Shapley meets Shapley

Haris Aziz[1] and Bart de Keijzer[2]

1   **NICTA and University of New South Wales**
    Sydney, NSW 2033, Australia
    `haris.aziz@nicta.com.au`
2   **Centrum Wiskunde & Informatica (CWI)**
    1098 XG, Amsterdam, The Netherlands
    `keijzer@cwi.nl`

─── **Abstract** ───────────────────────────────

This paper concerns the analysis of the *Shapley value* in *matching games*. Matching games constitute a fundamental class of cooperative games which help understand and model auctions and assignments. In a matching game, the value of a coalition of vertices is the weight of the maximum size matching in the subgraph induced by the coalition. The Shapley value is one of the most important solution concepts in cooperative game theory. After establishing some general insights, we show that the Shapley value of matching games can be computed in polynomial time for some special cases: graphs with maximum degree two, and graphs that have a small modular decomposition into cliques or cocliques (complete k-partite graphs are a notable special case of this). The latter result extends to various other well-known classes of graph-based cooperative games. We continue by showing that computing the Shapley value of unweighted matching games is #P-complete in general. Finally, a fully polynomial-time randomized approximation scheme (FPRAS) is presented. This FPRAS can be considered the best positive result conceivable, in view of the #P-completeness result.

1   Introduction

In economics and computer science, one of the most fundamental problems is the allocation of profits or costs based on contributions of the nodes in a network. The problem has assumed even more importance as networks have become ubiquitous. In this paper, we address this problem by simultaneously studying two concepts that can be traced to Lloyd S. Shapley — the *Shapley value* and *matching games*.

Lloyd S. Shapley is one of the most influential game theorists in history. Among his numerous contributions, two of them are the following: (i) formulating the *assignment game* as a rich and versatile class of cooperative games [25], and (ii) proposing the *Shapley value* as a highly desirable solution concept for cooperative games [24]. Both contributions have had far-reaching impact and were part of Shapley's Nobel Prize winning achievements. The assignment game is a cooperative game based on bipartite graphs, and models the interaction between buyers and sellers. It is the *transferable utility* version of the well-known stable marriage setting and is a fundamental model that is used for modelling exchange markets and auctions [23]. Assignment games were later generalized to *matching games*, for *non*-bipartite graphs (see e.g., [11, 17]). The main idea of a matching game is that each node represents an agent and the value of a coalition of nodes is the weight of the maximum weight matching in the subgraph induced by the coalition of nodes. Whereas the matching game is

one of the most natural and important cooperatives game, the Shapley value has been termed "the most important normative payoff division scheme" in cooperative game theory [28]. It is based on the idea that the payoff of an agent should be proportional to his marginal contributions to the payoff for the set of all players. For an excellent overview of the concept, we refer the reader to [Chapter 5, 21]. The Shapley value is the only solution concept that satisfies simultaneously the following properties: efficiency, symmetry, additivity, and dummy player property.

In this paper we address a gap in the computational cooperative game theory literature, and we initiate research on the computational aspects of the Shapley value in matching games. This gap is surprising on two fronts: (i) computational aspects of Shapley values have been extensively studied for a number of cooperative games (see e.g., [12, 15, 14]) and (ii) matching games are a well-established class of cooperative games, and the structure and computational complexity of computing important solution concepts such as the core, least core, and nucleolus have been examined in-depth for matching games (see e.g., [1, 26, 17, 10, 6, 5]).

Our results. We study the algorithmic aspects and computational complexity of the Shapley value for matching games for the first time. We establish first some general insights and some particular special cases for which the exact Shapley value can be computed in polynomial time for: graphs with a constant size decomposition into clique and coclique modules (these include e.g., complete k-partite graphs, for k constant), and for graphs with maximum degree two. The non-trivial algorithm required for graphs of maximum degree two illustrates that exact computation of the Shapley value quickly becomes rather complex, even for very simple graph classes. We then move on to the central results of this paper, which concerns the general problem: we prove that the computational complexity of computing the Shapley value of matching games is #P-complete even if the graph is unweighted. The proof relies on Berge's Lemma and the fact that a certain matrix related to the Pascal triangle has a non-zero determinant. We subsequently present an *FPRAS* (i.e., a *fully polynomial time randomized approximation scheme*) for computing the Shapley value of (weighted) matching games. In view of our #P-completeness result, the FPRAS is the best possible result we can hope for.

Related work. The complexity of computing the Shapley value of important classes of co-operative games has been the topic of detailed studies. The papers [12] and [15] present polynomial-time algorithms to compute the Shapley value of *graph games* and *marginal contribution nets* respectively. On the other hand, computing the Shapley value is known to be intractable for a number of cooperative games (see e.g., [14, 2]).

Among the classes of cooperative games, matching games are one of the most well-studied. The core of matching games is characterized in [11], where it is also shown that various computational problems regarding the core and the least core of matching games can be solved in polynomial time. For matching games, there has been considerable algorithmic research on the *nucleolus*: an alternative single valued solution concept (see e.g., [26, 17]).

As networks analysis becomes an increasingly important area, centrality indices of graphs have received immense interest (see e.g., [7]). The idea is to get a ranking of vertices according to their ability to connect with other vertices. Recently, a Shapley-values based game theoretic approach has been used to gauge the centrality or connectivity of vertices by representing different valuation functions with a graph [see e.g., 20]. The motivation is that the Shapley value of a vertex captures various synergies which standard centrality measures do not. In this vein, Shapley values of the matching game constitutes an interesting method of gauging centrality/connectivity of the vertices. In particular it quantifies in a principled manner the ability of a vertex to match with other vertices to increase the value of the coalition.

2   Preliminaries

We work throughout this text with undirected weighted graphs $G = (N, E, w)$, where N is the vertex set, E is the edge set, and $w : E \to \mathbb{R}_{\geq 0}$ is a weight function. For $S \subseteq N$, we denote by $G(S)$ the subgraph of G induced by S, i.e., the graph $(S, \{e \in E : e \in S \times S\})$. We assume for the remainder of this text that the reader is familiar with basic notions related to graphs and matchings.

A *cooperative game* consists of a set N of $n = |N|$ players and a characteristic function $v : 2^N \to \mathbb{R}$ associating a value $v(S)$ to every subset $S \subseteq N$. A subset of N is referred to as a *coalition* in this context. Deciding how to distribute the value $v(N)$ among the players in a fair and stable manner is an objective of central importance in the research area of cooperative games.

A *matching game* is a cooperative game (N, v) induced by an undirected weighted graph $G = (N, E, w)$ (with vertex set N, edge set E, and weight function $w : E \to \mathbb{R}_{\geq 0}$) such that for any $S \subseteq N$, $v(S)$ is the weight of a maximum weight matching of the subgraph $G(S)$. For a given graph G, we will denote by $MG(G)$ the matching game corresponding to graph G.

An *unweighted* matching game is a matching game for which all weights are 1 in the associated graph. In unweighted matching games, it holds that $v(S \cup \{i\}) - v(S) \in \{0, 1\}$ for all $S \subset N, i \in N \backslash S$. If, for an unweighted matching game (N, v), a player $i \in N$, and a coalition $S \subseteq N \backslash \{i\}$, it holds that $v(S \cup \{i\}) = v(S) + 1$, then we say that player i is *pivotal* (for coalition S, in game (N, v)). Similarly, if $\sigma : N \to N$ is a permutation on N, and i is pivotal for set of players $p(i, \sigma) = \{j : \sigma^{-1}(j) < \sigma^{-1}(i)\}$ (i.e., the players occurring before i in σ), then we say that σ is pivotal for i.

For the general case of weighted matching games, when S is a coalition not containing player i, we refer to the value $v(S \cup \{i\}) - v(S)$ as *the marginal contribution of i to S*. When σ is a permutation of N, we refer to the value $v(p(i, \sigma) \cup \{i\}) - v(p(i, \sigma))$ as *the marginal contribution of i to σ*.

The *Shapley value* of a player $i \in N$ in a cooperative game (N, v) is denoted by $\varphi_i(N, v)$, and is defined as follows.

$$\varphi_i(N, v) = \kappa_i(N, v)/n!, \qquad \kappa_i(N, v) = \sum_{S \subseteq N \backslash \{i\}} (|S|!)(n - |S| - 1)!(v(S \cup \{i\}) - v(S)). \qquad (1)$$

κ_i is called the *raw Shapley value*. It is well-known and straightforward to obtain that the raw Shapley value can be written as $\kappa_i(N, v) = \sum_{\sigma \in S_N} (v(p(i, \sigma) \cup \{i\}) - v(p(i, \sigma),))$, where S_N is the set of permutations on the player set N. For an unweighted matching game, the raw Shapley value of a player is thus equal to the number of pivotal permutations. We refer to the vectors $\varphi = (\varphi_1(N, v), \ldots \varphi_n(N, v))$ and $\kappa = (\kappa_1(N, v), \ldots, \kappa_n(N, v))$ respectively as the Shapley value and the raw Shapley value of the game (N, v).

The players $i, j \in N$ are called *symmetric* in (N, v) if $v(S \cup \{i\}) = v(S \cup \{j\})$ for any coalition $S \subseteq N \setminus \{i, j\}$. A player $i \in N$ is a *dummy* if $v(S \cup \{i\}) - v(S) = 0$ for all $S \subseteq N$. The Shapley value satisfies the following properties: (i) *Efficiency*: $\sum_{i \in N} \varphi_i(N, v) = v(N)$; (ii) *Symmetry*: if $i, j \in N$ are symmetric, then $\varphi_i(N, v) = \varphi_j(N, v)$; (iii) *Dummy*: if i is a dummy, then $\varphi_i(N, v) = 0$; (iv) *Additivity*: $\varphi_i(N, v^1 + v^2) = \varphi_i(N, v_1) + \varphi_i(N, v_2)$ for all $i \in N$;[1] and (v) *Anonymity*: relabeling the agents does not affect their Shapley value. We are interested in the following computational problem.

Shapley
Instance: A weighted graph $G = (N, E, w)$ and a specified player $i \in N$.
Question: Compute $\varphi_i(MG(G))$.

---

[1]   The sum of two characteristic functions v_1 and v_2 on the same player set is defined in the standard way: as $v_1(S) + v_2(S)$ for all $S \subseteq N$.

2.1   General insights

In this subsection, we gain some general insights about the Shapley value of matching games. First, if the graph is not connected, then the problem of computing the Shapley value of the graph reduces to computing the Shapley value of the respective connected components.

▶ **Lemma 1** (Shapley value in connected components). *Let $G = (N, E, w)$ be a weighted graph with k connected components, and let the respective vertex sets of these connected components be $N_1, \ldots, N_k$. Let v be the characteristic function of the matching game $MG(G)$ on that graph, and let $c : N \to [k]$ be the function that maps a vertex i to the number j such that $j \in N_k$.[2] Then, for every vertex i it holds that $\varphi_i(v) = \varphi_i(v_{c(i)})$, where v_j denotes the characteristic function of the matching game on the subgraph induced by N_j.*

It is rather straightforward to see that a vertex has a Shapley value zero if and only if it is not connected to any other vertex.

▶ Observation 1. A player in a matching game has a non-zero Shapley value if and only if there is an edge in the graph that contains the player. It can thus be decided in linear time whether a player in a matching game has a Shapley value of zero.

Next, we present another lemma concerning the Shapley value of unweighted matching games.

▶ **Lemma 2.** *Consider an unweighted matching game (N, v). If for each $s \in [n-1]$, the number of coalitions of size s for which player i is pivotal in (N, v) can be computed in time $f(n)$ for some function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$, then the Shapley value of i can be computed in time $n f(n)$.*

3   Exact algorithms for restricted graph classes

Some classes of matching games for which computing the Shapley value is trivial are symmetric graphs (e.g. cliques and cycles), and graphs with a constant number of vertices. We proceed to prove this for two additional special cases: weighted graphs that admit constant size (co)clique modular decompositions, and unweighted graphs with degree at most two.

3.1   Graphs with a constant number of clique or coclique modules

An important concept in the context of undirected graphs is that of a *module*. A subset of vertices $S \subseteq N$ is a module if all members of S have the same set of neighbors in $N \setminus S$. We can extend this notion to weighted graphs by requiring that all members of S are connected to the same set of neighbors, by edges of the same weight. A *modular decomposition* is a partition of the vertex set into modules.

A *clique module* (resp. *coclique module*) of a weighted graph is a module of which the vertices are pairwise connected by edges of the same weight (resp. pairwise disconnected). Note that every graph has a trivial modular decomposition into cliques (and cocliques): the partition of N into singletons.

We prove that if a weighted graph G has a size k modular decomposition consisting of only cliques or only cocliques, then the Shapley value of $MG(G)$ can be found in polynomial time. In fact, we will show that this holds for the more general class of *subgraph-based* games: We call a cooperative game (N, v) *subgraph-based* if there exists a weighted graph $G = (N, E, w)$ such that for $S, T \subset N$, it holds that $v(S) = v(T)$ if $G(S)$ and $G(T)$ are isomorphic.

---

[2] For $a \in \mathbb{N}$, we write $[a]$ to denote $\{b \in \mathbb{N} : 1 \leq b \leq a\}$.

▶ **Theorem 3.** *Consider a subgraph-based cooperative game (N, v). Then, the Shapley value of (N, v) can be computed in polynomial time if the following three conditions hold: i.) the weighted graph $G = (N, E, w)$ associated to (N, v) is given or can be computed in time polynomial in the size of the representation of (N, v); ii.) there exists a modular decomposition $\gamma(G)$ into k cocliques or k cliques and G is unweighted in the latter case; and iii.) $v(S)$ can be computed in polynomial time for all $S \subseteq N$.*

Proof. Note first that one can find for G in polynomial time a minimum cardinality modular decomposition into cocliques: simply check for each pair of vertices whether they are disconnected and connected to identical sets of vertices through edges with identical weights. If so, then they can be put in the same module. Similarly, a minimum cardinality modular decomposition into cliques can be found in polynomial time in case the graph is unweighted, by finding a minimum cardinality modular decomposition into cocliques in the complement of G (i.e., the graph that contains only those edges not in E).

A set of players S is said to be of the same *player type* if all player pairs in S are symmetric. We first show that all players in the same module of $\gamma(G)$ are of the same player type. Let i, j be two players in the same module M in $\gamma(G)$. Then, for every coalition $C \subseteq N\backslash\{i, j\}$, the subgraphs $G(C\cup\{i\})$ and $G(C \cup \{j\})$ are isomorphic (because $G(M)$ is a clique or coclique), so $v(C \cup \{i\}) = v(C \cup \{j\})$. Therefore, we know that the vertices can be divided into a constant number k of player types.

[27] showed that any cooperative game in which the value of a given coalition can be computed in polynomial time, and there is known size k partition of the players into sets of the same player type, then the Shapley value can be computed in polynomial time via dynamic programming. The number of player types in our game is constant number k of clique and coclique modules. Therefore the result of [27] can be applied, and this proves our claim. ◀

For matching games, the function v can be evaluated using any polynomial time maximum weight matching algorithm. Therefore, the above result implies that computing the Shapley value can be done in polynomial time for classes of graphs where we can find efficiently a size k modular decomposition into cliques or cocliques. This includes the class of complete k-partite graphs and any strong product[3] of an arbitrary size clique (or coclique) with a graph on k vertices.

▶ **Corollary 4.** *For matching games based on complete k-partite graphs, where k is a constant, the Shapley value can be computed in polynomial time.*

Theorem 3 also applies to cooperative games such as s-t vertex connectivity games and min-cost spanning tree games [10, 11], as these are subgraph-based games.

3.2 Graphs of degree at most two

We first examine *linear graphs* (or: "paths"), i.e., unweighted connected graphs in which two vertices have out-degree one and the remaining vertices have out-degree two.

▶ **Lemma 5.** *The Shapley value of a player in a matching game on an unweighted linear graph can be computed in $O(n^4)$ time.*

Proof. Assume without loss of generality that the vertex set is N and the edge set is $\{\{j, j + 1\} : j \in N\backslash\{n\}\}$, and that $i \in N$ is the player of whom we want to compute the Shapley value. Fix any $s \in [n - 1]$, and let η_i^s be the number of coalitions of size s for which vertex i is pivotal. We compute η_i^s by subdividing in separate cases and taking the sum of them:

---

[3] The *strong product* of two graphs $G_1 = (N, E_1)$ and $G_2 = (M, E_2)$ is defined as the graph $(N \times M, E')$, where $E' = \{\{(i_N, i_M), (j_N, j_M)\} \subseteq N \times M : i_M = j_M \wedge \{i_N, j_N\} \in E_1 \vee \{i_M, j_M\} \in E_2\}$.

- The number $\eta_i^{s,\text{left}} = |\{S \cup \{i + 1\} : S \subseteq N \backslash \{i, i - 1, i + 1\}, i \text{ is pivotal for } S\}|$. Intuitively: the number of coalitions S where i is pivotal such that adding i to S extends the left of a line segment.
- The number $\eta_i^{s,\text{right}} = |\{S \cup \{i - 1\} : S \subseteq N \backslash \{i, i - 1, i + 1\}, i \text{ is pivotal for } S\}|$.
- The number $\eta_i^{s,\text{connect}} = |\{S \cup \{i - 1, i + 1\} : S \subseteq N \backslash \{i, i - 1, i + 1\}, i \text{ is pivotal for } S\}|$. Intuitively: the number of coalitions S where i is pivotal, such that i connects two line segments.
- $\eta_i^{s,\text{isolated}} = |\{S : S \subseteq N \backslash \{i, i - 1, i + 1\}, i \text{ is pivotal for } S\}|$.

It is immediate that $\eta_i^{s,\text{isolated}} = 0$, since adding i to a coalition S not containing $i + 1$ nor $i - 1$ results in a coalition forming a subgraph in which i is an isolated vertex. For the remaining three values, $\eta_i^{s,\text{left}}$, $\eta_i^{s,\text{right}}$, and $\eta_i^{s,\text{connect}}$, we show below how to compute them efficiently.

- For $\eta_i^{s,\text{left}}$, observe that adding a vertex to the left of a (non-empty) line segment L increases the cardinality of a maximum matching if and only if L has an even number of edges (and thus an odd number of vertices). Therefore, define $\eta_i^{s,\text{left}}(k)$ to be the number of coalitions S of size s for which i is pivotal such that S contains the line segment $\{i + 1, \ldots, i + k + 1\}$, and does not contain $\{i - 1, i + k + 2\}$. The number $\eta_i^{s,\text{left}}(k)$ is easy to determine:

$$\eta_i^{s,\text{left}}(k) = \begin{cases} 0 & \text{if } k \text{ is odd}, \\ \binom{|N \backslash \{i-1,\ldots,i+k+2\}|}{s - |\{i-1,\ldots,i+k+1\} \cap N|} & \text{otherwise}. \end{cases}$$

We can then express $\eta_i^{s,\text{left}}$ as $\sum_{k=1}^{\max\{n-i-1, s-1\}} \eta_i^{s,\text{left}}(k)$. There is only a linear number of terms in this sum, and all of them can be computed in linear time.

- $\eta_i^{s,\text{right}}$ is computed in an analogous fashion.
- For $\eta_i^{s,\text{connect}}$, observe that adding a vertex i to a coalition such that i connects two line segments L_1 and L_2, increases the cardinality of a maximum matching if and only if L_1 and L_2 do not both have an odd number of edges (or equivalently: not both have an even number of vertices). Therefore, define $\eta_i^{s,\text{connect}}(k_1, k_2)$ to be the number of coalitions S of size s for which i is pivotal such that S contains the line segments $\{i - k_1 - 1, \ldots, i - 1\}$ and $\{i + 1, \ldots, i + k_2 + 1\}$, and does not contain $\{i - k_1 - 2, i + k_2 + 2\}$. The number $\eta_i^{s,\text{connect}}(k_1, k_2)$ is easy to determine:

$$\eta_i^{s,\text{connect}}(k_1, k_2) = \begin{cases} 0 & \text{if } k_1 \text{ and } k_2 \text{ are both odd}, \\ \binom{|N \backslash \{i-k-2,\ldots,i+k+2\}|}{s - |\{i-k-1,\ldots,i+k+1\} \cap N|} & \text{otherwise}. \end{cases}$$

We can then express $\eta_i^{s,\text{connect}}$ as $\sum_{k_1=1}^{\max\{i-2, s-1\}} \sum_{k_2=1}^{\max\{n-i-1, s-k_1-2\}} \eta_i^{s,\text{left}}(k_1, k_2)$. The number of terms in this sum is quadratic, and all of these terms can be computed in linear time. We can thus compute $\eta_i^{s,\text{connect}}$ in $O(n^3)$ time.

The claim now follows from Lemma 2. ◀

▶ **Theorem 6.** *For graphs with maximum degree* 2*, the Shapley value can be computed in polynomial time.*

Proof. A graph with degree at most two is a disjoint union of cycles and linear graphs. From Lemma 1, we can compute the Shapley value of the connected components separately. From Lemma 5, we know that the Shapley value of linear graphs can be computed in polynomial time. Due to anonymity, the Shapley value of a cycle is uniform. ◀

The above proof for linear graphs demonstrates that computation of the Shapley value of a matching game already becomes involved for even the simplest of graph structures. We would be interested in seeing an extension of this result that enables us to exactly compute the Shapley value in *any* non-trivial class of graphs that contains a vertex of degree at least three.

4 Computational complexity of the general problem

In this section, we examine the computational complexity of the general problem of computing the Shapley value for matching games. As we mentioned in Section 2, SHAPLEY is equivalent to the problem of counting the number of pivotal permutations for a player in an unweighted matching game, and is therefore a counting problem. It is moreover easy to see that this counting problem is a member of the complexity class #P.[4]

For certain cooperative games such as weighted voting games [14], intractability of computing the Shapley value can be established by proving that even checking whether a player gets non-zero Shapley value is NP-complete. Proposition 1 tells us that this is not the case for matching games. Before we proceed, we establish some notation. Let $G = (N, E)$ be a graph. Let $\alpha_k(G)$ be the number of vertex sets $S \subseteq N$ such that $|S| = k$ and the subgraph $G(S)$ of G induced by S admits a perfect matching. Then $\overline{\alpha_k}(G) = \binom{n}{k} - \alpha_k(G)$ is the number of subsets $S \subseteq N$ of size k such that $G(S)$ does not admit a perfect matching. In order to characterize the complexity of SHAPLEY, we first define the following problem.

#MATCHABLESUBGRAPHS$_k$
Instance: Undirected and unweighted graph $G = (N, E)$ and an even integer k.
Question: Compute $\alpha_k(G)$.

▶ **Lemma 7.** #MATCHABLESUBGRAPHS$_k$ *is* #P-*complete.*

Proof. In [9] it is proved that the following problem is #P-complete: Given an undirected and unweighted bipartite graph $G = (S \cup I, E)$, compute the number of subsets of $B \subseteq S$, such that $G(B \cup I)$ admits a perfect matching.[5] The problem is equivalent to #MatchableSubgraphs$_{2|I|}$. ◀

▶ **Theorem 8.** *Computing the Shapley value of a matching game on an unweighted graph is* #P-*complete.*

Proof. We present a polynomial-time Turing reduction from #MATCHABLESUBGRAPHS$_k$ to SHAPLEY. We show that if there exists a polynomial-time algorithm for SHAPLEY, then we can solve #MATCHABLESUBGRAPHS$_k$ for a given graph G in polynomial time, by solving SHAPLEY on a set of graphs that we construct from G. For each of these graphs, we show that a linear equation holds that relates the Shapley value of a vertex of G to the values α_k and $\overline{\alpha_k}$. The coefficient matrix of this system of equations will then turn out to be invertible, hence it can be solved in polynomial time via Gaussian elimination in order to compute the values α_k and $\overline{\alpha_k}$.

We remind the reader that the symbol κ is used to denotes the raw Shapley value, as defined in Section 2.

Let $G = (N, E)$ be the given graph, and let G_0 be the graph in which a new vertex y_0 is added to G that is connected to all vertices in N. For $i > 0$, let G_i be G_0 with i additional vertices $y_1, y_2, \ldots, y_i$ and i additional edges $\{\{y_j, y_{j-1}\} : j \in [i]\}$.

The first part of the proof consists of showing that the following set of equations hold:

$$\kappa_{y_i}(MG(G_i)) = \begin{cases} C(i) + \sum_{k=0}^{n}(k+i)!(n-k)!\overline{\alpha_k}(G) & \text{if } i \text{ is even,} \quad (2) \\ C(i) + \sum_{k=0}^{n}(k+i)!(n-k)!\alpha_k(G) & \text{if } i \text{ is odd,} \quad (3) \end{cases}$$

---

[4] Informally: #P is the class of computational problems that correspond to counting the number of accepting paths on a non-deterministic Turing machine. We refer the reader to any introductory text on complexity theory.

[5] The proof of Colbourn resolved "an exceptionally difficult problem" [9]. Interestingly, the corresponding decision problem of checking whether there exists a subgraph of size k that does not admit a perfect matching, appears to be open.

where

$$C(i) = \sum_{k=1}^{\lfloor i/2 \rfloor} \sum_{j=0}^{n+i-2k} (j + 2k - 1)!(n + i - j - 2k + 1)! \binom{n + i - 2k}{j}.$$

Define a *type 1 pivotal coalition for* y_i *in* $MG(G_i)$ as a pivotal coalition for i in $MG(G_i)$ that *does not* contain all players $y_0, \ldots, y_{i-1}$. Define a *type 2 pivotal coalition for* y_i *in* $MG(G_i)$ as a pivotal coalition for y_i in $MG(G_i)$ that *does* contain all players $y_0, \ldots, y_{i-1}$. Denote by $H_i^{\text{type 1}}(s)$ (resp. $H_i^{\text{type 2}}(s)$) the set of type 1 (resp. type 2) pivotal coalitions for i in $MG(G_i)$ that are of size s. From (1), it follows that

$$\kappa_{y_i}(MG(G_i)) = \sum_{s=1}^{n+i} s!(n + i - s)!|H_i^{\text{type 1}}(s)| + \sum_{s=1}^{n+i} s!(n + i - s)!|H_i^{\text{type 2}}(s)|. \tag{4}$$

First we characterize the coalitions in $H_i^{\text{type 2}}(s)$.

▶ **Lemma 9.** *If i is even, a coalition S of $MG(G_i)$ is in $H_i^{\text{type 2}}(s)$ if and only if $G(S \cap N)$ is not perfectly matchable (and $\{y_0, \ldots, y_{i-1}\} \subseteq S, |S| = s$). If i is odd, a coalition S of $MG(G_i)$ is in $H_i^{\text{type 2}}(s)$ if and only if $G(S \cap N)$ is perfectly matchable (and $\{y_0, \ldots, y_{i-1}\} \subseteq S, |S| = s$).*

Proof. *Case of even i.* ($\Rightarrow$) Let M be a maximum matching for $G_i(S)$. S is pivotal for y_i, so M is not a perfect matching. We can assume though, that all vertices $\{y_0, \ldots, y_{i-1}\}$ are matched to each other in the matched graph $(G_i(S), M)$, because $G_i(\{y_0, \ldots, y_{i-1}\})$ is a linear graph with an even number of vertices, and is thus perfectly matchable. It follows that the exposed nodes of $(G_i(S), M)$ are all in N, and therefore the matching M restricted to N is a maximum matching for $G(S \setminus \{y_0, \ldots, y_{i-1}\}) = G(S \cap N)$ that is non-perfect.

($\Leftarrow$) Let M be a maximum (non-perfect) matching for $G(S \cap N)$ and let y be an exposed vertex of $(G(S \cap N), M)$. Then $M' = M \cup \{\{y_j, y_{j+1}\} : j \text{ even} \wedge j < i\}$ is a maximum matching for $G_i(S)$, by Berge's Lemma, as it is clear that there is no augmenting path in $(G_i(S), M')$. Moreover, observe that in $(G_i(S), M')$ there is an even-length alternating path from y to y_{i-1}. Therefore, there is in (G_i, M') an augmenting path from y to y_i, and it follows again by Berge's lemma that S is pivotal.

Case of odd i. ($\Rightarrow$) Let M' be a maximum matching for $G_i(S)$. S is pivotal, so in $(G_i(S), M')$ there is an even-length alternating path P from an exposed node y to y_{i-1}. Obtain the matching M by augmenting M' along P. M is then a maximum matching for $G_i(S)$ in which y_{i-1} is exposed. $G_i(\{y_0, \ldots, y_{i-1}\})$ is a linear graph and M is maximum, so it follows that y_{i-1} is the only exposed node in $(G_i(S), M)$ among $\{y_0, \ldots y_{i-1}\}$. Therefore $S \cap N$ must be matched to each other in $(G(S), M)$ (for otherwise, in $(G_i(S), M)$ there would be an augmenting path from y_{i-1} to an exposed node of $S \cap N$, contradicting the fact that M is a maximum matching for $G_i(S)$). It follows that $G(S \cap N)$ is perfectly matchable.

($\Leftarrow$) Let M be a maximum perfect matching for $G(S \cap N)$. Let M' be a maximum matching for $G_i(\{y_0, \ldots, y_{i-1}\})$ in which y_{i-1} is the only exposed node. Then $M \cup M'$ is a matching for $G_i(S)$ in which y_{i-1} is the only exposed node. $M \cup M'$ is clearly a maximum matching, and in $(G_i, M \cup M')$ the edge $\{y_{i-1}, y_i\}$ is exposed. So S is pivotal. ◀

From the above lemma, it follows that the coalitions in $H_i^{\text{type 2}}(s)$ are precisely the coalitions of the form $T \cup \{y_0, \ldots, y_{i-1}\}$, where $T \subset N$ is such that for even i, $G(T)$ is not perfectly matchable, and for odd i, $G(T)$ is perfectly matchable. Therefore $|H_i^{\text{type 2}}(s)| = \overline{\alpha_{s-i}}(G)$ for even i and $|H_i^{\text{type 2}}(s)| = \alpha_{s-i}(G)$ for odd i, and this implies:

$$\sum_{s=1}^{n+i} s!(n + i - s)!|H_i^{\text{type 2}}(s)| = \begin{cases} \sum_{k=0}^{n}(k + i)!(n - k)!\overline{\alpha_k}(G) & \text{if } i \text{ is even,} \\ \sum_{k=0}^{n}(k + i)!(n - k)!\alpha_k(G) & \text{if } i \text{ is odd.} \end{cases}$$

In words: the second summation of (4) equals the summation of (2) when i is even, and the summation of (3) when i is odd. Therefore, it suffices to prove that the first summation of (4) equals $C(i)$.

For this sake, define $H_i^{\text{type 1}}(s, k)$ for $k \in [[i/2]]$ as $\{S \in H_i^{\text{type 1}}(s) : y_{i-2k} \notin S \wedge \{y_{i-1}, \ldots, y_{i-2k+1}\} \subseteq S\}$. Observe that $\{H_i^{\text{type 1}}(s, 1), \ldots, H_i^{\text{type 1}}(s, i/2)\}$ is a partition of $H_i^{\text{type 1}}(s)$. For a given k and s, note that the set $H_i^{\text{type 1}}(s, k)$ consists of all coalitions of the form $T \cup \{y_{i-1}, \ldots, y_{i-2k+1}\}$, where $T \subseteq N \cup \{y_0, \ldots, y_{i-2k-1}\}$, $|T| = s - 2k + 1$. Hence, $|H_i^{\text{type 1}}(s, k)| = \binom{n+i-2k}{s-2k+1}$ (defining $\binom{a}{b} = 0$ whenever $b < 0$ or $b > a$). Therefore:

$$
\begin{aligned}
\sum_{s=1}^{n+i} s!(n+i-s)!|H_i^{\text{type 1}}(s)| &= \sum_{k=1}^{\lfloor i/2 \rfloor} \sum_{s=2k-1}^{n+i-1} s!(n+i-s)!\binom{n+i-2k}{s-2k+1} \\
&= \sum_{k=1}^{\lfloor i/2 \rfloor} \sum_{j=0}^{n+i-2k} (j+2k-1)!(n+i-j-2k+1)!\binom{n+i-2k}{j}.
\end{aligned}
$$

This shows that (2) and (3) hold.

The second part of the proof consists of showing that all $\alpha_k(G), k \in N$ can be computed from $\kappa_{y_i}(MG(G_i))$ in polynomial time, using (2) and (3), for $i \in N \cup \{0\}$. This is sufficient to complete the proof, because the graphs $G_0, \ldots, G_n$ can clearly be constructed from G in polynomial time, hence a polynomial time algorithm that computes α_k from $\kappa_{y_i}(MG(G_i))$, $i \in N$ is a polynomial Turing reduction.

Let $\beta_i(G) = \alpha_i(G)$ for even i and let $\beta_i(G) = \overline{\alpha_i}(G)$ for odd i. We can represent (2) and (3) for $i \in N \cup \{0\}$ as the following system of equations:

$$
\begin{pmatrix}
0!n! & 1!(n-1)! & \cdots & n!0! \\
1!n! & & \cdots & (n+1)!0! \\
\vdots & \vdots & \ddots & \vdots \\
n!n! & & \cdots & (2n)!0!
\end{pmatrix}
\times
\begin{pmatrix}
\beta_0(G) \\
\beta_1(G) \\
\vdots \\
\beta_n(G)
\end{pmatrix}
=
\begin{pmatrix}
\kappa_{y_0}(MG(G_0)) - C(0) \\
\kappa_{y_1}(MG(G_1)) - C(1) \\
\vdots \\
\kappa_{y_n}(MG(G_n)) - C(n)
\end{pmatrix}
\tag{5}
$$

Denote by A the $(n+1) \times (n+1)$ matrix in the above equation. Recall that a scalar multiplication of a column by a constant c multiplies the determinant by c. Therefore, A is nonsingular if and only if nonsingularity also holds for the $(n+1) \times (n+1)$ matrix B, defined by $B_{ij} = (i+j)!$. B is a matrix that is related to Pascal's triangle, and it is known that its determinant is equal to $\prod_{i=0}^{n} i!^2 \neq 0$ [3, 2]. It follows that A is nonsingular, so our system of equations (5) is linearly independent and has a unique solution. Note that all entries in the system can be computed in polynomial time (assuming that the Shapley value of a matching game is polynomial time computable): The constants $C(i)$ consist of polynomially many terms, and all factorials and binomial coefficients that occur in (5) are taken over numbers of magnitude polynomial in n.

Therefore, we can use Gaussian elimination to solve (5) in $O(n^3)$ time. It follows that for all $i \in N$, $\beta_i(G)$ can be computed in polynomial time, and hence $\alpha_i(G)$ can be computed in polynomial time. Therefore, if there exists an algorithm that solves Shapley in polynomial time, then it can also be used to solve #MatchableSubgraphs$_k$ in polynomial time. ◄

5 An approximation algorithm

In this section, we show that although computing exactly the Shapley value of matching games is a hard problem, approximating it is much easier.

Let Σ be a finite alphabet in which we agree to describe our problem instances and solutions. A *fully polynomial time randomized approximation scheme (FPRAS)* for a function $f : \Sigma^* \to \mathbb{Q}$ is an algorithm that takes input $x \in \Sigma^*$ and a parameter $\epsilon \in \mathbb{Q}_{>0}$, and returns with probability at

least $\frac{3}{4}$ a number in between $f(x)/(1 + \epsilon)$ and $(1 + \epsilon)f(x)$. Moreover, an FPRAS is required to run in time polynomial in the size of x and $1/\epsilon$. The probability of $\frac{3}{4}$ is chosen arbitrarily: by a standard amplification technique, it can be replaced by an arbitrary number $\delta \in (1/2, 1)$. The resulting algorithm would then run in time polynomial in n, $1/\epsilon$, and $\log(1/\delta)$.

We will now formulate an algorithm that approximates the raw Shapley value of a player in a weighted matching game, and show that it is an FPRAS. Note that we cannot utilize approximation results in [18] and [4] since matching games are neither convex nor simple. Our FPRAS is based on Monte Carlo sampling, and works as follows: Let $(G = (N, E, w), i, \epsilon)$ be the input, where G is the weighted graph representing matching game $MG(G)$, $i \in N$ is a player in $MG(G)$, and ϵ is the precision parameter. For notational convenience, we write κ_i as a shorthand for $\kappa_i(MG(G))$. The algorithm first determines whether $\kappa_i = 0$ (Observation 1). If so, then it outputs 0 and terminates. If not, then it samples $\lceil 4n^2(n-1)^2/\epsilon^2 \rceil$ permutations of the player set uniformly at random. Denote this multiset of sampled permutations by P. The algorithm then outputs the average marginal contribution of player i over the permutations in P and terminates. Note that this average marginal contribution is efficiently computable: it is given by $1/\lceil 4n^2(n-1)^2/\epsilon^2 \rceil$ times the sum of the marginal contributions of player i to each of the sampled permutations. Determining these marginal contributions can be done in polynomial time, using any maximum weight matching algorithm. Denote our sampling algorithm by MatchingGame-Sampler.

MatchingGame-Sampler resembles the algorithms in [19, 18]: the differences are that the algorithm takes a different number of samples, and that it determines whether the Shapley value of player i is 0 prior to running the sampling procedure. Moreover, its proof of correctness requires different insights.[6]

▶ **Theorem 10.** MatchingGame-Sampler *is an FPRAS for the raw Shapley value in a weighted matching game.*

Proof. Denote by $\bar{\kappa}_i$ the output of the algorithm. If $\kappa_i = 0$, then MatchingGame-Sampler is guaranteed to output the right solution, so assume that $\kappa_i > 0$. Let $w_i^{\max}$ be the maximum weight among the edges attached to i, and let $e_i^{\max} \in E$ be an edge that is attached to i such that $w(e_i^{\max}) = w_i^{\max}$. Let X be a random variable that takes the value of $n!$ times the marginal contribution of player i in a uniformly randomly sampled permutation of the players. Note that $\mathbf{E}[X] = \kappa_i$. Note that the marginal contribution of a player in any permutation is at most $w_i^{\max}$, so X is at most $w_i^{\max} n!$.

Let j be the neighbor of i connected by $e_i^{\max}$. Observe that any permutation in which j is positioned first, and i is positioned second, is a permutation for i in which the marginal contribution of i is $w_i^{\max}$. There are $(n-2)!$ such permutations, so the raw Shapley value κ_i of i is at least $w_i^{\max}(n-2)!$. For the variance of X we obtain $\mathbf{Var}[X] = \mathbf{E}[X^2] - \mathbf{E}[X]^2 \leq \mathbf{E}[X^2] \leq (w_i^{\max})^2 n!^2 \leq n^2(n-1)^2\kappa_i^2$.

Observe that $\bar{\kappa}_i$ is a random variable that is equal to $\frac{\sum_{j=1}^{\lceil 4n^2(n-1)^2/\epsilon^2 \rceil} X_j}{\lceil 4n^2(n-1)^2/\epsilon^2 \rceil}$, where X_j are independent random variables with the same distribution as X. From this we obtain that $\mathbf{E}[\bar{\kappa}_i] = \mathbf{E}[X] = \kappa_i$. The desired approximation guarantee then follows from Chebyshev's inequality,[7] and completes the proof:

$$
\begin{aligned}
\mathbf{Pr}[|\bar{\kappa}_i - \kappa_i| \geq \epsilon\kappa_i] \quad \leq \quad & \frac{\mathbf{Var}[\bar{\kappa}_i]}{\epsilon^2\kappa_i^2} = \frac{\mathbf{Var}\left[\frac{1}{\lceil 4n^2(n-1)^2/\epsilon^2 \rceil} \sum_{j=1}^{\lceil 4n^2(n-1)^2/\epsilon^2 \rceil} X_j \right]}{\epsilon^2\kappa_i^2} \\
= \quad & \frac{\left(\frac{\mathbf{Var}[X]}{\lceil 4n^2(n-1)^2/\epsilon^2 \rceil} \right)}{\epsilon^2\kappa_i^2} \leq \frac{n^2(n-1)^2\kappa_i^2}{(4n^2(n-1)^2/\epsilon^2) \cdot \epsilon^2\kappa_i^2} \leq \frac{1}{4}.
\end{aligned}
$$

---

[6] To be precise, this applies only to [18]. For the sampling algorithm in [19], no proof or approximation-quality analysis of any kind is given.

[7] Here, one could also choose to apply Hoeffding's inequality instead of Chebyshev's inequality, but this will not result in an asymptotically better bound.

◄

▶ **Corollary 11.** *The algorithm that runs* MATCHINGGAME-SAMPLER *and returns its output scaled down by* $1/n!$, *is an FPRAS for the Shapley value of a weighted matching game.*

Observe that MATCHINGGAME-SAMPLER is an FPRAS in the strong sense that its running time does not depend on the weights of the edges. Due to the #P-completeness result stated in Theorem 7, this FPRAS is the best one can hope for, and provides us with a complete answer to the precise complexity of this problem (based on our best judgment).

6 Conclusions

In this paper, we examined the structure, algorithms, and computational complexity for the problem of computing the Shapley value in a matching game. There are many special cases of the problem that have not been treated in this paper, but nonetheless are potentially worthwhile to analyze: trees, bipartite graphs, connected regular graphs, and series-parallel graphs. Among these, bipartite graphs are especially interesting, since they model two-sided markets. There are some interesting computational problems related to Shapley value computation, such as the problem of comparing the Shapley values of two vertices. One may also pursue the same questions for *fractional matching games* in which the value of a coalition is the maximum size of a fractional matching [8]. Moreover, our study motivates the investigation of unexplored connections with some objects in matching theory. The matching polytope is one of the most-well studied objects in polyhedral combinatorics [22]. It will be interesting to identify any relation between the matching polytope of a graph and the Shapley values of the corresponding matching game. Secondly, network flows are fundamentally connected to matchings for the case of bipartite graphs. An interesting research direction is to explore the connection of network flow games [16] with matching games and whether computing Shapley values of one game is reducible to computing Shapley values of the other game, under certain conditions.

References

1 A. Alkan and D. Gale. The core of the matching game. *Games and Economic Behavior*, 2(3):203–212, 1990.

2 H. Aziz, O. Lachish, M. Paterson, and R. Savani. Power indices in spanning connectivity games. In *Proc. of 5th International Conference on Algorithmic Aspects in Information and Management (AAIM)*, volume 5564 of *LNCS*, pages 55–67. Springer, 2009.

3 R. Bacher. Determinants of matrices related to the Pascal triangle. *Journal de théorie des nombres de Bordeaux*, 14:19–41, 2002.

4 Y. Bachrach, E. Markakis, E. Resnick, A. D. Procaccia, J. S. Rosenschein, and A. Saberi. Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems*, 20:105–122, 2010.

5 P. Biró, M. Bornhoff, P. A. Golovach, W. Kern, and D. Paulusma. Solutions for the stable roommates problem with payments. *Theoretical Computer Science*, 2013.

6 P. Biró, W. Kern, and D. Paulusma. Computing solutions for matching games. *International Journal of Game Theory*, 41(1):75–90, 2011.

7 U. Brandes and T. Erlebach, editors. *Network Analysis*, volume 3418 of *LNCS*. Springer, 2005.

8 N. Chen, P. Lu, and H. Zhang. Computing the nucleolus of matching, cover and clique games. In *Proc. of 26th AAAI Conference*, 2012.

9 C. J. Colbourn, J. S. Provan, and D. Vertigan. The complexity of computing the Tutte polynomial on transversal matroids. *Combinatorica*, 15(1):1–10, 1995.

10 X. Deng and Z. Fang. Algorithmic cooperative game theory. In A. Chinchuluun, P. M. Pardalos, A. Migdalas, and L. Pitsoulis, editors, *Pareto Optimality, Game Theory And Equilibria*, volume 17 of *Springer Optimization and Its Applications*. Springer-Verlag, 2008.

11 X. Deng, T. Ibaraki, and H. Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999.

12 X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 12(2):257–266, 1994.

13 J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

14 E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. On the computational complexity of weighted voting games. *Annals of Mathematics and Artificial Intelligence*, 56(2):109–131, 2009.

15 S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proc. of 6th ACM-EC Conference*, pages 193–202. ACM Press, 2005.

16 E. Kalai and E. Zemel. Generalized network problems yielding totally balanced games. *Operations Research*, 30:998–1008, 1982.

17 W. Kern and D. Paulusma. Matching games: The least core and the nucleolus. *Mathematics of Operations Research*, 28(2):294–308, 2003.

18 D. Liben-Nowell, A. Sharp, T. Wexle, and K. Woods. Computing shapley value in supermodular coalitional games. In *Proc. of 18thCOCOON*, 2011.

19 I. Mann and L. S. Shapley. Values of large games, iv: Evaluating the electoral college by montecarlo techniques. Technical Report RM-2651, RAND Corporation, 1960.

20 T. P. Michalak, K. V. Aadithya, P. L. Szczepanski, B. Ravindran, and N. R. Jennings. Efficient computation of the shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research*, 46:607–650, 2013.

21 H. Moulin. *Fair Division and Collective Welfare*. The MIT Press, 2003.

22 M. D. Plummer. Matching theory - a sampler: from Dénes König to the present. *Discrete Mathematics*, 100:177–219, 1992.

23 A. Roth and M. A. O. Sotomayor. *Two-Sided Matching: A Study in Game Theoretic Modelling and Analysis*. Cambridge University Press, 1990.

24 L. S. Shapley. A value for n-person games. *Annals of Math Studies*, 28:307–317, 1953.

25 L. S. Shapley and M. Shubik. The Assignment Game I: The Core. *International Journal of Game Theory*, 1:111–130, 1972.

26 T. Solymosi and T. E. S. Raghavan. An algorithm for finding the nucleolus of assignment games. *International Journal of Game Theory*, 23:119–143, 1994.

27 S. Ueda, M. Kitaki, A. Iwasaki, and M. Yokoo. Concise characteristic function representations in coalitional games based on agent types. In T. Walsh, editor, *Proc. of 22nd IJCAI*, pages 393–399. AAAI Press, 2011.

28 E. Winter. The Shapley value. In *Handbook of Game Theory with Economic Applications*, chapter 53, pages 2025–2054. Elsevier, 2002.