Computing solutions for matching games*

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

[1] Institute of Economics, Hungarian Academy of Sciences,
H-1112, Budaörsi út 45, Budapest, Hungary
birop@econ.core.hu**

[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 member if the core is nonempty. This algorithm improves previous work based on the ellipsoid method and can also be used to compute stable solutions for instances of the stable roommates problem with payments. Second we show that the nucleolus of an n-player matching game with a nonempty core can be computed in $O(n^4)$ time. This generalizes the corresponding result of Solymosi and Raghavan for assignment games. Third we prove that is NP-hard to determine an imputation with minimum number of blocking pairs, even for matching games with unit edge weights, whereas the problem of determining an imputation with minimum total blocking value is shown to be polynomial-time solvable for general matching games.

Keywords. matching game; nucleolus; cooperative game theory.

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.

** Supported by EPSRC grant EP/E011993/1, by OTKA grant K69027 and by the Hungarian Academy of Sciences under its Momemtum Programme (LD-004/2010).

*** Supported by EPSRC grant EP/G043434/1.

* A preliminary version of this paper has been presented at TAMC 2010.

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.

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. 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. For instance, suppose that (i_1, i_2) and (i_3, i_4) are pairs in the matching. Then players i_1, i_2, i_3, and i_4 may decide to leave the matching if (i_1, i_3) forms a better pair than (i_1, i_2). They may even decide to do so if i_2 and i_4 cannot play together (for whatever reason). Contrary to the previous setting, i_1 and i_3 may compensate i_2 and i_4 for their loss of income. 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.

1.1 Cooperative game theory: definitions and terminology

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. 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 of 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 two solution concepts that are well known and that have been studied for matching games. We assume that profits must be maximized, because this is the case for matching games. We refer to Owen [19] for a general survey. First, the *core* of a game (N, v) consists of all allocations x with $x(S) \geq v(S)$ for all $S \in 2^N$. 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, 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 non-decreasing 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, it consists of exactly one imputation as shown by Schmeidler [21]. 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 [15] for more details.

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. Matching games defined on a bipartite graph are called *assignment games*.

1.2 Existing results on matching 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 [22] show that the core of an assignment game is always nonempty.

We will now discuss complexity aspects of solution concepts for matching games. First we recall a result of Gabow [10] which we will need later on.

Theorem 1 ([10]). *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 is easy to verify and can be found in several papers, see e.g. [5, 7, 20]. 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 $c(N) = \sum_{u \in N} c(u)$. Note that $c(N) \geq v(N)$ for the corresponding matching game (N, v), while $c(N) > v(N)$ is possible. Hence, a cover does not have to be an allocation of (N, v).

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 [13]. 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.

We will briefly survey the existing work on the nucleolus of a matching game. Note that an allocation x of a matching game (N, v) is an imputation if and only if x is nonnegative, because $v(\{i\}) = 0$ for all $i \in N$. This means that the set of imputations is nonempty, because it contains the n allocations that assign $v(N)$ to exactly one player and 0 to all the other players. Consequently, every matching game has a nucleolus.

Observation 1 implies that the size of the linear programs involved in the procedure of Maschler, Peleg and Shapley [15] is polynomial in the case that the matching game has a nonempty core [20]. 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 [24] compute the nucleolus of an assignment game without making use of the ellipsoid method. We state their result below, as we need it later on. For computing the nucleolus of assignment games defined on bipartite graphs that are unbalanced, a faster algorithm has been given by Matsui [16].

Theorem 2 ([24]). *The nucleolus of an n-player assignment game can be computed in $O(n^4)$ time.*

It is known [12] that the nucleolus of a simple matching game can be computed in polynomial time by using the standard procedure of Maschler, Peleg and Shapley [15], after reducing the size of the involved linear programs to be polynomial. This result has been extended to node matching games [20], i.e., matching games defined on a graph $G = (N, E)$ with an edge weighting w that allows a weighting $w^* : N \to \mathbb{R}_+$ such that $w(uv) = w^*(u) + w^*(v)$ for all $uv \in E$; note that every simple matching game is a node matching game by choosing $w^* \equiv \frac{1}{2}$.

Determining the computational complexity of finding the nucleolus for general matching games is an outstanding open problem, although there is still some hope for an efficient algorithm. This hope stems from the observation that the minimum excess of a matching game can be computed in polynomial time. This condition is sufficient to compute the nucleolus of a cooperative game in polynomial time if the core is nonempty [9]. It also stems from the result that an imputation in the nucleon can be computed in polynomial time for matching games, as shown by Faigle et al. [8]. The nucleon is a solution concept similar to the nucleolus. It is obtained by taking multiplicative excesses $e'(S, x) = x(S)/v(S)$ instead of additive excesses $e(S, x) = x(S) - v(S)$.

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. We say that a pair of adjacent vertices (u, v) is a *blocking pair* of a vector $x \in \mathbb{R}^N$ if $x_u + x_v < w(uv)$, and we define their *blocking value* with respect to x as $e_x(u, v)^+ = \max\{0, w(uv) - (x_u + x_v)\}$, which is to be interpreted as follows. If (u, v) is not blocking x then its blocking value $e_x(u, v)^+$ is zero. Otherwise, its blocking value expresses to which extent (u, v) is blocking x. Let $B(x) = \{(u, v) \mid x_u + x_v < w(uv)\}$ denote the *set of blocking pairs* of a vector $x \in \mathbb{R}^N$ and let $b(x) = \sum_{uv \in E} e_x(u, v)^+$ denote the *total blocking value* of x.

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. Note that $p(N) \leq v(N)$ for the corresponding matching game (N, v), while $p(N) < v(N)$ is possible. Hence, a payoff does not have be an allocation of (N, v).

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 $B(p) = \emptyset$, or equivalently, $b(p) = 0$. We also call such a pair *stable*. This problem is polynomially solvable by the following observation which is well known (cf. Eriksson and Karlander [7]) and easy to verify.

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

1.3 Our results

In Section 2, we give a new characterization of the core of a matching game. We also 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. By Observation 2 we can use our algorithm to find a stable solution for instances of Stable Roommates with Payments in $O(nm + n^2 \log n)$ time if such a solution exists.

Like the algorithm of Deng, Ibaraki and Nagamochi [5] for simple matching games, our algorithm for general matching games does not rely on the ellipsoid method. Instead it is based on the linear programming relaxation of the standard integer programming formulation for finding a maximum weight matching in a graph. Deng, Ibaraki and Nagamochi [5] show that the core of a matching game is nonempty if and only if the integrality gap is zero. Solving the dual of the relaxation yields a minimum weight cover. A classic result of Egerváry [6] shows that the maximum weight of a matching in a bipartite graph G is equal to the minimum weight of a cover of G. Consequently, the integrality gap is zero for bipartite graphs, and matching games on bipartite graphs, i.e., assignment games have a nonempty core as shown already by Shapley and Shubik [22]. In particular, minimum weight covers are core allocations in the case of assignment games due to Observation 1. Our approach for matching games is to make a translation from general graphs to bipartite graphs by using the well-known duplication technique of Nemhauser and Trotter [18].

In Section 3 we use the aforementioned duplication technique to show that the nucleolus of an n-player matching game with a nonempty core can be computed in $O(n^4)$ time. This generalizes the corresponding result of Solymosi and Raghavan [24] on computing the nucleolus of assignment games (Theorem 2).

We note that Klaus and Nichifor [14] investigate the relation of the core with other solution concepts for matching games. In particular, they express the need of a comparison of matching games with a nonempty core to assignment games and ask to which extent properties of assignment games are carried over to matching games with a nonempty core. As the results in Sections 2 and 3 are based on a duplication technique yielding bipartite graphs, our paper gives such a comparison with regards to computing a core allocation and the nucleolus.

Every core allocation of a matching game is an imputation with no blocking pairs, or equivalently, with total blocking value zero. In the final two sections of our paper, we consider matching games with an empty core. There, we try to minimize the number of blocking pairs and the total blocking value, respectively. This leads to the following two decision problems, which are trivially solvable for assignment games, because these games have a nonempty core [22]. For both problems we are only interested in imputations. This is justified by the fact that no player i will accept an allocation x with $x_i < 0 = v\{i\}$.

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

Blocking Value
Instance: a matching game (N, v) and a rational number $k \geq 0$.
Question: does (N, v) allow an imputation x with $b(x) \leq k$?

Our results on these two problems are as follows. 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 purposes. On the positive side, we show in Section 5 that the Blocking Value problem is solvable in polynomial time for general matching games.

2 The core of a matching game

As mentioned in the previous section, Shapley and Shubik [22] 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 but in a broader setting, namely for matching games after presenting a new characterization of their core, which may be empty. First we introduce some terminology.

Let $G = (N, E)$ be a graph with edge weighting $w : E \to \mathbb{R}_+$. We write $v \in e$ if v is an end vertex of edge e. A *fractional matching* is an edge mapping $f : E \to \mathbb{R}_+$ 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 $e \in E$. In this context, the *integrality gap* is defined as as the difference between the maximum weight of a fractional matching and the maximum weight of a matching.

2.1 The characterization

Solving the linear programming relaxation of the standard integer programming formulation for finding a maximum weight matching in a graph yields a maximum weight fractional matching. Solving the dual of the relaxation yields a minimum weight cover. These well-known observations lead us to two lemmas, both of which we need to prove our characterization. The first lemma is an application of the Duality Theorem (cf. Schrijver [23]). The second lemma is a special case of Theorem 1 from Deng, Ibaraki and Nagamochi [5]. It shows that the core of a matching game is nonempty if and only if the integrality gap is zero.

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.*

For our core characterization we also make use of the following theorem, which is a straightforward consequence of a result by Balinski [2]. In Section 2.2 we explain how his result can also be obtained by using the duplication technique of Nemhauser and Trotter [18].

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] characterize stable solutions for instances of the problem Stable Roommates with Payments in terms of forbidden minors. By Observation 2 we can apply Proposition 1 to find an alternative characterization, namely that a weighted graph G has a stable solution if and only if the maximum weight of a matching in G is equal to the maximum weight of a half-matching in G.

2.2 The algorithm

Proposition 1 tells us that we can decide whether the core of a matching game is nonempty by checking if the maximum weight of a matching equals the maximum weight of a half-matching. Due to Theorem 1 we can compute a maximum weight matching of a weighted n-vertex graph with m edges in $O(nm + n^2 \log n)$ time. What about computing the maximum weight of a half-matching? We will explain how to combine results from the literature in order to find an $O(nm + n^2 \log n)$ running time for computing the maximum weight of a half-matching and for computing a core allocation if the core is nonempty. The approach for doing this is based on a natural translation from general graphs to bipartite graphs introduced by Nemhauser and Trotter [18]. This translation is motivated by the following result of Egerváry [6].

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 is immediately useful for bipartite graphs. Combining it with Lemma 3 yields that the integrality gap is zero for bipartite graphs. As a matter of fact, combining it with Observation 1 yields that every minimum weight cover is a core allocation in the case of an assignment game.

Theorem 4 is useful for general graphs as well. It is well known how to use it for computing the maximum weight of a half-matching in $O(n^3)$ time for weighted graphs on n vertices (cf. Theorem 30.3 of Schrijver [23]). Nevertheless we explain this in detail below, because we need the arguments for improving the running time to $O(nm+n^2 \log n)$ and for computing a core allocation in the same time if the core is nonempty. We also need the arguments in the proof of Lemma 5 of Section 3.

Let (N, v) be a matching game defined on a graph $G = (N, E)$ with an edge weighting w. Let n and m denote the number of vertices and edges of G, respectively. First we construct a bipartite graph from G according to the duplication technique introduced by Nemhauser and Trotter [18] for finding a maximum weight independent set in a graph. 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 $G^d = (N^d, E^d)$ with $2n$ vertices and $2m$ edges. We call G^d the *duplicate* of G. Note that G^d can be constructed in $O(n + m)$ time.

We now compute a maximum weight matching f^d of G^d. Because G^d has $2n$ edges and $2m$ vertices, this takes $O(2n \cdot 2m + (2n)^2 \log 2n) = O(nm + n^2 \log n)$ 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 [23]). 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 $O(m)$ 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

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

Then f is a maximum weight half-matching due to Lemma 3, as desired. It took us $O(nm+n^2 \log n)$ time in total to compute f. As a side effect, we observe that equation (1) implies Theorem 3.

Recall that by Theorem 1 we can compute a maximum weight matching f^* of G in $O(nm + n^2 \log n)$ time, and that by Proposition 1 we just need to check whether $w(f^*) = w(f)$ in order to determine whether the core of (N, v) is nonempty.

Suppose that the core of (N, v) is nonempty, so $w(f^*) = w(f)$. Because $w(f) = c(N)$ and $w(f^*) = v(N)$, we obtain that

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

Hence, c is a core member due to Observation 1. It costed $O(nm+n^2 \log n)$ time in total to compute c. Summarizing, we have obtained 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 a simple matching game (N, v) defined on a n-vertex graph $G = (N, E)$ with m edges, we can improve the running time of the algorithm in Theorem 5 as follows. We use the $O(\sqrt{n}m)$ time algorithm of Micali and Vazirani [17] to compute a maximum matching f^* of G instead of using Theorem 1. We observe that every edge weight in the duplicate G^d of G is equal to $\frac{1}{2}$. This means that every maximum matching of G^d is a maximum weight matching of G^d. Hence, we can compute a maximum weight matching f^d of G^d in $O(\sqrt{n}m)$ time by using the algorithm of Micali and Vazirani [17] again. Given f^d we can compute the required cover c^d in $O(m)$ time (cf. Theorem 16.6 of Schrijver [23]). The rest of the algorithm stays the same.

The above leads to the following. If G has no isolated vertices, then $m = \Omega(n)$ and the overall running time becomes $O(\sqrt{n}m)$. Otherwise, we first remove every isolated vertex from G. In the case that we find a core allocation x of the remaining game, we can extend x by setting $x_v := 0$ for every isolated vertex v that we removed. We may do so because of the following. Let S be a coalition of players forming a maximum weight matching M of G. This means that $v(N) = w(M)$. Let x be a core allocation of (N, v). Then $x(N) = v(N)$ and $x(S) \geq w(M) = v(N)$ imply that $x_u = 0$ for every $u \in N \setminus S$.

3 The nucleolus of a matching game with a 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 vector $\overline{x}$ given by $\overline{x}_{u'} = \overline{x}_{u''} = \frac{1}{2}x_u$ for all $u \in N$.

Lemma 5. *Let (N, v) be a matching game with a nonempty core. Then a vector $x \in \mathbb{R}^N$ is an imputation of (N, v) if and only if $\overline{x}$ is an imputation of (N^d, v^d).*

Proof. By definition, $x_u \geq 0$ if and only if $\overline{x}_{u'} = \overline{x}_{u''} = \frac{1}{2}x_u \geq 0$ for all $u \in N$. Hence, we are left to show that $x(N) = v(N)$ if and only if $\overline{x}(N^d) = v^d(N^d)$. Because $x(N) = \overline{x}(N^d)$ by the definition of $\overline{x}$, this means that we must show that $v(N) = v^d(N^d)$.

Because G^d is bipartite, (N^d, v^d) is an assignment game. This means that (N^d, v^d) has a nonempty core [22]. Let y^d be a core allocation of (N^d, v^d). Observation 1 implies that y^d is a cover of G^d with weight $y^d(N^d) = v^d(N^d) = w^d(f^d)$, where f^d is a maximum weight matching of G^d. We apply Lemma 3 and find that y^d is a minimum weight cover of (N^d, v^d). Then the vector $y \in \mathbb{R}^N_+$ given by $y_u = y^d_{u'} + y^d_{u''}$ for all $u \in N$ satisfies $y(N) = v(N)$ due to equation (2). By the definition of y, we have that $y(N) = y^d(N^d)$. Because $y^d(N^d) = v^d(N^d)$, we then obtain that $v(N) = v(N^d)$, as desired. This completes the proof of Lemma 5. □

Lemma 6. *Let (N, v) be a matching game with a nonempty core. Then the nucleolus of (N^d, v^d) is the duplicate of the nucleolus of (N, v).*

Proof. Let η^d be the nucleolus of (N^d, v^d). Define η^* by $\eta^*_{u'} = \eta^d_{u''}$ and $\eta^*_{u''} = \eta^d_{u'}$ for all $u \in N$. Then $\theta(\eta^d) = \theta(\eta^*)$. Because η^d is unique as shown by Schmeidler [21], we find that $\eta^d_{u'} = \eta^d_{u''}$ for all $u \in N$. This makes it possible to define the vector η with $\overline{\eta} = \eta^d$.

By the definition of the nucleolus, η^d is an imputation of (N^d, v^d). We apply Lemma 5 and find that η is an imputation of (N, v). Let x be an arbitrary imputation of (N, v). Suppose that $\theta(x) \succ \theta(\eta)$, i.e., $\theta(x)$ is lexicographically greater than $\theta(\eta)$. By Lemma 5, we find that $\overline{x}$ is an imputation of (N^d, v^d). However, $\theta(x) \succ \theta(\eta)$ implies that $\theta(\overline{x}) \succ \theta(\overline{\eta}) = \theta(\eta^d)$. This is not possible, because η^d is the nucleolus of (N^d, v^d). Hence, $\theta(\eta) \succeq \theta(x)$. This means that η is the nucleolus of (N, v). This completes the proof of Lemma 6. □

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

Proof. Let (N, v) be an n-player matching game with a 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 $O((2n)^4) = O(n^4)$ time. Let η be the nucleolus of (N, v). By Lemma 6 we find that $\eta^d = \overline{\eta}$. This means that we can construct η in $O(n^2)$ time from η^d. Hence, the total time that we used is $O(n^4)$. This finishes the proof of Theorem 6. □

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 whether 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 7. *Let K be a complete graph with vertex set $\{1, \ldots, \ell\}$ for some odd integer ℓ and with unit edge weights. Let $x \in \mathbb{R}^K_+$. If $x(K) < \frac{\ell}{2}$ then $|B(x)| \geq \frac{\ell-1}{2}$.*

Proof. Write $\ell = 2q + 1$ and use induction on q. If $q = 0$ the statement holds. Suppose $q \geq 1$. We assume without loss of generality that $x_1 \leq x_2 \leq \cdots \leq x_{2q+1}$. Because $x(K) < \frac{\ell}{2}$, we have that $x_1 < \frac{1}{2}$. 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 $x_2 < \frac{1}{2}$. 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 is to test whether a graph $G = (V, E)$ 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 [11] 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 $n = |V|$.

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 $y^u_1, \ldots, y^u_p$ for each $u \in V$. We connect each y^u_i (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. The vertices of Y form "pendant stars", and we have added them to the vertices of G for the following two reasons. The first reason is to make it easier to compute the value of the grand coalition, as G^* has a maximum matching of size n. The second reason is that by choosing p sufficiently large we can ensure that we do not have to assign fractions to the vertices in G when minimizing the number of blocking pairs.

Now 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. This results in a graph $G' = (N, E')$, which consists of $2(n - k) + 1$ connected components, namely G^* and the $2(n - k)$ complete graphs $K^1, \ldots, K^{2(n-k)}$. We need these complete graphs for the following reason. By choosing ℓ large enough, no K^i will contain a blocking pair when we try to minimize the number of blocking pairs (cf. Lemma 7). Because ℓ is odd, each K^i will have to take away $\frac{1}{2}$ from what is available for distribution to the vertices of G^*. Consequently, what remains for G^* drops down from n to k. Because each vertex in G gets allocated either 0 or 1 due to the pendant stars, we can show that the number of blocking pairs is below a certain threshold if and only if the vertices that get allocated 1 unit form an independent set of size k. We explain this in detail below.

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, thereby 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

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

We now determine $|B(x)|$. By construction, $|\{(u, y_i^u) \mid u \in V \setminus U', 1 \leq i \leq p\}| = (n-k)p$. Because G is 3-regular, $|E| = \frac{3}{2}n$. Because $U' \subseteq U$ is an independent set, we then find that $|\{(u,v) \mid u,v \in V \setminus U' \text{ and } uv \in E\}| = \frac{3}{2}n - 3k$. Hence, $|B(x)| = (n-k)p + \frac{3}{2}n - 3k$, as desired.

"(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 start by proving a number of claims to show that x can be taken to be of the same form as the imputation that we constructed in the proof of the implication "(i) ⇒ (ii)".

Claim 1. We may assume without loss generality that $x_y = 0$ for each $y \in Y$.

We prove Claim 1 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 $x_u := x_u + x_y$. The resulting imputation has a smaller or equal number of blocking pairs. This proves Claim 1.

Claim 2. We may assume without loss of generality that $x(\bigcup_j K^j) = \ell(n-k)$.

We prove Claim 2 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. The resulting imputation has 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 7, there are at least $\frac{\ell-1}{2}$ blocking pairs in K^j. We choose $\ell = 2np + 2|E| + 2$. Recall that $|E| = \frac{3}{2}n$, because G is 3-regular. Then we obtain that $|B(x)| \geq \frac{\ell-1}{2} > np + |E| = np + \frac{3}{2}n$. Because $|B(x)| \leq (n-k)p + \frac{3}{2}n - 3k$, this is not possible. Hence, we have proven Claim 2.

Combining Claims 1 and 2 leads to

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

Claim 3. We may assume without loss of generality that $x_u \leq 1$ for all $u \in V$.

We prove Claim 3 as follows. Suppose that $x_u = 1 + \alpha$ for some $\alpha > 0$ for some $u \in V$. We set $x_u := 1$ and redistribute α over all vertices $v \in V$ with $x_v < 1$. When doing this we ensure that we do not increase the value of some x_v with more than $1 - x_v$. This is possible, because $x(V) = k < n$. The resulting imputation has a smaller or equal number of blocking pairs. This proves Claim 3.

Claim 4. We may assume without loss of generality that $x_u \in \{0, 1\}$ for all $u \in V$.

We prove Claim 4 as follows. By Claim 3, $x_u \leq 1$ for all $u \in V$. Suppose that $0 < x_u < 1$ for some $u \in V$. Recall that $x(V) = k$, which is an integer. This means that there exist one or more vertices in $V \setminus \{u\}$ that are each allocated a fraction between 0 and 1 such that we can give their allocation to u in order to set $x_u := 1$. By Claim 1, $x_y = 0$ for each $y \in Y$. Then the only extra blocking pairs that we introduce in this way are formed by the edges of G. Recall that $|E| = \frac{3}{2}n$. Hence, we get at most $\frac{3}{2}n$ new blocking pairs. However, we lose the p blocking pairs (u, y_h^u) for $h = 1, \ldots, p$. Then, by choosing $p > \frac{3}{2}n$, the resulting imputation has a smaller number of blocking pairs, which is not possible. This proves Claim 4.

We now continue with the proof. Let U consist of all vertices $u \in V$ with $x_u = 1$. Recall that $x(V) = k$. Then, by Claim 4, we find that $|U| = k$, and that $x_v = 0$ for all $v \in V \setminus U$. Recall that $|E| = \frac{3}{2}n$. Then $B(x) \geq (n-k)p + \frac{3}{2}n - 3|U| = (n-k)p + \frac{3}{2}n - 3k$, with equality only if U is an independent set. However, equality must hold because we assume that $B(x) \leq (n-k)p + \frac{3n}{2} - 3k$. Hence, U is an independent set of size k, as desired. This completes the proof of Theorem 7. □

5 The total blocking value

We show the following result.

Proposition 2. *The* Blocking Value *problem can be solved in polynomial time.*

Proof. Let (N, v) be a matching game defined on a graph $G = (N, E)$ with edge weighting w. Recall that $e_x(u, v)^+ = \max\{0, w(uv) - (x_u + x_v)\}$ and $b(x) = \sum_{ij \in E} e_x(i, j)^+$ for an imputation x. Hence, an optimal solution of the linear program

$$
\begin{array}{llll}
\text{(BV)} & \min \sum_{uv \in E} z_{uv} & & \\
& \text{s.t. } x_u + x_v + z_{uv} & \geq w(uv) & (uv \in E) \\
& x(N) & = v(N) & \\
& x_u & \geq 0 & (u \in N) \\
& z_{uv} & \geq 0 & (uv \in E)
\end{array}
$$

is a solution (x, z) such that x is an imputation with minimum total blocking value $b(x)$. Because (BV) can be solved in polynomial time by the ellipsoid method [13], we find that the BLOCKING VALUE problem is polynomial-time solvable. □

The minimum blocking value of an imputation x can be interpreted as the total utility that a higher power must supply to pairs of players in order to eliminate all blocking pairs. From this viewpoint, we could also try to minimize the total utility that such a power must supply to the individual players instead. We call this problem the BLOCKING PAIRS ELIMINATION problem. For a matching game (N, v) defined on a graph $G = (N, E)$ with edge weighting w, it can be formulated as the linear program

$$
\begin{array}{llll}
\text{(BPE)} & \min \sum_{u \in N} y_u & & \\
& \text{s.t. } x_u + x_v + y_u + y_v & \geq w(uv) & (uv \in E) \\
& x(N) & = v(N) & \\
& x_u & \geq 0 & (u \in N) \\
& y_u & \geq 0 & (u \in N).
\end{array}
$$

Consequently, BLOCKING PAIRS ELIMINATION can be solved in polynomial time as well by the ellipsoid method [13]. Alternatively, we can compute an optimal solution of (BPE) as follows. First we compute a minimum weight cover c of G. Then we choose an imputation $x^* \leq c$, and we take $y^* = c - x^*$. We claim that (x^*, y^*) is an optimal solution of (BPE) with $\sum_{u \in N} y^*_u = (x^*+y^*)(N) - x^*(N) = c(N) - v(N)$. In order to see this, let (x, y) be a solution of (BPE). Then $x + y$ is a cover by definition, which means that $(x + y)(N) \geq c(N)$. As a result we find that $\sum_{u \in N} y_u = (x + y)(N) - x(N) \geq c(N) - v(N) = \sum_{u \in N} y^*_u$, as desired. Note that $\sum_{u \in N} y^*_u = c(N) - v(N)$ is the integrality gap due to Lemma 3.

In order to show the difference between the two problems, we show that for optimal solutions (x, z) and (x', y) of (BV) and (BPE), respectively, the difference

$$\min \sum_{uv \in E} z_{uv} - \min \sum_{u \in N} y_u$$

can be made arbitrarily large. For this purpose, we consider the simple matching game (N, v) defined on the graph that consists of two connected components, namely a complete graph K_{2q+1} on vertices $v_1, \ldots, v_{2q+1}$ for some integer q and a star $K_{1,r}$ with center u_0 and leaves $w_1, \ldots, w_r$ for some integer r. We choose q and r such that $q > \frac{r}{2}$. Below we explain this in detail.

Let (x, z) be an optimal solution of (BV). Recall that x is an imputation of (N, v) that has minimum total blocking value $b(x) = \sum_{uv \in E} z_{uv}$. We will show that $b(x) = \frac{1}{2}r$ for x defined as $x \equiv \frac{1}{2}$ on $V(K_{2q+1})$, $x_{u_0} = \frac{1}{2}$ and $x \equiv 0$ on $V(K_{1,r}) \setminus \{u_0\}$.

First, we may assume without loss of generality that $x_w = 0$ for all $w \in V(K_{1,r}) \setminus \{u_0\}$; if not then we could increase x_{u_0} and decrease x_w without decreasing the total blocking value. Second, we may assume without loss of generality that $x_{u_0} \leq 1$; if not then we redistribute $x_{u_0} - 1$ over $x(K_{2q+1})$ without decreasing the total blocking value. Third, we may assume without loss of generality that $x(K_{2q+1}) \leq q + \frac{1}{2}$; if not then we assign $\frac{1}{2}$ to each v_i and redistribute what is left from $x(K_{2q+1})$ over x_{u_0} without decreasing the blocking value. Note that we can do this without making x_{u_0} larger than 1, because $x(N) = v(N) = q+1$. Because of this, we find that there exists an $\varepsilon \in \mathbb{R}$ with $0 \leq \epsilon \leq \frac{1}{2}$ such that $x(K_{2q+1}) = q + \epsilon$, $x_{u_0} = 1 - \epsilon$ and $x_w = 0$ for all $w \in V(K_{1,r}) \setminus \{u_0\}$. By definition, $z_{uv} \geq 1 - x_u - x_v$ for every edge uv, and we deduce that

$$
\begin{aligned}
b(x) = \textstyle\sum_{uv \in E} z_{uv} &= \sum_{uv \in E(K_{2q+1})} z_{uv} + \sum_{w \in V(K_{1,r}) \setminus \{u_0\}} z_{u_0 w} \\
&\geq \sum_{uv \in E(K_{2q+1})} (1 - (x_u + x_v)) + \sum_{w \in V(K_{1,r}) \setminus \{u_0\}} \varepsilon \\
&= (2q+1)q - 2qx(K_{2q+1}) + \varepsilon r \\
&= (2q+1)q - 2q(q + \epsilon) + \varepsilon r \\
&= q - 2q\varepsilon + \varepsilon r \\
&= q + (r - 2q)\varepsilon.
\end{aligned}
$$

Recall that $q > \frac{r}{2}$. Then $r - 2q < 0$, and consequently, $b(x) = q + (r - 2q)\varepsilon$ is minimized for $\varepsilon = \frac{1}{2}$, which yields that the minimum $b(x) = \frac{1}{2}r$ is achieved by x given by $x \equiv \frac{1}{2}$ on $V(K_{2q+1})$, $x_{u_0} = \frac{1}{2}$ and $x_w \equiv 0$ on $V(K_{1,r}) \setminus \{u_0\}$, as desired. In contrast, the solution (x, y) of (BPE) that consists of the same imputation x and the vector y defined by $y_u = \frac{1}{2}$ and $y \equiv 0$ on $N \setminus \{u\}$ leads to $\sum_{u \in N} y_u = \frac{1}{2}$, which is a constant, and thus independent of q and r.

We leave the problem of finding a combinatorial proof for Proposition 2 as an open problem.

Acknowledgements. We would like to thank David Manlove and the two anonymous reviewers for their useful suggestions that helped us to improve the readability of our paper.

References

1. D.J. Abraham, P. Biró and D.F. Manlove, “Almost stable” matchings in the Roommates problem, In: *Proceedings of WAOA 2005*, LNCS 3879 (2006) 1–14.
2. M.L. Balinski, Integer programming: Methods, uses, computation, *Management Science* **12** (1965) 253–313.
3. J.M. Bilbao, *Cooperative games on combinatorial structures*, Kluwer Academic, Norwell, Massachusetts, 2000.
4. P. Biró, *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. X. Deng, T. Ibaraki and H. Nagamochi, Algorithmic aspects of the core of combinatorial optimization games, *Math. Oper. Res.* **24** (1999) 751–766.
6. J. Egerváry, Matrixok kombinatorius tulajdonságairól, *Matematikai és Fizikai Lapok* **38** (1931) 16–28.
7. K Eriksson and J. Karlander, Stable outcomes of the roommate game with transferable utility, *Internat. J. Game Theory* **29** (2001) 555–569.
8. U. Faigle, W. Kern, S. Fekete, and W. Hochstättler, The nucleon of cooperative games and an algorithm for matching games, *Math. Program.* **83** (1998) 195–211.
9. U. Faigle, W. Kern, and J. Kuipers, On the computation of the nucleolus of a cooperative game, *Internat. J. Game Theory* **30** (2001) 79–98.
10. H.N. Gabow, Data structures for weighted matching and nearest common ancestors with linking, In: *Proceedings of SODA 1990* (1990) 434–443.
11. M.R. Garey, D.S. Johnson, and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoret. Comput. Sci.* **1** (1976) 237–267.
12. W. Kern and D. Paulusma, Matching games: the least core and the nucleolus, *Math. Oper. Res.* **28** (2003) 294–308.
13. L.G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Mathematics Doklady* **20** (1979) 191–194.
14. B. Klaus and A. Nichifor, Consistency for one-sided assignment problems, *Social Choice and Welfare* **35** (2010) 415–433.
15. M. Maschler, B. Peleg and L.S. Shapley, Geometric properties of the kernel, nucleolus, and related solution concepts, *Math. Oper. Res.* **4** (1979) 303–338.
16. T. Matsui, A note on the nucleolus of assignment games, In: 5th International Conference on Nonlinear Analysis and Convex Analysis, World Scientific, Singapore (1998) 253–260.
17. S. Micali and V.V. Vazirani, An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs, In *Proceedings of FOCS 1980* (1980) 17–27.
18. G.L. Nemhauser and L.E. Trotter, Vertex packings: structural properties and algorithms, *Math. Program.* **8** (1975) 232–248.
19. G. Owen, *Game theory*, Academic Press, San Diego, California, 1995.
20. D. Paulusma, *Complexity aspects of cooperative games*, PhD Thesis, University of Twente, Enschede, the Netherlands, 2001.
21. D. Schmeidler, The nucleolus of a characteristic function game, *SIAM J. Appl. Math.* **17** (1969) 1163–1170.
22. L.S. Shapley and M. Shubik, The assignment game I: the core, *Internat. J. Game Theory* **1** (1972) 111–130.
23. A. Schrijver, *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, Algorithms and Combinatorics 24, Springer-Verlag, Berlin, 2003.
24. T. Solymosi and T.E.S. Raghavan, An algorithm for finding the nucleolus of assignment games, *Internat. J. Game Theory* **23** (1994) 119–143.