., Rothe, J., & Zuckerman, M. (2018). Bounds on the cost of stabilizing a cooperative game. *Journal of Artificial Intelligence Research, 63*, 987–1023.

- Balinski, M. L. (1965). Integer programming: Methods, uses, computation. *Management Science*, 12, 253–313.
- Bateni, M., Hajiaghayi, M., Immorlica, N., & Mahini, H. (2010). The cooperative game theory foundations of network bargaining games. *Proceedings ICALP 2010, Lecture Notes in Computer Science*, 6198, 67–78.
- Benedek, M., Biró, P., Kern, W., Pálvölgyi, D., & Paulusma, D. (2023). Partitioned matching games for international kidney exchange. *CoRR*, *abs/2301.13181*.
- Benedek, M., Biró, P., Kern, W., & Paulusma, D. (2022). Computing balanced solutions for large international kidney exchange schemes. *Proceedings AAMAS 2022*, 82–90.
- Benedek, M., Fliege, J., & Nguyen, T. (2021). Finding and verifying the nucleolus of cooperative games. *Mathematical Programming*, 190, 135–170.
- Biró, P., Bomhoff, M., Golovach, P. A., Kern, W., & Paulusma, D. (2014). Solutions for the stable roommates problem with payments. *Theoretical Computer Science*, 540, 53–61.
- Biró, P., & Fleiner, T. (2016). Fractional solutions for capacitated NTU-games, with applications to stable matchings. *Discrete Optimization*, 22, 241–254.
- Biró, P., Gyetvai, M., Klimentova, X., Pedroso, J. P., Pettersson, W., & Viana, A. (2020). Compensation scheme with Shapley value for multi-country kidney exchange programmes. *Proceedings ECMS 2020*, 129–136.
- Biró, P., Haase-Kromwijk, B., Andersson, T., & et al. (2019a). Building kidney exchange programmes in Europe – an overview of exchange practice and activities. *Transplantation*, 103, 1514–1522.
- Biró, P., Kern, W., Pálvölgyi, D., & Paulusma, D. (2019b). Generalized matching games for international kidney exchange. *Proceedings AAMAS 2019*, 413–421.
- Biró, P., Kern, W., & Paulusma, D. (2012). Computing solutions for matching games. *International Journal of Game Theory*, 41, 75–90.
- Biró, P., Kern, W., Paulusma, D., & Wojuteczky, P. (2018). The stable fixtures problem with payments. *Games and Economic Behavior*, 108, 245–268.
- Biró, P., Manlove, D. F., & Mittal, S. (2010). Size versus stability in the marriage problem. *Theoretical Computer Science*, 411(16-18), 1828–1841.
- Bock, A., Chandrasekaran, K., Könemann, J., Peis, B., & Sanità, L. (2015). Finding small stabilizers for unstable graphs. *Mathematical Programming*, 154, 173–196.
- Boehmer, N., Bredereck, R., Heeger, K., & Niedermeier, R. (2021). Bribery and control in stable marriage. *Journal of Artificial Intelligence Research*, 71, 993–1048.
- Bondy, J. A., & Murty, U. S. R. (2008). *Graph Theory*, Vol. 244 of *Graduate Texts in Mathematics*. Springer.
- Bousquet, N. (2015). The Shapley value of matching games on trees. *Manuscript*.
- Chalkiadakis, G., Elkind, E., & Wooldridge, M. (2011). *Computational Aspects of Cooperative Game Theory*. Morgan and Claypool Publishers.

- Chandrasekaran, K. (2017). Graph stabilization: A survey. In *Combinatorial Optimization and Graph Algorithms*. Springer.
- Chandrasekaran, K., Gottschalk, C., Könemann, J., Peis, B., Schmand, D., & Wierz, A. (2019). Additive stabilizers for unstable graphs. *Discrete Optimization*, 31, 56–78.
- Chen, J., Skowron, P., & Sorge, M. (2021). Matchings under preferences: Strength of stability and tradeoffs. *ACM Transactions on Economics and Computation*, 9, 1–55.
- Chen, N., Lu, P., & Zhang, H. (2012). Computing the nucleolus of matching, cover and clique games. *Proceedings AAAI 2012*, 1319–1325.
- Conitzer, V., & Sandholm, T. (2006). Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170, 607–619.
- Deng, X., Ibaraki, T., & Nagamochi, H. (1999). Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24, 751–766.
- Diestel, R. (2012). *Graph Theory, 4th Edition*, Vol. 173 of *Graduate Texts in Mathematics*. Springer.
- Edmonds, J. (1965). Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards Section B*, 69B, 125–130.
- Eiben, E., Gutin, G., Neary, P. R., Rambaud, C., Wahlström, M., & Yeo, A. (2023). Preference swaps for the stable matching problem. *Theoretical Computer Science*, 940, 222–230.
- Eriksson, K., & Karlander, J. (2001). Stable outcomes of the roommate game with transferable utility. *International Journal of Game Theory*, 29, 555–569.
- Faigle, U., Kern, W., Fekete, S. P., & Hochstättler, W. (1998). The nucleon of cooperative games and an algorithm for matching games. *Mathematical Programming*, 83, 195–211.
- Farczadi, L. (2015). *Matchings and Games on Networks*. Ph.D. thesis, University of Waterloo, Ontario, Canada.
- Farczadi, L., Georgiou, K., & Könemann, J. (2013). Network bargaining with general capacities. *Proceedings ESA 2013, Lecture Notes in Computer Science*, 8125, 433–444.
- Gillies, D. B. (1959). Solutions to general non-zero-sum games. *Annals of Mathematics Studies*, 40, 47–85.
- Gourvès, L., Monnot, J., & Pascual, F. (2013). Cooperation in multiorganization matching. *Algorithmic Operations Research*, 7, 111–124.
- Greco, G., Lupia, F., & Scarcello, F. (2020). Coalitional games induced by matching problems: Complexity and islands of tractability for the Shapley value. *Artificial Intelligence*, 278.
- Grötschel, M., Lovász, L., & Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1, 169–197.
- Hardwick, J. (2017). *Graphical Algorithms for Finding the Nucleolus of Binary-Valued Matching Games*. Ph.D. thesis, University of Illinois, Chicago, Illinois, USA.

- Ito, T., Kakimura, N., Kamiyama, N., Kobayashi, Y., & Okamoto, Y. (2017). Efficient stabilization of cooperative matching games. *Theoretical Computer Science*, 677, 69–82.
- Kaneko, M., & Wooders, M. H. (1982). Cores of partitioning games. *Mathematical Social Sciences*, 3, 313–327.
- Kern, W., & Paulusma, D. (2003). Matching games: The least core and the nucleolus. *Mathematics of Operations Research*, 28, 294–308.
- Khachiyan, L. G. (1979). A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20, 191–194.
- Király, T., & Mészáros-Karkus, Z. (2019). Complexity of the NTU international matching game. *EGRES Technical Report*, 2019-12.
- Kleinberg, J. M., & Tardos, É. (2008). Balanced outcomes in social exchange networks. *Proceedings STOC 2008*, 295–304.
- Klimentova, X., Viana, A., Pedroso, J. a. P., & Santos, N. (2021). Fairness models for multi-agent kidney exchange programmes. *Omega*, 102, 102333.
- Koh, Z. K., & Sanità, L. (2020). Stabilizing weighted graphs. *Mathematics of Operations Research*, 45, 1318–1341.
- Könemann, J., Larson, K., & Steiner, D. (2015). Network bargaining: Using approximate blocking sets to stabilize unstable instances. *Theory of Computing Systems*, 57, 655–672.
- Könemann, J., Pashkovich, K., & Toth, J. (2020). Computing the nucleolus of weighted cooperative matching games in polynomial time. *Mathematical Programming*, 183, 555–581.
- Könemann, J., & Toth, J. (2020). A general framework for computing the nucleolus via dynamic programming. *Proceedings SAGT 2020, Lecture Notes in Computer Science*, 12283, 307–321.
- Könemann, J., Toth, J., & Zhou, F. (2021). On the complexity of nucleolus computation for bipartite b -matching games. *Proceedings SAGT 2021, Lecture Notes in Computer Science*, 12885, 171–185.
- Koopmans, T. C., & Beckmann, M. (1957). Assignment problems and the location of economic activities. *Econometrica*, 25, 53–76.
- Kumabe, S., & Maehara, T. (2020a). Convexity of b -matching games. *Proceedings IJCAI 2020*, 261–267.
- Kumabe, S., & Maehara, T. (2020b). Convexity of hypergraph matching game. *Proceedings AAMAS 2020*, 663–671.
- Maschler, M., Peleg, B., & Shapley, L. S. (1971). The kernel and bargaining set for convex games. *International Journal of Game Theory*, 1, 73–93.
- Maschler, M., Peleg, B., & Shapley, L. S. (1979). Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research*, 4, 303–338.

- Nemhauser, G. L., & Trotter Jr., L. E. (1975). Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8, 232–248.
- Owen, G. (2013). *Game Theory*. Emerald Group Publishing.
- Paulusma, D. (2001). *Complexity Aspects of Cooperative Games*. Ph.D. thesis, University of Twente, Enschede, The Netherlands.
- Schmeidler, D. (1969). The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17, 1163–1170.
- Shapley, L. S. (1953). A value for n -person games. *Annals of Mathematical Studies*, 28, 307–317.
- Shapley, L. S. (1971). Cores of convex games. *International Journal of Game Theory*, 1, 11–26.
- Shapley, L. S., & Shubik, M. (1971). The assignment game I: The core. *International Journal of Game Theory*, 1, 111–130.
- Solymosi, T., & Raghavan, T. E. S. (1994). An algorithm for finding the nucleolus of assignment games. *International Journal of Game Theory*, 23, 119–143.
- Sotomayor, M. (1992). The multiple partners game. In *Equilibrium and dynamics: essays in honor of David Gale*. Macmillan Press Ltd, New York.
- Tan, J. J. (1990). A maximum stable matching for the roommates problem. *BIT Numerical Mathematics*, 30, 631–640.
- Tutte, W. T. (1954). A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics*, 6, 347–352.
- Vazirani, V. V. (2022a). The general graph matching game: Approximate core. *Games and Economic Behavior*, 132, 478–486.
- Vazirani, V. V. (2022b). Insights into the core of the assignment game via complementarity. *CoRR*, [abs/2202.00619](https://arxiv.org/abs/2202.00619).
- Wako, J. (2006). Another proof that assignment games have singleton cores only if multiple optimal matchings exist. *Economic Theory*, 29, 213–217.
- Xiao, H., Lu, T., & Fang, Q. (2021). Approximate core allocations for multiple partners matching games. *CoRR*, [abs/2107.01442](https://arxiv.org/abs/2107.01442).