Artificial Intelligence 278 (2020) 103180

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Coalitional games induced by matching problems: Complexity and islands of tractability for the Shapley value

Gianluigi Greco *, Francesco Lupia, Francesco Scarcello

University of Calabria, Italy

ARTICLE INFO

Article history:
Received 4 February 2019
Received in revised form 8 August 2019
Accepted 25 September 2019
Available online 3 October 2019

Keywords:
Coalitional games
Matching theory
Shapley value
Computational complexity
Tree decompositions
Monadic second order logic

ABSTRACT

The paper focuses on cooperative games where the worth of any coalition of agents is determined by the optimal value of a matching problem on (possibly weighted) graphs. These games come in different forms that can be grouped in two broad classes, namely of *matching* and *allocation* games, and they have a wide spectrum of applications, ranging from two-sided markets where buyers and sellers are encoded as vertices in a graph, to allocation problems where indivisible goods have to be assigned (matched) to agents in a fair way, possibly using monetary compensations.

The Shapley value and the related notion of Banzhaf value have often been identified as appropriate solution concepts for many applications of matching/allocation games, but their computation is intractable in general. It is known that these concepts can be computed in polynomial time for matching games on unweighted trees and on graphs having degree at most two. However, it was open whether or not such positive results could be extended to the more general case of graphs having bounded treewidth, and to the case of allocation problems on weighted graphs.

The paper provides a positive answer to these questions, by showing that computing the Shapley value and the Banzhaf value is feasible in polynomial time for the following classes of games: matching games over unweighted graphs having bounded treewidth, allocation games over weighted graphs having bounded treewidth, and allocation games over weighted graphs and such that each good is of interest for two agents at most. Without such structural restrictions, computing these solution concepts on allocation games is instead shown to be #P-hard, even in the case of unweighted graphs.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Coalitional games and matching problems

Coalitional game theory provides a rich framework for understanding and reasoning about environments populated by autonomous agents who may want to collaborate with each other in order to obtain higher worth than by acting in isolation. While being traditionally grounded in the economic literature, coalitional games gained popularity in the last few years within the artificial intelligence research community, where they have been intensively analyzed from the computational viewpoint [1–5] and where they have been used in a wide spectrum of applications, such as task allocation [6], rideshar-

* Corresponding author.
E-mail addresses: gianluigi.greco@unical.it (G. Greco), francesco.lupia@unical.it (F. Lupia), francesco.scarcello@unical.it (F. Scarcello).

https://doi.org/10.1016/j.artint.2019.103180
0004-3702/© 2019 Elsevier B.V. All rights reserved.

ing [7], worth distribution problems based on agents' skills [8], and definition of mechanisms to pool scarce resources [9, 10].

In abstract terms, a *coalitional game* $\mathcal{G}$ can be formalized as a tuple $\langle N, v\rangle$, where N is a set of agents and v is a function associating each coalition $C \subseteq N$ with the worth $v(C) \in \mathbb{R}$ that agents in C can guarantee to themselves. This function is usually given in a compact way, rather than by explicitly listing all possible (exponentially-many) coalitions with their worth [11–14]. In particular, for each coalition C, the value $v(C)$ often comes from the solution to an underlying computation/optimization problem, whose precise formulation depends on the specific games that are considered (see, e.g., [12,15–19] and the references therein). As an example, in the well-known class of *induced subgraph games* [11], we are given a weighted graph whose nodes correspond to the agents and where the value $v(C)$ is defined as the sum of the weights of the edges in the subgraph induced by C. Other classes of games that have been intensively studied in earlier literature are still defined on top of graphs/networks, but involve more complex optimization problems related, for instance, to the computation of *minimum spanning trees* [20], *maximal flows* [21–23], *optimal connectivity properties* [24,25], *optimal placements of facilities* [26], *minimum traveling salesman paths* [27], and *minimum vertex covers* [28], just to name a few.

In the paper, we focus on this kind of combinatorial approach to succinctly specify coalitional games, and we consider two classes of games, namely *matching* and *allocation* games, where the underlying optimization problem is the computation of *maximum matchings*. Being defined by the same optimization problem, matching and allocation games share many conceptual similarities and can be analyzed by using similar technical machineries. Indeed, the main difference between these classes of games comes in how we constraint and interpret the underlying graph where maximum matchings have to be computed: while in matching games the graph can be arbitrary and its nodes one-to-one correspond to agents, in allocation games the graph is always bipartite, with nodes on one "side" corresponding to agents, and nodes on the other "side" corresponding to "goods" (or whatever agents are actually interested in).

The interest in matching games goes back to the seventies, when they have been firstly advocated by Shapley and Shubik [29] as a natural model to study two-sided markets [30]. In this model, the underlying graph G is bipartite and its edges encode the interactions between a set of buyers and a set of sellers, each of them supplying or demanding exactly one unit of an indivisible good. The value $v(C)$ of a coalition C, which can mix together buyers and sellers, is the maximum number of trades that can occur among the members of C—and, indeed, it coincides with the cardinality of the maximum matching in the subgraph of G induced by the nodes in C, denoted hereafter by $G[C]$. The extension of this model to arbitrary graphs (i.e., not necessarily bipartite) was firstly proposed by LeBreton and Weber [31] in the context of an application for school location and, subsequently, by Eriksson and Karlander [32] to deal with a roommate problem. In these extensions, moreover, the edges of the underlying graph G are weighted (reflecting the utility of matching the agents associated to the endpoints), so that $v(C)$ is defined as the weight of a maximum weighted matching in the subgraph of G induced by the nodes in C. The resulting games go generally under the name of *weighted-matching* games and they have been the subject of several research works, which have been aimed at exploiting them in concrete domains, as well as at analyzing their intriguing theoretical underpinnings (see [33] and the references therein).

Compared to (weighted-)matching games, the formalization of allocation games is more recent. These games have been proposed in the nineties by Moulin [34] as a framework to analyze fair division problems where monetary compensations are allowed and utilities are quasi-linear. Their applications range from house allocation to room assignment-rent division, to (cooperative) scheduling and task allocation, to protocols for wireless communication networks, and to queuing problems. Similarly to the matching games by Shapley and Shubik [29], we are given a bipartite graph, but here the two "sides" of the partition are semantically asymmetric: one of them still models the given set of agents (who form the coalitions), while the other one comprises a number of "goods" to be allocated to agents in an optimal way. An edge connecting an agent i to a good g means that i is interested in the good g and, in weighted graphs, a weight on the edge reflects i's valuation of g. In fact, in most applications, such as in [35–38], goods values do not depend on agents, but are only determined by objective properties of the goods (e.g., the value of a house, or the quality of a research product in [35,39]). Therefore, they can be modeled as bipartite graphs with weights on the goods or, viewed in the classical matching framework, as edge-weighted (bipartite) graphs where, for each good, its incident edges have the same weight. We refer to these games as the *weighted-allocation games*, and we do not deal in the paper with the case of arbitrary weights on edges, which is left for future work.

For every kind of game considered in the paper, we provide a fine-grained analysis to distinguish the results that hold in the setting where the underlying graphs are weighted and the coalition worth is determined by maximum-weight matchings, from the results that apply to the unweighted version where the coalition worth is determined by maximum-cardinality matchings.

1.2. Solutions concepts and the Shapley value

In a coalitional game $\mathcal{G} = \langle N, v\rangle$ (*with transferable utility*), it is assumed that the agents forming a coalition C can distribute the payoff $v(C)$ among themselves in any way. An outcome can be, then, modeled as a profile assigning some payoff x_i to each agent $i \in N$, in a way that the total worth available to the grand-coalition N is distributed, i.e., $x_1 + \cdots + x_n = v(N)$. The crucial question is to identify those outcomes, called *solution concepts*, which can embody some desirable and intuitive stability and/or fairness properties in the worth distribution.

Table 1
Summary of complexity results for graphs with and without weights on the edges—numbers refer to the theorems where the results are proven. Specific entries are reported for results provided on graphs without further restrictions (ANY), on classes of graphs having bounded treewidth (BTW), and on classes of graphs whose degree is at most two (DEGREE ≤ 2)—for allocation games, the requirement on the degree refers to the nodes corresponding to the "goods" that have to be allocated (that is, it identifies the class of *binary-clashes* allocation games). $^{*,+}$ results are proven in [47], [38], respectively.

	GRAPHS			WEIGHTED GRAPHS		
	ANY	BTW	DEGREE ≤ 2	ANY	BTW	DEGREE ≤ 2
MATCHING	**#P**-hard*	in **P** (4.10)	in **P***	**#P**-hard*	open	in **P***
ALLOCATION	**#P**-hard (3.2)	in **P** (5.11)	in **P** (6.1)	**#P**-hard$^+$	in **P** (5.11)	in **P** (6.1)

The most well-studied solution concept is arguably the *core*, formalized by Gillies [40] on top of an intuition that goes back to the work of Edgeworth [41]. The idea is to single out those outcomes that are "stable" w.r.t. deviations of sets of agents which might want to leave the grand-coalition and form an independent coalition, that is, $\sum_{i \in C} x_i \geq v(C)$ must hold for each $C \subseteq N$. On allocation games and on matching games defined over bipartite graphs, it has been shown that there always exists (at least) one outcome satisfying these conditions [29,34]. Instead, this is not the case for arbitrary matching games [42], as the reader can immediately check by considering three agents connected in a triangle (whose edges have unitary weight), for which the conditions $x_1 + x_2 \geq 1$, $x_1 + x_3 \geq 1$, $x_2 + x_3 \geq 1$, and $x_1 + x_2 + x_3 = 1$ cannot be simultaneously satisfied. However, polynomial-time algorithms have been exhibited to check whether the core of a (possibly weighted) matching game is not empty and to compute a core outcome (if any exists) [15,43].

Noticeable solution concepts that refine the stability condition of the core are the *least-core* and the *nucleolus* [44]. These concepts have been put under the lens of computational complexity on matching-based games, too. Indeed, they have been shown to be computable in polynomial time on matching games without weights [45] and on any class of weighted matching games for which the core is guaranteed not to be empty [43].

Other solution concepts focus on fairness properties rather than on stability conditions. In this context, the prototypical solution concept is the *Shapley* value [46], which is based—very roughly—on the idea that the payoff of an agent should be somehow proportional to the marginal contributions she provides to the payoff of all possible coalitions that can form. This idea leads to very desirable properties for a division scheme, such as equal treatment of equals and additivity (see Section 2 for a more formal treatment). However, the price to be paid is that, differently from the core (and related concepts), the Shapley value is computationally intractable.

Indeed, Aziz and de Keijzer [47] have shown that, on matching games, computing the Shapley value is **#P**-complete, hence not feasible in polynomial time under the standard complexity assumption that **P**≠**NP**. However, on the positive side, they showed that the computation is feasible in polynomial time if the maximum degree of the underlying graphs is two, and they asked whether this result can be extended to some non-trivial class of graphs of degree at least three. The question has been then addressed by Bousquet [48], who showed that the Shapley value of matching games over trees can be computed in polynomial time, and left open the problem of assessing whether the result can be extended to larger classes of graphs having a low degree of cyclicity, formally, to classes of graphs having *bounded treewidth* [49].

Similar bad news come from allocation problems, for which the Shapley value is known to be intractable (**#P**-complete) when the underlying graphs are weighted [38], but it was not known whether such an intractability result extends to the case of unweighted graphs. Moreover, no analysis was so far conducted to isolate subclasses of practical interest where the computation can be carried out efficiently. In particular, prior to our work, it was not assessed whether the Shapley value of allocation problems can be computed efficiently on trees and, more-generally, on graphs having bounded treewidth, or on graphs whose nodes have a small degree.

1.3. Contribution

In the paper, we address the above research questions related to matching and allocation games, and we provide a comprehensive analysis of the intrinsic complexity of the Shapley value by devoting special care to the identification of islands of tractability.

A summary of our results is reported in Table 1. Note that we have considered all the settings previously addressed in the literature and discussed in Section 1.2, by distinguishing weighted graphs from unweighted ones, and by dealing with graphs without structural restrictions, with graphs having bounded treewidth, and with graphs having bounded degree (in particular, degree at most two). The picture that emerges is rather clear and complete, with the only missing entry referring to the complexity of weighted-matching games on graphs having bounded treewidth.[1]

[1] Assessing whether our tractability result in Theorem 4.10 can be extended to weighted-matching games is one of the challenging open problems that we leave for further research—see Section 8.

In more details, our contribution is as follows:

(1) We take a closer look at the complexity of allocation games and we show that **#P**-hardness still holds even if the underlying bipartite graphs are not weighted. The reduction is entirely different from the one exhibited in the weighted case [38].
(2) We analyze matching and allocation games on graphs having bounded treewidth, by providing polynomial-time algorithms that are able to compute the Shapley value and, with minor modifications, the *Banzhaf value* [50], a closely-related solution concept. On the two classes of games, tractability results are established via a similar proof scheme that evidences the connection that exists between them. Indeed, such results are established *(i)* by pointing out novel and useful properties for matchings, *(ii)* by using these properties to get rid of matching optimization problems and solve instead suitable counting problems, and *(iii)* by exploiting known tractability results for Monadic Second Order logic (MSO) formulas [51] on graphs having bounded treewidth. The first two points required rather involved technical elaborations and the graph-theoretical results we have derived are of independent interest. In fact, these tractability results are the main technical achievements of this paper and our proof techniques might well find applications in different contexts. It is worthwhile noting that the tractability result we establish on matching games positively answers the open problem posed by Aziz and de Keijzer [47] and Bousquet [48]. Moreover, the tractability result on allocation games works for weighted games, too (where weights encode objective values of the goods). With this respect, it is relevant to observe that dealing with weights poses additional challenges in order to use the ingredient *(iii)* discussed above, as they cannot be supported "natively" in the MSO formalism.
(3) We show that the Shapley value of weighted-allocation games can be computed in polynomial time on *binary-clashes allocation games*, that is, on those games where at most two agents can be interested in the same good, without any restriction on the number of goods each agent is interested in. This tractability result about allocation games complements the analogous structural result that was already proven in the literature for matching games where, however, *all* nodes are required to have degree at most two [47].

An interesting feature of our approach to show tractability of the Shapley value is that, since polynomial-time algorithms for evaluating MSO formulas on graphs having bounded treewidth have been implemented in systems that are already available to the research community, the MSO encoding provides us with a declarative, yet "executable" specification for its computation. Therefore, on a more practical level,

(4) We implement the algorithms discussed in (2), by delegating the evaluation of the MSO encodings to the state-of-the-art MSO solver *Sequoia* [52]. The effectiveness of the resulting prototype is then tested on a number of benchmark graphs, derived either synthetically or from a concrete application for the evaluation of the quality of research in Italy [35,39]. Results of this experimentation are discussed in the paper, too.

Note that the analysis we carry out in this paper extends some preliminary results on the complexity of the Shapley value we discussed in [53,54]. In particular, the polynomial time algorithm we present for weighted-allocation games over graphs having bounded treewidth is entirely novel and significantly generalizes[2] the one illustrated in the conference version [54]. Moreover, results of a comprehensive experimental activity were missing in [53,54], where only a few preliminary tests of synthetic datasets had been considered.

1.4. Organization

The rest of the paper is organized as follows. Section 2 presents the notation and overviews some basic concepts related to coalitional games and to the notion of treewidth. The complexity of allocation games over arbitrary graphs is discussed in Section 3. Structural tractability results for matching and weighted-allocation games are presented in Section 4 and Section 5, respectively. The analysis of weighted-allocation games defined over graphs for which the degree of the nodes associated with goods is at most two is conducted in Section 6. Our prototype system and the experimental activity we have carried out are discussed in Section 7. Conclusions are eventually drawn in Section 8, where a number of avenues for further research are also illustrated.

2. Preliminaries

In this section, we recall some background concepts related to coalitional games and to structural decomposition methods that we shall use in the rest of the paper.

[2] A different approach has been used in [54], where we proved tractability only for classes of games whose *interaction* graphs (whose nodes are the agents and where edges connect those pair of agents that are interested in the same good) have bounded treewidth. To appreciate the significant improvement of the result presented in this extended version, observe that a bipartite graph with one good g connected to n distinct agents is actually a tree, but the corresponding interaction graph is the n-nodes clique, whose treewidth is $n-1$.

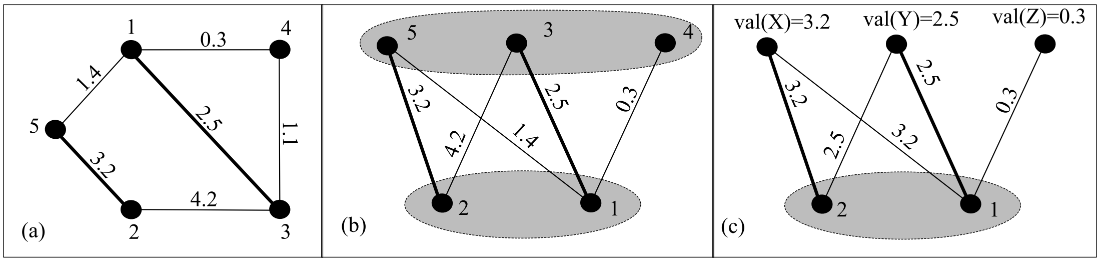


Fig. 1. Illustration of examples in Section 2: (a) the graph G' defining the weighted-matching game $\mathcal{G}^m_{G'}$; (b) the graph G'' defining $\mathcal{G}^m_{G''}$; and (c) the graph G''' defining the weighted-allocation game $\mathcal{G}^a_{G'''}$.

2.1. Weighted-matching and weighted-allocation games

A coalitional game can be formalized as a tuple $\mathcal{G} = \langle N, v\rangle$ where $N = \{1, ..., n\}$ is a set of agents and where v is a function associating each coalition $C \subseteq N$ with the worth $v(C) \in \mathbb{R}$ that agents in C can guarantee to themselves by collaborating with each other; in particular, the worth function satisfies $v(\emptyset) = 0$.

In the paper, we consider classes of coalitional games defined in terms of matching problems of graphs. To formalize these classes, let us first introduce some notions.

Let $G = (V, E, w)$ be a weighted graph, where V is a set of nodes, E is a set of (undirected) edges, and $w : E \mapsto \mathbb{R}$ is a function equipping each edge $e \in E$ with the value $w(e) \in \mathbb{R}^+ \cup \{0\}$. A set $M \subseteq E$ of edges is said to be a *matching* in the graph G if $e_1 \cap e_2 = \emptyset$, for each pair of distinct edges $e_1, e_2 \in M$. The *weight* of M is the value $\sum_{e\in M} w(e)$, shortly denoted as $w(M)$. If $w(M) \geq w(\bar{M})$ holds for each matching $\bar{M}$, then we say that M is a *maximum weighted* matching and we denote the value $w(M)$ as $\mathtt{max}(G)$. These notions are illustrated below.

Example 2.1. Consider the weighted graph $G' = (V', E', w')$ shown in Fig. 1(a), with $V' = \{1, 2, 3, 4, 5\}$ being the set of its nodes and with the matching $M = \{\{1, 3\}, \{2, 5\}\}$ being depicted in bold. Note that $w(M) = w(\{1, 3\}) + w(\{2, 5\}) = 2.5 + 3.2 = 5.7$. In fact, it can be checked easily that M is the maximum weighted matching in G' (so that $\mathtt{max}(G') = 5.7$). For instance, for the matching $\bar{M} = \{\{1, 5\}, \{2, 3\}\}$, we have $w(\bar{M}) = 5.6 < w(M)$. $\lhd$

Assume now that a weighted graph $G = (V, E, w)$ is given. The *weighted-matching game* induced by $G = (V, E, w)$ is the tuple $\mathcal{G}^m_G = \langle V, v^m_G\rangle$, where nodes are transparently viewed as agents and where v^m_G is the function associating with each coalition $C \subseteq V$ the value of the maximum matching of the subgraph $G[C]$ induced by the nodes in C, that is, $v^m_G(C) = \mathtt{max}(G[C])$.

Example 2.2. Consider the game $\mathcal{G}^m_{G'} = \langle V', v^m_{G'}\rangle$ induced by graph G' in Example 2.1: for the grand coalition V' we have $v^m_{G'}(V') = \mathtt{max}(G'[V']) = \mathtt{max}(G') = 5.7$, as witnessed by the maximum weighted matching shown in Fig. 1(a); instead, for the coalition $\{1, 2, 3\}$, we have $v^m_{G'}(\{1, 2, 3\}) = 4.2$, since $\{\{2, 3\}\}$ is the maximum weighted matching in the subgraph induced by the nodes $\{1, 2, 3\}$.

For another example, consider the game $\mathcal{G}^m_{G''} = \langle V'', v^m_{G''}\rangle$ induced by the weighted graph G'' shown in Fig. 1(b). Note that G'' is bipartite[3] and its "sides" are the sets $\{1, 2\}$ and $\{3, 4, 5\}$: for the grand coalition we have $v^m_{G''}(V'') = \mathtt{max}(G'') = 5.7$; for the coalition $\{1, 2, 4, 5\}$, we have $v^m_{G''}(\{1, 2, 4, 5\}) = 3.5$, because of the matching $M'' = \{\{1, 4\}, \{2, 5\}\}$. $\lhd$

Similarly to the case of the game $\mathcal{G}^m_{G''}$ discussed in Example 2.2, in *allocation games* we are given a weighted bipartite graph $G = (V_1 \cup V_2, E, w)$ such that $|e \cap V_1| = |e \cap V_2| = 1$ holds, for each $e \in E$. However, the two "sides" V_1 and V_2 consist of nodes of different kinds: V_1 is the set of agents, while V_2 is a set of goods (or whatever the agents are interested in). Moreover, in this paper, we consider weights interpreted as objective values of the goods, that is, values that depends only on properties of goods rather than on subjective valuations attributed to them by the various agents. These values can be viewed equivalently as weights on the nodes in V_2, or as weights on the edges of the graph, provided that the weights of all edges incident to any good $g \in V_2$ are equal. Such a weighted bipartite graph defines a *weighted-allocation game* $\mathcal{G}^a_G = \langle V_1, v^a_G\rangle$ where, for each coalition $C \subseteq V_1$, $v^a_G(C) = \mathtt{max}(G[C \cup V_2])$. That is, the worth of C is the one associated with the best way to match the nodes in C (agents) with the nodes of the other side V_2 (the goods).

Example 2.3. Consider the game $\mathcal{G}^a_{G'''} = \langle\{1, 2\}, v^a_{G'''}\rangle$ induced by the bipartite weighted graph G''' shown in Fig. 1(c), where 1 and 2 are agents and X, Y, and Z are goods (with some associated values). The edges shown in the figure mean that agent

[3] Weighted-matching games over bipartite graphs are usually called *assignment* games [29]. For completeness, recall that $G = (V, E, w)$ is bipartite if $V = V_1 \cup V_2$ and $|e \cap V_1| = |e \cap V_2| = 1$ holds, for each $e \in E$.

1 is interested in all goods, while agent 2 is interested only in goods X and Y. For the grand coalition, we have $v^a_{G'''}(\{1,2\}) = 5.7 = \mathtt{max}(G'''[\{1,2,X,Y,Z\}])$ because of the (global) optimum matching $M''' = \{\{2,X\},\{1,Y\}\}$, which in particular assigns the good Y with value 2.5 to agent 1. On the other hand, $v^a_{G'''}(\{1\}) = 3.2$, because in this case agent 1 can take the best good X, whose value is indeed 3.2. The same holds for agent 2, hence $v^a_{G'''}(\{2\}) = 3.2$, too. ◁

If the function w of a weighted graph $G = (V, E, w)$ is such that $w(e) = 1$, for each $e \in E$, then we say that G is an *unweighted graph* (or simply a graph) and, for brevity, we write $G = (V, E)$. In the following, the games discussed above are just called matching and allocation games, if defined over (unweighted) graphs.

2.2. Solution concepts for coalitional games

A fundamental problem for a coalitional game $\mathcal{G} = \langle N, v\rangle$ is to single out the most desirable outcomes, usually called *solution concepts*, in terms of appropriate notions of worth distributions, i.e., of payoff vectors of the form $(x_1, \ldots, x_{|N|}) \in \mathbb{R}^{|N|}$ where $x_i + \cdots + x_{|N|}$ equals the worth associated with the whole set N of agents. This question was studied in economics and game theory with the aim of providing arguments and counterarguments about why such proposals are reasonable mathematical renderings of the intuitive concepts of fairness and stability [55].

In the paper, we focus on the *Shapley value* [46], which is a well-known solution concept such that the payoff associated with each agent $i \in N$ is given by the following weighted average of her marginal contributions to all coalitions:

$$\phi_i(\mathcal{G}) = \sum_{C \subseteq N\setminus\{i\}} \frac{|C|!(|N|-|C|-1)!}{|N|!}\Big(v(C\cup\{i\}) - v(C)\Big),$$

where $v(C\cup\{i\}) - v(C)$ is the *marginal contribution* of i to the coalition $C \cup \{i\}$.

Example 2.4. For the allocation game $\mathcal{G}^a_{G'''}$ induced by the graph G''' shown in Fig. 1(c), the Shapley value of agent 1 is given by

$$\begin{aligned}\phi_1(\mathcal{G}^a_{G'''}) &= \tfrac{1!0!}{2!}\Big(v^a_{G'''}(\{1,2\}) - v^a_{G'''}(\{2\})\Big) + \tfrac{0!1!}{2!}\Big(v^a_{G'''}(\{1\}) - v^a_{G'''}(\emptyset)\Big) = \\ &= \tfrac{1}{2}\Big(5.7-3.2\Big) + \tfrac{1}{2}\Big(3.2-0\Big) = \tfrac{5.7}{2}\end{aligned}$$

As a further example, for a game $\hat{\mathcal{G}} = \langle N, \hat{v}\rangle$ defined over agents in $N = \{1,2,3\}$ and such that $\hat{v}(\{1,2\}) = \hat{v}(\{2,3\}) = \hat{v}(\{1,3\})$, $\hat{v}(\{1\}) = \hat{v}(\{2\}) = \hat{v}(\{3\})$ and $\hat{v}(\emptyset) = 0$, we have:

$$\begin{aligned}\phi_1(\hat{\mathcal{G}}) = &\tfrac{2!0!}{3!}\Big(\hat{v}(\{1,2,3\}) - \hat{v}(\{2,3\})\Big) + \tfrac{1!1!}{3!}\Big(\hat{v}(\{1,2\}) - \hat{v}(\{2\})\Big) + \\ &\tfrac{1!1!}{3!}\Big(\hat{v}(\{1,3\}) - \hat{v}(\{3\})\Big) + \tfrac{0!2!}{3!}\Big(\hat{v}(\{1\}) - \hat{v}(\emptyset)\Big) = \tfrac{\hat{v}(\{1,2,3\})}{3}\end{aligned}$$

Moreover, note that $\phi_1(\hat{\mathcal{G}}) = \phi_2(\hat{\mathcal{G}}) = \phi_3(\hat{\mathcal{G}})$. ◁

Distributions computed according to the Shapley value are often perceived as "fair", as they satisfy the following desirable properties:

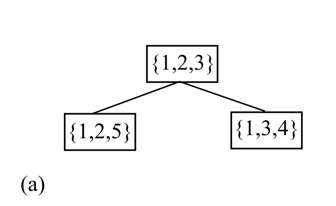
- $\phi_1(\mathcal{G}) + \cdots + \phi_{|N|}(\mathcal{G}) = v(\mathcal{G})$; *(efficiency)*
- If $v(C\cup\{i\}) - v(C) = 0$ for each $C \subseteq N$, then $\phi_i(\mathcal{G}) = 0$; *(nullity)*
- If $v(C\cup\{i\}) = v(C\cup\{j\})$ for each $C \subseteq N \setminus \{i,j\}$, then $\phi_i(\mathcal{G}) = \phi_j(\mathcal{G})$; *(symmetry)*
- If there are two games $\mathcal{G}_1 = \langle N, v_1\rangle$ and $\mathcal{G}_2 = \langle N, v_2\rangle$ such that $v(C) = v_1(C) + v_2(C)$ for each $C \subseteq N$, then $\phi_i(\mathcal{G}) = \phi_i(\mathcal{G}_1) + \phi_i(\mathcal{G}_2)$. *(additivity)*

A solution concept related to the Shapley value is the *Banzhaf value* [50], which for each agent i in the game $\mathcal{G} = \langle N, v\rangle$ is given by the (simple) average of her marginal contributions to all coalitions:

$$\beta_i(\mathcal{G}) = \frac{1}{2^{|N|-1}} \sum_{C \subseteq N\setminus\{i\}} \Big(v(C\cup\{i\}) - v(C)\Big).$$

2.3. Structural properties and MSO formulas

Many intractable problems are known to be solvable in polynomial time when restricted to classes of instances whose structures are represented as acyclic graphs or even hypergraphs. However, typical instances arising from real applications are hardly precisely acyclic. Yet, they are often not very intricate and, in fact, tend to exhibit some limited degree of cyclicity, which sometimes suffices to retain most of the nice properties of acyclic ones. Therefore, several efforts have been



$$\begin{aligned}\exists R,B,Y,\ & R\cup B\cup Y=N\wedge\\ & R\cap B=\emptyset\ \wedge\ R\cap Y=\emptyset\ \wedge\ B\cap Y=\emptyset\wedge\\ & \forall x,\ x\in B\rightarrow(\forall y,(x,y)\in r_E\rightarrow\neg(y\in B))\ \wedge\\ & \forall x,\ x\in R\rightarrow(\forall y,(x,y)\in r_E\rightarrow\neg(y\in R))\ \wedge\\ & \forall x,\ x\in Y\rightarrow(\forall y,(x,y)\in r_E\rightarrow\neg(y\in Y))\end{aligned}$$

(b)

Fig. 2. A width-2 tree decomposition of G' in Example 2.5 and the formula $\xi_{3\text{COL}}$ in Example 2.6.

spent to investigate invariants that are suited to identify nearly-acyclic (hyper)graphs, leading to the definition of a number of so-called *(purely) structural decomposition-methods* [56], such as the *tree decompositions* [49] and, after some years, the *(generalized) hypertree* [57], *fractional hypertree* [58], *component hypertree* [59], and *greedy hypertree* [60,61] decompositions.

These methods aim at transforming a given cyclic (hyper)graph into an acyclic one, by organizing its edges or its nodes into a polynomial number of clusters, and by suitably arranging these clusters as a tree, called decomposition tree. The original problem instance can then be evaluated over such a tree, with a cost that is exponential with respect to a measure of the complexity of the clusters, also called *width* of the decomposition, and polynomial if this width is bounded by some constant. In the paper, we focus on the treewidth notion, according to which the width of a decomposition is given by the maximum cardinality (minus one) of the set of nodes occurring in any of the clusters of the decomposition tree.

Formally, a *tree decomposition* of a graph[4] $G=(V,E)$ is a pair $\langle T,\chi\rangle$, where T is a tree and χ is a labeling function assigning to each vertex p in T a set of nodes $\chi(p)\subseteq V$, such that the following conditions are satisfied: (1) for each node $x\in V$, there exists p in T such that $x\in\chi(p)$; (2) for each edge $\{x,y\}\in E$, there exists p in T such that $\{x,y\}\subseteq\chi(p)$; and, (3) for each node $x\in V$, the subgraph of T induced by all nodes p such that $x\in\chi(p)$ is connected.

The *width* of $\langle T,\chi\rangle$ is the number $\max_{p\in T}(|\chi(p)|-1)$. The *treewidth* of G, denoted by $tw(G)$, is the minimum width over all its tree decompositions. Treewidth is known to be a true generalization of acyclicity: G is acyclic if and only if $tw(G)=1$. A class of graphs has bounded treewidth if there is some k with $tw(G)\leq k$, for every graph G in the class.

Example 2.5. Consider again the graph G' in Example 2.1. The treewidth of this graph is 2. Indeed, $tw(G')>1$ holds, because the graph contains cycles; and $tw(G)\leq 2$, because there is a tree decomposition, shown in Fig. 2(a), having width 2. ◁

A general approach to show tractability results over graphs having bounded treewidth is to exploit some deep connections that have been established in the literature to relate tree decompositions with logical sentences expressed in terms of *Monadic Second Order Logic (MSO)*.

Indeed, any graph $G=(V,E)$ can be viewed as a relational structure, whose domain is V and where the set of its edges is encoded as a binary (symmetric) relation $r_E=\bigcup_{\{x,y\}\in E}\{(x,y),(y,x)\}$. Over structures of this kind, we can build *monadic second order* (MSO) logic formulas by using the relation r_E, individual variables, the logical connectives $\vee$, $\wedge$, and $\neg$, and the quantifiers $\exists$ and $\forall$. Second order formulas allow the use of *node* and *edge* variables, ranging over the possible subsets of N and r_E, respectively, of the membership relation $\in$, and of the quantifiers $\exists$ and $\forall$ over node and edge variables. In addition, it is often convenient to use symbols like $=$, $\subseteq$, $\subset$, $\cap$, $\cup$, and $\rightarrow$ with their usual meaning, as abbreviations. The fact that a Monadic Second Order formula ξ holds over a graph G is denoted by $G\models\xi$.

Example 2.6. The well-known 3-coloring problem on a graph $G=(V,E)$ can be encoded via the MSO formula $\xi_{3\text{COL}}$ reported in Fig. 2(b): $G\models\xi_{3\text{COL}}$ if and only if there exists a partition of the nodes in V into three disjoint sets of nodes R, B, and Y (corresponding to nodes colored red, blue, and yellow) and such that no adjacent nodes take the same color. ◁

An important property of bounded treewidth graphs for MSO is given by Courcelle's theorem.

Theorem 2.7 *(cf. [51]). Let ξ be a fixed* MSO *sentence, let k be a fixed constant, and let $\mathcal{C}_k$ be a class of graphs having treewidth bounded by k. Then, for each $G\in\mathcal{C}_k$, deciding whether $G\models\xi$ is feasible in linear time.*

For instance, from the above theorem and Example 2.6, we can immediately conclude that 3-colorability is a property that can be checked in linear time on classes of graphs having bounded treewidth—in general, the problem is known to be NP-complete (see, e.g., [62]).

To our ends, we need a recently proposed histogram version of Courcelle's theorem, which is about counting solutions to MSO formulas having a given size, on graphs having bounded treewidth. Let $\xi(C)$ be a MSO sentence, where C is a *free*, i.e., not quantified, node set variable and, for each $\bar{C}\subseteq V$, let $\xi(\bar{C})$ denote the formula where C is fixed to be the set $\bar{C}$. For

[4] On weighted graphs, we shall just ignore the weight associated with the edges.

instance, if $\xi_{3\mathtt{COL}}(R)$ is the formula obtained from the one reported in Fig. 2(b) by dropping the existential quantification on R, then $\xi_{3\mathtt{COL}}(\{1,3\})$ asks whether the graph can be colored in a way that nodes 1 and 3, and only those nodes, get color red. Now, for each graph $G=(V,E)$ and natural number $d \in \{1,\ldots,|V|\}$, define $\mathtt{histogram}(\xi(C),G,d)$ as the number of subsets $\bar{C} \subseteq V$ such that $|\bar{C}|=d$ and $G \models \xi(\bar{C})$. For instance, $\mathtt{histogram}(\xi_{3\mathtt{COL}}(R),G,3)$ returns the number of the (valid) colorings where there are precisely three nodes having the color red. The following result shows that histogram can be computed in polynomial time whenever applied on classes of graphs having bounded treewidth.

Theorem 2.8 *(simplified from [63]). Let $\xi(C)$ be a fixed* MSO *sentence, let k be a fixed constant, and let $\mathcal{C}_k$ be a class of graphs having treewidth bounded by k. Then, for each $G=(V,E) \in \mathcal{C}_k$ and for each natural number $d \in \{1,\ldots,|V|\}$, computing* $\mathtt{histogram}(\xi(C),G,d)$ *is feasible in deterministic logspace (hence, polynomial time).*

Before leaving this section, we note that the tractability results we have mentioned have been proven in the literature by means of techniques that are entirely constructive and are, very roughly, based on dynamic programming algorithms that efficiently process tree decompositions in a bottom-up fashion (that is, from the leaves to an arbitrarily chosen root). Details on these algorithms can be found in the references we have mentioned above—see, also, [64] and the references therein. In fact, such algorithms have been implemented in systems that are already available to the research community, and on top of which we built our prototype (see Section 7).

3. Intractability of computation

In the last few years, the Shapley value has been intensively put under the lens of the computational complexity. Indeed, a number of works have pointed out the intrinsic complexity of computing this value over different compact encodings for coalitional games. For instance, we know that it is tractable on *induced subgraph games* [11], on games defined via *(read-once) marginal contribution networks* [65,66], on *multi-attribute coalitional games* [67], on games defined over *multi-issue domains* [68], and on a number of classes of games defined to formalize certain *game-theoretic centrality measures* [69] (see, also, [70] for further complexity results on such game-theoretic centrality measure and for fast algorithms for their computation). On the other hand, computing the Shapley value is known to be **NP**-hard on *network flow games* [22] and on *knapsack budgeted games* [71], while it is known to be **#P**-complete on a number of classes of games, including *minimum base games on matroids* [72] and *voting games* [11,73,74]. For the latter class of games approximation algorithms have been also studied [75], while fully polynomial-time randomized approximation scheme are known to exist for *supermodular coalitional games* [76].

Concerning the encodings we consider in the paper, it is known that the Shapley value is intractable, formally **#P**-complete, on matching games [47]. Moreover, the same complexity result has been proven to hold over weighted-allocation games [38].

In this section, we sharpen the result discussed above and we show that **#P**-hardness still holds[5] if we consider the restriction to the (unweighted) allocation games, which are induced by unweighted graphs. To establish the result, we proceed by first proving the intractability of the Banzhaf value.

Theorem 3.1. *On allocation games, computing the Banzhaf value is **#P**-hard.*

Proof. Let $G=(V_1 \cup V_2, E)$ be a bipartite graph with $V_1=\{1,\ldots,n\}$ and $V_1 \cap V_2 = \emptyset$. Recall that computing the number of subsets $C \subseteq V_1$ of nodes to which all nodes in V_2 can be *matched* is **#P**-hard [78].

Consider the bipartite graph $G'=(V_1 \cup \{n+1\} \cup V_2, E \cup \{\{n+1,x\} \mid x \in V_2\})$, which is obtained by adding to G' a novel fresh node $(n+1)$ connected to all nodes in V_2. Consider then the allocation game $\mathcal{G}^a_{G'} = \langle V_1 \cup \{n+1\}, v^a_{G'} \rangle$ induced by G', where $V_1 \cup \{n+1\}$ is the given set of agents. Observe that, for any given coalition $C \subseteq V_1$, $v^a_{G'}(C \cup \{n+1\}) - v^a_{G'}(C) = 0$ if and only if $C \subseteq V_1$ is a set of nodes to which all nodes in V_2 can be matched. Eventually, $\beta_{n+1}(\mathcal{G}^a_{G'}) \times 2^n$ is the number of subsets $C \subseteq V_1$ for which some node in V_2 cannot be matched, and $2^n - \beta_{n+1}(\mathcal{G}^a_{G'}) \times 2^n$ is the desired number, which can be computed in polynomial time once the Banzhaf value $\beta_{n+1}(\mathcal{G}^a_{G'})$ is known. □

Now, we show that the Banzhaf value of allocation games can be computed in polynomial time based on the knowledge of the Shapley value, so that this latter concept turns out to be **#P**-hard too. This property was already known to hold over (certain) *simple* games [79]. For its proof, we shall exploit the fact that, for each agent $i \in N$, the Shapley value can be rewritten as follows:

$$\phi_i(\mathcal{G}) = \sum_{h=0}^{n-1} \frac{h!(n-h-1)!}{n!} \beta_i(\mathcal{G},h), \tag{1}$$

[5] As commonly done in the literature, hardness results for **#P** are given under Turing reductions (see, e.g., [77]).

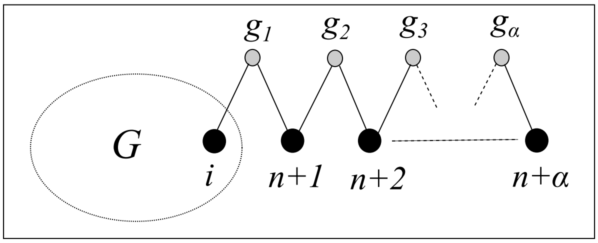


Fig. 3. Illustration in the proof of Theorem 3.2: Construction of the graph G_α based on G.

where, for each $h \in \{0, ..., n-1\}$,

$$\beta_i(\mathcal{G}, h) = \sum_{C \subseteq N \setminus \{i\},\ |C|=h} (v(C \cup \{i\}) - v(C)).$$

Theorem 3.2. *On allocation games, computing the Shapley value is* **#P***-hard.*

Proof. The line of the proof is to exhibit a reduction from the Banzhaf to the Shapley value.[6] Let $G = (V_1 \cup V_2, E)$ be a bipartite graph with $V_1 = \{1, ..., n\}$ and $V_1 \cap V_2 = \emptyset$. Let i be an agent in V_1 and, for each $\alpha \in \{1, ..., n\}$, consider the bipartite graph $G_\alpha = (V_1 \cup \{n+1, ..., n+\alpha\} \cup V_2 \cup \{g_1, ..., g_\alpha\}, E \cup E_\alpha)$ depicted in Fig. 3 and such that $E_\alpha = \{\{i, g_1\}, \{g_1, n+1\}, \{n+1, g_2\}, ..., \{g_\alpha, n+\alpha\}\}$. Consider the allocation games $\mathcal{G}^a_G = \langle V_1, v^a_G \rangle$ and $\mathcal{G}^a_{G_\alpha} = \langle V_1 \cup \{n+1, ..., n+\alpha\}, v^a_{G_\alpha} \rangle$ induced by G and G_α, respectively.

Let $C \subseteq V_1 \cup \{n+1, ..., (n+\alpha-1)\}$ and observe that if $\{i, n+1, ..., (n+\alpha-1)\} \not\subseteq C$, then $v^a_{G_\alpha}(C \cup \{n+\alpha\}) - v^a_{G_\alpha}(C) = 1$. Otherwise, i.e., if $\{i, n+1, ..., (n+\alpha-1)\} \subseteq C$, then $v^a_{G_\alpha}(C \cup \{n+\alpha\}) - v^a_{G_\alpha}(C) = v^a_G(C' \cup \{i\}) - v^a_G(C')$, where $C' = C \setminus \{i, n+1, ..., n+\alpha\}$. Therefore, for each $h \in \{0, ..., (n+\alpha-1)\}$, we have that:

$$\beta_{n+\alpha}(\mathcal{G}^a_{G_\alpha}, h) = \begin{cases} \kappa_{n+\alpha}(h) + \beta_i(\mathcal{G}^a_G, h-\alpha) & \text{if } h \geq \alpha \\ \kappa_{n+\alpha}(h) & \text{if } h < \alpha \end{cases}$$

where $\kappa_{n+\alpha}(h)$ is the number of coalitions C such that $|C| = h$ and such that $\{i, n+1, ..., (n+\alpha-1)\} \not\subseteq C$. Note that $\kappa_{n+\alpha}(h)$ can be computed in polynomial time.

By using the above expression for $\beta_{n+\alpha}(\mathcal{G}^a_{G_\alpha}, h)$ in Equation (1), we can derive that:

$$\phi_{n+\alpha}(\mathcal{G}^a_{G_\alpha}) = \sum_{h=0}^{n+\alpha-1} \frac{h!(n+\alpha-h-1)!}{(n+\alpha)!} \kappa_{n+\alpha}(h) + \sum_{h=0}^{n-1} \frac{(h+\alpha)!(n-h-1)!}{(n+\alpha)!} \beta_i(\mathcal{G}^a_G, h).$$

Eventually, by instantiating α with each value in $\{1, ..., n\}$, we get a system of n linear equations over the variables $\beta_i(\mathcal{G}^a_G, 0), ..., \beta_i(\mathcal{G}^a_G, n-1)$. By multiplying by $(n+\alpha)!$ all the terms of the corresponding equation, we get a system where $(h+\alpha)!(n-h-1)!$ is the coefficient of any term of the form $\beta_i(\mathcal{G}^a_G, h)$. Given the form of these coefficients, the equations are easily seen to be linear independent—see [47,79].

Therefore, the values of $\beta_i(\mathcal{G}^a_G, 0), ..., \beta_i(\mathcal{G}^a_G, n-1)$ for which the system admits a solution are univocally determined. Moreover, they might be computed in polynomial time (e.g., by Bareiss's implementation of Gaussian elimination [80]), if we were able to compute in polynomial time $\phi_{n+\alpha}(\mathcal{G}^a_{G_\alpha})$, for each $\alpha \in \{1, ..., n\}$. To conclude, just notice that the Banzhaf value of agent i in $\mathcal{G}_\mathcal{A}$ is precisely given by $\beta_i(\mathcal{G}^a_G) = \frac{1}{2^{n-1}} \sum_{h=0}^{n-1} \beta_i(\mathcal{G}^a_G, h)$. □

4. Matching games over graphs of bounded treewidth

The question of whether the Shapley value is tractable on classes of graphs having bounded treewidth was stated as an open problem in earlier literature [47]. In this section, we provide a positive answer to the question by exhibiting a rather involved algorithm that exploits three distinct ingredients. First, in Section 4.1 we establish some graph-theoretic results on matchings on graphs. After these results, we are able to get rid of the optimization problems, and to identify instead in Section 4.2 a different way to deal with marginal contributions in matching games. Finally, in Section 4.3 we show that the Shapley value (and the Banzhaf value) can be computed by counting suitable solutions (corresponding to coalitions) of an MSO formula, so that tractability can be established via the histogram-based version of Courcelle's Theorem (cf. Theorem 2.8).

[6] Note that the arguments in the proof can be used to prove similar reductions on different classes of games, provided there is a way to extend the game at hand with a single player that has a fixed effect on the Banzhaf value.

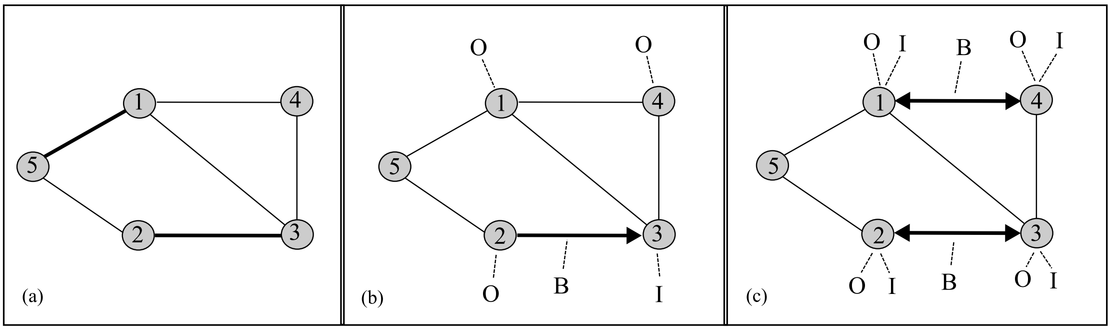


Fig. 4. Illustration of examples in Section 4: (a) the graph G and the matching M, identified by the edges depicted in bold; (b) a matching that does not block node 5; and (c) a maximum matching that blocks node 5.

4.1. Basic results on matchings

Assume hereinafter that a graph $G = (V, E)$ is given. Let M be a matching in G. A node $x \in V$ is said *M-matched* if there is some edge $e \in M$ with $x \in e$; otherwise, x is *M-exposed*. A path is a sequence of distinct nodes $\pi_1, \ldots, \pi_m$, with $m > 1$, such that $\{\pi_j, \pi_{j+1}\} \in E$, for each $j \in \{1, \ldots, m-1\}$. The path is said *M-alternating* if each pair of adjacent edges satisfy $\{\pi_j, \pi_{j+1}\} \in M$ and $\{\pi_{j+1}, \pi_{j+2}\} \notin M$, or vice-versa ($j \in \{1, \ldots, m-2\}$). Moreover, the path is *M-augmenting* if it is M-alternating, and π_1 and π_m are M-exposed; in fact, they are the only M-exposed nodes occurring in the path.

Note that M is a *maximum matching* (i.e., one including the maximum possible number of edges over all matchings) if and only if there is no M-augmenting path [81].

Example 4.1. Consider the graph G and the matching $M = \{\{1, 5\}, \{2, 3\}\}$ shown in Fig. 4(a). Note that node 4 is M-exposed and, for instance, the path 5, 1, 4 is M-alternating. However, node 5 is M-matched and thus the path is not M-augmenting. In fact, there is no M-augmenting path, since the reader can check that M is a maximum matching. $\lhd$

The following definition introduces a crucial notion for analyzing matching games.

Definition 4.2. A node $x \in V$ is *blocked* by a matching M if

- x is M-exposed, and
- for each M-alternating path $\pi_1, \ldots, \pi_m$ with $\pi_1 = x$, it holds that π_m is M-matched.

Example 4.3. Consider again the setting of Example 4.1. It is easy to check that node 5 is not blocked by the matching $M = \{\{1, 5\}, \{2, 3\}\}$. Moreover, note that 5 is not blocked by the matching $\{\{2, 3\}\}$, while it is blocked by $\{\{1, 4\}, \{2, 3\}\}$. $\lhd$

Actually, since any single edge is an alternating path, the following is an equivalent characterization of Definition 4.2.

Fact 4.4. A node $x \in V$ is blocked by M if and only if for each M-alternating path $\pi_1, \ldots, \pi_m$ with $\pi_1 = x$, it holds that π_m is M-matched and $\{\pi_1, \pi_m\} \notin M$.

We next show that, for this notion, it is not necessary to focus on maximum matchings.

Lemma 4.5. *A node $i \in V$ is blocked by some maximum matching if and only if it is blocked by some matching.*

Proof. Assume that i is blocked by a matching M_1 for which there exists an M_1-augmenting path $\pi'_1, \ldots, \pi'_h$. Consider the matching M_2 obtained from M_1 by removing all edges traversed by $\pi'_1, \ldots, \pi'_h$ and by adding all edges in $E \setminus M_1$ traversed by $\pi'_1, \ldots, \pi'_h$. That is, M_2 is the matching obtained by "augmenting" M_1 via the path $\pi'_1, \ldots, \pi'_h$. Clearly, $|M_2| > |M_1|$. Moreover, note that each node π'_j, with $j \in \{1, \ldots, h\}$, is not M_2-exposed. Now, let $\bar{\pi}_1, \ldots, \bar{\pi}_q$ be any M_2-alternating path with $\bar{\pi}_1 = i$. We shall show that the node $\bar{\pi}_q$ is not M_2-exposed and $\{\bar{\pi}_1, \bar{\pi}_q\} \notin M_2$.

Claim 4.5.(A). *$\bar{\pi}_1, \ldots, \bar{\pi}_q$ and $\pi'_1, \ldots, \pi'_h$ do not share any edge.* By contradiction, consider the smallest index $k \in \{1, \ldots, q-1\}$ and an index $j \in \{1, \ldots, h-1\}$ such that $\pi'_j = \bar{\pi}_k$ and $\pi'_{j+1} = \bar{\pi}_{k+1}$. Note that if $k = 1$, then the path $\bar{\pi}_k, \pi'_{j+1}, \ldots, \pi'_h$ would witness that i is not blocked by M_1, which is impossible. Indeed, we would have that $\bar{\pi}_k = i$ and that $\bar{\pi}_k, \pi'_{j+1}, \ldots, \pi'_h$ is M_1-alternating, with π'_h being M_1-exposed (for this latter property, just recall that $\pi'_1, \ldots, \pi'_h$ is M_1-augmenting). This means that we necessarily have $k > 1$, and that $\bar{\pi}_1, \ldots, \bar{\pi}_k$ is an M_1-alternating path. Now, in the case where $j = 1$ holds, we have

$\bar{\pi}_k = \pi'_1$. Therefore, the path $\bar{\pi}_1, ..., \bar{\pi}_k$ would again witness that i is not blocked by M_1, which is impossible. So, we have $j > 1$. Hence, the node $\bar{\pi}_k = \pi'_j$ is adjacent to the nodes $\bar{\pi}_{k-1}$, π'_{j-1}, and $\bar{\pi}_{k+1} = \pi'_{j+1}$. Recall that $\pi'_1, ..., \pi'_h$ is M_1-alternating and assume first that the edge $\{\pi'_j, \pi'_{j-1}\}$ is in M_1. Then, $\{\bar{\pi}_{k-1}, \pi'_j\} \notin M_1$ because π'_j $(= \bar{\pi}_k)$ is matched to someone else. Because M_1 and M_2 are equal for edges in the path $\bar{\pi}_1, ..., \bar{\pi}_k$ below k, it follows that $\bar{\pi}_1, ..., \bar{\pi}_{k-1}, \pi'_j, \pi'_{j-1}, ..., \pi'_1$ is M_1-alternating. The same holds if we assume instead that the edge $\{\pi'_j, \pi'_{j-1}\}$ is not in M_1. Indeed, in this case $\{\pi'_j, \pi'_{j+1}\} \in M_1$ and thus, by definition of M_2, $\{\bar{\pi}_k, \bar{\pi}_{k+1}\} \notin M_2$ and $\{\bar{\pi}_{k-1}, \bar{\pi}_k\} \in M_2$ by alternation, that is, $\{\bar{\pi}_{k-1}, \pi'_j\} \in M_1$, because the two matchings are equal below k. Again, we get that $\bar{\pi}_1, ..., \bar{\pi}_{k-1}, \pi'_j, \pi'_{j-1}, ..., \pi'_1$ is M_1-alternating. Since π'_1 is M_1-exposed, this contradicts that i is blocked by M_1. ◇

Given the above claim, we derive that $\bar{\pi}_1, ..., \bar{\pi}_q$ is also M_1-alternating; hence, $\{\bar{\pi}_1, \bar{\pi}_q\} \notin M_1$. It then follows that $\{\bar{\pi}_1, \bar{\pi}_q\} \notin M_2$, because we have observed that $\bar{\pi}_1 = i$ cannot occur in the path $\pi'_1, ..., \pi'_h$ and thus cannot become M_2-matched. To conclude, we show that $\bar{\pi}_q$ is M_2-matched. This is trivial if $\bar{\pi}_q \in \{\pi'_1, ..., \pi'_h\}$. Otherwise, if $\bar{\pi}_q \notin \{\pi'_1, ..., \pi'_h\}$, then $\bar{\pi}_q$ is M_2-matched because it is M_1-matched and it does not occur in the path where the two matchings may differ.

Now, either M_2 is a maximum matching, or we can repeat the above construction, until we eventually get a sequence $M_1, M_2, ..., M_\ell$ of matchings such that i is blocked by each of them and $|M_1| < |M_2| < ... < |M_\ell|$, with the latter being a maximum matching of the given graph. □

Example 4.6. Recall from Example 4.3 that node 5 is blocked by the matching $\{\{1, 4\}, \{2, 3\}\}$. This is actually a maximum matching. In fact, it can be checked that $\{\{1, 4\}, \{2, 3\}\}$ is the only matching blocking node 5. ◁

4.2. Characterization of marginal contribution

Given the graph $G = (V, E)$, recall from Section 2 that $\mathcal{G}^m_G = (V, v^m_G)$ denotes the matching game induced by G. The second ingredient of our proof is a connection between the value of the marginal contribution $v^m_G(C \cup \{i\}) - v^m_G(C)$ and the fact that agent $i \in V$ is blocked by some matching in the subgraph $G[C \cup \{i\}]$ (induced over the nodes in $C \cup \{i\}$, where $C \subseteq V$).

Lemma 4.7. *The following statements are equivalent:*

(1) $v^m_G(C \cup \{i\}) - v^m_G(C) = 0$;
(2) *i is blocked by some matching M in $G[C \cup \{i\}]$.*

Proof. *(1)⇒(2)* Let M be a maximum matching in $G[C]$. Since $v^m_G(C \cup \{i\}) - v^m_G(C) = 0$, M is also a maximum matching in $G[C \cup \{i\}]$. Consider, then, any M-alternating path $\pi_1, ..., \pi_m$ in $G[C \cup \{i\}]$ with $\pi_1 = i$. Since M is a matching in $G[C]$ and $i \notin C$, then π_1 is clearly M-exposed in $G[C \cup \{i\}]$. In particular, $\{\pi_1, \pi_m\} \notin M$. Assume now, by contradiction, that π_m is M-exposed, too. Then, $\pi_1, ..., \pi_m$ is M-augmenting in $G[C \cup \{i\}]$, which is impossible.

(2)⇒(1) Because of Lemma 4.5, we can assume that M is a maximum matching in $G[C \cup \{i\}]$. Because i is blocked by M, we know that i must be M-exposed. Then, M is also a maximum matching in $G[C]$ and we have concluded, since $v^m_G(C \cup \{i\}) - v^m_G(C) = 0$ holds. □

In fact, we can continue the chain of equivalences of Lemma 4.7, by providing a number of conditions that can easily be checked. To this end, recall that the edges of G can be encoded as the binary (symmetric) relation r_E defined as the set of pairs $\bigcup_{\{x,y\} \in E} \{(x, y), (y, x)\}$.

Lemma 4.8. *The following statements are equivalent:*

(2) *i is blocked by some matching M in $G[C \cup \{i\}]$;*
(3) *There are sets $B \subseteq r_E$, $I \subseteq V$, and $O \subseteq V$ satisfying on G the following conditions:*
  (a) $i \notin I$;
  (b) $\forall x, y \in C \cup \{i\},\ \big((x, y) \in B \big) \rightarrow (y \in I)$;
  (c) $\forall x \in C \cup \{i\}, \forall y \in C,\ \big((x, y) \in r_E \wedge (x \in I) \big) \rightarrow (y \in O)$;
  (d) $\forall x \in C \cup \{i\},\ x \in O \rightarrow \Big(\exists z \in C \text{ such that } (x, z) \in B \Big)$;
  (e) $\forall x, y, y' \in C \cup \{i\},\ \Big(\big((x, y) \in B \wedge (x, y') \in B\big) \vee \big((y, x) \in B \wedge (y', x) \in B\big) \vee \big((y, x) \in B \wedge (x, y') \in B\big) \Big) \rightarrow (y = y')$;
  (f) $\forall x \in C,\ (i, x) \in r_E \rightarrow \Big(\big((i, x) \notin B \big) \wedge (x \in O) \Big)$.

Before detailing the proof of the result, we provide an intuition on the above conditions (encoding that i is blocked by some matching M in $G[C \cup \{i\}]$) and we propose an exemplification.

A crucial role in conditions **(a)**,...,**(f)** is played by the set B. This set consists of the edges in M, which are however oriented according to the way they are traversed along some M-alternating path starting at node i; in particular, note that condition **(e)** states that, by looking at the undirected version of B, every node cannot have more than one incident edge. The remaining conditions are instead meant to check that i is M-exposed and that every M-alternating path starting at i terminates in a node that is M-matched. Concerning the latter property, note that because of condition **(b)** the set I includes all destination nodes for edges in B, that is, I contains all M-matched nodes that can be reached by traversing some M-alternating path starting at i. Of course, starting from a node in I, we might extend the given M-alternating path by traversing one further edge. The set O then includes all destination nodes of such edges **(c)**, and we require that such nodes are M-matched too **(d)**. Finally, given the above definitions of I and O, it is immediate to check that conditions **(a)**, **(b)** and **(f)** guarantee that i is M-exposed.

Example 4.9. Consider the coalition $C = \{1, 2, 3, 4\}$ and Fig. 4, which shows two examples of sets B, I, and O. Intuitively, B encodes a matching, whose edges are however oriented according to the way they are traversed along some alternating path starting at node 5. Note that certain edges can be traversed in both directions. Indeed, for the matching $\{\{1, 4\}, \{2, 3\}\}$, the paths $5, 1, 4, 3, 2$ and $5, 2, 3, 4, 1$ witness that the two edges $\{1, 4\}$ and $\{2, 3\}$ are traversed in both directions.

Note that the sets in Fig. 4(b), with $B = \{(2, 3)\}$ violate condition **(d)**. Instead, the sets in Fig. 4(c), with $B = \{(1, 4), (2, 3), (4, 1), (3, 2)\}$ satisfy all conditions and, by Lemma 4.8, they witness that node 5 is blocked by some matching (in fact, precisely by the maximum matching $\{\{1, 4\}, \{2, 3\}\}$). $\lhd$

Proof of Lemma 4.8. *(2)⇒(3)* Assume that i is blocked by some matching M in $G[C \cup \{i\}]$, and consider the sets B, I, and O built as follows. For each edge $\{x, y\}$ in $G[C \cup \{i\}]$ if there is an M-alternating path $\pi_1, ..., \pi_m$ in this subgraph with $\pi_1 = i$ and such that $x = \pi_j$ and $y = \pi_{j+1}$, for some index $j \in \{1, ..., m-1\}$, and $\{x, y\}$ is in M, then (x, y) is in B, $x \in O$, and $y \in I$. For each edge $\{i, y\} \in E \setminus M$, we set $y \in O$. No other element is in B, I, or O. Note that the sets trivially satisfy (a), (b), and (e).

Concerning (c), let $\pi_1, ..., \pi_m$ be an M-alternating path in $G[C \cup \{i\}]$ with $\pi_1 = i$ such that, for some index $j \in \{2, ..., m\}$, $\pi_j = x$ with $x \in I$ and $\{\pi_{j-1}, \pi_j\} \in M$ hold. Note that $\pi_{j-1} \in O$. Then, assume that $\{x, y\}$ is in E with $y \in C$ and $y \neq \pi_{j-1}$. We have that $\{x, y\} \notin M$ and $\pi_1, ..., \pi_j, y$ is M-alternating. Since i is blocked by M, y is M-matched and $\{\pi_1, y\} \notin M$. Therefore, there is an edge $\{y, z\} \in M$ and the path $\pi_1, ..., \pi_j, y, z$ is M-alternating, too. By construction, this entails that $y \in O$.

Concerning (d), assume first that there is an M-alternating path $\pi_1, ..., \pi_m$ with $\pi_1 = i$ such that, for some index $j \in \{1, ..., m-1\}$, $\pi_j = x$ with $x \in O$ and where $\{\pi_j, \pi_{j+1}\} \in M$ holds. In this case, $(x, \pi_{j+1}) \in B$ holds, by construction. The other possibility is that $\{i, x\} \in E \setminus M$. In this case, since i is blocked by M, there must be a node $z \neq i$ such that $\{x, z\} \in M$. Again, by construction, we have $(x, z) \in B$.

Concerning (f), note that there is no $x \in V$ with $\{i, x\} \in M$; otherwise, the path i, x would witness that i is not blocked by M. By construction, we hence have $(i, x) \notin B$ and $x \in O$.

(3)⇒(2) Assume that B, I, and O satisfy all the properties listed in the statement. Consider the set $M = \{\{x, y\} \mid (x, y) \in B, x \in C \cup \{i\}, y \in C \cup \{i\}\}$. Because of (e), it is immediate that M is a matching in $G[C \cup \{i\}]$. Moreover, by (a), (b), and (f), we derive that there is no node $y \in V$ such that $\{i, y\}$ is in M, that is, i is M-exposed. Consider then any M-alternating path $\pi_1, ..., \pi_m$ of $G[C \cup \{i\}]$ with $\pi_1 = i$. We shall show that π_m is M-matched.

To this end, we first claim that, for each natural number $h \geq 1$ such that $2h + 1 \leq m$, it holds that $\{\pi_{2h-1}, \pi_{2h}\} \notin M$, $(\pi_{2h}, \pi_{2h+1}) \in B$, $\pi_{2h} \in O$. For $h = 1$, we have just to observe that $\{\pi_1, \pi_2\} \notin M$ and, by (f) $\pi_2 \in O$. Moreover, from (d) we necessarily have $(\pi_2, z) \in B$ with $z = \pi_3$, because $\{\pi_2, \pi_3\} \in M$, by alternation, and (e) must hold. So, let us assume that the property holds at some natural number h, and we show that it holds at $h + 1$ with $2(h + 1) + 1 \leq m$, too. Indeed, we know that $(\pi_{2h}, \pi_{2h+1}) \in B$ and hence $\{\pi_{2h}, \pi_{2h+1}\} \in M$, by construction. Moreover, by (b), $\pi_{2h+1} \in I$. By (c), $\pi_{2h+2} \in O$. By alternation, $\{\pi_{2h+1}, \pi_{2h+2}\} \notin M$. Then, as in the base case, by (d) and (e), $(\pi_{2h+2}, \pi_{2h+3}) \in B$ and $\{\pi_{2h+2}, \pi_{2h+3}\} \in M$.

In the light of this result, if m is odd, then we can immediately conclude that π_m is M-matched. Otherwise, we know that $(\pi_{m-2}, \pi_{m-1}) \in B$. By (b), $\pi_{m-1} \in I$. By (c), $\pi_m \in O$. By (d), there is a node z such that $(\pi_m, z) \in B$. Then, we conclude that π_m is M-matched. $\square$

4.3. MSO encoding

We are now ready to discuss the third ingredient of our elaboration, namely the use of tractability results for the evaluation of monadic second-order formulas on bounded-treewidth graphs.

Theorem 4.10. *On matching games defined over classes of graphs having bounded treewidth, the Shapley value can be computed in polynomial time.*

Proof. Let $G = (V, E)$ be a graph. Consider Equation (1) in Section 3 and note that, for each agent $i \in V$, the Shapley value of the matching game $\mathcal{G}_G^m = \langle V, v_G^m \rangle$ can be computed in polynomial time if we are able to compute in polynomial time, for each coalition cardinality $h \in \{1, \ldots, |V| - 1\}$, the values

$$\beta_i(\mathcal{G}_G^m, h) = \sum_{C \subseteq V \setminus \{i\},\ |C|=h} \Big(v_G^m(C \cup \{i\}) - v_G^m(C)\Big).$$

We do that by defining a Monadic Second Order (MSO) formula $F_i(C) = \exists B, I, O\ \Phi$, to be evaluated on the graph G, viewed as a relational structure with relation symbol r_E and domain V. In this second-order formula, Φ is the conjunction of (the first-order) Conditions **(a)**–**(f)** in Lemma 4.8, B is an existentially-quantified edge-set variable, I and O are existentially-quantified node-set variables, and C is a free node-set variable. Then, given Lemma 4.7 and Lemma 4.8, the number of coalitions $C \subseteq V \setminus \{i\}$ having $|C| = h$ for which $v_G^m(C \cup \{i\}) - v_G^m(C) = 0$ holds coincides with the number of node-sets $\hat{C} \subseteq V \setminus \{i\}$ with $|\hat{C}| = h$ for which $G \models F_i(\hat{C})$. This number can be computed in polynomial time because of Theorem 2.8 and, based on it, $\beta_i(\mathcal{G}_G^m, h)$ can be computed easily (in polynomial time, too) by observing that $v_G^m(C \cup \{i\}) - v_G^m(C) \in \{0, 1\}$, for each $C \subseteq V \setminus \{i\}$. □

A similar approach can be used to prove the tractability of the Banzhaf value on graphs having bounded treewidth. In this case we are even able to derive a linear time algorithm, rather than a polynomial-time one.

Theorem 4.11. *On matching games defined over classes of graphs having bounded treewidth, the Banzhaf value can be computed in linear time.*

Proof. We shall show that, for each node $i \in V$, we can compute in linear time the value of the expression $\sum_{C \subseteq V \setminus \{i\}} (v_G^m(C \cup \{i\}) - v_G^m(C))$. Indeed, we can proceed as in the proof of Theorem 4.10, by eventually noticing that the number of coalitions $C \subseteq V \setminus \{i\}$ for which $v_G^m(C \cup \{i\}) - v_G^m(C) = 0$ holds coincides with the number of node-sets $\hat{C} \subseteq V \setminus \{i\}$ for which $G \models F_i(\hat{C})$. Then, the statement follows easily, by recalling that counting all instantiations of a free node-set variable leading to satisfy a given MSO formula was shown to be feasible in linear time over graphs having bounded treewidth [82]. □

5. Weighted-allocation games over graphs of bounded treewidth

In this section we show that the Shapley value can be computed in polynomial time over weighted-allocation games induced by graphs having bounded treewidth. The high-level idea is similar to the one we have exhibited for matching games in Section 4: we need some way to avoid the computation of maximum matching problems for evaluating the marginal contributions to the various coalitions and, instead, identify suitable properties of the coalitions that can be checked by a Monadic Second Order formula. Then, the desired values can be computed by counting solutions of MSO formulas over bounded-treewidth graphs, which is feasible in polynomial time. Contrasted with the properties used in the previous section, the technical machinery developed for dealing with weighted-allocation problems is quite different.

Recall that we deal with weighted-allocation games induced by bipartite graphs of the form $G = (N \cup \mathbb{G}, E, w)$ where $N = \{1, \ldots, n\}$ encodes a set of agents, $\mathbb{G}$ encodes a set of goods (or whatever the agents are interested in), and each good $g \in \mathbb{G}$ is characterized by a value, denoted by $\texttt{val}(g)$, which depends on its properties only, rather than by subjective valuations from the different agents. Formally, all edges incident to g have the same weight, that is, $w(e) = w(e') = \texttt{val}(g)$, for each pair of edges with $e \cap e' = \{g\}$.[7] By slightly abusing notation, whenever $S \subseteq \mathbb{G}$ is a set of goods, $\texttt{val}(S)$ denotes the sum of their values (with $\texttt{val}(\emptyset) = 0$, by convention).

In this section any matching M, which is by definition a set of edges, will be viewed also as a function mapping each agent $i \in N$ to the set $\{g \in \mathbb{G} \mid \{i, g\} \in M\}$ containing the good allocated to i, if any. Thus, $M(i)$ is either a singleton or the empty set. Moreover, let $\texttt{img}(M) = \{g \in \mathbb{G} \mid M(i) = \{g\} \text{ for some } i \in N\}$, and denote by $\texttt{val}(M)$ the sum of the values of these goods.

Example 5.1. Consider the weighted graph $G = (N \cup \mathbb{G}, E, w)$ shown in Fig. 5(a), where $N = \{1, 2, 3, 4, 5\}$ and $\mathbb{G} = \{g_1, g_2, g_3\}$. The edges in bold in the figure depict the matching M with $M(1) = \{g_2\}$, $M(4) = \{g_3\}$, and $M(2) = M(3) = M(5) = \emptyset$. Note that $\texttt{val}(M) = \texttt{val}(g_2) + \texttt{val}(g_3) = 5.3 + 3.2$. ◁

In the following, let $\{w_1, \ldots, w_m\} = \{\texttt{val}(g) \mid g \in \mathbb{G}\} \cup \{0\}$ be the set of all values associated with goods in $\mathbb{G}$ (plus the null value 0, if not present), and assume that $w_1 < w_2 < \cdots < w_m$, with $w_1 = 0$. Plain (unweighted) allocation games correspond to the special case of this setting where $m = 2$ and $w_2 = 1$, that is, we have only the binary values $\{0, 1\}$.

5.1. Dependent agents and marginal contributions

The first ingredient we put in place for establishing the desired tractability result is a property that allows us to get rid of the optimization problems needed to obtain the maximum weighted matching involved in the computation of marginal contributions. Here, a crucial gadget is provided by the following definition, which allows us to identify a set of agents that "depends" on the agent i, when considering a coalition C and a matching M.

[7] We assume, w.l.o.g., that each good $g \in \mathbb{G}$ has at least one incident edge.

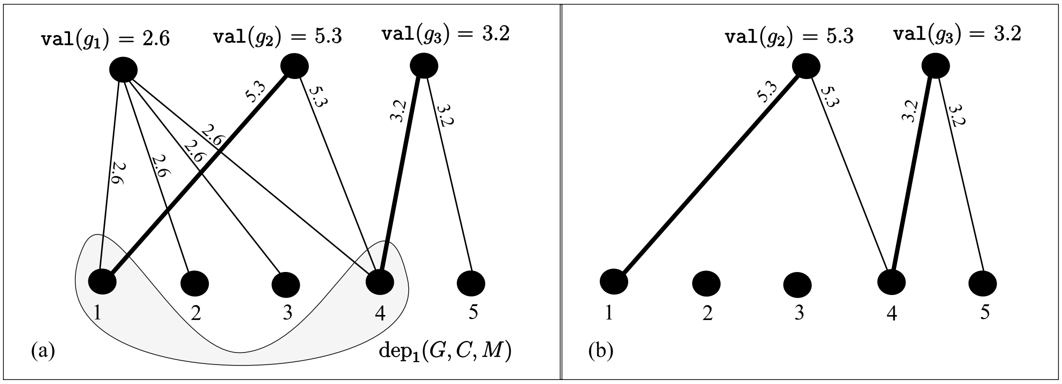


Fig. 5. Illustration of examples in Section 5: (a) the graph G and the matching M; and (b) the graph $G_{3.2}$.

Definition 5.2. Let $C \subseteq N$ be a set of agents and let M be a matching in $G = (N \cup \mathbb{G}, E, w)$. Then, the set $\text{dep}_i(G, C, M)$ of agents that are *dependent on player $i \in N$ with respect to C and M* is the smallest set satisfying the following properties:

1. $i \in \text{dep}_i(G, C, M)$;
2. if $j \in \text{dep}_i(G, C, M)$, $M(j) = \{g\}$, $z \in C$, and $\{g, z\} \in E$, then $z \in \text{dep}_i(G, C, M)$.

The second property is also called the *closure condition* of the dependency set.

Example 5.3. Consider again the weighted graph G and the matching M shown in Fig. 5(a), plus the coalition $C = \{1, 2, 3, 4\}$. Then, $\text{dep}_1(G, C, M) = \{1, 4\}$. Indeed, agent 1 belongs to $\text{dep}_1(G, C, M)$ by definition, $M(1) = \{g_2\}$ and $\{g_2, 4\}$ is in E. Note that agent 5 does not belong to $\text{dep}_1(G, C, M)$ because 5 does not belong to the coalition C (otherwise it would be included in the set, too, by the closure condition). $\lhd$

We next show that the existence of a matching for such dependent agents where all of them get goods having value at least $\sigma > 0$ is a necessary and sufficient condition to say that the marginal contribution of i to C is at least σ. To prove the result, we need to recall a property that is already known to hold on allocation games and to elaborate a novel technical lemma—which is of interest in its own in the context of matching theory and is, indeed, stated here in a way more general than it is actually required for our subsequent elaborations.

Lemma 5.4 *(cf. [34,38]). Let C and C' be sets of agents such that $C' \subseteq C \subseteq N$. Then, $v^a_G(C) \geq v^a_G(C')$. Moreover, if M is a maximum weighted matching in $G[C \cup \mathbb{G}]$, then there is a maximum weighted matching M' in $G[C' \cup \mathbb{G}]$ such that $\texttt{img}(M') \subseteq \texttt{img}(M)$.*

Lemma 5.5. *Let $C \subseteq N$, let $i \in N \setminus C$, let $\sigma \in \mathbb{R}$ be a real number with $\sigma > 0$, and assume there is a matching M in $G[C \cup \{i\} \cup \mathbb{G}]$ such that $\texttt{val}(M(j)) \geq \sigma$, for each $j \in \text{dep}_i(G, C, M)$. Then, every maximum weighted matching $\bar{M}$ in $G[C \cup \{i\} \cup \mathbb{G}]$ is such that $\texttt{val}(\bar{M}(j)) \geq \sigma$ holds, for each $j \in \text{dep}_i(G, C, \bar{M})$.*

Proof. Consider any maximum weighted matching $\bar{M}$ in $G[C \cup \{i\} \cup \mathbb{G}]$. Notice preliminarily that, for each agent $z \in C \setminus \text{dep}_i(G, C, M)$, $\bar{M}(z)$ cannot be a singleton $\{\hat{g}\}$ where $\hat{g} = M(\hat{z})$ holds for some agent $\hat{z} \in \text{dep}_i(G, C, M)$. Otherwise, this would entail that z belongs to $\text{dep}_i(G, C, M)$, by the closure property of this set. We now claim the following two properties.

Claim 5.5.(A). *It holds that* $\texttt{val}(\bar{M}(i)) \geq \sigma$. Assume by contradiction that $\texttt{val}(\bar{M}(i)) < \sigma$. We claim that there is a sequence of agents $j_1, \ldots, j_h$ belonging to $\text{dep}_i(G, C, M)$ and starting at $i = j_1 \in \text{dep}_i(G, C, M)$ such that:

1. for each index $x \in \{1, \ldots, h\text{-}1\}$, $\bar{M}(j_{x+1}) = M(j_x)$;
2. $M(j_h) \cap \texttt{img}(\bar{M}) = \emptyset$.

Note first that $M(j_1) \neq \emptyset$, since $j_1 \in \text{dep}_i(G, C, M)$ and, hence, $\texttt{val}(M(j_1)) \geq \sigma > 0$. Let $\{g_1\} = M(j_1)$ and recall that g_1 cannot be assigned by $\bar{M}$ to any agent outside $\text{dep}_i(G, C, M)$. Then, either $\{g_1\} \cap \texttt{img}(\bar{M}) = \emptyset$, or g_1 is assigned to some agent, say $j_2 \in \text{dep}_i(G, C, M)$, for which we have $\bar{M}(j_2) = \{g_1\} = M(j_1)$. In the former case, we immediately stop the construction of the sequence; otherwise, we continue by applying the same reasoning on j_2. Indeed, since j_2 is in $\text{dep}_i(G, C, M)$, $M(j_2) \neq \emptyset$. In particular, by letting $\{g_2\} = M(j_2)$, then either $\{g_2\} \cap \texttt{img}(\bar{M}) = \emptyset$, or g_2 is assigned to some agent, say $j_3 \in \text{dep}_i(G, C, M)$. Again, in the former case we stop with $h = 2$, otherwise we continue.

After at most $|\text{dep}_i(G, C, M)|$ steps, we eventually reach some agent $j_h \in \text{dep}_i(G, C, M)$ such that $\{g_h\} = M(j_h)$ and $\{g_h\} \cap \texttt{img}(\bar{M}) = \emptyset$. Moreover, $\texttt{val}(M(j_h)) \geq \sigma > \texttt{val}(\bar{M}(j_1))$. By using this sequence, define the matching $\bar{M}' : C \cup \{i\} \to \mathbb{G}$ as $\bar{M}'(j_x) = \bar{M}(j_{x+1})$, for each $x \in \{1, \ldots, h\text{-}1\}$; $\bar{M}'(j_h) = M(j_h)$; and $\bar{M}'(j') = \bar{M}(j')$, for any other agent $j' \in (C \cup \{i\}) \setminus \{j_1, \ldots, j_h\}$.

Note that $\bar{M}'$ is a matching in $C \cup \{i\}$ and $\mathtt{val}(\bar{M}') = \mathtt{val}(\bar{M}) - \mathtt{val}(\bar{M}(j_1)) + \mathtt{val}(M(j_h)) > \mathtt{val}(\bar{M})$, which contradicts the optimality of $\bar{M}$. $\diamond$

Claim 5.5.(B). For each $p \in \mathrm{dep}_i(G, C, \bar{M})$, $\mathtt{val}(\bar{M}(p)) \geq \sigma$. To prove the claim, let D be the minimal subset of $\mathrm{dep}_i(G, C, \bar{M})$ such that:

- $i \in D$;
- if $j \in D$, $z \in \mathrm{dep}_i(G, C, \bar{M})$, $\mathtt{val}(\bar{M}(z)) \geq \sigma$, $\bar{M}(j) = \{g\}$, and $\{g, z\} \in E$, then $z \in D$.

By construction of D and Claim 5.5.(A) about $\bar{M}(i)$, all agents in D get goods having value at least σ according to $\bar{M}$. However, if the statement of the claim does not hold, then there exists some agent $p_{k+1} \in \mathrm{dep}_i(G, C, \bar{M}) \setminus D$ such that $\mathtt{val}(\bar{M}(p_{k+1})) < \sigma$ and $\{g_k, p_{k+1}\} \in E$ for some good $g_k \in \mathbb{G}$ such that $\bar{M}(p_k) = g_k$ and $p_k \in D$. Intuitively, p_{k+1} is at the "external boundary" of D, and cannot enter D because $\mathtt{val}(\bar{M}(p_{k+1})) < \sigma$.

Now, if p_{k+1} belongs to the set $\mathrm{dep}_i(G, C, M)$, then we can apply precisely the same construction as in the proof of Claim 5.5.(A), by just using $p_{k+1} = j_1$ instead of $i = j_1$, in order to contradict the optimality of $\bar{M}$. Hence, we next consider the case where $p_{k+1} \notin \mathrm{dep}_i(G, C, M)$. In this case, by construction of D and since $p_k \in D$, there is a sequence of agents $p_{k+1}, p_k, \ldots, p_1 = i$, such that $p_s \in D$, $\bar{M}(p_s) = \{g_s\}$ (hence with $\mathtt{val}(g_s) \geq \sigma$), and $\{g_s, p_{s+1}\} \in E$, for each $s \in \{1, \ldots, k\}$.

We complete the proof of the claim by showing that the existence of this sequence again leads to a contradiction. Consider the good $g_k = \bar{M}(p_k)$ and notice that g_k cannot be assigned by M to some agent in $\mathrm{dep}_i(G, C, M)$, otherwise p_{k+1} should belong to this set (but we are now considering the case where $p_{k+1} \notin \mathrm{dep}_i(G, C, M)$). Then, consider the agent p_{k-1} and the good g_{k-1} such that $\bar{M}(p_{k-1}) = \{g_{k-1}\}$. Either g_{k-1} is not used by M for agents in $\mathrm{dep}_i(G, C, M)$, or p_k belongs to $\mathrm{dep}_i(G, C, M)$. By continuing this way along the path from p_{k+1} to i, we eventually can find the minimum value of $r \geq 0$ (in the worst case with $r = k - 1$) for which the agent $p_{k-r} \in \mathrm{dep}_i(G, C, M) \cap D$ is such that $\bar{M}(p_{k-r}) = \{g_{k-r}\}$ and g_{k-r} is not assigned by M to some agent in $\mathrm{dep}_i(G, C, M)$. Now, as in the proof of Claim 5.5.(A), we can find a sequence of agents $j_1, \ldots, j_h$, belonging to $\mathrm{dep}_i(G, C, M)$ and starting at $p_{k-r} = j_1$ such that:

1. for each index $x \in \{1, \ldots, h\text{-}1\}$, $\bar{M}(j_{x+1}) = M(j_x)$;
2. $M(j_h) \cap \mathtt{img}(\bar{M}) = \emptyset$.

By using this sequence, define the matching $\bar{M}' : C \cup \{i\} \to \mathbb{G}$ as follows:

- $\bar{M}'(j_h) = M(j_h)$;
- if $h > 1$, $\bar{M}'(j_x) = \bar{M}(j_{x+1})$, for each $x \in \{1, \ldots, h\text{-}1\}$;
- $\bar{M}'(p_{x+1}) = \bar{M}(p_x)$, for each $x \in \{k - r, \ldots, k\}$; and
- $\bar{M}'(j') = \bar{M}(j')$, for any other agent $j' \in (C \cup \{i\}) \setminus \{j_1, \ldots, j_h, p_{k-r+1}, \ldots, p_{k+1}\}$.

Note that $\bar{M}'$ is a matching in $G[C \cup \{i\} \cup \mathbb{G}]$ and it total value is $\mathtt{val}(\bar{M}') = \mathtt{val}(\bar{M}) - \mathtt{val}(\bar{M}(p_{k+1})) + \mathtt{val}(M(j_h)) > \mathtt{val}(\bar{M})$. Indeed, $\mathtt{val}(\bar{M}(p_{k+1})) < \sigma$ while $\mathtt{val}(M(j_h)) \geq \sigma$, since $j_h \in \mathrm{dep}_i(G, C, M)$. This contradicts the optimality of $\bar{M}$. $\diamond$

The property of interest immediately follows by the claim above. $\square$

We have now all the technical ingredients in place to characterize marginal contributions in terms of a property of the dependent agents.

Theorem 5.6. *The following statements are equivalent:*

(1) $v_G^a(C \cup \{i\}) - v_G^a(C) \geq \sigma$*;*
(2) *there is a matching* M *in* $G[C \cup \{i\} \cup \mathbb{G}]$ *such that* $\mathtt{val}(M(j)) \geq \sigma$*, for each* $j \in \mathrm{dep}_i(G, C, M)$*.*

Proof. *(1)⇒(2)* Assume by contradiction that *(2)* does not hold. Let $\bar{M}$ be a maximum weighted matching in $G[C \cup \{i\} \cup \mathbb{G}]$. Hence, there exists some $j \in \mathrm{dep}_i(G, C, \bar{M})$ with $val(\bar{M}(j)) < \sigma$. Consider then the two following cases.

Case $j = i$*:* in this case, $\mathtt{val}(\bar{M}(i)) < \sigma$. Because the restriction of $\bar{M}$ to $G[C \cup \mathbb{G}]$ is a matching in $G[C \cup \mathbb{G}]$, we have $v_G^a(C) \geq \mathtt{val}(\bar{M}) - \mathtt{val}(\bar{M}(i)) > \mathtt{val}(\bar{M}) - \sigma = v_G^a(C \cup \{i\}) - \sigma$, and hence $v_G^a(C \cup \{i\}) - v_G^a(C) < \sigma$, which contradicts *(1)*.

Case $j \neq i$*:* in this case, $\mathtt{val}(\bar{M}(i)) \geq \sigma$ and $\mathtt{val}(\bar{M}(j)) < \sigma$. By definition of $\mathrm{dep}_i(G, C, \bar{M})$, there exists a sequence of agents $i = j'_1, j'_2, \ldots, j'_h = j$ such that $\bar{M}(j'_x) = \{g_x\}$ and $\{g_x, j'_{x+1}\} \in E$, for each $x \in \{1, \ldots, h\text{-}1\}$, and $\mathtt{val}(\bar{M}(j'_h)) < \sigma$. By using this sequence, define the matching $\bar{M}_{\text{-}i}$ such that $\bar{M}_{\text{-}i}(j'_{x+1}) = \bar{M}(j'_x)$, for each $x \in \{1, \ldots, h\text{-}1\}$; and $\bar{M}_{\text{-}i}(j'') = \bar{M}(j'')$, for any other agent $j'' \in C \setminus \{j'_2, \ldots, j'_h\}$. Note that $\bar{M}_{\text{-}i}$ is a matching in $G[C \cup \mathbb{G}]$ and its total value is $\mathtt{val}(\bar{M}_{\text{-}i}) = \mathtt{val}(\bar{M}) - \mathtt{val}(\bar{M}(j'_h))$. For the new matching we have $v_G^a(C) \geq \mathtt{val}(\bar{M}) - \mathtt{val}(\bar{M}(j'_h)) > \mathtt{val}(\bar{M}) - \sigma$. Again, we get $v_G^a(C \cup \{i\}) - v_G^a(C) < \sigma$, which contradicts *(1)*.

(2)⇒(1) Let M be a matching in $G[C \cup \{i\} \cup \mathbb{G}]$ such that $\mathtt{val}(M(j)) \geq \sigma$ for every $j \in dep_i(G, C, M)$. By Lemma 5.5, there exists a maximum weighted matching $\bar{M}$ in $G[C \cup \{i\} \cup \mathbb{G}]$ such that $\mathtt{val}(\bar{M}(j)) \geq \sigma$ still holds for each $j \in dep_i(G, C, \bar{M})$. Consider now the coalition C. By Lemma 5.4, there exists a maximum weighted matching $\bar{M}_{\text{-}i}$ in $G[C \cup \mathbb{G}]$ such that $\mathtt{img}(\bar{M}_{\text{-}i}) \subseteq \mathtt{img}(\bar{M})$. Recall that $v_G^a(C \cup \{i\}) = \mathtt{val}(\mathtt{img}(\bar{M}))$ and $v_G^a(C) = \mathtt{val}(\mathtt{img}(\bar{M}_{\text{-}i}))$. Hence, if $\bar{M}(i) \cap \mathtt{img}(\bar{M}_{\text{-}i}) = \emptyset$, then $v_G^a(C \cup \{i\}) - v_G^a(C) = \mathtt{val}(\mathtt{img}(\bar{M})) - \mathtt{val}(\mathtt{img}(\bar{M}_{\text{-}i})) = \mathtt{val}(g') \geq \sigma$. So, assume that $\bar{M}(i) \subseteq \mathtt{img}(\bar{M}_{\text{-}i})$, which entails the existence of a sequence of agents $i = j_1, \dots, j_h$ belonging to $C \cup \{i\}$ and such that $\bar{M}_{\text{-}i}(j_{x+1}) = \bar{M}(j_x)$, for each $x \in \{1, \dots, h-1\}$, and $\bar{M}(j_h) \cap \mathtt{img}(\bar{M}_{\text{-}i}) = \emptyset$. In fact, by definition of dependent agents, we actually have that $j_h \in dep_i(G, C, \bar{M})$. Hence, $\mathtt{val}(\bar{M}(j_h)) \geq \sigma$ holds. Again, we derive $v_G^a(C \cup \{i\}) - v_G^a(C) = \mathtt{val}(\mathtt{img}(\bar{M})) - \mathtt{val}(\mathtt{img}(\bar{M}_{\text{-}i})) = \mathtt{val}(\bar{M}(j_h)) \geq \sigma$. □

5.2. MSO encoding

The encoding in terms of Monadic Second Order logic will exploit not only the characterization provided by Theorem 5.6, but also a rewriting of the Shapley value on the basis of the various possible values (cf. the real number σ in the statement of the theorem) that marginal contributions can assume. In fact, we next show that it is enough to consider the distinct values of the goods in the given instance, whose number is thus bounded by the input size.

More formally, recall that $\{w_1, \dots, w_m\}$ is the set of all values associated with goods in $\mathbb{G}$ with $0 = w_1 < w_2 < \cdots < w_m$. Let $i \in N$ be an agent, let $h \in \{0, \dots, n-1\}$, let $\ell \in \{1, \dots, m\}$, and let us denote by $\#\mathrm{coal}_i^\ell(G, h)$ the number of coalitions C such that $|C| = h$ and $v_G^a(C \cup \{i\}) - v_G^a(C) \geq w_\ell$. Then, the coefficients $\beta_i(\mathcal{G}_G^a, h)$ in the expression for the Shapley value illustrated in Equation (1) can be rewritten in terms of these values $\#\mathrm{coal}_i^\ell(G, h)$.

Theorem 5.7. *For each agent $i \in N$ and $h \in \{0, \dots, n-1\}$,*

$$\beta_i(\mathcal{G}_G^a, h) = w_m \times \#\mathrm{coal}_i^m(G, h) + \sum_{\ell=1}^{m-1} w_\ell \times \left(\#\mathrm{coal}_i^\ell(G, h) - \#\mathrm{coal}_i^{\ell+1}(G, h)\right).$$

Proof. Recall that, for each $i \in N$ and $h \in \{0, \dots, n-1\}$,

$$\beta_i(\mathcal{G}_G^a, h) = \sum_{C \subseteq N \setminus \{i\},\ |C| = h} (v_G^a(C \cup \{i\}) - v_G^a(C)).$$

Consider an arbitrary coalition $C \subseteq N \setminus \{i\}$ with $|C| = h$. Let M be a maximum weighted matching in $G[C \cup \{i\} \cup \mathbb{G}]$ and define $I = \{g \in \mathbb{G} \mid \{g, x\} \in M\}$. Because of Lemma 5.4, there is a maximum weighted matching M' in $G[C \cup \mathbb{G}]$ such that, by letting $I = \{g \in \mathbb{G} \mid \{g, x\} \in M\}$, we get $I' \subseteq I$. Now, if $I = I'$, then we clearly have $v_G^a(C \cup \{i\}) - v_G^a(C) = 0$. Otherwise, $I \setminus I'$ is a singleton $\{g\}$, and $v_G^a(C \cup \{i\}) - v_G^a(C) = \mathtt{val}(g)$. In both cases, the marginal contribution is a value w_ℓ with $\ell \in \{1, \dots, m\}$.

After the above observation, we can rewrite the expression for the coefficients as $\beta_i(\mathcal{G}_G^a, h) = \sum_{\ell=1}^m w_\ell \times K_\ell^i(G, h)$, where $K_\ell^i(G, h)$ is the number of coalitions C with $|C| = h$ and such that $v_G^a(C \cup \{i\}) - v_G^a(C) = w_\ell$. By definition of $\#\mathrm{coal}_i^\ell(G, h)$, we get $K_m^i(G, h) = \#\mathrm{coal}_i^m(G, h)$ and, for each $\ell \in \{1, \dots, m-1\}$ we have $K_\ell^i(\mathcal{G}_\mathcal{A}, h) = \#\mathrm{coal}_i^\ell(G, h) - \#\mathrm{coal}_i^{\ell+1}(G, h)$. □

The level graph G_ℓ For every level $\ell \in \{1, \dots, m\}$, let graph $G_\ell = (N \cup \mathbb{G}_\ell, E_\ell)$ be the subgraph of G induced over the nodes in $N \cup \mathbb{G}_\ell$, where $\mathbb{G}_\ell = \{g \in \mathbb{G} \mid \mathtt{val}(g) \geq w_\ell\}$. For instance, Fig. 5(b) illustrates the graph $G_{3.2}$ derived from the graph G shown in (a).

The MSO formula $F_i^\ell(C)$ Let $F_i^\ell(C)$ be the following MSO formula, where C is a free node-set variable meant to encode the coalitions such that $v_G^a(C \cup \{i\}) - v_G^a(C) \geq w_\ell$:

$$\exists M \subseteq r_{E_\ell}\ \exists D \subseteq N\ (C \subseteq N \wedge i \in D \wedge dep(C, M, D) \wedge goodM(M, D)), \quad \text{where}$$

$$dep(C, M, D) := \forall j \in D\ \forall z \in C$$
$$\Big(\exists g \in \mathbb{G}\ \ (\{z, g\} \in E_\ell \wedge \{j, g\} \in M) \rightarrow z \in D\Big)$$
$$goodM(M, D) := \forall x \in D\ \exists g \in \mathbb{G}_\ell$$
$$\Big(\{x, g\} \in M \wedge$$
$$\forall x' \in N\ (x' \neq x \rightarrow \{x', g\} \notin M) \wedge$$
$$\forall g' \in \mathbb{G}\ (g' \neq g \rightarrow \{x, g'\} \notin M)\Big)$$

The idea is to evaluate this formula on the level graph G_ℓ associated with each value w_ℓ, with M being a possible matching, and D a superset of the dependent set of agents $dep_i(G_\ell, C, M)$. The subformula $goodM(M, D)$ checks that M is indeed a matching where all agents in D get a good having value at least w_ℓ. In this case, we say that M and D witness that $G_\ell \models F_i^\ell(C)$.

After Theorem 5.6, we easily get the following result.

Lemma 5.8. *Let $C \subseteq N$, let $i \in N \setminus C$, and let $\ell \in \{1,\dots,m\}$. Then, $G_\ell \models F_i^\ell(C)$ if and only if $v_G^a(C \cup \{i\}) - v_G^a(C) \geq w_\ell$.*

Proof. Assume that $G_\ell \models F_i^\ell(C)$. Then, there is a matching M and a set of agents D that witness it. In particular all agents in D, and hence all agents in $\mathrm{dep}_i(G_\ell, C, M)$, get a good whose value is at least w_ℓ. Note that, because the matching M involves only goods of value at least w_ℓ, using the level graph G_ℓ or the original graph G is immaterial, so that $\mathrm{dep}_i(G_\ell, C, M) = \mathrm{dep}_i(G, C, M)$ clearly holds. By Theorem 5.6, we then conclude that $v_G^a(C \cup \{i\}) - v_G^a(C) \geq w_\ell$. For the other direction, assume that $v_G^a(C \cup \{i\}) - v_G^a(C) \geq w_\ell$. From Theorem 5.6, we know there exists a matching M in $G[C \cup \{i\} \cup \mathbb{G}]$ such that $\mathtt{val}(M(j)) \geq w_\ell$, for each $j \in \mathrm{dep}_i(G, C, M)$. By choosing $D = \mathrm{dep}_i(G, C, M)$, the formula $F_i^\ell(C)$ is clearly satisfied by the graph G_ℓ. □

Simplifying the level graph Before establishing the main result of this section, it is relevant to observe that the level graph can be actually simplified—without loss of generality—to reduce its treewidth as long as there are goods connected to only one agent. We shall show in Section 7 that this leads to an impressive speed-up in real-world applications with hundreds of agents.

Let *Simplify* be the procedure that modifies G_ℓ by repeating the following steps as long as possible: take any vertex $g \in \mathbb{G}$ having degree-1 in G_ℓ and let j the one agent connected to it; remove from G_ℓ all edges incident to j, but the edge $\{g, j\}$. Let G'_ℓ be the output of *Simplify*(G_ℓ). The correctness of the simplification is established below.

Lemma 5.9. *Let $C \subseteq N$, let $i \in N \setminus C$, and let $\ell \in \{1,\dots,m\}$. Then, $G'_\ell \models F_i^\ell(C)$ if and only if $v_G^a(C \cup \{i\}) - v_G^a(C) \geq w_\ell$.*

Proof. We prove the lemma by induction, on each stage of the procedure *Simplify*.

Assume that this is true until a certain step, and let $\bar{G}_\ell$ be the current version of the level graph. Consider the next step of simplification, where we select a good $g \in \mathbb{G}$ with $\mathtt{val}(g) \geq w_\ell$ that is connected to only one agent j in the level graph $\bar{G}_\ell$. Consider the new graph G'_ℓ obtained from $\bar{G}_\ell$ by removing all edges incident to j, but the edge $\{g, j\}$. After the induction hypothesis and Lemma 5.8, it suffices to show that, for any coalition C, the following statements are equivalent:

(1) $\bar{G}_\ell \models F_i^\ell(C)$;
(2) $G'_\ell \models F_i^\ell(C)$.

(2)⇒(1) Consider a pair of sets M' and D' witnessing $G'_\ell \models F_i^\ell(C)$. Let $M = M' \cup \{\{g, j\}\}$ and $D = D' \cup \{j\}$. Note that M assigns a good of value at least w_ℓ to every agent in D, and that $D \supseteq D' \supseteq \mathrm{dep}_i(G'_\ell, C, M')$. Moreover, $D \supseteq \mathrm{dep}_i(G'_\ell, C, M)$ actually holds, because j is connected only to the good g in G'_ℓ. We now claim that $D \supseteq \mathrm{dep}_i(\bar{G}_\ell, C, M)$. Indeed, the missing edges from j to the goods other than g (which are the only difference between G_ℓ and G'_ℓ) cannot lead to include agents in $\mathrm{dep}_i(\bar{G}_\ell, C, M)$ that are not already included in $\mathrm{dep}_i(G'_\ell, C, M)$, but the one agent j that belongs to D. Therefore, M assigns a good of value at least w_ℓ to every agent in D and $D \supseteq \mathrm{dep}_i(\bar{G}_\ell, C, M)$, thereby witnessing that $\bar{G}_\ell \models F_i^\ell(C)$.

(1)⇒(2) Consider a pair of sets M and D witnessing $\bar{G}_\ell \models F_i^\ell(C)$. In the case where $\{g, j\} \in M$, then M and D clearly witness $G'_\ell \models F_i^\ell(C)$, too. Consider then the case where $\{g, j\} \notin M$. Let M' be the set obtained from M by removing the edge involving agent j in M, if any, and by including the edge $\{g, j\}$. *We claim that the set $D' = D \cup \{j\}$ is a superset of* $\mathrm{dep}_i(G'_\ell, C, M')$. Indeed, if j is not in D, then we can observe that $\mathrm{dep}_i(G'_\ell, C, M') = \mathrm{dep}_i(\bar{G}_\ell, C, M) \cup \{j\}$, because $j \notin \mathrm{dep}_i(\bar{G}_\ell, C, M))$ and j is connected in G'_ℓ only to the good g, which is not connected to any other agent. The claim then follows because $D \supseteq \mathrm{dep}_i(\bar{G}_\ell, C, M)$. Consider now the case where $j \in D$, so that $D' = D$. We first observe that $D \supseteq \mathrm{dep}_i(\bar{G}_\ell, C, M)$ trivially entails that $D' = D \supseteq \mathrm{dep}_i(G'_\ell, C, M)$. The claim then follows because $\mathrm{dep}_i(G'_\ell, C, M) \supseteq \mathrm{dep}_i(G'_\ell, C, M')$; in fact, moving from M to M' has the effect of isolating j in the computation of the dependent agents. Eventually, because $\mathtt{val}(g) \geq w_\ell$, we can also conclude that the set M' assigns goods of values at least w_ℓ to all agents belonging to the set D'. Therefore, M' and D' witness that $G'_\ell \models F_i^\ell(C)$. □

In fact, as shown in [83], other simplifications might be possible, by removing specific substructures that does not change the Shapley value. Each simplification leads to smaller graphs and can reduce the treewidth, thus speeding-up the computation. Investigating more complex forms of simplifications is beyond the scope of this paper, but is an interesting avenue of further research.

Main results All the technical ingredients are now in place, and we can establish the main result of this section, namely that the Shapley value and the Banzhaf value can be computed in polynomial time over weighted-allocation games induced by (simplified) graphs having bounded treewidth.

Theorem 5.10. *On weighted-allocation games such that, for every level $\ell \in \{1,\dots,m\}$, the graph Simplify(G_ℓ) has treewidth bounded by some constant, the Shapley value and the Banzhaf value can be computed in polynomial time.*

Proof. From the characterization given by Equation (1) in Section 3, in order to compute the Shapley value and the Banzhaf value in polynomial time for any player $i \in N$, it is sufficient to compute the values $\beta_i(\mathcal{G}^a_G, h)$ in polynomial time, for each coalition cardinality value $h \in \{0, \ldots, n-1\}$. By Theorem 5.7, we can do that by computing in polynomial time the (polynomially-many) values $\#\text{coal}^\ell_i(\mathcal{G}^a_G, h)$, for each $h \in \{0, \ldots, |N|-1\}$, and for each level ℓ associated with the value w_ℓ of some good in $\mathcal{G}$.

Now, recall that $\#\text{coal}^\ell_i(\mathcal{G}^a_G, h)$ is the number of coalitions C such that $|C| = h$ and $v^a_G(C \cup \{i\}) - v^a_G(C) \geq w_\ell$. Given Lemma 5.9, the number of coalitions $C \subseteq N \setminus \{i\}$ for which $v_{\mathcal{A}}(C \cup \{i\}) - v_{\mathcal{A}}(C) \geq w_\ell$ holds and with $|C| = h$ coincides with the number of coalitions $\hat{C} \subseteq N \setminus \{i\}$ with $|\hat{C}| = h$ for which $G'_\ell \models F^\ell_i(\hat{C})$, where G'_ℓ is the output of *Simplify*(G_ℓ). By hypothesis, the treewidth of G'_ℓ is bounded by a constant and thus, by Theorem 2.8, for every cardinality value, the number of coalitions $\hat{C}$ such that $G'_\ell \models F^\ell_i(\hat{C})$ can be computed in polynomial time, that is, $\#\text{coal}^\ell_i(\mathcal{G}^a_G, h)$ can be computed in polynomial time. □

Now, note that $tw(\textit{Simplify}(G_\ell)) \leq tw(G_\ell) \leq tw(G)$ clearly holds, for any level ℓ. Hence, the following comes as an immediate corollary of Theorem 5.10.

Corollary 5.11. *On weighted-allocation games defined over classes of graphs having bounded treewidth, the Shapley value and the Banzhaf value can be computed in polynomial time.*

6. Degree-based tractability results

So far, we have studied islands of tractability for the Shapley value of matching-based coalitional games, by singling out them based on the notion of tree decomposition. In this section, we study different classes of games defined by restricting the underlying graphs on the basis of the degree of their nodes. The analysis moves from the observation that weighted-matching games are known to be tractable when restricted to classes of graphs where each node has at most degree two (i.e., it has two adjacent nodes) [47]. Our goal is to show that a similar condition—in fact, a much less restrictive condition as we shall discuss below—can guarantee tractability of weighted-allocation games, too.

We use the same notation as in Section 5. Assume that a bipartite graph G with nodes $N \cup \mathbb{G}$ is given, and let $\{w_1, \ldots, w_m\} = \{\texttt{val}(g) \mid g \in \mathbb{G}\} \cup \{0\}$ be the set of all values associated with goods in $\mathbb{G}$, where $0 = w_1 < w_2 < \cdots < w_m$. Any matching M is also viewed as a function mapping each agent $i \in N$ to the set $M(i) = \{g \in \mathbb{G} \mid \{i, g\} \in M\}$. Moreover, denote by $\Omega(i)$ the set $\{g \in \mathbb{G} \mid \{i, g\} \in E\}$ of the goods to which i is connected in G.

A *binary-clashes allocation game* is a weighted-allocation game such that at most two agents can be interested in the same good (without any restriction on the number of goods each agent is interested in). That is, for every good $g \in \mathbb{G}$, $|\{i \in N \mid g \in \Omega(i)\}| \leq 2$.

Theorem 6.1. *The Shapley value and the Banzhaf value of binary-clashes allocation games can be computed in polynomial time.*

Before proving the result, we point out that it guarantees tractability even for classes of binary-clashes allocation games defined by unbounded treewidth graphs, for which the elaborations in Section 5 cannot be applied.

Example 6.2. Consider a class of binary-clashes allocation games defined by graphs of the form $\{G_n \mid n > 0\}$ where, for each G_n, the agent set is $\{1, \ldots, n\}$, the good set is $\mathbb{G}_n = \{g_{i,j} \mid i, j \in \{1, \ldots, n\} \wedge i \neq j\}$, and for each good node $g_{i,j} \in \mathbb{G}_n$, the only two edges incident to it are $\{i, g_{i,j}\}$ and $\{j, g_{i,j}\}$. That is, only the two agents i and j are interested in such a good $g_{i,j}$. Note however that every agent is interested in $n-1$ different goods.

It is immediate to check that, for any n, every tree decomposition of G_n must necessarily include a vertex covering all nodes in $\{1, \ldots, n\}$; therefore, $tw(G_n) \geq n-1$. This means that Theorem 5.10 does not apply to this class of games, on which the tractability of the Shapley value (and Banzhaf value) follows instead by Theorem 6.1. ◁

We now start the elaboration of the proof of Theorem 6.1. To this end, we need some further notation. For each level $\ell \in \{1, \ldots, m\}$ over the possible values, define the graph $G_\ell = G[N \cup \mathbb{G}_\ell]$, where $\mathbb{G}_\ell = \{g \in \mathbb{G} \mid \texttt{val}(g) \geq w_\ell\}$. For any pair of distinct agents j', j'', define the goods above this level they are jointly interested in as $\Omega_\ell(j', j'') = \Omega(j') \cap \Omega(j'') \cap \{g \mid \texttt{val}(g) \geq w_\ell\}$.

We next focus on some agent i for which we would like to compute the Shapley value. Denote by $\text{con}_i(G_\ell)$ the set of all agents (including i) that are reachable from agent i in G_ℓ. Define the agent interaction graph $G^\star_{i,\ell}$ as the graph having the agents in $\text{con}_i(G_\ell)$ as the set of its nodes, and where two agents j' and j'' are connected with an edge if and only if $\Omega_\ell(j', j'') \neq \emptyset$.

We now start the proof of Theorem 6.1, by first pointing out a useful property that relates the set we have just defined with the coefficients $\#\text{coal}^\ell_i(G, h)$, which are the number of coalitions C such that $|C| = h$ and $v^a_G(C \cup \{i\}) - v^a_G(C) \geq w_\ell$. We use, as a technical ingredient, the notion of dependent agents and Theorem 5.6 in Section 5.

Lemma 6.3. *Let $h \in \{0, \ldots, n-1\}$ and assume that one of the following conditions holds:*

(a) *There are j and j' in* $\mathrm{con}_i(G_\ell)$ *with* $|\Omega_\ell(j, j')| \geq 2$;
(b) *There is an agent j in* $\mathrm{con}_i(G_\ell)$ *and a good* $g \in \Omega(j)$ *with* $\mathtt{val}(g) \geq w_\ell$ *such that* $g \notin \Omega(j')$*, for each* $j' \neq j$;
(c) *The graph* $G^\star_{i,\ell}$ *contains a cycle.*

Then, $\#\mathrm{coal}^\ell_i(G, h) = \frac{(n-1)!}{(n-1-h)!h!}$.

Proof. *(a)* Assume that there are two agents j' and j'' in $\mathrm{con}_i(G_\ell)$, and a pair of (distinct) goods $\{g', g''\} \subseteq \Omega_\ell(j, j')$.

Consider a spanning tree T of the graph $G^\star_{i,\ell}$ rooted at agent j'. Define a matching $\bar{M}$ as follows: $\bar{M}(j') = g'$, $\bar{M}(j'') = g''$; and for any other agent $j_c \neq j''$ whose parent in T is j_p, let $\bar{M}(j_c)$ be an arbitrary good in $\Omega_\ell(j_c, j_p)$. The matching $\bar{M}$ is well-defined because of the binary-clashes property, which states that $|\{i \in N \mid g \in \Omega(i)\}| \leq 2$, for each $g \in \mathbb{G}$. Indeed, just note that the good g' assigned to the root j' cannot be useful for any other children of j' in T (but the single node j'', which already have assigned the good g''). Thus, every child $j''' \neq j''$ of j' can be assigned any good g''' in $\Omega_\ell(j', j''')$ that is of interest only for j' and j''' (and having value at least w_ℓ). In particular, g''' must be different from g' and g'', as well as from any other good that j' has in common with other agents connected with this node in $G^\star_{i,\ell}$. The same reasoning can then be applied to see that $\bar{M}$ correctly assigns one good to any node j_c with parent j_p, along T.

Now, by considering Definition 5.2, note that $\mathrm{con}_i(G_\ell) \supseteq \mathrm{dep}_i(G, C, \bar{M})$ for each possible set $C \subseteq N \setminus \{i\}$. Hence, $\bar{M}$ satisfies condition *(2)* in Theorem 5.6, and we conclude $v^a_G(C \cup \{i\}) - v^a_G(C) \geq w_\ell$ independently of the coalition $C \subseteq N \setminus \{i\}$. So, $\#\mathrm{coal}^\ell_i(G, h)$ is the number of coalitions C, with $|C| = h$, that we can build from a set of $|N \setminus \{i\}| = n - 1$ agents.

(b) Let j be the agent for which exists a good $g \in \Omega(j)$ with $\mathtt{val}(g) \geq w_\ell$ and such that $g \notin \Omega(j')$, for each $j' \neq j$. Consider a spanning tree T of the graph $G^\star_{i,\ell}$ rooted at node j, and define a matching $\bar{M}$ as follows: $\bar{M}(j) = g$; for any other agent j_c whose parent in T is j_p, let $\bar{M}(j_c)$ be an arbitrary good in $\Omega_\ell(j_c, j_p)$. Again $\bar{M}$ is well-defined and, by applying the same line of reasoning as above, we get the desired characterization of $\#\mathrm{coal}^\ell_i(G, h)$.

(c) Let T be a spanning tree of $G^\star_{i,\ell}$. Because we are considering the case $G^\star_{i,\ell}$ is cyclic, at least one edge $\{j, j'\}$ of this graph does not belong to the spanning tree T. Let us root T at node j, and define the matching $\bar{M}$ such that: $\bar{M}(j) = g$ with $g \in \Omega_\ell(j, j')$; for any other agent j_c whose parent in T is j_p, let $\bar{M}(j_c)$ be an arbitrary good in $\Omega_\ell(j_c, j_p)$. To conclude as in the previous cases, just note that $\bar{M}$ is well-defined. □

We next address the cases that are not covered by the previous lemma.

Lemma 6.4. *Assume that none of the conditions (a), (b) and (c) of Lemma 6.3 holds. Let $i \in N$ be an agent and let $C \subseteq N \setminus \{i\}$. Then,* $v^a_G(C \cup \{i\}) - v^a_G(C) < w_\ell$ *if and only if* $\mathrm{con}_i(G_\ell) \subseteq C \cup \{i\}$.

Proof. Assume that $\mathrm{con}_i(G_\ell) \subseteq C \cup \{i\}$. By definition of $\mathrm{con}_i(G_\ell)$, since $|\{i \in N \mid g \in \Omega(i)\}| \leq 2$ for each $g \in \mathbb{G}$, and none of the conditions (a), (b) and (c) of Lemma 6.3 holds, $G^\star_{i,\ell}$ is a tree and the goods in $\bigcup_{j \in \mathrm{con}_i(G_\ell)} \Omega(j)$ having value at least w_ℓ have a one-to-one correspondence with the edges of $G^\star_{i,\ell}$. Hence, the number of these goods is $|\mathrm{con}_i(G_\ell)| - 1$. Therefore, there is no matching M such that $\mathtt{val}(M(j)) \geq w_\ell$, for each $j \in \mathrm{con}_i(G_\ell)$. By Theorem 5.6, we conclude that $v^a_G(C \cup \{i\}) - v^a_G(C) < w_\ell$.

For the opposite direction, let C' be the connected component of $G^\star_{i,\ell}[C \cup \{i\}]$ that contains the agent i. Observe that $\mathrm{con}_i(G_\ell) \nsubseteq C \cup \{i\}$ entails that there exists an agent $j \in \mathrm{con}_i(G_\ell) \setminus (C \cup \{i\})$ that is connected in $G^\star_{i,\ell}$ to some agent $j' \in C'$. Let us root the tree $G^\star_{i,\ell}[C']$ at node j', and define a matching $\bar{M}$ as follows: $\bar{M}(j') = g$ with $g \in \Omega_\ell(j, j')$; for any other agent j_c whose parent in $G^\star_{i,\ell}[C']$ is j_p, let $\bar{M}(j_c) = g_c$ with $g_c \in \Omega_\ell(j_c, j_p)$. Note that $\bar{M}$ is well-defined and all goods assigned by this matching have value at least w_ℓ. Moreover, note that $C' = \mathrm{dep}_i(G, C, \bar{M})$, so that we can apply Theorem 5.6 in order to conclude that $v^a_G(C \cup \{i\}) - v^a_G(C) \geq w_\ell$. □

Corollary 6.5. *Assume that none of the conditions (a), (b) and (c) of Lemma 6.3 holds. Let i be an agent in N, and let $h \in \{0, \ldots, n-1\}$. If $h + 1 < |\mathrm{con}_i(G_\ell)|$, then* $\#\mathrm{coal}^i_\ell(\mathcal{G}_\mathcal{A}, h) = \frac{(n-1)!}{(n-1-h)!h!}$. *Otherwise,* $\#\mathrm{coal}^i_\ell(\mathcal{G}_\mathcal{A}, h) = \frac{(n-1)!}{(n-1-h)!h!} - \frac{(n-|\mathrm{con}_i(G_\ell)|)!}{(n-h-1)!(h+1-|\mathrm{con}_i(G_\ell)|)!}$.

Proof. By Lemma 6.4, the coefficient $\#\mathrm{coal}^i_\ell(\mathcal{G}_\mathcal{A}, h)$ can be written as the number of coalitions of cardinality h that we can build from a set of $|N \setminus \{i\}| = n - 1$ agents minus the number of coalitions C for which $C \cup \{i\} \supseteq \mathrm{con}_i(G_\ell)$ holds. For $h + 1 < |\mathrm{con}_i(G_\ell)|$, no coalition C with $|C| = h$ can enjoy this latter property. Whenever $h + 1 \geq |\mathrm{con}_i(G_\ell)|$, the number of coalitions C with $|C| = h$ for which $\mathrm{con}_i(G_\ell) \subseteq C \cup \{i\}$ can be obtained by counting all possible coalitions that can be obtained by extending $\mathrm{con}_i(G_\ell)$ with further $h - |\mathrm{con}_i(G_\ell)|$ agents. □

With the above technical ingredients in place, we can now prove Theorem 6.1. Let i be an agent and consider each value level w_ℓ. By Corollary 6.5 and Lemma 6.3, we compute in polynomial time the numbers $\#\mathrm{coal}^i_\ell(\mathcal{G}_\mathcal{A}, h)$, for every cardinality $h \in \{0, \ldots, n-1\}$. Then, the result immediately follows by using Theorem 5.7 for the computation of the coefficients needed to get the Shapley and the Banzhaf value.

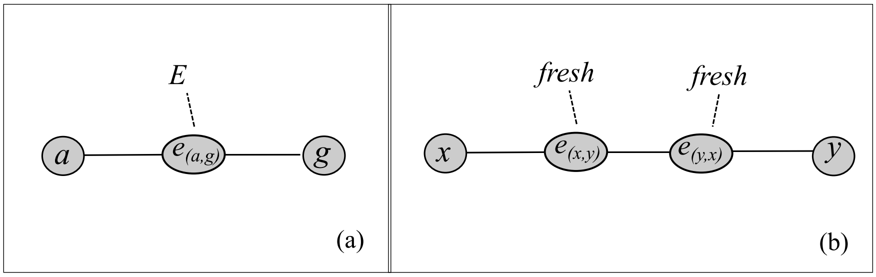


Fig. 6. Using node labels for encoding second-order quantification on edge-sets: (a) undirected edges in weighted-allocation games; and (b) directed edges in matching games.

7. Experimental activity

The (MSO-based) algorithms described in the previous sections have been implemented in a Java system prototype built on top of the MSO solver *Sequoia* [52].

We consider two experimental settings. The first one, described in Section 7.1, aims at evaluating the performance of the approach we have proposed to deal with weighted-allocation games. Instead, the second setting, discussed in Section 7.2, is meant to assess the effectiveness of our approach on matching games. In both cases, we consider a benchmark of synthetic instances and of instances taken from real-world graphs. Experiments have been performed on a dedicated machine equipped with an Intel Core i7-3770k 3.5 GHz processor, 12 GB (DDR 1600 MHz) of RAM, and running Linux Debian Jessie. Our Java algorithms were executed on the JDK 1.8.0 05-b13, and the GNU g++-4.9 compiler was used to compile *Sequoia*.

Note that *Sequoia* is a complete MSO solver that however uses some randomized heuristics, so that different runs (while producing the same result) might actually require different computation times. Accordingly, unless stated otherwise, computation times discussed in the rest of the section (and depicted in the charts) are the average values over 5 runs—this number has been chosen by noticing that, in our application, average values does not change significantly when more than 3 runs are performed.

7.1. Weighted-allocation games

Implementation issues On allocation games, our system prototype first builds the bipartite graph relating agents with goods.[8] Then, for every value ℓ, the prototype computes the graph *Simplify*$(\mathcal{G}_\ell)$—see, again, Lemma 5.9—and prepares it for its subsequent use in *Sequoia*. Indeed, the solver *Sequoia* does not support natively second order quantification on sets of edges. However, it supports the evaluation of MSO formulas over undirected graphs that may have labels on their vertices. In fact, this feature has been used, in a rather standard way, in order to encode the edges as new vertices of the graph with the distinguished label *"E"*, as shown in Fig. 6(a).

Note that this encoding affects the number of the nodes, but it does not increase the treewidth of the original graph. Eventually, the MSO formula described in Section 5.2 is evaluated on the resulting graph, by running *Sequoia* with the "CardCount" option that allows us to calculate, with a single execution, a vector that contains, for each cardinality value, the number of those distinct instantiations of the (free variable) set C for which the formula is satisfiable. Once the counting problems on the different levels are solved, the prototype calculates the desired Shapley value, by a straightforward implementation of the arguments in the proof of Theorem 5.10.

Datasets For our experimentation, we consider two benchmarks.

The first one refers to a setting described by [35,39] concerning the feasibility of computing the Shapley value of a weighted-allocation game arising in the most recent Italian research assessment program (namely, VQR 2011-2014). In this application, the agents are the researchers (and the professors) of the Italian Universities, and the goods are their research products, whose values are determined according to criteria published by ANVUR (the Italian agency in charge of the VQR program). A product can be assigned at most to one researcher, so that the setting naturally gives rise to a weighted-allocation game. Actually, in the research assessment program VQR 2011-2014, at most two products can be assigned to each researcher, therefore this program does not fit precisely our setting. We are planning to deal with such a generalization in a future work. Nevertheless, the real-world agents-goods graphs coming from VQR instances are a natural benchmark for our algorithms, where of course we have to consider weighted-allocation games with one good for each agent. In particular, we consider the case of a very large University ("La Sapienza" of Rome) for which we used the actual VQR graph, comprising thousands of researchers and publications. Fig. 7 provides a graphical representation of the agents

[8] The system is also able to perform the problem simplification described in [83]. This phase is not relevant to our aims and details are omitted; we refer the interested reader to [83].

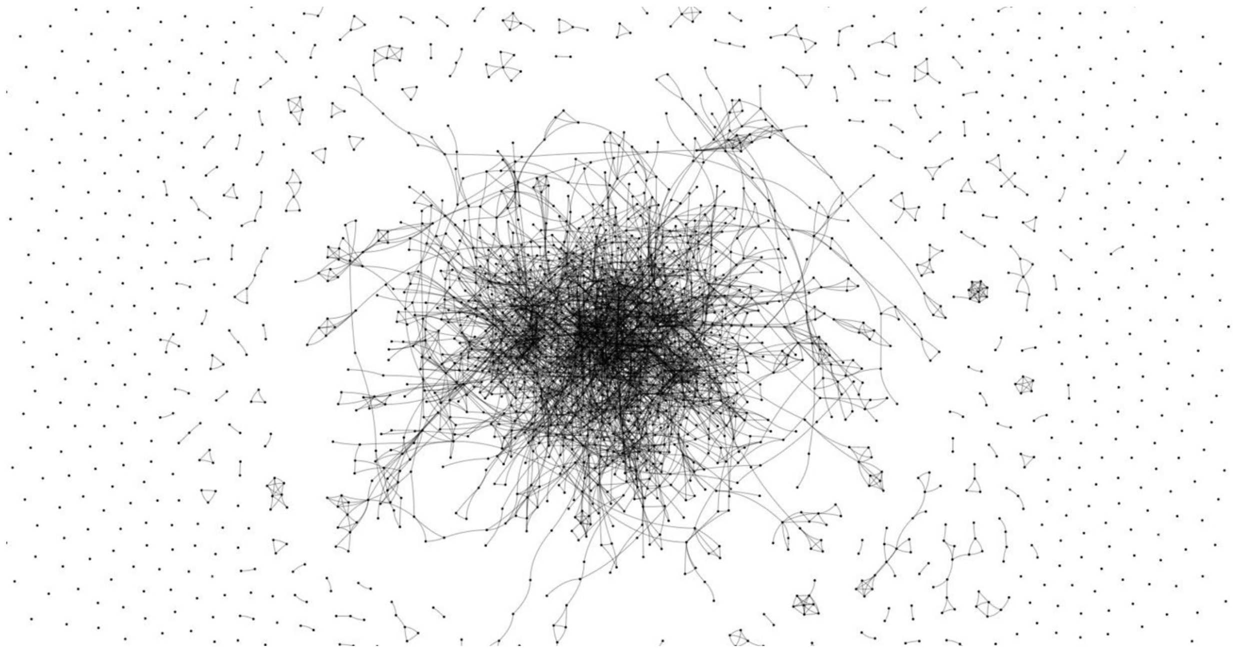


Fig. 7. Co-authorship graph for the University "La Sapienza" of Rome.

LEVEL	INDEPENDENT AGENTS WITH EXCLUSIVE GOODS	INDEPENDENT AGENTS WITHOUT GOODS	DEPENDENT AGENTS (WITH SHARED GOODS)
0	685	0	0
0.1	665	0	20
0.4	610	35	40
0.7	511	92	82
1	338	228	119

Fig. 8. Statistics on the level graphs: An agent is *independent* if either *(i)* is connected to one good that is furthermore not shared with other agents, or *(ii)* has no available goods at all for the given level.

involved in this game, where two researchers are connected if they are coauthors of some research product, that is, if they are jointly interested in some good in the associated allocation game.

The second benchmark is made instead of synthetic *ladder* graphs, whose treewidth is 2.

Results on the VQR graph The VQR graph of "La Sapienza" consists of 3562 researchers and 5909 publications. Fig. 7 shows that there are many isolated authors as well as some small groups isolated from the rest of the researchers. Indeed, we identified 156 connected components, with the largest one having 685 researchers and 1352 publications, while all the others being rather small (155 components with an average size of 3). We get rid of the small components that can be solved easily, and we focus on the largest one, whose treewidth is 47.

The values associated with the research products according to the VQR guidelines and occurring in the instance are $0, 0.1, 0.4, 0.7$, and 1. For these values, we build the corresponding level graphs, comprising $5235, 4665, 4197, 3541$, and 2363 edges, respectively. For these graphs, the numbers of agents are $685, 685, 650, 593$, and 457, while the treewidths are $47, 47, 30, 18$, and 6, respectively. The level graphs associated with values in $\{0.4, 0.7, 1\}$ consist of different connected components, some of which have large sizes (up to 3941 edges and 600 agents). A few statistics on these graphs are reported in Fig. 8. The level graphs have been then pre-processed by means of the procedure *Simplify* (cf. Lemma 5.9), in order to reduce their size and their treewidth. The characteristics of the graphs resulting from this phase are reported in Fig. 9.

Finally, by exploiting these simplified graphs and the approach we discussed in Section 5, we computed the Shapley values for all the agents in the VQR graph. Interestingly, our algorithm founds the exact solution in about 19 minutes, which is surprisingly fast when compared with the performances of the approximation algorithm discussed by [83] (that does not exploit the treewidth) and that requires about 160 hours to get an approximation that is expected to be good enough. To point out the scalability of our approach, we extracted a set of samples of connected subgraphs from the largest component of the VQR graph, with varying sizes and with treewidths ranging from 1 to 3. The time required to compute

Level	Number of connected components	Maximum number of edges	Maximum number of agents	Maximum treewidth
0	0	0	0	0
0.1	7	43	6	2
0.4	16	59	7	2
0.7	30	44	6	4
1	40	57	9	3

Fig. 9. Characteristics of the graphs resulting from the simplification, for each level of value. Maximum values are taken over all possible connected components.

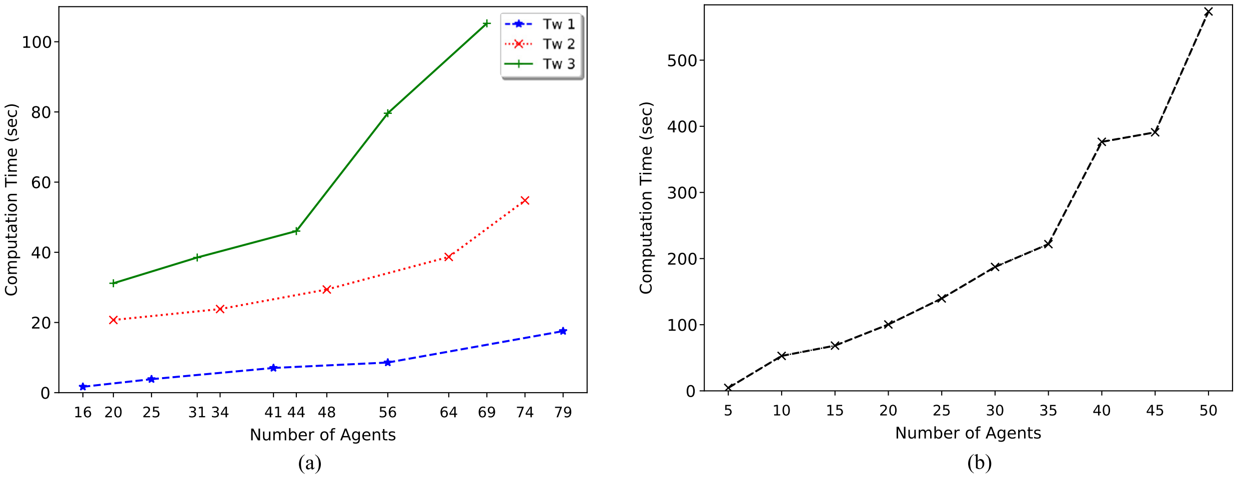


Fig. 10. Time required to compute the Shapley value on weighted-allocation games.

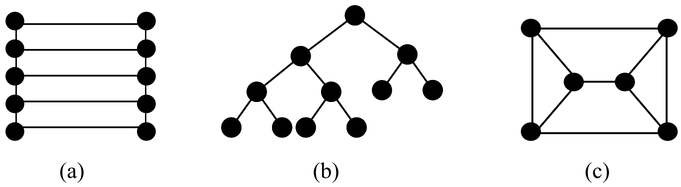


Fig. 11. Syntectic graphs in Section 7: (a) ladder graphs, (b) complete binary trees, and (c) Halin graphs.

the Shapley values on each of these samples is reported in Fig. 10(a), which evidences the nice scaling w.r.t. the size of the graph (and a higher impact of the treewidth).

Results on ladder graphs The second setting we have considered for our experimentation refers to a number of synthetic ladder graphs, with up to 50 agents. More formally, each graph $G_n = (N \cup \mathbb{G}, E, w)$ consists of $N = \{1, \ldots, n\}$ agents and $\mathbb{G} = \{g_1, \ldots, g_n\}$ goods such that:

- for each $i \in N$ and $j \in \{i-1, i, i+1\} \cap \{1, \ldots, n\}$, $\{i, g_j\}$ is in E; and,
- values to the goods are randomly assigned, by taking them from the set $\{0, 0.40, 0.70, 1\}$.

Note that the treewidth of these graphs is 2 (see Fig. 11(a), for an illustration—in each of two vertical chains, agents and goods are alternated with one chain starting from an agent and the other from a good). Indeed, G_n consists of set of $n-1$ (pairwise adjacent) cycles, each one consisting of the edges $\{x, g_x\}$, $\{g_x, x+1\}$, $\{x+1, g_{x+1}\}$, and $\{g_{x+1}, x\}$ for $x \in \{1, \ldots, n-1\}$. The time required to compute the Shapley values in this setting is reported in Fig. 10(b). Again, it clearly emerges that the scaling is basically linear w.r.t. the number of agents.

7.2. Matching games

Implementation issues Similarly to the case of weighted-allocation games, some encoding tricks have been required to use *Sequoia* for evaluating the MSO formulas we associated in Section 4 with matching games. In particular, in that formulas we

Instance	N. of nodes	N. of edges	tw	N. of blocked coalitions
tree	10	9	1	2.86E+02
tree	20	19	1	2.92E+05
tree	30	29	1	3.11E+08
tree	40	39	1	3.15E+11
tree	50	49	1	3.22E+14
tree	60	59	1	3.29E+17
tree	70	69	1	1.47E+20
tree	80	79	1	3.26E+20
tree	90	89	1	4.59E+20
tree	100	99	1	5.56E+20
ladder-9	18	25	2	5.92E+04
ladder-15	30	43	2	2.39E+08
ladder-21	42	61	2	9.75E+11
ladder-24	48	70	2	6.23E+13
ladder-30	60	88	2	2.55E+17
halin-6	6	9	3	1.40E+01
halin-8	8	12	3	5.45E+01
halin-10	10	16	3	2.18E+02
halin-15	15	24	3	5.87E+03

(a) Synthetic Graphs

Instance	N. of nodes	N. of edges	tw	N. of blocked coalitions
VQR-1	10	11	2	2.64E+02
VQR-1b	10	14	3	3.29E+02
VQR-2	15	15	2	9.58E+03
VQR-2b	15	20	3	6.76E+03
VQR-3	20	21	2	2.98E+05
VQR-3b	20	25	3	2.79E+05
VQR-4	25	26	2	1.02E+07
VQR-4b	25	29	3	4.82E+06
VQR-5	30	32	2	2.70E+08
VQR-5b	30	36	3	2.94E+08
VQR-6	35	37	2	9.00E+09
VQR-6b	35	42	3	8.30E+09
VQR-7	40	46	2	2.37E+11
VQR-7b	40	45	3	3.03E+11
VQR-8	45	51	2	1.10E+13
VQR-8b	45	52	3	1.04E+13

(b) VQR Graphs

Fig. 12. Statistics on instances for matching games.

need the ability to quantify over directed edges (to deal with the set B of ordered pairs in Lemma 4.8) which is a feature not available in the solver. The feature has been simulated as follows: First, for every undirected edge $\{x, y\}$ in the input graph, we add two auxiliary nodes with the label "*fresh*", as depicted in Fig. 6(b). Then, in the formula implementing the conditions reported in Lemma 4.8, a node set B_N can be used to simulate the set of directed edges B, with the intended meaning that $(x, y) \in B$ if a fresh node $e_{(x,y)}$, adjacent to x, belongs to B_N; conversely, $(y, x) \in B$ if a fresh node $e_{(y,x)}$, adjacent to y, belongs to B_N.

Datasets Experiments over matching games have been conducted over synthetic random graphs belonging to the classes of (a) ladder graphs, (b) complete binary trees, and (c) Halin graphs (see Fig. 11), whose treewidth is 2, 1, and 3, respectively. Experiments over the VQR scenario described above have been performed, too. A summary of some statistics for the graphs considered in the analysis is reported in Fig. 12. There, we also reported the number of "blocked coalitions" that *Sequoia* has discovered, that is, the number of coalitions for which $v^m_G(C \cup \{i\}) - v^m_G(C) = 0$ holds in Lemma 4.7 (which is used in the algorithm discussed in the proof of Theorem 4.10). The reader can notice that these numbers are quite huge and might also want to observe that we have considered graphs with up to 100 nodes, for which a naïve computation method would explore 2^{100} possible coalitions. Contrasted with these numbers, the results on the computation times we shall discuss below are quite impressive.

Results on syntectic graphs For the first set of experiments, we computed the Shapley value by varying the number of agents, for each of the three classes of graphs we have considered. Results are reported in Fig. 13. Note that, rather than just reporting the average computation time over five runs, the figure also reports the maximum and the minimum computation times we actually registered. It can be checked that, in all cases, the prototype scales nicely with respect to the size of the given graphs.

Results on VQR graphs Finally, we run our prototype on subgraphs of the largest connected component of the VQR graph of the University "La Sapienza" of Rome (as discussed above for the weighted-allocation games), with varying sizes and with treewidths 2 or 3. Results are reported in Fig. 14. As expected, it clearly emerges that the treewidth impacts significantly on the running time, but the algorithm nicely scales with respect to the size of the graph.

8. Discussion and conclusion

We have studied the problem of computing the Shapley value (and the Banzhaf value) of coalitional games implicitly (and succinctly) specified in terms of matching problems. It turns out that the problem is **#P**-complete, even in stringent settings. Motivated by this result and by further negative results in the literature, we have then identified islands of tractability by focusing on games defined over classes of graphs having bounded treewidth and on binary-clashes allocation games. Our

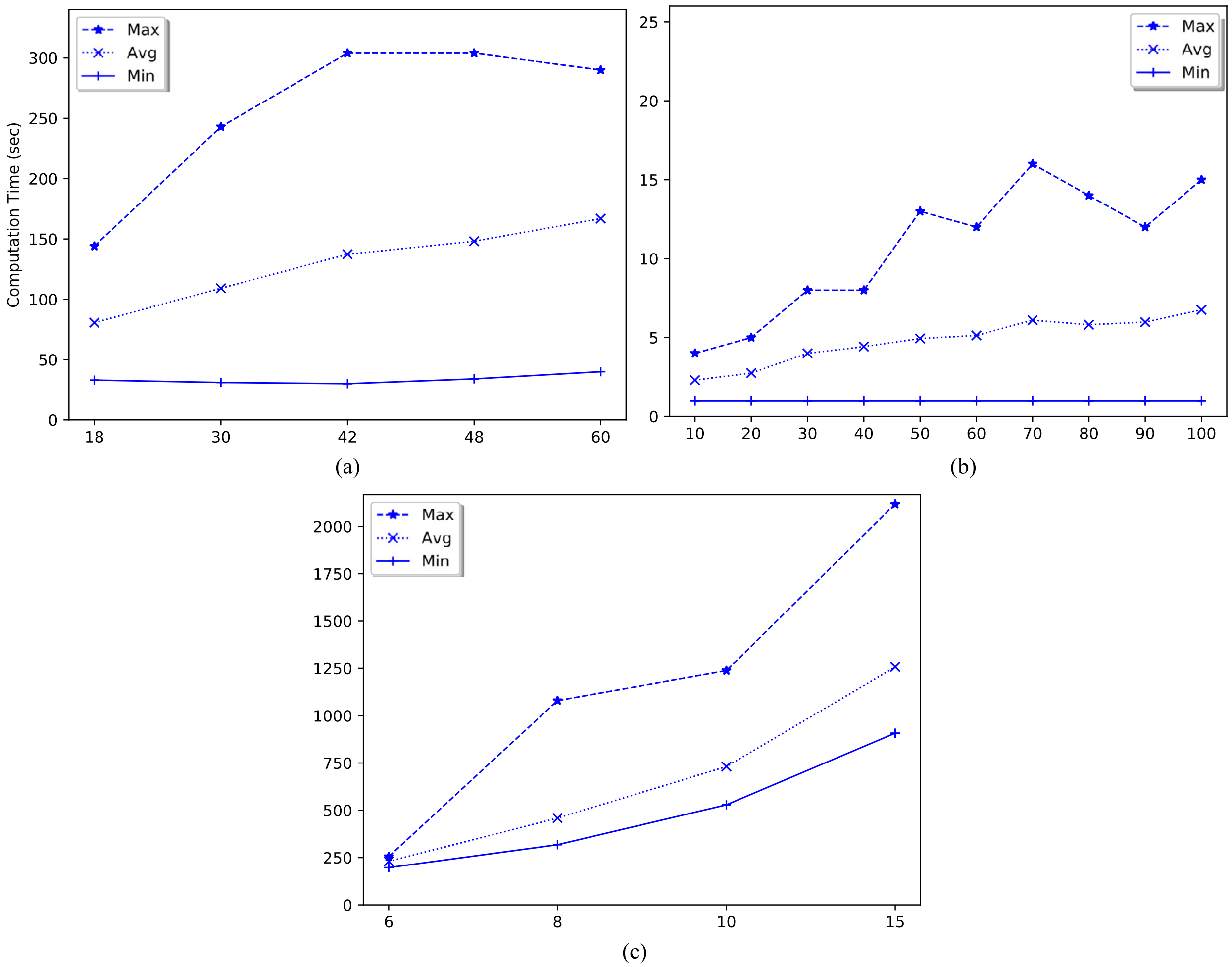


Fig. 13. Computation times for matching games w.r.t. the number of agents, on randomly weighted (a) ladder graphs, (b) complete binary trees, and (c) Halin graphs.

polynomial-time algorithms exploit novel and useful properties for matchings, which are then encoded in terms of Monadic Second Order logic (MSO). These algorithms have been implemented on top of a state-of-the-art MSO solver.

By looking at our findings, a number of avenues of further research clearly emerge. First, our work might stimulate further research to analyze, on classes of graphs having bounded treewidth, the complexity of other solution concepts for weighted-matching and weighted-allocation games, such as the least core and the nucleolus. With this respect, we can already notice that since our algorithms found on the fact that the computation of the Shapley value (and the Banzhaf value) can be reduced to counting the number of coalitions of a given size enjoying some MSO-definable property, then they can be smoothly applied to derive tractability results for different solution concepts that can be recast as *semivalues* [84].[9]

Second, we observe that weighted-matching (and weighted-allocation) games $\mathcal{G}_G^m$, with $G = (V, E, w)$, can be also thought as *Myerson*'s graph-restricted games [85]. Indeed, in the model proposed by Myerson, (V, E) defines the underlining communication structure enabling the cooperation of the agents, and it is required that the worth of any coalition C is the sum of the worth of the maximal coalitions $C' \subseteq C$ for which the subgraph induced over C' is connected. This property clearly holds on weighted-matching (and weighted-allocation) games, as noticed in the context of a study focusing on the core as a solution concept [86]—and, of course, the specific solution concept being considered does not play any role here. In fact, the Shapley and the Banzhaf values on Myerson's graph-restricted games have been studied too, in particular over graphs having bounded treewidth and by focusing on *weighted voting games* [87]. An interesting avenue of research is to assess whether effective MSO encodings can be defined for this class, as well as for other classes of games that can be naturally defined over graphs.

[9] A semivalue χ of a game $\mathcal{G}$ is such that $\chi_i(\mathcal{G}) = \sum_{C \subseteq N \setminus \{i\}} \gamma(|C|)\Big(v(C \cup \{i\}) - v(C)\Big)$, for some properly chosen function γ depending only on the size of the given coalition—see [84], for a formal treatment.

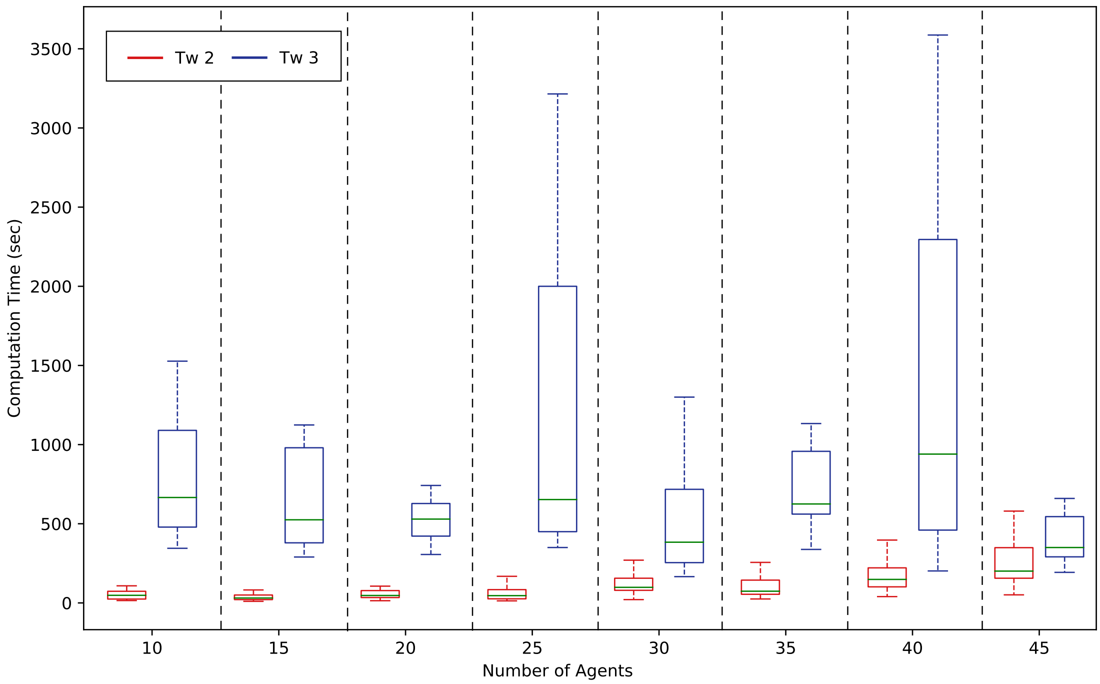


Fig. 14. Computation times for matching games w.r.t. the number of agents on VQR graphs. For each given number of agents, times are reported for components having treewidth 2 (on the left) and having treewidth 3 (on the right).

From an algorithmic viewpoint, it would be interesting to study the entry that is still marked as "open" in Table 1, that is, to study whether our tractability results in Section 4 can be extended to weighted matchings. It is also interesting to investigate whether our tractability result in Section 6 is tight or whether, instead, it holds even for classes of graphs where each good/node has degree three at most.[10] Moreover, it remains to look for islands of tractability for those allocation problems where agents have subjective values on the goods, which are modeled by bipartite graphs with arbitrary weights. Eventually, another interesting research question is whether, on games defined by bounded-treewidth graphs, the Shapley value is not only tractable, but even *fixed-parameter* tractable, that is, solvable in time $f(k) \cdot ||G||^{O(1)}$, for each graph G with $tw(G) \leq k$, where f is a function depending only on the parameter k.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

We are grateful to the anonymous referees for their useful suggestions that helped us to improve the quality of the paper. Moreover, we are indebted to Angelo Mendicelli for his support in setting-up the framework for the experimental analysis, in particular for dealing with Sequoia.

Gianluigi Greco and Francesco Scarcello were partially supported by Regione Calabria under POR projects "Explora Process" and "OSME: The Open Source Monetization Ecosystem", respectively.

References

[1] G. Chalkiadakis, E. Elkind, M. Wooldridge, Computational Aspects of Cooperative Game Theory (Synthesis Lectures on Artificial Intelligence and Machine Learning), 1st edition, Morgan & Claypool Publishers, 2011.
[2] G. Chalkiadakis, G. Greco, E. Markakis, Characteristic function games with restricted agent interactions: core-stability and coalition structures, Artif. Intell. 232 (2016) 76–113.
[3] A. Iwasaki, S. Ueda, N. Hashimoto, M. Yokoo, Finding core for coalition structure utilizing dual solution, Artif. Intell. 222 (C) (2015) 49–66.
[4] T. Ågotnes, W. van der Hoek, M. Wooldridge, Reasoning about coalitional games, Artif. Intell. 173 (1) (2009) 45–79.

[10] One may recall that computing the number of perfect matchings (cf. proof of Theorem 3.1) is **#P**-hard even on graph whose nodes have degree 3 [88]. Hence, a negative answer to the question would not be surprising.

[5] M. Wooldridge, P.E. Dunne, On the computational complexity of qualitative coalitional games, Artif. Intell. 158 (1) (2004) 27–73.
[6] O. Shehory, S. Kraus, Methods for task allocation via agent coalition formation, Artif. Intell. 101 (1–2) (1998) 165–200.
[7] F. Bistaffa, A. Farinelli, G. Chalkiadakis, S.D. Ramchurn, A cooperative game-theoretic approach to the social ridesharing problem, Artif. Intell. 246 (2017) 86–117.
[8] Y. Bachrach, D.C. Parkes, J.S. Rosenschein, Computing cooperative solution concepts in coalitional skill games, Artif. Intell. 204 (2013) 1–21.
[9] P.E. Dunne, S. Kraus, E. Manisterski, M. Wooldridge, Solving coalitional resource games, Artif. Intell. 174 (1) (2010) 20–50.
[10] M. Wooldridge, P.E. Dunne, On the computational complexity of coalitional resource games, Artif. Intell. 170 (10) (2006) 835–871.
[11] X. Deng, C.H. Papadimitriou, On the complexity of cooperative solution concepts, Math. Oper. Res. 19 (2) (1994) 257–266.
[12] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, On the complexity of core, kernel, and bargaining set, Artif. Intell. 175 (12–13) (2011) 1877–1910.
[13] V. Conitzer, T. Sandholm, Complexity of constructing solutions in the core based on synergies among coalitions, Artif. Intell. 170 (6–7) (2006) 607–619.
[14] H. Aziz, F. Brandt, H.G. Seedig, Computing desirable partitions in additively separable hedonic games, Artif. Intell. 195 (2013) 316–334.
[15] X. Deng, T. Ibaraki, H. Nagamochi, Algorithmic aspects of the core of combinatorial optimization games, Math. Oper. Res. 24 (3) (1999) 751–766.
[16] J.M. Bilbao, Cooperative games on combinatorial structures, in: Theory and Decision Library C, vol. 26, Kluwer Academinc Publishers, Reading, MA, USA, 2000.
[17] X. Deng, Ch. Combinatorial optimization games, in: C.A. Floudas, P.M. Pardalos (Eds.), Encyclopedia of Optimization, Springer US, 2009, pp. 387–391.
[18] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, Non-transferable utility coalitional games via mixed-integer linear constraints, J. Artif. Intell. Res. 38 (2010) 633–685.
[19] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, The complexity of the nucleolus in compact games, ACM Trans. Comput. Theory 7 (1) (2014) 3, 52 pp.
[20] U. Faigle, W. Kern, S.P. Fekete, W. Hochstättler, On the complexity of testing membership in the core of min-cost spanning tree games, Int. J. Game Theory 26 (3) (1997) 361–366.
[21] Y. Bachrach, The least-core of threshold network flow games, in: Proc. of MFCS, 2011, pp. 36–47.
[22] Y. Bachrach, J.S. Rosenschein, Power in threshold network flow games, Auton. Agents Multi-Agent Syst. (2009) 106–132.
[23] X. Deng, Q. Fang, X. Sun, Finding nucleolus of flow game, J. Comb. Optim. 18 (1) (2009) 64–86.
[24] Y. Bachrach, E. Porat, J.S. Rosenschein, Sharing rewards in cooperative connectivity games, J. Artif. Intell. Res. 47 (2013) 281–311.
[25] Y. Bachrach, E. Porat, Path disruption games, in: Proc. of AAMAS, 2010, pp. 1123–1130.
[26] M.X. Goemans, M. Skutella, Cooperative facility location games, J. Algorithms 50 (2) (2004) 194–214.
[27] U. Faigle, S.P. Fekete, W. Hochstättler, W. Kern, On approximately fair cost allocation in Euclidean tsp games, OR Spektrum 20 (1) (1998) 29–37.
[28] Q. Fang, L. Kong, Core stability of vertex cover games, in: Proc. of WINE, 2007, pp. 482–490.
[29] L.S. Shapley, M. Shubik, The assignment game I: the core, Int. J. Game Theory 1 (1) (1971) 111–130.
[30] A.E. Roth, M. Sotomayor, Two-sided matching, in: Handbook of Game Theory with Economic Applications, vol. 1, Elsevier, 1992, pp. 485–541.
[31] M. Le Breton, S. Weber, Stability of coalition structures and the principle of optimal partitioning, in: Social Choice, Welfare and Ethics, Cambridge University Press, 1995, pp. 301–319.
[32] K. Eriksson, J. Karlander, Stable outcomes of the roommate game with transferable utility, Int. J. Game Theory 29 (4) (2001) 555–569.
[33] Q. Fang, B. Li, X. Sun, J. Zhang, J. Zhang, Computing the least-core and nucleolus for threshold cardinality matching games, Theor. Comput. Sci. 609 (P2) (2016) 500–510.
[34] H. Moulin, An application of the Shapley value to fair division with money, Econometrica 60 (6) (1992) 1331–1349.
[35] C. Demetrescu, F. Lupia, A. Mendicelli, A. Ribichini, F. Scarcello, M. Schaerf, On the Shapley value and its application to the Italian VQR research assessment exercise, J. Informetr. 13 (1) (2019) 87–104.
[36] D. Mishra, B. Rangarajan, Cost sharing in a job scheduling problem, Soc. Choice Welf. 29 (3) (2007) 369–382.
[37] F. Maniquet, A characterization of the Shapley value in queueing problems, J. Econ. Theory 109 (1) (2003) 90–103.
[38] G. Greco, F. Scarcello, Mechanisms for fair allocation problems: no-punishment payment rules in verifiable settings, J. Artif. Intell. Res. 49 (2014) 403–449.
[39] G. Greco, F. Scarcello, Fair division rules for funds distribution: the case of the Italian research assessment program (VQR 2004-2010), Intell. Artif. 7 (1) (2013) 45–56.
[40] D.B. Gillies, Solutions to general non-zero-sum games, in: A.W. Tucker, R.D. Luce (Eds.), Contributions to the Theory of Games, in: Annals of Mathematics Studies, vol. 40, Princeton University Press, 1959, pp. 47–85.
[41] F.Y. Edgeworth, Mathematical Psychics: an Essay on the Mathematics to the Moral Sciences, C. Kegan Paul & Co., London, 1881.
[42] A. Alkan, D. Gale, The core of the matching game, Games Econ. Behav. 2 (3) (1990) 203–212.
[43] P. Biró, W. Kern, D. Paulusma, Computing solutions for matching games, Int. J. Game Theory 41 (1) (2012) 75–90.
[44] D. Schmeidler, The nucleolus of a characteristic function game, SIAM J. Appl. Math. 17 (6) (1969) 1163–1170.
[45] W. Kern, D. Paulusma, Matching games: the least core and the nucleolus, Math. Oper. Res. 28 (2) (2003) 294–308.
[46] L.S. Shapley, A value for n-person games, in: H.W. Kuhn, A.W. Tucker (Eds.), Contributions to the Theory of Games II, Princeton University Press, 1953, pp. 307–317.
[47] H. Aziz, B. de Keijzer, Shapley meets Shapley, in: Proc. of STACS, 2014, pp. 99–111.
[48] N. Bousquet, The Shapley value of matching games on trees, in: http://www.math.mcgill.ca/nbousquet/shapley.pdf, 2015.
[49] N. Robertson, P. Seymour, Graph minors. III. Planar tree-width, J. Comb. Theory, Ser. B 36 (1) (1984) 49–64.
[50] J.F. Banzhaf, Weighted voting doesn't work: a mathematical analysis, Rutgers Law Rev. 19 (1965) 317–343.
[51] B. Courcelle, Graph rewriting: an algebraic and logic approach, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, The MIT Press, 1990, pp. 193–242.
[52] A. Langer, Fast Algorithms for Decomposable Graphs, Ph.D. thesis, RWTH Aachen University, 2013.
[53] G. Greco, F. Lupia, F. Scarcello, The tractability of the Shapley value over bounded treewidth matching games, in: Proc. of IJCAI, 2017, pp. 1046–1052.
[54] G. Greco, F. Lupia, F. Scarcello, Structural tractability of Shapley and Banzhaf values in allocation games, in: Proc. of IJCAI, 2015, pp. 547–553.
[55] M.J. Osborne, A. Rubinstein, A Course in Game Theory, The MIT Press, Cambridge, MA, USA, 1994.
[56] G. Gottlob, G. Greco, F. Scarcello, Treewidth and hypertree width, in: Tractability: Practical Approaches to Hard Problems, Cambridge University Press, 2014, pp. 3–38.
[57] G. Gottlob, N. Leone, F. Scarcello, Hypertree decompositions and tractable queries, J. Comput. Syst. Sci. 64 (3) (2002) 579–627.
[58] M. Grohe, D. Marx, Constraint solving via fractional edge covers, ACM Trans. Algorithms 11 (1) (2014) 4, 20 pp.
[59] G. Gottlob, Z. Miklós, T. Schwentick, Generalized hypertree decompositions: NP-hardness and tractable variants, J. ACM 56 (6) (2009) 30, 32 pp.
[60] G. Greco, F. Scarcello, Greedy strategies and larger islands of tractability for conjunctive queries and constraint satisfaction problems, Inf. Comput. 252 (2017) 201–220.
[61] G. Greco, F. Scarcello, The power of local consistency in conjunctive queries and constraint satisfaction problems, SIAM J. Comput. 46 (3) (2017) 1111–1145.
[62] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, NY, USA, 1979.
[63] M. Elberfeld, A. Jakoby, T. Tantau, Logspace versions of the theorems of bodlaender and courcelle, in: Proc. of FOCS, 2010, pp. 143–152.

[64] A. Langer, F. Reidl, P. Rossmanith, S. Sikdar, Practical algorithms for MSO model-checking on tree-decomposable graphs, Comput. Sci. Rev. 13–14 (2014) 39–74.
[65] S. Ieong, Y. Shoham, Marginal contribution nets: a compact representation scheme for coalitional games, in: Proc. of EC, 2005, pp. 193–202.
[66] E. Elkind, L.A. Goldberg, P.W. Goldberg, M. Wooldridge, A tractable and expressive class of marginal contribution nets and its applications, Math. Log. Q. 55 (4) (2009) 362–376.
[67] S. Ieong, Y. Shoham, Multi-attribute coalitional games, in: Proc. of EC, 2006, pp. 170–179.
[68] V. Conitzer, T. Sandholm, Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains, in: Proc. of AAAI, 2004, pp. 219–225.
[69] T.P. Michalak, K.V. Aadithya, P.L. Szczepanski, B. Ravindran, N.R. Jennings, Efficient computation of the Shapley value for game-theoretic network centrality, J. Artif. Intell. Res. 46 (1) (2013) 607–650.
[70] P.L. Szczepanski, Fast algorithms for game-theoretic centrality measures, CoRR abs/1512.01764, 2015.
[71] S. Bhagat, A. Kim, S. Muthukrishnan, U. Weinsberg, The Shapley value in knapsack budgeted games, in: Proc. of WINE, 2014, pp. 106–119.
[72] H. Nagamochi, D.-Z. Zeng, N. Kabutoya, T. Ibaraki, Complexity of the minimum base game on matroids, Math. Oper. Res. 22 (1997) 146–164.
[73] K. Prasad, J.S. Kelly, Np-completeness of some problems concerning voting games, Int. J. Game Theory 19 (1) (1990) 1–9.
[74] M. Zuckerman, P. Faliszewski, Y. Bachrach, E. Elkind, Manipulating the quota in weighted voting games, Artif. Intell. 180–181 (2012) 1–19.
[75] S.S. Fatima, M. Wooldridge, N.R. Jennings, A linear approximation method for the Shapley value, Artif. Intell. 172 (14) (2008) 1673–1699.
[76] D. Liben-Nowell, A. Sharp, T. Wexler, K. Woods, Computing Shapley value in supermodular coalitional games, in: Proc. of COCOON, Springer, Berlin, Heidelberg, 2012, pp. 568–579.
[77] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1993.
[78] C.J. Colbourn, J.S. Provan, D. Vertigan, The complexity of computing the tutte polynomial on transversal matroids, Combinatorica 15 (1) (1995) 1–10.
[79] H. Aziz, O. Lachish, M. Paterson, R. Savani, Power indices in spanning connectivity games, in: Proc. of AAIM, 2009, pp. 55–67.
[80] E.H. Bareiss, Sylvester's identity and multistep integer-preserving Gaussian elimination, Math. Comput. 22 (1968) 565.
[81] C. Berge, Two theorems in graph theory, Proc. Natl. Acad. Sci. USA 43 (9) (1957) 842–844.
[82] B. Courcelle, J. Makowsky, U. Rotics, On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic, Discrete Appl. Math. 108 (1–2) (2001) 23–52.
[83] F. Lupia, A. Mendicelli, A. Ribichini, F. Scarcello, M. Schaerf, Computing the Shapley value in allocation problems: approximations and bounds, with an application to the Italian VQR research assessment program, J. Exp. Theor. Artif. Intell. 30 (4) (2018) 505–524.
[84] P. Dubey, A. Neyman, R.J. Weber, Value theory without efficiency, Math. Oper. Res. 6 (1) (1981) 122–128.
[85] R.B. Myerson, Graphs and cooperation in games, Math. Oper. Res. 2 (3) (1977) 225–229.
[86] R. Meir, Y. Zick, E. Elkind, J.S. Rosenschein, Bounding the cost of stability in games over interaction networks, in: Proc. of AAAI, 2013, pp. 690–696.
[87] O. Skibski, T.P. Michalak, Y. Sakurai, M. Yokoo, A pseudo-polynomial algorithm for computing power indices in graph-restricted weighted voting games, in: Proc. of IJCAI, 2015, pp. 631–637.
[88] P. Dagum, M. Luby, Approximating the permanent of graphs with large factors, Theor. Comput. Sci. 102 (2) (1992) 283–305.