Computing the Nucleolus of Weighted Cooperative Matching Games in Polynomial Time

Jochen Könemann[1], Kanstantsin Pashkovich[2], and Justin Toth[1](✉)

[1] University of Waterloo, Waterloo, ON N2L 3G1, Canada
{jochen,wjtoth}@uwaterloo.ca
[2] University of Ottawa, Ottawa, ON K1N 6N5, Canada
kpashkov@uottawa.ca

Abstract. We provide an efficient algorithm for computing the nucleolus for an instance of a weighted cooperative matching game. This resolves a long-standing open question posed in [Faigle, Kern, Fekete, Hochstättler, Mathematical Programming, 1998].

Keywords: Combinatorial optimization · Algorithmic game theory · Matchings

1 Introduction

Imagine a network of players that form partnerships to generate value. For example, maybe a tennis league pairing players to play exhibition matches [3], or people making trades in an exchange network [39]. These are examples of what are called *matching games*. In a (weighted) matching game, we are given a graph $G = (V, E)$, weights $w : E \to \mathbb{R}_{\geq 0}$, the player set is the set V of nodes of G, and $w(uv)$ denotes the value earned when u and v collaborate. Each coalition $S \subseteq V$ is assigned a *value* $\nu(S)$ so that $\nu(S)$ is equal to the value of a maximum weight matching on the induced subgraph $G[S]$. The special case of matching games where $w = \mathbb{1}$ is the all-ones vector, and G is bipartite is called an *assignment game* and was introduced in a classical paper by Shapley and Shubik [39], and was later generalized to general graphs by Deng, Ibaraki, and Nagamochi [11].

We are interested in what a fair redistribution of the total value $\nu(V)$ to the players in the network looks like. The field of *cooperative game theory* gives us the language to make this question formal. A vector $x \in \mathbb{R}^V$ is called an *allocation* if

This work was done in part while the second author was visiting the Simons Institute for the Theory of Computing. Supported by DIMACS/Simons Collaboration on Bridging Continuous and Discrete Optimization through NSF grant #CCF-1740425.
We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG).

A. Lodi and V. Nagarajan (Eds.): IPCO 2019, LNCS 11480, pp. 413–426, 2019.
https://doi.org/10.1007/978-3-030-17953-3_31

$x(V) = \nu(V)$ (where we use $x(V)$ as a short-hand for $\sum_{i \in V} x(i)$ as usual). Given such an allocation, we let $x(S) - \nu(S)$ be the *excess* of coalition $S \subseteq V$. This quantity can be thought of as a measure of the satisfaction of coalition S. A fair allocation should maximize the bottleneck excess, i.e. maximize the minimum excess, and this can be accomplished by an LP:

$$\begin{aligned} \max \quad & \varepsilon && && (P) \\ \text{s.t.} \quad & x(S) \geq \nu(S) + \varepsilon && \text{for all} \quad S \subseteq V \\ & x(V) = \nu(V) \\ & x \geq \mathbb{0}. \end{aligned}$$

Let ε^* be the optimum value of (P), and define $P(\varepsilon^*)$ to be the set of allocations x such that (x, ε^*) is feasible for (P). The set $P(\varepsilon^*)$ is known as the *leastcore* [33] of the given cooperative game, and the special case when $\varepsilon^* = 0$, $P(0)$ is the well-known *core* [21] of (V, ν). Intuitively, allocations in the core describe payoffs in which no coalition of players could profitably deviate from the *grand coalition* V.

Why stop at maximizing the bottleneck excess? Consider an allocation which, subject to maximizing the smallest excess, maximizes the second smallest excess, and subject to that maximizes the third smallest excess, and so on. This process of successively optimizing the excess of the worst-off coalitions yields our primary object of interest, the *nucleolus*. For an allocation $x \in \mathbb{R}^V$, let $\theta(x) \in \mathbb{R}^{2^V - 2}$ be the vector obtained by sorting the list of excess values $x(S) - \nu(S)$ for any $\varnothing \neq S \subset V$ in non-decreasing order[1]. The *nucleolus*, denoted $\eta(V, \nu)$ and defined by Schmeidler [38], is the unique allocation that lexicographically maximizes $\theta(x)$:

$$\eta(V, \nu) := \arg \operatorname{lex} \max\{\theta(x) : x \in P(\varepsilon^*)\}.$$

We refer the reader to Appendix B for an example instance of the weighted matching game with its nucleolus. We now have sufficient terminology to state our main result:

Theorem 1. *Given a graph $G = (V, E)$ and weights $w : E \to \mathbb{R}$, the nucleolus $\eta(V, \nu)$ of the corresponding weighted matching game can be computed in polynomial time.*

Despite its intricate definition the concept of the nucleolus is surprisingly ancient. Its history can be traced back to a discussion on bankruptcy division in the Babylonian Talmud [1]. Modern research interest in the nucleolus stems not only from its geometric beauty [33], or several practical applications (e.g., see [5,32]), but from the strange way problems of computing the nucleolus fall in the complexity landscape, seeming to straddle the NP vs P boundary.

[1] It is common within the literature, for instance in [26], to exclude the coalitions for $S = \varnothing$ and $S = V$ in the definition of the nucleolus. On the other hand, one could also consider the definition of the nucleolus with all possible coalitions, including $S = \varnothing$ and $S = V$. We note that the two definitions of the nucleolus are equivalent in all instances of matching games except for the trivial instance of a graph consisting of two nodes joined by a single edge.

Beyond being one of the most fundamental problems in combinatorial optimization, starting with the founding work of Kuhn on the Hungarian method for the assignment problem [29], matching problems have historically teetered on the cusp of hardness. For example, prior to Edmonds' celebrated Blossom Algorithm [12,13] it was not clear whether Maximum Matching belonged in P. For another example, until Rothvoß' landmark result [37] it was thought that the matching polytope could potentially have sub-exponential extension complexity. In cooperative game theory, matchings live up to their historical pedigree of representing a challenging problem class. The long standing open problem in this area was whether the nucleolus of a weighted matching game instance can be computed in polynomial time. The concept of the nucleolus has been known since 1969 [38], and the question was posed as an important open problem in multiple papers. In 1998, Faigle, Kern, Fekete, and Hochstättler [15] mention the problem in their work on the nucleon, a multiplicative-error analog to the nucleolus which they show is polynomial time computable. Kern and Paulusma state the question of computing the nucleolus for general matching games as an important open problem in 2003 [26]. In 2008, Deng and Fang [9] conjectured this problem to be NP-hard, and in 2017 Biró, Kern, Paulusma, and Wojuteczky [4] reaffirmed this problem as an interesting open question. Theorem 1 settles the question, providing a polynomial-time algorithm to compute the nucleolus of a general instance of a weighted cooperative matching game.

Prior to our work, the nucleolus was known to be polynomial-time computable only in structured instances of the matching game. Solymosi and Raghavan [40] showed how to compute the nucleolus in an (unweighted) assignment game instance in polynomial time. Kern and Paulusma [26] later provided an efficient algorithm to compute the nucleolus in general unweighted matching game instances. Paulusma [35] extended the work in [26] and gave an efficient algorithm to compute the nucleolus in matching games where edge weights are induced by node potentials. Farczadi [20] finally extended Paulusma's framework further using the concept of *extendible allocations*. We note also that it is easy to compute the nucleolus in weighted instances of the matching game with non-empty core. For such instances, the leastcore has a simple compact description that does not include constraints for coalitions of size greater than 2. Thus it is relatively straightforward to adapt the iterative algorithm of Maschler [33] to a polynomial-time algorithm for computing the nucleolus (e.g., see [20, Chapter 2.3] for the details, Sect. 1.2 for an overview).

In a manner analogous to how we have defined matching games, a wide variety *combinatorial optimization games* can be defined [11]. In such games, the value of a coalition S of players is succinctly given as the optimal solution to an underlying combinatorial optimization problem. It is natural to conjecture that the complexity of computing the nucleolus in an instance of such a game would fall in lock-step with the complexity of the underlying problem. Surprisingly this is not the case. For instance, computing the nucleolus is known to be NP-hard for network flow games [10], weighted threshold games [14], and spanning tree games [16,19]. On the other hand, polynomial time algorithms are known for finding the nucleolus of special cases of flow games, certain classes of matching games, fractional matching games, and convex games [2,6,10,17,20,22,23,26,30,34–36,40].

The nucleolus is known to lie in the prekernel [38], a solution concept representing allocations which, speaking intuitively, reflect a balance of power between players. The prekernel of a cooperative game is known to be non-convex and even disconnected in general [28,41]. Despite this, Faigle, Kern and Kuipers [17] showed how to compute a point in the intersection of prekernel and leastcore in polynomial time under the reasonable assumption that the game has a polynomial time oracle to compute the minimum excess coalition for a given allocation. Later the same authors [18] refine their result to computing a point in the intersection of the core and lexicographic kernel, a set which is also known to contain the nucleolus. Bateni et al. [2] pose as an open question the existence of an efficiently computable, balanced and *unique* way of sharing profit in a network bargaining setting. The nucleolus is always unique [38], and balanced in the sense of lying in the leastcore intersect prekernel. Theorem 1 therefore resolves the latter open question left in [2].

1.1 Leastcore and Core of Matching Games

It is straightforward to see that (P) can be rewritten equivalently as

$$\begin{aligned} \max\ & \varepsilon & & & (P_1)\\ \text{s.t. } & x(M) \geq w(M) + \varepsilon & & \text{for all} \quad M \in \mathcal{M} \\ & x(V) = \nu(G) \\ & x \geq \mathbb{0}\,, \end{aligned}$$

where $\mathcal{M}$ is the set of all matchings M on G, and $x(M)$ is a shorthand for $x(V(M))$.

The separation problem for the linear program (P_1) can be reduced to finding a maximum weight matching in the graph G with edge weights $w(uv) - x(uv)$, $uv \in E$ (where we use $x(uv)$ as a shorthand for $x(u)+x(v)$). Since the maximum weight matching can be found in polynomial time [12], we know that the linear program (P_1) can be solved in polynomial time as well [25].

We use ε_1 to denote the optimal value of (P_1) and $P_1(\varepsilon_1)$ for the set of allocations x such that (x, ε_1) is feasible for the leastcore linear program (P_1). In general, for a value ε and a linear program Q on variables in $\mathbb{R}^V \times \mathbb{R}$ we denote by $Q(\varepsilon)$ the set $\{x \in \mathbb{R}^V : (x, \varepsilon) \text{ is feasible for } Q\}$.

Note that $\varepsilon_1 \leq 0$. Indeed, $\varepsilon \leq 0$ in any feasible solution (x, ε) to (P_1) as otherwise $x(M)$ would need to exceed $w(M)$ for all matchings M. In particular this would also hold for a maximum weight matching on G, implying that $x(V) > \nu(G)$. If $\varepsilon_1 = 0$ then the core of the cooperative matching game is non-empty. One can see that $\varepsilon_1 = 0$ if and only if the value of a maximum weight matching on G with weights w equals the value of a maximum weight fractional matching. This follows since $x \in P_1(\varepsilon_1)$ is a fractional weighted node cover of value $\nu(G)$ when $\varepsilon_1 = 0$. When $\varepsilon_1 < 0$, we say that the cooperative matching game has an empty core. The matching game instance given in Appendix B has an empty core.

In this paper, we assume that the cooperative matching game (G, w) has an empty core, as computing the nucleolus is otherwise well-known to be doable in polynomial time [35].

1.2 Maschler's Scheme

As discussed, our approach to proving Theorem 1 relies on Maschler's scheme. The scheme requires us to solve a linear number of LPs: $\{(P_j)\}_{j\geq 1}$ that we now define. (P_1) is the leastcore LP that we have already seen in Sect. 1.1. LPs (P_j) for $j \geq 2$ are defined inductively. Crucial in their definition is the notion of *fixed* coalitions that we introduce first. For a polyhedron $Q \subseteq \mathbb{R}^V$ we denote by $\mathrm{Fix}(Q)$ the collection of sets $S \subseteq V$ such that $x(S)$ is constant over the polyhedron Q, i.e.

$$\mathrm{Fix}(Q) := \{S \subseteq V \ : \quad x(S) = x'(S) \quad \text{for all} \quad x, x' \in Q\}\,.$$

With this we are now ready to state LP (P_j) for $j \geq 2$:

$$\begin{aligned} \max \quad & \varepsilon && && (P_j)\\ \text{s.t.} \quad & x(S) - \nu(S) \geq \varepsilon && \text{for all} \quad S \subset V,\, S \notin \mathrm{Fix}(P_{j-1}(\varepsilon_{j-1})) \\ & \qquad\qquad x \in P_{j-1}(\varepsilon_{j-1})\,, \end{aligned}$$

where ε_j be the optimal value of the linear program (P_j). Let j^* be the minimum number j such that $P_j(\varepsilon_j)$ contains a single point. This point is the nucleolus of the game [8]. It is well-known [33] that $P_{j-1}(\varepsilon_{j-1}) \subset P_j(\epsilon_j)$ and $\varepsilon_{j-1} < \varepsilon_j$ for all j. Since the dimension of the polytope describing feasible solutions of (P_j) decreases in every round until the dimension becomes zero, we have $j^* \leq |V|$ [33], [35, Pages 20–24].

Therefore, in order to find the nucleolus of the cooperative matching game efficiently it suffices to solve each linear program (P_j), $j = 1, \ldots, j^*$ in polynomial time. We accomplish this by providing polynomial-size formulations for (P_j) for all $j \geq 1$.

In Sect. 2 we introduce the concept of *universal matchings* which are fundamental to our approach, and give a compact formulation for the first linear program in Maschler's Scheme, the leastcore. We also present our main technical lemma, Lemma 5, which provides a crucial symmetry condition on the values allocations can take over the vertices of blossoms in the graph decomposition we use to describe the compact formulation. In Sect. 3 we describe the successive linear programs in Maschler's Scheme and provide a compact formulation for each one in a matching game.

2 Leastcore Formulation

In this section we provide a polynomial-size description of (P_1). It will be useful to define a notation for *excess*: for any $x \in P_1(\varepsilon_1)$ and $M \in \mathcal{M}$ let $\mathrm{excess}(x, M) := x(M) - w(M)$.

2.1 Universal Matchings, Universal Allocations

For each $x \in P_1(\varepsilon_1)$ we say that a matching $M \in \mathcal{M}$ is an x-tight matching whenever $\text{excess}(x, M) = \varepsilon_1$. We denote by $\mathcal{M}^x$ the set of x-tight matchings.

A *universal matching* $M \in \mathcal{M}$ is a matching which is x-tight for all $x \in P_1(\varepsilon_1)$. We denote the set of universal matchings on G by $\mathcal{M}_{uni}$. A *universal allocation* $x^* \in P_1(\varepsilon_1)$ is a leastcore point whose x^*-tight matchings are precisely the set of universal matchings, i.e. $\mathcal{M}^{x^*} = \mathcal{M}_{uni}$.

Lemma 1. *There exists a universal allocation $x^* \in P_1(\varepsilon_1)$.*

Proof. Indeed, it is straightforward to show that every x^* in the relative interior (see [42, Lemma 2.9(ii)]) of $P_1(\varepsilon_1)$ is a universal allocation. If the relative interior is empty then $P_1(\varepsilon_1)$ is a singleton, which trivially contains a universal allocation. In the arxiv version [27] we provide a combinatorial proof. ■

Lemma 2. *A universal allocation $x^* \in P_1(\varepsilon_1)$ can be computed in polynomial time.*

Proof. A point x^* in the relative interior of $P_1(\varepsilon_1)$ can be found in polynomial time using the ellipsoid method (Theorem 6.5.5 [24], [7]). Since any allocation x^* from the relative interior of $P_1(\varepsilon_1)$ is a universal allocation, this implies the statement of the lemma. ■

Given a non-universal allocation x and a universal allocation x^*, we observe that $\mathcal{M}^{x^*} \subset \mathcal{M}^x$ and so $\theta(x^*)$ is strictly lexicographically greater than $\theta(x)$. Thus the nucleolus is a universal allocation. We emphasize that $\mathcal{M}^{x^*} = \mathcal{M}_{uni}$ is invariant under the (not necessarily unique) choice of universal allocations x^*. Henceforth we fix a universal allocation $x^* \in P_1(\varepsilon_1)$.

2.2 Description for Convex Hull of Universal Matchings.

By the definition of universal allocation x^*, a matching M is universal if and only if it is x^*-tight. Thus, M is a universal matching if and only if its characteristic vector lies in the optimal face of the matching polytope corresponding to (the maximization of) the linear objective function assigning weight $-\text{excess}(x^*, uv) = w(uv) - x^*(uv)$ to each edge $uv \in E$. Let $\mathcal{O}$ be the set of node sets $S \subseteq V$ such that $|S| \geq 3$, $|S|$ is odd. Edmonds [12] gave a linear description of the matching polytope of G as the set of $y \in \mathbb{R}^E$ satisfying:

$$\begin{aligned} y(\delta(v)) &\leq 1 && \text{for all} \quad v \in V \\ y(E(S)) &\leq (|S|-1)/2 && \text{for all} \quad S \in \mathcal{O} \\ y &\geq 0\,. \end{aligned}$$

Thus, a matching $M \in \mathcal{M}$ is universal if and only if it satisfies the constraints

$$\begin{aligned} M \cap \delta(v) &= 1 && \text{for all} \quad v \in W \\ M \cap E(S) &= (|S|-1)/2 && \text{for all} \quad S \in \mathcal{L} \\ M \cap \{e\} &= 0 && \text{for all} \quad e \in F\,, \end{aligned} \tag{1}$$

where W is some subset of V, $\mathcal{L}$ is a subset of $\mathcal{O}$, and F is a subset of E. Using an uncrossing argument, as in [31, Pages 141–150], we may assume that the collection of sets $\mathcal{L}$ is a laminar family of node sets; i.e., for any two distinct sets $S, T \in \mathcal{L}$, either $S \cap T = \varnothing$ or $S \subseteq T$ or $T \subseteq S$.

Lemma 3. *For every node $v \in V$ there exists $M \in \mathcal{M}_{uni}$ such that v is exposed by M. Hence, $W = \varnothing$.*

Proof. Assume for a contradiction that there exists a node $v \in V$ such that $v \in W$.

First, note that there always exists a non-universal matching $M \in \mathcal{M} \setminus \mathcal{M}_{uni}$ since otherwise the empty matching would be universal, and thus

$$0 = x^*(\varnothing) = w(\varnothing) + \varepsilon_1,$$

implying that the core of the given matching game instance is non-empty.

Suppose first that there exists a node $u \in V$ exposed by some matching $M' \in \mathcal{M}_{uni}$ such that $x^*_u > 0$. We define

$$\delta_0 := \min\{\text{excess}(x^*, M) - \varepsilon_1 \ : \quad M \in \mathcal{M} \setminus \mathcal{M}_{uni}\}.$$

Recall that $\mathcal{M}_{uni}$ is the set of maximum weight matchings on G with respect to the node weights $w(uv) - x^*(uv)$, $uv \in E$, i.e. $\mathcal{M}_{uni}$ is the set of x^*-tight matchings. Moreover, recall that $\text{excess}(x^*, M) = \varepsilon_1$ for $M \in \mathcal{M}_{uni}$. Thus, we have $\delta_0 > 0$.

We define $\delta := \min\{\delta_0, x^*_u\} > 0$ and a new allocation x' as follows:

$$x'_r := \begin{cases} x^*_r + \delta, & \text{if } r = v \\ x^*_r - \delta, & \text{if } r = u \\ x^*_r, & \text{otherwise.} \end{cases}$$

Since all universal matchings contain v, the excess with respect to x' of any universal matching is no smaller than their excess with respect to x^*. Therefore, by our choice of δ, (x', ϵ_1) is a feasible, and hence optimal, solution for (P_1). But M' is not x'-tight, since M' covers v and exposes u. This contradicts that M' is a universal matching.

Now consider the other case: for all $u \in V$ if u is exposed by a universal matching then $x^*_u = 0$. Then, for every universal matching $M \in \mathcal{M}_{uni}$ we have

$$\varepsilon_1 = \text{excess}(x^*, M) = x^*(V) - w(M) = \nu(G) - w(M).$$

Since $\nu(G)$ is the maximum weight of a matching on G with respect to the weights w, we get that $\varepsilon_1 \geq 0$. Thus x^* is in the core, contradicting our assumption that the core is empty. ■

2.3 Description of Leastcore

We denote inclusion-wise maximal sets in the family $\mathcal{L}$ as $S_1^*, S_2^*, \ldots, S_k^*$. We define the edge set E^+ to be the set of edges in G such that at most one of its nodes is in S_i^* for every $i \in [k] := \{1, \ldots, k\}$, i.e.

$$E^+ := E \setminus \big(\bigcup_{i=1}^{k} E(S_i^*)\big).$$

Lemma 4. *For every choice of $v_i \in S_i^*$, $i \in [k]$, there exists a universal matching $M \in \mathcal{M}_{uni}$ such that the node set covered by M is as follows*

$$\bigcup_{i=1}^{k} S_i^* \backslash \{v_i\}\,.$$

Proof. By Lemma 3, we know that for every $i \in [k]$ there exists a universal matching $M_{v_i} \in \mathcal{M}_{uni}$ such that v_i is exposed by M_{v_i}. Now, for every $i \in [k]$, let us define

$$M_i := E(S_i^*) \cap M_{v_i}.$$

Since M_i satisfies all laminar family constraints in $\mathcal{L}$ for subsets of S_i^* we have that

$$\bigcup_{i=1}^{k} M_i$$

is a matching satisfying all the constraints (1), and hence is a universal matching covering the desired nodes. ■

For each $i \in [k]$ fix a unique *representative* node $v_i^* \in S_i^*$. By Lemma 4, there exists a universal matching M^* covering precisely $\bigcup_{i \in [k]} S_i^* \backslash \{v_i^*\}$. For any $x \in P_1(\varepsilon_1)$ and $S \subseteq V$ we use $\text{diff}(x, S)$ to denote

$$\text{diff}(x, S) := x(S) - x^*(S)\,.$$

For single nodes we use the shorthand $\text{diff}(x, v) = \text{diff}(x, \{v\})$. We now prove the following crucial structure result on allocations in the leastcore.

Lemma 5. *For every leastcore allocation x, i.e. for every $x \in P_1(\varepsilon_1)$, we have that*

(i) for all $i \in [k]$, for all $u \in S_i^*$: $\quad \text{diff}(x, u) = \text{diff}(x, v_i^*)$,
(ii) for all $e \in E^+$: $\quad \text{excess}(x, e) \geq 0$.

Proof. Consider $u \in S_i^*$, and note that we may use Lemma 4 to choose a universal matching M_u covering precisely

$$S_i^* \backslash \{u\} \cup \bigcup_{j \neq i} S_j^* \backslash \{v_j^*\}.$$

Hence we have $V(M_u) \cup \{u\} = V(M^*) \cup \{v_i^*\}$, and since M^* and M_u are universal, $x(M^*) = x^*(M^*)$ and $x(M_u) = x^*(M_u)$. Using these observations we see that

$$\begin{aligned} sym(x,u) &= x(u) + x(M_u) - (x^*(u) - x^*(M_u)) \\ &= x(v_i^*) + x(M^*) - (x^*(v_i^*) - x^*(M^*)) = \text{diff}(x, v_i^*). \end{aligned}$$

showing (i).

Now we prove (ii). Consider $e \in E^+$ where $e = \{u, v\}$. Since $e \notin E(S_i^*)$ for all $i \in [k]$, we can choose a universal matching M exposing u and v by Lemma 4. Thus $M \cup \{e\}$ is also a matching. Notice M is x-tight, and so we have

$$\text{excess}(x,e) = \underbrace{\text{excess}(x, M \cup \{e\})}_{\geq \varepsilon_1} - \underbrace{\text{excess}(x,M)}_{=\varepsilon_1} \geq 0$$

as desired. ■

Lemma 6. *Let $x \in P_1(\varepsilon_1)$ and let $M \in \mathcal{M}$ be a matching such that $M \subseteq \bigcup_{i \in [k]} E(S_i^*)$. Then there exists $M' \subseteq M^*$ such that* excess$(x, M') \leq$ excess(x, M) *and for all $i \in [k]$, $|M' \cap E(S_i^*)| = |M \cap E(S_i^*)|$.*

Proof. See the arxiv version [27]. ■

Recall that x^* is a fixed universal allocation in $P_1(\varepsilon_1)$. Let $E^* \subseteq E$ denote the union of universal matchings, i.e. $E^* = \cup_{M \in \mathcal{M}_{uni}} M$. We now define linear program ($\overline{P}_1$).

$$\max \quad \varepsilon \qquad (\overline{P}_1)$$

$$\begin{aligned} \text{s.t.} \quad & \text{diff}(x,u) = \text{diff}(x, v_i^*) && \text{for all} \quad u \in S_i^*,\ i \in [k] \qquad (2) \\ & \text{excess}(x,e) \leq 0 && \text{for all} \quad e \in E^* \\ & \text{excess}(x,e) \geq 0 && \text{for all} \quad e \in E^+ \\ & \text{excess}(x, M^*) = \varepsilon \\ & x(V) = \nu(G) \\ & x \geq 0\,. \end{aligned}$$

Let $\overline{\varepsilon}_1$ be the optimal value of the linear program ($\overline{P}_1$). We now show that $\overline{P_1}(\overline{\varepsilon}_1)$ is indeed a compact description of the leastcore $P_1(\varepsilon_1)$.

Theorem 2. *We have $\varepsilon_1 = \overline{\varepsilon}_1$ and $P_1(\varepsilon_1) = \overline{P}_1(\overline{\varepsilon}_1)$.*

Proof. See Appendix A.

3 Computing the Nucleolus

The last section presented a polynomial-size formulation for the leastcore LP (P_1). In this section we complete our polynomial-time implementation of Maschler's scheme by showing that (P_j) has the following compact reformulation:

$$\begin{array}{lrll} \max & \varepsilon & & (\overline{P}_j) \\ \text{s.t.} & \text{excess}(x,e) \geq \varepsilon - \varepsilon_1 & \text{for all} & e \in E^+,\ e \notin \text{Fix}(\overline{P}_{j-1}(\overline{\varepsilon}_{j-1})) \\ & x(v) \geq \varepsilon - \varepsilon_1 & \text{for all} & v \in V,\ v \notin \text{Fix}(\overline{P}_{j-1}(\overline{\varepsilon}_{j-1})) \\ & \text{excess}(x,e) \leq \varepsilon_1 - \varepsilon & \text{for all} & e \in E^*,\ e \notin \text{Fix}(\overline{P}_{j-1}(\overline{\varepsilon}_{j-1})) \\ & x \in \overline{P}_{j-1}(\overline{\varepsilon}_{j-1})\,, & & \end{array}$$

where $\overline{\varepsilon}_j$ is the optimal value of the linear program $(\overline{P}_j)$.

Theorem 3. *For all $j = 1, \ldots, j^*$, we have $\varepsilon_j = \overline{\varepsilon}_j$ and $P_j(\varepsilon_j) = \overline{P}_j(\overline{\varepsilon}_j)$.*

Proof. We refer the reading to the arxiv version [27] for the proof.

With Theorem 3 we can replace each linear program (P_j) with $(\overline{P}_j)$ in Maschler's Scheme. Since the universal allocation x^*, the node sets S_i^*, $i \in [k]$, the edge set E^+, and the edge set E^* can all be computed in polynomial time, we have shown that the nucleolus of any cooperative matching game with empty core can be computed in polynomial time. Therefore we have shown Theorem 1.

Open Questions. Matching Games generalize naturally to b-matching games, where instead the underlying optimization problem is to find an edge subset M with $|M \cap \delta(v)| \leq b_v$ for each node v. Biro, Kern, Paulusma, and Wojuteczky [4] showed that the core-separation problem when $b_v > 2$ for some vertex v, is coNP-Hard. Despite this, the complexity of computing the nucleolus of these games is open.

Our algorithm relies heavily on the ellipsoid method. When the core is non-empty, there is a combinatorial algorithm for finding the nucleolus [3]. It would be interesting to develop a combinatorial algorithm for nucleolus computation of matching games in general.

Acknowledgements. The authors thank Umang Bhaskar, Daniel Dadush, and Linda Farczadi for stimulating and insightful discussions related to this paper.

Appendix

A Proof of Theorem 2

Proof. First, we show that $P_1(\varepsilon_1) \subseteq \overline{P}_1(\varepsilon_1)$. Consider $x \in P_1(\varepsilon_1)$. By Lemma 5(i) we have

$$\text{diff}(x,u) = \text{diff}(x,v_i^*) \qquad \text{for all} \quad u \in S_i^*,\ i \in [k].$$

Lemma 5(ii) shows that $\text{excess}(x, e) \geq 0$ for all $e \in E^+$, and $\text{excess}(x, M^*) = \varepsilon_1$ holds by the universality of M^*. It remains to show that

$$\text{excess}(x, e) \leq 0 \quad \text{for all} \quad e \in E^* .$$

Suppose for contradiction there exists $e \in E^*$ such that $\text{excess}(x, e) > 0$. By the definition of E^*, there exists a universal matching M' containing e. Since M' is universal, $\text{excess}(x, M') = \varepsilon_1$. But by our choice of e,

$$\text{excess}(x, M' \setminus \{e\}) < \text{excess}(x, M') = \varepsilon_1$$

contradicting that x is in $P_1(\varepsilon_1)$. Thus we showed that (x, ε_1) is feasible for $(\overline{P}_1)$, i.e. we showed that $P_1(\varepsilon_1) \subseteq \overline{P}_1(\varepsilon_1)$.

To complete the proof we show that $\overline{P}_1(\overline{\varepsilon}_1) \subseteq P_1(\overline{\varepsilon}_1)$. Let x be an allocation in $\overline{P}_1(\overline{\varepsilon}_1)$. Due to the description of the linear program (P_1), it is enough to show that for every matching $M \in \mathcal{M}$ we have

$$\text{excess}(x, M) \geq \overline{\varepsilon}_1 .$$

Since $\text{excess}(x, e) \geq 0$ for all$e \in E^+$, it suffices to consider only the matchings M, which are unions of matchings on the graphs $G[S_i^*]$, $i \in [k]$. Let $t_i := |M \cap E(S_i^*)|$. By Lemma 6 applied to x^* there exists $M' \subseteq M^*$ such that $\text{excess}(x^*, M) \geq \text{excess}(x^*, M')$ and $|M' \cap E(S_i^*)| = t_i$, for all $i \in [k]$. Then due to constraints (2) in $(\overline{P}_1)$ we have

$$\begin{aligned}\text{excess}(x, M) &= \underbrace{\sum_{i=1}^{k} 2t_i \,\text{diff}(x, v_i^*)}_{=\text{diff}(x, M')} + \underbrace{\text{excess}(x^*, M)}_{\geq \text{excess}(x^*, M')} \\ &\geq \text{excess}(x, M') \geq \text{excess}(x, M^*) = \overline{\varepsilon}_1 ,\end{aligned}$$

where the last inequality follows since $M' \subseteq M^*$ and $\text{excess}(x, e) \leq 0$ for all $e \in E^*$.

Thus, we showed that $P_1(\varepsilon_1) \subseteq \overline{P}_1(\varepsilon_1)$ and $\overline{P}_1(\overline{\varepsilon}_1) \subseteq P_1(\overline{\varepsilon}_1)$. Recall, that ε_1 and $\overline{\varepsilon}_1$ are the optimal values of the linear programs (P_1) and $(\overline{P}_1)$ respectively. Thus, we have $\varepsilon_1 = \overline{\varepsilon}_1$ and $P_1(\varepsilon_1) = \overline{P}_1(\overline{\varepsilon}_1)$. ■

B Example of a Matching Game With Empty Core

Consider the example in Fig. 1. This graph $G = (V, E)$ is a 5-cycle with two adjacent edges 15 and 12 of weight 2, and the remaining three edges of weight 1. Since the maximum weight matching value is $\nu(G) = 3$, but the maximum weight fractional matching value is $\frac{7}{2}$, the core of this game is empty. The allocation x^* defined by $x^*(1) = \frac{7}{5}$ and $x^*(2) = x^*(3) = x^*(4) = x^*(5) = \frac{2}{5}$ lies in the leastcore. Each edge has the same excess, $-\frac{1}{5}$, and any coalition of four vertices yields a minimum excess coalition with excess $-\frac{2}{5}$. Hence the leastcore value of this game is $\varepsilon_1 = -\frac{2}{5}$.

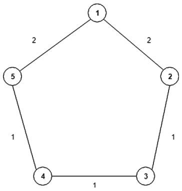

Fig. 1. Matching game with empty core

In fact, we can see that x^* is the nucleolus of this game. To certify this we can use the result of Schmeidler [38] that the nucleolus lies in the intersection of the leastcore and the prekernel. For this example, the prekernel condition that for all $i \neq j \in V$,

$$\begin{aligned} &\max\{x(S \cup \{i\}) - \nu(S \cup \{i\}) : S \subseteq V\backslash\{j\}\} \\ = &\max\{x(S \cup \{j\}) - \nu(S \cup \{j\}) : S \subseteq V\backslash\{i\}\} \end{aligned}$$

reduces to the condition that the excess values of non-adjacent edges are equal. Since G is an odd cycle, this implies that all edges has equal excess, i.e.

$$\begin{aligned} \text{excess}(x, 12) = \text{excess}(x, 23) &= \text{excess}(x, 34) \\ &= \text{excess}(x, 45) = \text{excess}(x, 15). \end{aligned}$$

Combining the four equations above with the leastcore condition that $x(V) = \nu(G)$ we obtain a system of equations with the unique solution x^*. Hence the intersection of the leastcore and prekernel is precisely $\{x^*\}$, and so by Schmeidler, x^* is the nucleolus.

References

1. Aumann, R.J., Maschler, M.: Game theoretic analysis of a bankruptcy problem from the talmud. J. Econ. Theory **36**(2), 195–213 (1985)
2. Bateni, M.H., Hajiaghayi, M.T., Immorlica, N., Mahini, H.: The cooperative game theory foundations of network bargaining games. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6198, pp. 67–78. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14165-2_7
3. Biró, P., Kern, W., Paulusma, D.: Computing solutions for matching games. Int. J. Game Theory **41**, 75–90 (2012)
4. Biró, P., Kern, W., Paulusma, D., Wojuteczky, P.: The stable fixtures problem with payments. Games Econ. Behav. **11**(9), 245–268 (2017)
5. Brânzei, R., Solymosi, T., Tijs, S.: Strongly essential coalitions and the nucleolus of peer group games. Int. J. Game Theory **33**(3), 447–460 (2005)
6. Chen, N., Lu, P., Zhang, H.: Computing the nucleolus of matching, cover and clique games. In: AAAI (2012)
7. Dadush, D.: Personal communication (2017)
8. Davis, M., Maschler, M.: The kernel of a cooperative game. Naval Res. Logist. Q. **12**(3), 223–259 (1965)
9. Deng, X., Fang, Q.: Algorithmic cooperative game theory. In: Chinchuluun, A., Pardalos, P.M., Migdalas, A., Pitsoulis, L. (eds.) Pareto Optimality, Game Theory and Equilibria. SOIA, vol. 17, pp. 159–185. Springer, New York (2008). https://doi.org/10.1007/978-0-387-77247-9_7

10. Deng, X., Fang, Q., Sun, X.: Finding nucleolus of flow game. J. Comb. Optim. **18**(1), 64–86 (2009)
11. Deng, X., Ibaraki, T., Nagamochi, H.: Algorithmic aspects of the core of combinatorial optimization games. Math. Oper. Res. **24**(3), 751–766 (1999)
12. Edmonds, J.: Maximum matching and a polyhedron with 0, 1-vertices. J. Res. Natl. Bur. Stan. B **69**(125–130), 55–56 (1965)
13. Edmonds, J.: Paths, trees, and flowers. Can. J. Math. **17**(3), 449–467 (1965)
14. Elkind, E., Goldberg, L.A., Goldberg, P., Wooldridge, M.: Computational complexity of weighted threshold games. In: Proceedings of the National Conference on Artificial Intelligence, p. 718 (2007)
15. Faigle, U., Kern, W., Fekete, S.P., Hochstättler, W.: The nucleon of cooperative games and an algorithm for matching games. Math. Program. **83**(1–3), 195–211 (1998)
16. Faigle, U., Kern, W., Kuipers, J.: Note computing the nucleolus of min-cost spanning tree games is NP-hard. Int. J. Game Theory **27**(3), 443–450 (1998)
17. Faigle, U., Kern, W., Kuipers, J.: On the computation of the nucleolus of a cooperative game. Int. J. Game Theory **30**(1), 79–98 (2001)
18. Faigle, U., Kern, W., Kuipers, J.: Computing an element in the lexicographic kernel of a game. Math. Methods Oper. Res. **63**(3), 427–433 (2006)
19. Faigle, U., Kern, W., Paulusma, D.: Note on the computational complexity of least core concepts for min-cost spanning tree games. Math. Methods Oper. Res. **52**(1), 23–38 (2000)
20. Farczadi, L.: Matchings and games on networks. University of Waterloo (2015)
21. Gillies, D.B.: Solutions to general non-zero-sum games. Contrib. Theory Games **4**(40), 47–85 (1959)
22. Granot, D., Granot, F., Zhu, W.R.: Characterization sets for the nucleolus. Int. J. Game Theory **27**(3), 359–374 (1998)
23. Granot, D., Maschler, M., Owen, G., Zhu, W.R.: The kernel/nucleolus of a standard tree game. Int. J. Game Theory **25**(2), 219–244 (1996)
24. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, Algorithms and Combinatorics: Study and Research Texts, vol. 2. Springer, Heidelberg (1988). https://doi.org/10.1007/978-3-642-97881-4
25. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, vol. 2. Springer, New York (2012)
26. Kern, W., Paulusma, D.: Matching games: the least core and the nucleolus. Math. Oper. Res. **28**(2), 294–308 (2003)
27. Koenemann, J., Pashkovich, K., Toth, J.: Computing the nucleolus of weighted cooperative matching games in polynomial time. arXiv preprint arXiv:1803.03249 (2018)
28. Kopelowitz, A.: Computation of the kernels of simple games and the nucleolus of n-person games. Technical report, Hebrew Univ Jerusalem (Israel) Dept of Mathematics (1967)
29. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)
30. Kuipers, J., Solymosi, T., Aarts, H.: Computing the nucleolus of some combinatorially-structured games. Math. Program. **88**(3), 541–563 (2000)
31. Lau, L.C., Ravi, R., Singh, M.: Iterative Methods in Combinatorial Optimization, vol. 46. Cambridge University Press, Cambridge (2011)
32. Lemaire, J.: An application of game theory: cost allocation. ASTIN Bull. J. IAA **14**(1), 61–81 (1984)

33. Maschler, M., Peleg, B., Shapley, L.S.: Geometric properties of the kernel, nucleolus, and related solution concepts. Math. Oper. Res. **4**(4), 303–338 (1979)
34. Megiddo, N.: Computational complexity of the game theory approach to cost allocation for a tree. Math. Oper. Res. **3**(3), 189–196 (1978)
35. Paulusma, D.: Complexity Aspects of Cooperative Games. Twente University Press, Enschede (2001)
36. Potters, J., Reijnierse, H., Biswas, A.: The nucleolus of balanced simple flow networks. Games Econ. Behav. **54**(1), 205–225 (2006)
37. Rothvoß, T.: The matching polytope has exponential extension complexity. J. ACM (JACM) **64**(6), 41 (2017)
38. Schmeidler, D.: The nucleolus of a characteristic function game. SIAM J. Appl. Math. **17**(6), 1163–1170 (1969)
39. Shapley, L.S., Shubik, M.: The assignment game I: the core. Int. J. Game Theory **1**(1), 111–130 (1971)
40. Solymosi, T., Raghavan, T.E.: An algorithm for finding the nucleolus of assignment games. Int. J. Game Theory **23**(2), 119–143 (1994)
41. Stearns, R.E.: Convergent transfer schemes for n-person games. Trans. Am. Math. Soc. **134**(3), 449–459 (1968)
42. Ziegler, G.M.: Lectures on Polytopes, vol. 152. Springer, New York (2012)