On Solution Concepts for Matching Games

Péter Biró[1,⋆], Walter Kern[2], and Daniël Paulusma[3,⋆⋆]

[1] Department of Computing Science, Sir Alwyn Williams Building, University of Glasgow, Glasgow G12 8QQ, Scotland
pbiro@dcs.gla.ac.uk

[2] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, NL-7500 AE Enschede
w.kern@math.utwente.nl

[3] Department of Computer Science, University of Durham, Science Laboratories, South Road, Durham DH1 3EY, England
daniel.paulusma@durham.ac.uk

Abstract. A matching game is a cooperative game (N, v) defined on a graph $G = (N, E)$ with an edge weighting $w : E \to \mathbb{R}_+$. The player set is N and the value of a coalition $S \subseteq N$ is defined as the maximum weight of a matching in the subgraph induced by S. First we present an $O(nm + n^2 \log n)$ algorithm that tests if the core of a matching game defined on a weighted graph with n vertices and m edges is nonempty and that computes a core allocation if the core is nonempty. This improves previous work based on the ellipsoid method. Second we show that the nucleolus of an n-player matching game with nonempty core can be computed in $O(n^4)$ time. This generalizes the corresponding result of Solymosi and Raghavan for assignment games. Third we show that determining an imputation with minimum number of blocking pairs is an NP-hard problem, even for matching games with unit edge weights.

1 Introduction

Consider a group N of tennis players that will participate in a doubles tennis tournament. Suppose that each pair of players can estimate the expected prize money they could win together if they form a pair in the tournament. Also suppose that each player is able to negotiate his share of the prize money with his chosen partner, and that each player wants to maximize his own prize money. Can the players be matched together such that no two players have an incentive to leave the matching in order to form a pair together? This is the example Eriksson and Karlander [7] used to illustrate the stable roommates problem with payments. In this paper we consider the situation in which groups of *possibly more than two* players in a doubles tennis tournament can distribute their total prize money among each other. For instance, suppose that three players always form pairs among themselves. Then they may decide that only two of them will

⋆ Supported by EPSRC grant EP/E011993/1 and by OTKA grant K69027.
⋆⋆ Supported by EPSRC grants EP/D053633/1 and EP/G043434/1.

J. Kratochvil et al. (Eds.): TAMC 2010, LNCS 6108, pp. 117–127, 2010.

form a pair in this tournament, but that the third player will be compensated for his loss of income. Now the question is whether the players can be matched together such that no group of players will be better off when leaving the matching. This scenario is an example of a matching game. Matching games are well studied within the area of Cooperative Game Theory. In order to explain these games and how they are related to the first problem setting, we first state the necessary terminology and formal definitions.

A *cooperative game* (N, v) is given by a set N of n *players* and a *value function* $v : 2^N \to \mathbb{R}$ with $v(\emptyset) = 0$. A *coalition* is any subset $S \subseteq N$. We refer to $v(S)$ as the *value* of coalition S, i.e., the maximal profit or the minimal costs that the players in S achieve by cooperating with each other. In the first case we also speak of (N, v) as a *profit game* and in the second case we also say that (N, v) is a *cost game*. The v-values of many cooperative games are derived from solving an underlying discrete optimization problem (cf. Bilbao [3]). It is often assumed that the *grand coalition* N is formed, because in many games the total profit or costs are optimized if all players work together. The central problem is then how to allocate the *total value* $v(N)$ to the individual players in N. An *allocation* is a vector $x \in \mathbb{R}^N$ with $x(N) = v(N)$, where we adopt the standard notation $x(S) = \sum_{i \in S} x_i$ for $S \subseteq N$. A *solution concept* $\mathcal{S}$ for a class of cooperative games Γ is a function that maps each game $(N, v) \in \Gamma$ to a set $\mathcal{S}(N, v)$ of allocations for (N, v). These allocations are called $\mathcal{S}$-*allocations*.

The choice for a specific solution concept $\mathcal{S}$ not only depends on the notion of "fairness" specified within the decision model but also on certain computational aspects, such as the computational complexity of testing nonemptiness of $\mathcal{S}(N, v)$, or computing an allocation in $\mathcal{S}(N, v)$. Here we take the size of the underlying discrete structure as the natural input size, instead of the 2^n v-values themselves.

We will now define some well-known solution concepts in terms of profit games; see Owen [18] for a survey. First, the *core* of a game (N, v) consists of all allocations x with $x(S) \geq v(S)$ for all $S \in 2^N \backslash \{\emptyset, N\}$. Core allocations are fair in the sense that every nonempty coalition S receives at least its value $v(S)$. Therefore, players in a coalition S do not have any incentive to leave the grand coalition (recall the doubles tennis tournament). However, for many games, the core might be empty. Therefore, other solution concepts have been designed, such as the nucleolus and the nucleon, both defined below.

Let (N, v) be a cooperative game. The *excess* of a nonempty coalition $S \subsetneq N$ regarding an allocation $x \in \mathbb{R}^N$ expresses the satisfaction of S with x and is defined as $e(S, x) := x(S) - v(S)$. We order all excesses $e(S, x)$ into a nondecreasing sequence to obtain the *excess vector* $\theta(x) \in \mathbb{R}^{2^n - 2}$. The *nucleolus* of (N, v) is then defined as the set of allocations that lexicographically maximize $\theta(x)$ over all *imputations*, i.e., over all allocations $x \in \mathbb{R}^N$ with $x_i \geq v(\{i\})$ for all $i \in N$. The nucleolus is not defined if the set of imputations is empty. Otherwise, Schmeidler [20] showed it consists of exactly one allocation. Note that, by definition, the nucleolus lies in the core if the core is nonempty. The standard procedure for computing the nucleolus proceeds by solving up to n

linear programs, which have exponential size in general. We refer to Maschler, Peleg and Shapley [14] for more details. Taking multiplicative excesses $e'(S,x) = x(S)/v(S)$ in the definition of the nucleolus leads to the *nucleon* of a game.

Matching games. In a *matching game* (N,v), the underlying discrete structure is a finite undirected graph $G=(N,E)$ that has no loops and no multiple edges and that is *weighted*, i.e., on which an edge weighting $w: E \to \mathbb{R}_+$ has been defined. The players are represented by the vertices of G, and for each coalition S we define $v(S) = w(M) = \sum_{e \in M} w(e)$, where M is a maximum weight matching in the subgraph of G induced by S. If $w \equiv 1$, then $v(S)$ is equal to the size of a maximum matching and we call (N,E) a *simple* matching game. If there exists a weighting $w^*: N \to \mathbb{R}_+$ such that $w(uv) = w^*(u) + w^*(v)$ for all $uv \in E$, then we obtain a *node matching game*. Note that every simple matching game is a node matching game by choosing $w^* \equiv \frac{1}{2}$. Matching games defined on a bipartite graph are called *assignment games*.

The core of a matching game can be empty. In order to see this, consider a simple matching game (N,v) on a triangle with players a,b,c. An allocation x in the core must satisfy $x_a+x_b \geq 1$, $x_a+x_c \geq 1$, and $x_b+x_c \geq 1$, and consequently, $x(N) = x_a+x_b+x_c \geq \frac{3}{2}$. However, this is not possible due to $x(N) = v(N) = 1$. Shapley and Shubik [21] show that the core of an assignment game is always nonempty.

We will now discuss complexity aspects of solution concepts for matching games. First let us recall the following result of Gabow [9].

Theorem 1 ([9]). *A maximum weight matching of a weighted graph on n vertices and m edges can be computed in $O(nm + n^2 \log n)$ time.*

The following observation can be found in several papers, see e.g. [5,7,19]. Here, a *cover* of a graph $G=(N,E)$ with edge weighting w is a vertex mapping $c: N \to \mathbb{R}_+$ such that $c(u)+c(v) \geq w(uv)$ for each edge $uv \in E$. The *weight* of c is defined as $c(N) = \sum_{u \in N} c(u)$.

Observation 1. *Let (N,v) be a matching game on a weighted graph $G=(N,E)$. Then $x \in \mathbb{R}^N$ is in the core of (N,v) if and only if x is a cover of G with weight $v(N)$.*

Observation 1 and Theorem 1 imply that testing core nonemptiness can be done in polynomial time for matching games by using the ellipsoid method for solving linear programs [12]. Deng, Ibaraki and Nagamochi [5] characterize when the core of a simple matching game is nonempty. In this way they can compute a core allocation of a simple matching game in polynomial time, without having to rely on the ellipsoid method. Eriksson and Karlander [7] characterize the extreme points of the core of a matching game. Observation 1 implies that the size of the linear programs involved in the procedure of Maschler, Peleg and Shapley [14] is polynomial in the case that the matching game has a nonempty core [19]. Hence the nucleolus of such matching games can be computed in polynomial time by using the ellipsoid method at most n times. Solymosi and Raghavan [23] compute the nucleolus of an assignment game without making use of the ellipsoid method.

Matsui [15] present a faster algorithm for computing the nucleolus of assignment games defined on bipartite graphs that are unbalanced.

Theorem 2 ([23]). *The nucleolus of an n-player assignment game can be computed in $O(n^4)$ time.*

It is known [11] that the nucleolus of a simple matching game can be computed in polynomial time by using the standard procedure of Maschler, Peleg and Shapley [14], after reducing the size of the involved linear programs to be polynomial. This result is extended to node matching games [19]. For matching games, Faigle et al. [8] show how to compute an allocation in the nucleon in polynomial time. Determining the computational complexity of finding the nucleolus for general matching games is still open.

Connection to the stable roommates problem. We refer to a survey [4] for more on this problem. Here, we only define the variant with payments. Let $G = (N, E)$ be a graph with edge weighting w. Let $B(x) = \{(u, v) \mid x_u + x_v < w(uv)\}$ denote the set of *blocking pairs* of a vector $x \in \mathbb{R}^N$. A vector $p \in \mathbb{R}^N$ with $p_u \geq 0$ for all $u \in N$ is said to be a *payoff* with respect to a matching M in G if $p_u + p_v = w(uv)$ for all $uv \in M$, and $p_u = 0$ for each u that is not incident to an edge in M. The problem Stable Roommates with Payments tests if a weighted graph allows a *stable solution*, i.e., a pair (M, p), where p is a payoff with respect to matching M such that $B(p) = \emptyset$. We also call such a pair *stable.* This problem is polynomially solvable due to following well-known observation (cf. Eriksson and Karlander [7]).

Observation 2. *An allocation x of the matching game defined on a weighted graph G is a core allocation if and only if there exists a matching M in G such that (M, x) is stable.*

Any core allocation of a matching game is an imputation with an empty set of blocking pairs. If the core is empty, then we can try to minimize the number of blocking pairs. This leads to the following decision problem, which is trivially solvable for assignment games, because these games have a nonempty core [21].

Blocking Pairs
Instance: a matching game (N, v) and an integer $k \geq 0$.
Question: does (N, v) allow an imputation x with $|B(x)| \leq k$?

Our results and paper organization. In Section 2 we present an $O(nm + n^2 \log n)$ time algorithm that tests if the core of a matching game on a weighted n-vertex graph with m edges is nonempty and that computes a core allocation if it exists. Like the algorithm of Deng, Ibaraki and Nagamochi [5] for simple matching games, our algorithm does not rely on the ellipsoid method. By Observation 2 we can use this algorithm to find stable solutions for instances of Stable Roommates with Payments. In Section 3 we generalize Theorem 2 by showing that the nucleolus of an n-player matching game with nonempty core can be computed in $O(n^4)$ time. Klaus and Nichifor [13] investigate the relation of the core with other solution concepts for matching games and express the need of a comparison of matching games with nonempty core to assignment games. As the results in Sections 2 and 3 are based on a duplication technique yielding bipartite graphs, our paper gives such a comparison. In Section 4 we show that the Blocking Pairs problem is NP-complete, even for simple matching games. We note that, in the context of stable matchings without payments, minimizing the number of blocking pairs is NP-hard as well [1]. This problem setting is quite different from ours, and we cannot use the proof of this result for our means.

2 The Core of a Matching Game

As mentioned in the previous section, Shapley and Shubik [21] showed that every assignment game has a nonempty core. However, they did not analyze the computational complexity of finding a core allocation. In this section we consider this question for matching games. First we introduce some terminology.

Let $G = (N, E)$ be a graph with edge weighting $w : E \rightarrow \mathbb{R}_+$. We write $v \in e$ if v is an end vertex of edge e. A *fractional matching* is an edge mapping $f : E \rightarrow \mathbb{R}_+$ such that $\sum_{e:v\in e} f(e) \leq 1$ for each $v \in N$. The weight of a fractional matching f is defined as $w(f) = \sum_{e\in E} w(e)f(e)$. We call f a *matching* if $f(e) \in \{0, 1\}$ for all $e \in E$, and we call f a *half-matching* if $f(e) \in \{0, \frac{1}{2}, 1\}$ for all $e \in E$.

We state two useful lemmas. The first lemma is a direct application of the Duality Theorem (cf. Schrijver [22]). The second lemma is a special case of Theorem 1 from Deng, Ibaraki and Nagamochi [5].

Lemma 3. *Let $G = (N, E)$ be a graph with edge weighting w. Let f be a fractional matching of G and let c be a cover of G. Then $w(f) \leq c(N)$, with equality if and only if f has maximum weight and c has minimum weight.*

Lemma 4 ([5]). *Let (N, v) be a matching game on a weighted graph $G = (N, E)$. Then the core of (N, v) is nonempty if and only if the maximum weight of a matching in G equals the maximum weight of a fractional matching in G.*

We also need the following theorem, which is a straightforward consequence of a result by Balinski [2].

Theorem 3 ([2]). *Let G be a weighted graph. Then the maximum weight of a half-matching of G is equal to the minimum weight of a cover of G.*

Lemma 3 and 4 together with Theorem 3 characterize the core of a matching game.

Proposition 1. *Let (N, v) be a matching game on a weighted graph $G = (N, E)$. The core of (N, v) is nonempty if and only if the maximum weight of a matching in G is equal to the maximum weight of a half-matching in G.*

Eriksson and Karlander [7] consider the problem Stable Roommates with Payments and characterize stable solutions in terms of forbidden minors. We can translate their characterization as follows. A weighted G allows a stable pair (M, x) if and only if the weight of M cannot be improved via any half-integer alternating path. Such an improvement is possible if and only if the maximum weight of a half-matching in G is greater than the maximum weight of a matching in G. Hence their result [7] follows directly from Observation 2 and Proposition 1.

We will use Proposition 1 in the proof of Theorem 5, in which we present a polynomial-time algorithm for finding a core allocation of a matching game, if such an allocation exists. We also need the following result by Egerváry [6], which holds for bipartite graphs.

Theorem 4 ([6]). *Let G be a weighted bipartite graph. Then the maximum weight of a matching in G is equal to the minimum weight of a cover of G.*

Theorem 4 can also be used in an alternative proof of Theorem 3 based on a duplication technique, as described by Nemhauser and Trotter [17]. Since we need this technique in our proof of Theorem 5, we discuss it below.

Let (N, v) be a matching game defined on graph $G = (N, E)$ with edge weighting w. We replace each vertex u by two copies u', u'' and each edge $e = uv$ by two edges $e' = u'v''$ and $e'' = u''v'$. We define edge weights $w^d(e') = w^d(e'') = \frac{1}{2}w(e)$ for each $e \in E$. This yields a weighted bipartite graph $G^d = (N^d, E^d)$ in $O(n^2)$ time. We call G^d the *duplicate* of G. We compute a maximum weight matching f^d of G^d in $O(nm + n^2 \log n)$ time due to Theorem 1. Given f^d, we compute a minimum weight cover c^d of G^d in the same time (cf. Theorem 17.6 from Schrijver [22]). We compute the half-matching f in G defined by $f(e) := \frac{f^d(e')+f^d(e'')}{2}$ for each $e \in E$ in $O(n^2)$ time and note that

$$w(f) = \sum_{e \in E} w(e)f(e) = \sum_{e \in E} \left(w^d(e')f^d(e') + w^d(e'')f^d(e'')\right) = w^d(f^d).$$

We define $c : N \to \mathbb{R}_+$ in $O(n)$ time by $c(u) := c^d(u') + c^d(u'')$ for all $u \in N$ and deduce that $c(u)+c(v) = c^d(u')+c^d(u'')+c^d(v')+c^d(v'') \geq w^d(u'v'')+w^d(u''v') = \frac{1}{2}w(uv)+\frac{1}{2}w(uv) = w(uv)$. This means that c is a cover of G with $c(N) = c^d(N^d)$, and by Theorem 4, we deduce that

$$w(f) = w^d(f^d) = c^d(N^d) = c(N). \tag{1}$$

We observe that f is a maximum weight half-matching due to Lemma 3. Hence equation (1) implies Theorem 3. We compute a maximum weight matching f^* of G in $O(nm + n^2 \log n)$ time due to Theorem 1. Then, by Proposition 1, we just need to check whether $w(f^*) = w(f)$ in order to find out if the core of (N, v) is nonempty. Suppose the core of (N, v) is nonempty. Since $w(f) = c(N)$ and $w(f^*) = v(N)$, we obtain

$$c(N) = v(N). \tag{2}$$

Hence, c is a core member due to Observation 1, and we get the following result.

Theorem 5. *There exists an $O(nm + n^2 \log n)$ time algorithm that tests if the core of a matching game on a graph with n vertices and m edges is nonempty and that computes a core allocation in the case that the core is nonempty.*

For simple matching games, we improve the running time of the algorithm in Theorem 5 to $O(\sqrt{n}m)$ by using the $O(\sqrt{n}m)$ time algorithm of Micali and Vazirani [16] for computing a maximum matching in a graph with n vertices and m edges.

3 The Nucleolus of a Matching Game with Nonempty Core

We start with some extra terminology. For a matching game (N, v) defined on a weighted graph $G = (N, E)$ we define its *duplicate* as the assignment game (N^d, v^d) defined on G^d with edge weights w^d. The *duplicate* of a vector $x \in \mathbb{R}^N$ is the *symmetric* vector x^d given by $x^d_{u'} = x^d_{u''} = \frac{1}{2}x_u$ for all $u \in N$.

Lemma 5. *Let (N, v) be a matching game with nonempty core. Then the nucleolus of (N^d, v^d) is the duplicate of the nucleolus of (N, v).*

Proof. Let η^* be the nucleolus of (N^d, v^d). Define $\overline{\eta}$ by $\overline{\eta}_{u'} = \eta^*_{u''}$ and $\overline{\eta}_{u''} = \eta^*_{u'}$ for all $u \in N$. Then $\theta(\eta^*) = \theta(\overline{\eta})$. Because η^* is unique as shown by Schmeidler [20], we find that η^* must be symmetric. Let η be such that $\eta^d = \eta^*$. We observe that $\theta(x) \preceq \theta(y)$ if and only if $\theta(x^d) \preceq \theta(y^d)$ for two imputations x and y of (N, v). Hence, we are left to prove that η is an imputation.

Note that $\eta_u \geq 0$ for all $u \in N$. We now show $\eta(N) = v(N)$. By definition, η^d is in the core of (N^d, v^d). Observation 1 implies that η^d is a cover of G^d with weight $\eta^d(N^d) = v^d(N^d) = w^d(f^d)$, where f^d is a maximum weight matching of G^d. By Lemma 3, η^d is a minimum weight cover of (N^d, v^d). Then, by equation (2), we derive $\eta(N) = v(N)$. Hence, η is indeed an imputation of (N, v). $\square$

Theorem 6. *The nucleolus of an n-player matching game with nonempty core can be computed in $O(n^4)$ time.*

Proof. Let (N, v) be an n-player matching game with nonempty core that is defined on a graph G with edge weighting w. We create G^d and w^d in $O(n^2)$ time. Note that $|N^d| = 2n$. By Theorem 2 we compute the nucleolus η^d of (N^d, v^d) in $O(n^4)$ time. From η^d we construct η in $O(n^2)$ time. By Lemma 5 we find that η is the nucleolus of (N, v). This finishes the proof of Theorem 6. $\square$

4 Blocking Pairs in a Matching Game

Fixing parameter k makes the Blocking Pairs problem polynomially solvable. This can be seen as follows. We choose a set B of k blocking pairs. Then we use the ellipsoid method to check in polynomial time if there exists an imputation x with $x_u + x_v \geq w(uv)$ for all pairs $uv \notin B$. Because k is fixed, the total number of choices is bounded by a polynomial in n. What happens when k is part of the input? Before we present our main result, we start with a useful lemma.

Lemma 6. *Let K be a complete graph with vertex set $\{1, \ldots, \ell\}$ for some odd integer ℓ, and let $x \in \mathbb{R}^K_+$. If $x(K) < \frac{\ell}{2}$ then $|B(x)| \geq \frac{\ell-1}{2}$ holds.*

Proof. Write $\ell = 2q + 1$ and use induction on q. If $q = 1$ the statement holds. Suppose $q \geq 2$. We assume without loss of generality that $x_1 \leq x_2 \leq \cdots \leq x_{2q+1}$ holds. Since $x(K) < \frac{\ell}{2}$, we have $x_1 < \frac{1}{2}$. If $x_1 + x_{2q+1} < 1$ then $x_1 + x_i < 1$ for $2 \leq i \leq 2q+1$. Hence, we have at least $2q$ blocking pairs. Suppose $x_1 + x_{2q+1} \geq 1$. Then $x_2 + \cdots + x_{2q} < \frac{2q-1}{2}$. By induction this yields $q - 1$ blocking pairs. Note that $x_2 < \frac{1}{2}$ holds. Hence $x_1 + x_2 < 1$, and we have at least q blocking pairs. □

Theorem 7. Blocking Pairs *is* NP*-complete, even for simple matching games.*

Proof. Clearly, this problem is in NP. To prove NP-completeness, we reduce from Independent Set, which tests if a graph G contains an *independent* set of size at least k, i.e., a set U (with $|U| \geq k$) such that there is no edge in G between any two vertices of U. Garey, Johnson and Stockmeyer [10] show that the Independent Set problem is already NP-complete for the class of *3-regular* connected graphs, i.e., graphs, in which all vertices are of degree three. So, we may assume that G is 3-regular and connected. Let $n = |V|$. We may without loss of generality assume that $k \leq \frac{1}{2}n$; otherwise G does not have an independent set of size at least k.

From G we construct the following graph. First, we introduce a set Y of np new vertices for some integer p, the value of which we will determine later. We denote the vertices in Y by $y_1^u, \ldots, y_p^u$ for each $u \in V$. We connect each y_i^u (only) to its associated vertex u. This yields a graph G^*, in which all vertices of G now have degree $3 + p$, and all vertices of Y have degree one. Second, let K be a complete graph on ℓ vertices, where ℓ is some *odd* integer larger than np, the value of which will be made clear later on. We add $2(n - k)$ copies $K^1, \ldots, K^{2(n-k)}$ of K to G^* without introducing any further edges. The resulting graph consists of $2(n - k) + 1$ components and is denoted by $G' = (N, E')$. We denote the simple matching game on G' by (N, v). We observe that $\{uy_1^u \mid u \in V\}$ is a maximum matching in G^* of size n. Because of this and because ℓ is odd, we obtain that $v(N) = \frac{1}{2}(\ell-1)2(n-k)+n = \ell(n-k)+k$. We show that the following statements are equivalent for suitable choices of ℓ and p, hereby proving Theorem 7.

(i) G has an independent set U of size $|U| \geq k$.
(ii) $|B(x)| \leq (n - k)p + \frac{3}{2}n - 3k$ for some imputation x of (N, v).

"(i) ⇒ (ii)" Suppose G has an independent set U of size $|U| \geq k$. We define an imputation x as follows, $x \equiv \frac{1}{2}$ on each K^h, $x \equiv 1$ on U' for some subset $U' \subseteq U$ of size $|U'| = k$ and $x \equiv 0$ otherwise. Then the set of blocking pairs is

$$\{(u, y_i^u) \mid u \in V \backslash U', 1 \leq i \leq p\} \cup \{(u, v) \mid u, v \in V \backslash U' \text{ and } uv \in E\}.$$

We observe that $|\{(u, y_i^u) \mid u \in V \backslash U', 1 \leq i \leq p\}| = (n - k)p$. Furthermore, because G is 3-regular, $U' \subseteq U$ is an independent set and $k \leq \frac{1}{2}n$, we find that $|\{(u, v) \mid u, v \in V \backslash U'\}| = |E| - 3k = \frac{3}{2}n - 3k \geq 0$. Hence, $|B(x)| = (n - k)p + \frac{3}{2}n - 3k$.

"(ii) ⇒ (i)" Suppose $|B(x)| \leq (n - k)p + \frac{3}{2}n - 3k$ for some imputation x of (N, v). We may without loss of generality assume that x has minimum number of blocking pairs. We first prove a number of claims.

Claim 1. We may without loss of generality assume that $x_i \leq 1$ for all $i \in N$.

We prove Claim 1 as follows. Suppose $x_i = 1+\alpha$ for some $\alpha > 0$ for some $i \in N$. We set $x_i := 1$ and redistribute α over all vertices $j \in N \backslash Y$ with $x_j < 1$. When doing this we ensure that we do not increase the value of some x_j with more than $1-x_j$. This is possible, since $x(N) = v(N) = \ell(n-k)+k < 2\ell(n-k)+np+n = |N|$. The resulting allocation would be an imputation with a smaller or equal number of blocking pairs. This proves Claim 1.

Claim 2. We may without loss of generality assume that $x_y = 0$ for each $y \in Y$.

We prove Claim 2 as follows. Suppose $x_y > 0$ for some $y \in Y$. Let u be the (unique) neighbor of y. We set $x_y := 0$ and $x_u := \min\{x_u + x_y, 1\}$. If necessary, we redistribute the remainder over $V \cup \bigcup_j K^j$ without violating Claim 1. This is possible, since $x(N) = \ell(n-k)+k < 2\ell(n-k)+n = |\bigcup_j K^j| + |V|$. The resulting imputation would have a smaller or equal number of blocking pairs. This proves Claim 2.

Claim 3. $x(\bigcup_j K^j) = \ell(n-k)$.

We prove Claim 3 as follows. First suppose $x(\bigcup_j K^j) > \ell(n-k)$. Then we set $x_i := \frac{1}{2}$ for each $i \in \bigcup_j K^j$ and redistribute the remainder over V without violating Claim 1. This is possible, since after setting $x_i := \frac{1}{2}$ for each $i \in \bigcup_j K^j$, we have $x(N) - x(\bigcup_j K^j) = \ell(n-k)+k-\ell(n-k) = k \leq n = |V|$. The resulting imputation would have a smaller or equal number of blocking pairs. Hence, we may assume that $x(\bigcup_j K^j) \leq \ell(n-k)$ holds.

Suppose $x(\bigcup_j K^j) < \ell(n-k)$. Then there is some K^j with $x(K^j) < \frac{\ell}{2}$. By Lemma 6, there are at least $\frac{\ell-1}{2}$ blocking pairs in K^j. We choose $\ell = 2np+2|E|+2$ and obtain $|B(x)| \geq \frac{\ell-1}{2} > (n-k)p+|E|$. However, let x^* be given by $x^* \equiv \frac{1}{2}$ on $\bigcup_j K^j$, $x^* \equiv 0$ on Y, $x^* \equiv 1$ on some $U \subseteq V$ of size $|U| = k$ and $x^* \equiv 0$ on $V \backslash U$. Then x^* is an imputation as $x^*_i \geq 0$ for all $i \in N$ and $x^*(N) = \ell(n-k)+k = v(N)$. We observe that $|B(x^*)| < (n-k)p+|E|$. Hence x is not an imputation with minimum number of blocking pairs. This proves Claim 3.

We now continue with the proof. Combining Claims 2 and 3 leads to

$$x(V) = x(N) - x\left(\bigcup_{j=1}^{2(n-k)} K^j\right) - x(Y) = v(N) - \ell(n-k) = \ell(n-k)+k-\ell(n-k) = k.$$

Let R be the set of vertices v in G with $x_v < 1$. We first show that $|R| \leq n-k$. Suppose $|R| \geq n-k+1$. Since $x(Y) = 0$ due to Claim 2, we find that $(n-k)p + \frac{3}{2}n - 3k \geq |B(x)| \geq |R|p \geq (n-k)p + p$. This is not possible if we choose $p = 2n$. Hence, indeed $|R| \leq n-k$ holds.

Let U consist of all vertices $u \in V$ with $x_u = 1$. Note that $U = V \backslash R$ due to Claim 1. Since $x(V) = k$ as we deduced above, we find that $|U| \leq k$, and thus $|R| \geq n-k$. As we already deduced that $|R| \leq n-k$, we find that $|R| = n-k$, and consequently, $|U| = k$. The latter equality implies that $x_v = 0$ for all $v \in R$.

Because G is 3-regular, G has $\frac{3n}{2}$ edges. Then $B(x) \geq (n-k)p + \frac{3n}{2} - 3|U|$, with equality only if U is an independent set. Equality must hold since we assume that $B(x) \leq (n-k)p + \frac{3n}{2} - 3k$ and $|U| = k$. Hence, indeed U is an independent set of size k. This completes the proof of Theorem 7. □

5 Future Research

Beside the notorious problem of determining the complexity of computing the nucleolus of matching games with empty core, we would like to mention one other open problem. Let x be an imputation of a matching game (N, v) defined on a graph $G = (N, E)$ with edge weighting w. We say that a pair of adjacent vertices u, v has *blocking value* $e_x(i,j)^+ = \max\{0, w_{i,j} - (x_i + x_j)\}$. We let $b(x) = \sum_{ij \in E} e_x(i,j)^+$ be the *total blocking value* of x. The complexity status of the problem that asks if a matching game (N, v) has an imputation with total blocking value at most k, where k is a positive integer part of the input, is open.

References

1. Abraham, D.J., Biró, P., Manlove, D.F.: "Almost stable" matchings in the Roommates problem. In: Erlebach, T., Persinao, G. (eds.) WAOA 2005. LNCS, vol. 3879, pp. 1–14. Springer, Heidelberg (2006)
2. Balinski, M.L.: Integer programming: Methods, uses, computation. Management Science 12, 253–313 (1965)
3. Bilbao, J.M.: Cooperative games on combinatorial structures. Kluwer Academic, Norwell (2000)
4. Biró, P.: The stable matching problem and its generalizations: an algorithmic and game theoretical approach, PhD Thesis, Budapest University of Technology and Economics, Budapest, Hungary (2007)
5. Deng, X., Ibaraki, T., Nagamochi, H.: Algorithmic aspects of the core of combinatorial optimization games. Math. Oper. Res. 24, 751–766 (1999)
6. Egerváry, J.: Matrixok kombinatorius tulajdonságairól. Matematikai és Fizikai Lapok 38, 16–28 (1931)
7. Eriksson, K., Karlander, J.: Stable outcomes of the roommate game with transferable utility. Internat. J. Game Theory 29, 555–569 (2001)
8. Faigle, U., Kern, W., Fekete, S., Hochstättler, W.: The nucleon of cooperative games and an algorithm for matching games. Math. Program. 83, 195–211 (1998)
9. Gabow, H.N.: Data structures for weighted matching and nearest common ancestors with linking. In: Proceedings of SODA 1990, pp. 434–443 (1990)
10. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. Theoret. Comput. Sci. 1, 237–267 (1976)
11. Kern, W., Paulusma, D.: Matching games: the least core and the nucleolus. Math. Oper. Res. 28, 294–308 (2003)
12. Khachiyan, L.G.: A polynomial algorithm in linear programming. Soviet Mathematics Doklady 20, 191–194 (1979)
13. Klaus, B., Nichifor, A.: Consistency and monotonicity in one-sided assignment problems (2009) (preprint)
14. Maschler, M., Peleg, B., Shapley, L.S.: Geometric properties of the kernel, nucleolus, and related solution concepts. Math. Oper. Res. 4, 303–338 (1979)

15. Matsui, T.: A note on the nucleolus of assignment games. In: 5th International Conference on Nonlinear Analysis and Convex Analysis, pp. 253–260. World Scientific, Singapore (1998)
16. Micali, S., Vazirani, V.V.: An $O(\sqrt{|V|}\cdot|E|)$ algorithm for finding maximum matching in general graphs. In: Proceedings of FOCS 1980, pp. 17–27 (1980)
17. Nemhauser, G.L., Trotter, L.E.: Vertex packings: structural properties and algorithms. Math. Program. 8, 232–248 (1975)
18. Owen, G.: Game theory. Academic Press, San Diego (1995)
19. Paulusma, D.: Complexity aspects of cooperative games, PhD Thesis, University of Twente, Enschede, the Netherlands (2001)
20. Schmeidler, D.: The nucleolus of a characteristic function game. SIAM J. Appl. Math. 17, 1163–1170 (1969)
21. Shapley, L.S., Shubik, M.: The assignment game I: the core. Internat. J. Game Theory 1, 111–130 (1972)
22. Schrijver, A.: Combinatorial optimization. In: Polyhedra and efficiency. Algorithms and Combinatorics 24, vol. A. Springer, Berlin (2003)
23. Solymosi, T., Raghavan, T.E.S.: An algorithm for finding the nucleolus of assignment games. Internat. J. Game Theory 23, 119–143 (1994)