Computing the Shapley Value in Allocation Problems: Approximations and Bounds, with an Application to the Italian VQR Research Assessment Program

Francesco Lupia[1], Angelo Mendicelli[1], Andrea Ribichini[2], Francesco Scarcello[1], and Marco Schaerf[3]

[1] Dept. of Computer Science, Modeling, Electronics and Systems Engineering, University of Calabria, 87036, Rende, Italy
{lupia,a.mendicelli,scarcello}@dimes.unical.it
[2] Dept. of Physics, Sapienza University, 00185, Rome, Italy
ribichini@dis.uniroma1.it
[3] Dept. of Computer, Control, and Management Engineering Antonio Ruberti, Sapienza University, 00185, Rome, Italy
marco.schaerf@uniroma1.it

Abstract. In allocation problems, a given set of goods are assigned to agents in such a way that the social welfare is maximized, that is, the largest possible global worth is achieved. When goods are indivisible, it is possible to use money compensation to perform a fair allocation taking into account the actual contribution of all agents to the social welfare. Coalitional games provide a formal mathematical framework to model such problems, in particular the Shapley value is a solution concept widely used for assigning worths to agents in a fair way. Unfortunately, computing this value is a #P-hard problem, so that applying this good theoretical notion is often quite difficult in real-world problems.
In this paper, we first review the application of the Shapley value to an allocation problem that models the evaluation of the Italian research structures with a procedure known as VQR. For large universities, the problem involves thousands of agents and goods (here, researchers and their research products). We then describe some useful properties that allow us to greatly simplify many such large instances. Moreover, we propose new algorithms for computing lower bounds and upper bounds of the Shapley value, which in some cases provide the exact result and that can be combined with approximation algorithms. The proposed techniques have been tested on large real-world instances of the VQR research evaluation problem.

1 Introduction

1.1 Coalitional Game Theory

Coalitional games provide a rich mathematical framework to analyze interactions between intelligent agents. We consider coalitional games of the form $\mathsf{G} = \langle N, v \rangle$, consisting of a set N of n agents and a characteristic function v. The latter maps each coalition $C \subseteq N$ to the worth that agents in C can obtain by collaborating with each other. In this context, the crucial problem is to find a mechanism to allocate the worth $v(N)$, i.e., the value of the grand-coalition N, in a way that is fair for all players and that additionally satisfies some further important properties such as efficiency: we distribute precisely the available budget $v(N)$ to players (not more and not less). Moreover, for stability reasons, it is usually required that every group of agents C gets at least the worth $v(C)$ that it can guarantee to the game.

Several solution concepts have been considered in the literature as “fair allocation” schemes and, among them, a prominent one is the *Shapley value* [16]. According to this notion, the worth of any agent i is determined by considering its actual contribution to all the possible coalitions of agents. More precisely, it is considered the so-called *marginal contribution* to any coalition C, that is, the difference between what can be obtained when i collaborates with the agents in C and what can be obtained without the contribution of i. More formally, the Shapley value of a player $i \in N$ is defined by the following weighted average of all such marginal contributions:

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

1.2 Allocation Games

Among the various classes of coalitional games, we focus in this paper on *allocation games*, which is a setting for analyzing fair division problems where monetary compensations are allowed and utilities are quasi-linear [11]. Allocation games naturally arise in various application domains, ranging from house allocation to room assignment-rent division, to (cooperative) scheduling and task allocation, to protocols for wireless communication networks, and to queuing problems (see, e.g., [4,9,10,11,13] and the references therein).

Computing the Shapley value of such games is a difficult problem, indeed it is #P-hard even if goods can only have two different possible values [1]. In this paper we focus on large instances of this problem, involving thousands of agents and goods, for which no algorithm described in the literature is able to provide an exact solution. There are however some promising recent advances that identify islands of tractability for the allocation problems where at most one good is allocated to each agent: it has been recently shown that those instances where the treewidth of the agents' interaction-graph is bounded by some constant (i.e., have a low degree of cyclicity) can be solved in polynomial-time [1]. The result is based on recent advances on counting solutions of conjunctive queries with existential variables [3]. Unfortunately, if the structure is quite cyclic this technique cannot be applied to large instances, because its computational complexity has an exponential dependency on the treewidth.

In some applications, one can be satisfied with some approximations of the Shapley value. With this respect, things are quite good in principle, since we know there exists a fully polynomial-time randomised approximation scheme to compute the Shapley value in supermodular games [8]. The algorithm can thus be tuned to obtain the desired maximum expected error. However, not very surprisingly, for very large instances one has to consider a huge number of samples, in order to stay below a reasonable expected error.

1.3 Contribution

In order to attack large instances of allocation problems, we start by proving some useful properties of these problems that allow us to decompose instances into smaller pieces, which can be solved independently. Moreover, some of these properties identify cases where the computation of the worth function can be obtained in a very efficient way.

With these properties, we are able to use the randomised approximation algorithm even on instances that (when not decomposed) are very large.

Furthermore, we note that in some applications one may prefer to determine a guaranteed interval for the Shapley value, rather than one probably good point. Therefore, we propose algorithms for computing a lower bound and an upper bound of the Shapley value for allocation problems. In most cases the distance between the two bounds is quite small, and often they coincide, which means that we actually computed the exact value. We also used these algorithms together with the approximation algorithm, to provide a more accurate evaluation of the maximum error of the randomised solution, for the considered instances.

1.4 The Case Study

The proposed techniques have been tested on large real-world instances of the VQR2011-2014 Italian research assessment exercise: Every Italian research structure R has to select some research products, and submit them to an evaluation agency called ANVUR. While doing so, the structure R is in competition with all other Italian research structures, as the outcome of the evaluation will be used to proportionally transfer the funds allocated by the Ministry to support research activities in the next years (until the subsequent evaluation process). Every structure R is therefore interested in selecting and submitting its *best* research products. For the sake of simplicity, we next simply speak of publications instead of research products (which can also be patents, books, etc.), and of universities and departments instead of structures and substructures (which can be other research subjects). The programme is articulated in two phases: (1) Based on the self-evaluations[4] of the authors, R selects and submits to ANVUR (at most) two publications for each one of its authors[5], in such a way that any product is formally associated with at most one author. (2) ANVUR formulates its independent quality judgment about the submitted publications (the score assigned to each publication is made known only to its authors), and the sum of their "true" scores (i.e., those resulting by ANVUR actual evaluation) is then the VQR score of R. Eventually, R will receive funds in subsequent years proportional to this score. Furthermore, for the VQR2004-2010 research assessment exercise (data for VQR2011-2014 have not been released yet) ANVUR also published an evaluation of all departments, based on the product scores (the score of each department was computed as the sum of the scores of the products formally assigned to the authors in that department). Finally, the scores were

[4] To be more precise, ANVUR provided reference tables to assist in the evaluation of products, at least for some research areas. To our ends, this detail is immaterial.

[5] There are exceptions to this rule: in specific circumstances, fewer than two publications are expected for some authors. To our ends, this detail is immaterial.

also used for evaluating individual researchers that had been recently hired by R (this also greatly influenced R's funds in subsequent years), as well as those researchers that were members of PhD committees. Scores for recently hired researchers were computed as the sum of the scores of the products formally assigned to them; data in this respect were published by ANVUR in aggregated form only, for each department and for each scientific disciplinary sector. Evaluations for researchers that were members of PhD committees were computed as the sum of the scores of the best publications each one of them had coauthored, among all the publications submitted for the VQR (for this evaluation, the formal assignment of publications to authors was irrelevant); data in this respect were published by ANVUR in aggregated form only, for each PhD committee.

The way ANVUR currently uses product scores, for the purposes described above, yields evaluations that do not satisfy the desirable properties outlined in Section 4. In order to deal with this issue, we have modeled the problem as an allocation game [2], with a fair way to divide the total score of the university among researchers, groups, departments based on the Shapley value. The proposed rule enjoys many desirable properties, such as the independence of the specific allocation of research products, the independence of the preliminary (optimal) products selection, the guarantee of the actual (marginal) contribution, and so on. Unfortunately, due to the high computational complexity of the approach we propose, it proved unfeasible to compute the desired allocation for our largest test case (namely, the researchers of Sapienza University of Rome who participated in the research assessment exercise VQR2011-2014), even using the approximation algorithms known in the literature. We are therefore proposing improvements and modifications to the approximate algorithms, in an attempt to solve this problem. Our approach has yielded some very encouraging results (even though, as reported in Section 7.2, we are still not able to compute, or even approximate within acceptable bounds, an allocation for a non–negligible fraction of the agents in the Sapienza test case), outlining a promising line of research.

2 Preliminaries

In the setting considered in this paper, a game is defined by an allocation scenario $\mathcal{A} = \langle N, \mathbb{G}, \Omega, \mathtt{val}, k\rangle$ comprising a set of agents N and a set of goods $\mathbb{G}$, whose values are given by the real function $\mathtt{val}$. The function Ω associates each agent with the set of goods (s)he is interested in. Moreover, the natural number k provides the maximum number of goods that can be assigned to each agent. Goods are indivisible and unsharable.

For a coalition $C \subseteq N$, a (feasible) allocation $\pi_{\mathcal{A}}[C]$ is a mapping from C to sets of goods from $\mathbb{G}$ such that: each agent $i \in C$ gets a set of goods $\pi_{\mathcal{A}}(i) \subseteq \Omega(i)$ with $|\pi_{\mathcal{A}}(i)| \leq k$, and $\pi_{\mathcal{A}}(i) \cap \pi_{\mathcal{A}}(j) = \emptyset$, for any other agent $j \in C$ (each good can be assigned to one agent at most).

We denote by $\mathtt{img}(\pi_{\mathcal{A}}[C])$ the set of all goods in the image of $\pi_{\mathcal{A}}[C]$, that is, $\mathtt{img}(\pi_{\mathcal{A}}[C]) = \bigcup_{i \in C} \pi_{\mathcal{A}}[C](i)$. With a slight abuse of notation, we denote by $\mathtt{val}(S)$ the sum of all the values of a set of goods $S \subseteq \mathbb{G}$, and by $\mathtt{val}(\pi_{\mathcal{A}}[C])$ the value $\mathtt{val}(\mathtt{img}(\pi_{\mathcal{A}}[C]))$. An allocation $\pi_{\mathcal{A}}[C]$ is optimal if there exists no allocation $\pi'_{\mathcal{A}}[C]$ with $\mathtt{val}(\pi'_{\mathcal{A}}[C]) > \mathtt{val}(\pi_{\mathcal{A}}[C])$. The total value of such an optimal allocation for the coalition C is denoted by $\mathtt{opt}_{\mathcal{A}}(C)$. The budget available for $\mathcal{A}$, also called the (maximum) social welfare is $\mathtt{opt}_{\mathcal{A}}(N)$, that is, the value of any optimal allocation for the whole set of agents N (the grand-coalition). The coalitional game defined by the scenario $\mathcal{A}$ is the pair $\langle N, \mathtt{opt}_{\mathcal{A}}\rangle$, that is, the game where the worth of any coalition is given by the value of any of its optimal allocations. Note that $\mathtt{opt}_{\mathcal{A}}(C) \geq 0$ holds, for each $C \subseteq N$, since the allocation where no agent receives any goods is a feasible one (the value of an empty set of goods is 0). The definition trivializes for $C = \emptyset$, with $\mathtt{opt}_{\mathcal{A}}(\emptyset) = 0$.

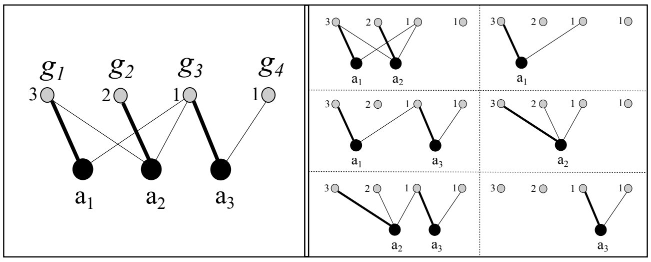

Fig. 1: Allocation scenario $\mathcal{A}_0$ in Example 1.

Example 1. Consider the allocation scenario $\mathcal{A}_0$, depicted in a graphical way in Figure 1. We have a set $\{g_1, g_2, g_3, g_4\}$ of goods that have to be allocated to three agents. For simplicity assume that it is possible to allocate just one good to each agent, therefore each edge connects an agent to a good she is interested in. Edges in bold identify an optimal allocation, i.e., a feasible allocation whose sum of values of the allocated goods is the maximum possible one. The value of this allocation is $\mathtt{val}(g_1) + \mathtt{val}(g_2) + \mathtt{val}(g_3) = 3 + 2 + 1 = 6$.

For each $C \subset \{1, 2, 3\}$ with $C \neq \emptyset$, an optimal allocation restricted to the agents in C is also reported. Then, the associated coalitional game is $\mathsf{G}_{\mathcal{A}_0} = \langle\{1, 2, 3\}, v_{\mathcal{A}_0}\rangle$, where $v_{\mathcal{A}_0}(\{1, 2, 3\}) = 6$, $v_{\mathcal{A}_0}(\{1, 2\}) = 5$, $v_{\mathcal{A}_0}(\{1, 3\})$ = $v_{\mathcal{A}_0}(\{2, 3\}) = 4$, $v_{\mathcal{A}_0}(\{1\}) = v_{\mathcal{A}_0}(\{2\}) = 3$, and $v_{\mathcal{A}_0}(\{3\}) = 1$. $\lhd$

For any allocation scenario $\mathcal{A} = \langle N, \mathbb{G}, \Omega, \mathtt{val}, k\rangle$, we define the *agents graph* as the undirected graph $G(\mathcal{A}) = (N, E)$ such that $\{i, j\} \in E$ if there is a good $g \in \Omega(i) \cap \Omega(j)$.

Notice that, in this specific case, the agents graph contains an edge between two agents (authors) if they are coauthors of at least one publication (we could call it the collaboration graph).

3 The VQR Allocation Game

Note that the VQR research assessment exercise can be naturally modeled as an allocation scenario $\mathcal{A} = \langle\mathcal{R}, \mathcal{P}, \mathit{products}, \mathtt{val}, 2\rangle$ where $\mathcal{R}$ is the set of researchers affiliated with a certain university R, $\mathcal{P}$ is the set of publications selected by R for the assessment exercise, *products* maps authors to the set of publications they have written, and `val` assigns a value to each publication. In the current VQR programme (covering years 2011-2014), the range of `val` is $\{0, 0.1, 0.4, 0.7, 1\}$, with the latter value reserved to the *excellent* products.

In the submission phase, the values are estimated by the universities according to criteria published by ANVUR. The final result may sometimes be different, in particular for those publications that undergo a peer-review process by experts selected by ANVUR, which clearly introduces a subjective factor in the evaluation. This is however immaterial for the purpose of this paper, so that we assume that the values in the preliminary phase do coincide with the final ANVUR evaluation for those products. At the end of the program, R will receive an amount of funds proportional to $V_R = \mathtt{val}(\mathcal{P})$, that is, to the considered measure of the quality of the research produced by the university R. The first combinatorial problem, which is easily seen to be a weighted matching problem, is to identify the best allocation scenario for the university. That is, to select a set of publications $\mathcal{P}$ to be submitted, having the maximum possible total value among all those authored by $\mathcal{R}$ in the considered period.

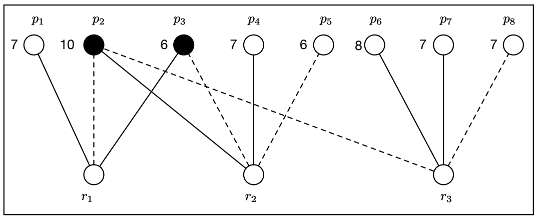

Fig. 2: Authors and products in Example 2.

Example 2. Let us consider the weighted bipartite graph in Figure 2, whose vertices are the researchers $\mathcal{R} = \{r_1, r_2, r_3\}$ of a university R and all the publications they have written. Edges encode the authorship relation, and weights encode the publications values. Consider the allocation ψ such that $\psi(r_1) = \{p_1, p_3\}$, $\psi(r_2) = \{p_2, p_4\}$, and $\psi(r_3) = \{p_6, p_7\}$, encoded by the solid lines in the figure. Then, the set of publications submitted for the evaluation is $\mathcal{P}_\psi = \{p_1, p_2, p_3, p_4, p_6, p_7\}$. Note that p_2 is co-authored by r_1, r_2, and r_3, while p_3 is co-authored by r_1 and r_2. According to the values of each product , we get $\mathtt{opt}(\mathcal{R}) = 45$. $\lhd$

The problem that we face is how to compute, from the total value obtained by R, a fair score for individual researchers, or groups, or departments, and so on. As mentioned above, product scores are currently used for evaluating the hiring policy of universities and the PhD committees, and it seems that such scores will contribute to evaluate the quality of teaching, too. Unfortunately, as we will argue, this is currently done in a way that fails to satisfy the properties that we outline below. Instead, following [2], we propose to use the Shapley value of the allocation game defined by the scenario selected by the given structure R as the *division rule* to distribute the available total value (or budget) to all the participating agents. For the allocation scenario in Example 2, we get $\phi_{r_1} = \frac{29}{2}$, $\phi_{r_2} = \frac{29}{2}$, and $\phi_{r_3} = 16$. Notice that the Shapley value is not a percentage assignment of publications to authors, but takes into account all possible coalitions of agents.

We will now recall the main desirable properties enjoyed by this solution. We refer the interested reader to [2] for a more detailed description and discussion of these properties.

Budget-balance. *The division rule precisely distributes the VQR score of R over all its members, i.e., $\sum_{r \in \mathcal{R}} \phi_r = V_R$.*

Fairness. *The division rule is indifferent w.r.t. the specific optimal allocation used to submit the products to ANVUR. In particular, the score of each researcher is independent of the particular products assigned to him in the submission phase; moreover, it is independent of the specific set of products $\mathcal{P}$ selected by the university, as long as the choice is optimal (i.e., with the same maximum value V_R).*

Marginality. *For any group of researchers $\mathcal{S} \subseteq \mathcal{R}$, $\phi_{\mathcal{S}} \geq marg(\mathcal{S}, \mathcal{R})$, where $\phi_{\mathcal{S}} = \sum_{i \in \phi_{\mathcal{S}}} \phi_i$ and $marg(\mathcal{S}, \mathcal{R}) = \texttt{opt}(\mathcal{R}) - \texttt{opt}(\mathcal{R} \setminus \mathcal{S})$. That is, every group is granted at least its marginal contribution to the performance of the grand-coalition $\mathcal{R}$.*

We remark the importance of the *fairness* property, as the choice of a specific optimal set of products is immaterial for R, but it may lead to quite different scores for individuals (and for their aggregations). Indeed, this happens for the division rules adopted by ANVUR for the evaluation of both departments and newly hired researchers (see Section 1.4), which are not based on the Shapley value. *Budget-balance*, on the other hand, is violated by the division rule for evaluating researchers who are members of PhD committees.

4 Useful properties to deal with large real-world problems

Recall that computing the Shapley value is #P-hard for many classes of games (see, e.g., [6,5,7,12]), including the allocation games, even if goods may have only two possible values [4]. In our case study, where agents are researchers and goods are research products, for large universities we may have to deal with thousands of agents and goods. The brute-force approach would need to compute, for each agent, $2^{|N|}$ optimization problems, which is clearly infeasible for our application.

In this section we describe some properties of the Shapley value, in particular for allocation problems, which allow us to solve some large real-world instances or at least to obtain good approximations for them. Let us consider in this section an allocation game $\mathsf{G} = \langle N, v \rangle$, whose agents graph is $G = (N, E)$.

Fact 1 (Modularity) *Let C_1 and C_2 be two sets of agents that induce two (disjoint) connected components of the agents graph, and denote by $\mathsf{G}_1 = \langle C_1, v_1 \rangle$ (resp., $\mathsf{G}_2 = \langle C_2, v_2 \rangle$) the coalitional game restricted to agents in C_1 (resp., C_2). Then, for each agent $i \in N$, $\phi_i(\mathsf{G}) = \phi_i(\mathsf{G}_1) + \phi_i(\mathsf{G}_2)$.*

Proof. It follows by the additivity property of the Shapley value: indeed, for each coalition $C \subseteq N$, we have $v(C) = v_1(C \cap C_1) + v_2(C \cap C_2)$.

Modularity is an important properties when dealing with large instances, because it states that we can solve in a separate way the problems associated with each component.

The next properties that we point out below try to leverage such property by removing agents interactions having no actual impact on the game output.

Recall that we assumed that all goods are non-negative. Then the following property, which trivially holds, says that we can always assume that any good with value 0 is not shared by agents.

Fact 2 (No shared null goods) *Let B the set of goods having value 0. By removing all such goods from G we get an equivalent allocation game. Also, if it is useful in algorithms, we can add any number of goods having no value to each agent, with all such goods being of interest to one agent only.*

Fact 3 (Disconnected agent) *Consider an agent i and a coalition C that is disconnected from i, that is, such that $Neigh(i) \cap C = \emptyset$. Then, $\texttt{opt}(\{i\} \cup C) = \texttt{opt}(\{i\}) + \texttt{opt}(C)$.*

Theorem 4 (Separability). *Let C be any coalition such that $\texttt{opt}(C) + \texttt{opt}(N \setminus C) \leq \texttt{opt}(N)$. Then, we can define two allocation scenarios over agents C and $N \setminus C$ that can be solved separately, that is, such that, for each player $i \in N$, its Shapley value on the restricted allocation scenario where i occurs is the same as in the original game.*

Proof. (Sketch.) Denote $N \setminus C$ by $\bar{C}$, and consider the games $\mathsf{G}_1 = \langle C, v_1 \rangle$ and $\mathsf{G}_2 = \langle \bar{C}, v_2 \rangle$ restricted to agents in C and $\bar{C}$, respectively. Since the allocation games are supermodular [4], we have $\texttt{opt}(C) + \texttt{opt}(\bar{C}) \geq \texttt{opt}(N)$, which entails that, for the considered coalition, $\texttt{opt}(C) + \texttt{opt}(\bar{C})$ is actually equal to $\texttt{opt}(N)$. This means that the values of the goods not used in any optimal allocation for $\bar{C}$ is equal to the sum of the values of the best goods for the agents in C.

We claim that in this case, for each optimal allocation π for N, a set of goods $S \subseteq \Omega(C)$ with $\texttt{val}(S) = \texttt{opt}(C)$ is assigned to players in C, so that both C and $\bar{C}$ get their best goods according to π. Assume that C gets a value $v = \texttt{val}(S) \leq \texttt{opt}(C)$, while $\bar{C}$ gets a value $\bar{v} \leq \texttt{opt}(\bar{C})$. We know that $\texttt{opt}(C) + \texttt{opt}(N \setminus C) = \texttt{opt}(N)$ and, by the optimality of π, $v + \bar{v} = \texttt{opt}(N)$, from which the claim follows.

Consider now any coalition $C' \subseteq N$, and let $C_a = C' \cap C$ and $C_b = C' \cap \bar{C}$. Let π' be an optimal allocation for C'. We claim that there is an optimal allocation π_a mapping goods from S to C with $\texttt{val}_{\pi_a}(C_a) = \texttt{val}_{\pi'}(C_a)$, and an optimal allocation π_b mapping goods not in S to $\bar{C}$ with $\texttt{val}_{\pi_b}(C_b) = \texttt{val}_{\pi'}(C_b)$. Assume this is not the case. Then at least one of those allocations lead to values smaller than those in π' (note that π' cannot be worse, because the union of the two restricted allocations is a valid candidate mapping for C'). Assume C_a gets smaller values (the other case is symmetrical), that is, $\texttt{val}_{\pi_a}(C_a) < \texttt{val}_{\pi'}(C_a)$. Then, there exists some agent i and a good $p \notin S$ so that $p \in \pi'(i)$. By using Theorem 4.4 in [4], we can show that this would contradict the fact that $\texttt{val}(S) = \texttt{opt}(C)$. Intuitively, goods such as p that are shared with agents outside C and that allows us to get a better value for the agents in $C_a \subseteq C$, could be used to improve the choice of the available goods S for the full set C.

The statement then follows easily by the additivity of the Shapley value, by considering two allocation scenarios restricted to C with the goods in S and to $\bar{C}$ with the goods outside S, respectively.

As a special case, when C is a singleton $\{i\}$, we can immediately drop from the game such an agent i for which the value of her best goods is equal to her marginal contribution to the grand-coalition. Note that we also immediately get $\phi(i) = \texttt{opt}(\{i\})$.

5 Approximating the Shapley Value

In order to approximate the Shapley value, we use the Fully Polynomial-time Randomized Approximation Scheme (FPRAS) proposed in [8]: for any $\epsilon > 0$ and $\delta > 0$, it is possible to compute in polynomial-time an $\epsilon-$approximation of the Shapley value with probability of failure at most δ. The technique works for supermodular and monotone coalitional games, and it can be shown that our allocation games indeed meet these properties [4].

The method is based on generating a certain number of permutations (of all agents) and computing the marginal contribution of each agent to the coalition of agents occurring before her (him) in the considered permutation. Then the Shapley value of each player is computed as the average of all such marginal contributions. The above procedure is repeated $O(\log(1/\delta))$ times, in indepedent runs, with the result for each agent consisting of the median of all computed values for her (him). Finally, the obtained values are scaled (i.e., they are all multiplied by a common numerical factor) to ensure that the budget-balance property is not violated.

Clearly enough, the more permutations are considered, the closer to the Shapley value the result will be. We next report a slightly modified version of the basic procedure of this algorithm, where we avoid the computation of some marginal contributions, if we can obtain the result by using Fact 3.

As a preliminary, we compute the required number of permutations m to meet the required error guarantee. In each of the m iterations, the algorithm generates a random permutation from the set of agents N. We then iterate through this permutation and compute the marginal contribution of each agent j to the set of agents C occurring before j in the permutation at hand. If some neighbour of j in the agents graph occurs in C, the algorithm proceeds as usual by resolving the optimisation problem to compute $v_{\mathcal{A}}(C \cup \{j\})$ (i.e., to get the value $\texttt{opt}(C \cup \{j\})$ of an optimal allocation for $C \cup \{j\}$). Note that to get the desired marginal contribution we need to solve one allocation problem only, because the value $\texttt{opt}(C)$ for all the preceding agents is known from the previous step. Moreover, by Fact 3, we know that for those permutations in which all the players in $Neigh(j)$ follow j, the marginal contribution of j is just $\texttt{opt}(\{j\})$ (see step 10). Finally, at steps 16–18 for each agent the algorithm divides the sum of her contributions by the number of performed iterations m. The correctness of the whole algorithm follows from Theorem 4 in [8].

Computation Time Analysis. Let $n = |N|$ be the number of agents, and let m be the required number of iterations. The cost of the algorithm is $O(m \times n \times margBlock)$, where $margBlock$ denotes the cost of computing each marginal contribution (steps 7–11). This requires the computation of an optimal weighted matching in a bipartite graph, which is feasible in $O(n^3)$, via the classical Hungarian algorithm. However, if the current agent is disconnected from the rest of the coalition, the cost is simply given by the lookup in the cache where the best allocation for each single agent is stored.

Algorithm 1 Shapley value approximation in allocation games

```
**Input:** An allocation game $\mathsf{G}_{\mathcal{A}} = \langle N, v_{\mathcal{A}} \rangle$;
**Parameters:** Real numbers $1 > \epsilon > 0$ and $1 > \delta > 0$;
**Output:** A vector $\tilde{\phi}$ that is an $\epsilon$-approximation of the Shapley value of $\mathsf{G}_{\mathcal{A}}$, with probability $1-\delta$;
$m = \frac{|N|\cdot(|N|-1)}{\delta\cdot\epsilon^2}$;
$i = 0$;
while $i < m$ do
    $\mathit{shuffle}(N)$;
    $C := \{\emptyset\}$;
    for all $j \in N$ do
        if $\mathit{Neigh}(j) \cap C \neq \emptyset$ then
            $\tilde{\phi}_j \mathrel{+}= v_{\mathcal{A}}(C \cup \{j\}) - v_{\mathcal{A}}(C)$;
        else
            $\tilde{\phi}_j \mathrel{+}= v_{\mathcal{A}}(\{j\})$;
        end if
        $C := C \cup \{j\}$;
        $i = i + 1$;
    end for
end while
for all $j \in N$ do
    $\tilde{\phi}_j = \frac{\tilde{\phi}_j}{m}$;
end for
return $\tilde{\phi}$;
```

6 Lower and Upper Bounds for the Shapley value

In this section we describe the computation of exact (not probabilistic) bounds for the Shapley value of any given allocation game $\mathsf{G}_{\mathcal{A}} = \langle N, v_{\mathcal{A}} \rangle$. Preliminarily recall that two very simple bounds are easily obtained by looking at the definition of the Shapley value: for each player i, $marg(\{i\}, N) \leq \phi_i \leq \mathtt{opt}(\{i\})$.

To obtain tighter bounds we observe that, intuitively, the neighbours of i in a coalition C are the agents having the higher influence on the marginal contribution of i to C. Indeed, they are precisely those agents interested in using the goods of i when (s)he does not belong to the coalition. We already observed that in the extreme case that no neighbours are present then i contributes with all her (his) best goods. Therefore, the first idea is to consider the power-set of $\mathit{Neigh}(i)$ to see what happens for every combination of these agents. The second observation is that, for each pair of coalitions $C_1 \subseteq C_2$, $marg(\{i\}, C_2) \leq marg(\{i\}, C_1)$. Putting it all together, we proceed as follows. Let P' be a set of neighbors of i, let $Z = \mathit{Neigh}(i) \setminus P'$, let $C = N \setminus (\mathit{Neigh}(i) \cup \{i\})$ and let $l = |C|$. We compute the marginal contribution of i to $C \cup P'$, but use this same value for the marginal contributions of i to each coalition of the form $C' \cup X$, for any $C' \subseteq C$. This is performed by multiplying such contribution by a factor y computed as $\sum_{k=0}^{l} \frac{(l-k+|P'|)!\cdot(|Z|+k)!}{|N|!} \cdot \binom{l}{k}$. Thus, for each element P' of the power set of the neighbours, we consider the worst situation for i where all other agents in C are present.

The case of the upper bound is obtained in the dual way, by using instead the most favorable case where only the neighbours in P' belong to the considered coalitions (i.e., $C = \emptyset$). We also use the same factor y, by using the property $\binom{l}{k} = \binom{l}{l-k}$.

Then, we can show the next algorithm compute, for each agent i, a lower bound LB_i and an upper bound UB_i for her/his Shapley value.

Algorithm 2 Computing Bounds for the Shapley Value in Allocation Games

Input: An allocation game $\mathsf{G}_{\mathcal{A}} = \langle N, v_{\mathcal{A}} \rangle$;
Output: A pair of vectors (LB, UB) encoding, respectively, a lower bound and an upper bound of the Shapley value of $\mathsf{G}_{\mathcal{A}}$;

```
 1: for all $i \in N$ do
 2:     $P := Powerset(Neigh(i))$;
 3:     $C := N \setminus (Neigh(i) \cup \{i\})$;
 4:     $l = |C|$;
 5:     for all $P' \in P$ do
 6:         $Z := Neigh(i) \setminus P'$;
 7:         $y = \sum_{k=0}^{l} \frac{(l-k+|P'|)! \cdot (|Z|+k)!}{|N|!} \cdot \binom{l}{k}$;
 8:         $LB_i \mathrel{+}= y \cdot (v_{\mathcal{A}}(C \cup P' \cup \{i\}) - v_{\mathcal{A}}(C \cup P'))$;
 9:         $UB_i \mathrel{+}= y \cdot (v_{\mathcal{A}}(P' \cup \{i\}) - v_{\mathcal{A}}(P'))$;
10:     end for
11: end for
12: return $(LB, UB)$;
```

Computation Time Analysis. The for loops in steps 5–10 show that both algorithms are exponential in the maximum degree of the agents graphs, that is, $\max_{i \in N} |Neigh(i)|$. In practice, after the simplifications that we perform on the instance, these sets are not large and the bounds can be computed in a reasonable time (see the next section).

7 Implementation Details and Experimental Evaluation

7.1 Parallel Implementation of the Approximate Algorithm

We found the approximate Shapley value algorithm of Niben-Nowell et al. [8] (we briefly discussed their approach in Section 5) to be amenable to parallel implemtation. We engineered our parallel implementation as follows. Besides the input allocation game, and the two parameters δ and ϵ, we added a third parameter, the *thread pool size*. During the execution of the algorithm, each thread (there are as many threads as the thread pool size dictates) is responsible for generating a certain number of permutations according to the requested approximation factor and, for each permutation, it computes the marginal contributions of all authors to that permutation, and saves them in a local cache. When a thread has generated its assigned number of permutations, it delivers its local cache of computed scores to a synchronized output acceptor (which increments the overall score of each author accordingly), and then shuts itself down, as its work is completed. When all threads have shut down, each entry of the acceptor's output vector is averaged over the total number of permutations, yielding the final approximate Shapley vector for that run. The above procedure is repeated for each independent run. When all runs are done, the component-wise median of all final approximate Shapley vectors is computed, and the resulting vector is scaled (i.e., all entries are multiplied by a number such that the Shapley values budget balance property is enforced), yielding the desired approximation with the desired probability.

7.2 Experimental Results

Hardware and software configuration. Experiments have been performed on two dedicated machines. In particular, sequential algorithms were run on a machine with an Intel Core i7-3770k 3.5 GHz processor, 12 GB (DDR3 1600 MHz) of RAM, and operating system Linux Debian Jessie. We tested the parallel implementation on a machine equipped with two Intel Xeon E5-4610 v2 @ 2.30GHz with 8 cores and 16 logical processors each, for a total of 32 logical processors, 128 GB of RAM, and operating system Linux Debian Wheezy. Algorithms were implemented in Java, and the code was executed on the JDK 1.8.0 05-b13, for the Intel Core i7 machine, and on the OpenJDK Runtime Environment (IcedTea 2.6.7) (7u111-2.6.7-1 deb7u1), for the Intel Xeon machine.

Dataset description. We applied the algorithms to the computation of a fair division of the scores for the researchers of Sapienza University of Rome who partecipated in the research assessment exercise VQR2011-2014. Sapienza contributors to the exercise were 3562 and almost all were required to submit 2 publications for review. We computed the scores of each publication by applying, when available, the bibliographic assessment tables provided by ANVUR.

Preprocessing. The analysis was carried out by preliminarily simplifying the input using the properties discussed in Section 4, as explained next.

Starting with a setting with 3562 researchers and 5909 publications, first we removed each researcher having no publications for review. After this step a total of 370 authors were removed. Then, by exploiting the simplification described in Fact 2, we removed 2323 publications. By using Theorem 4, the graph was subsequently filtered removing each author whose marginal contribution to the grand coalition coincides with the optimal allocation restricted to the author himself. After this step 2427 researchers out of 3562 were removed. Then we divided the resulting agents graph into connected components obtaining a total number of 156 connected components and we discovered only two components consisting of more than 10 agents. The sizes of these components are 691 and 15. Eventually, the components were further decomposed as follows: For any agent i, a good g is removed from its set $\Omega(i)$ if $\texttt{val}(g) + \max_{g' \in \Omega(i) \setminus \{g\}} \texttt{val}(g') < v(N) - v(N \setminus \{i\})$. After the whole preprocessing phase, we obtained a total of 159 connected components with the largest one having 685 nodes. The size of the second largest is just 15 while all the others remain very small (less than 10 nodes). In the rest of the section, we shall illustrate results of experimental activity conducted over the various methods. To this end, we fixed the value $\delta = 0.01$.

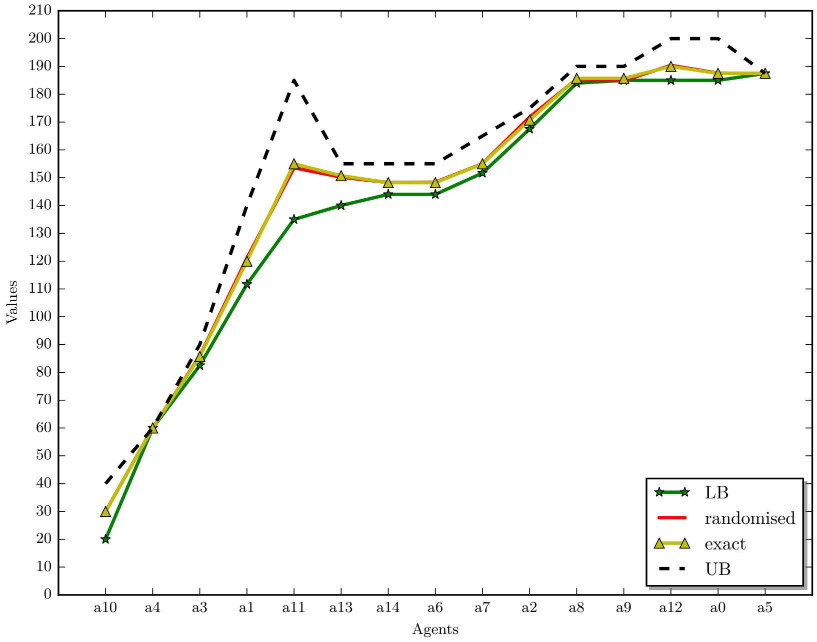

Fig. 3: Methods comparison ($n = 15$)

The value was chosen heuristically based on a series of tests conducted on various CNR Areas of Sapienza, where CNR Areas are (large) scientific disciplines such as Math and Computer Science (Area 01) or Physics (Area 02).

Test with components of variable size. As we have already pointed out, after the preprocessing step we obtained very small connected components (less than 10 nodes) except for the first two (685 and 15 nodes, respectively). For all components with less than 10 nodes, the exact algorithm (of which we used the sequential implementation) tended to perform very well (a few milliseconds), therefore we omit the analysis here. In order to test all the other algorithms, besides the two largest components, we randomly extracted a sample of (distinct) nodes out of the original graph, to produce different subgraphs with size $|N| \in \{23, 26, 30, 40\}$.

For the the considered cases we do not find significant enough differences between the randomized method and the exact one, see, e.g., figures 3 and 4. Notably, apart from a small number of cases, our bounds (especially the lower bounds) are always very close to the exact value. In particular, for $n = 26$ we were able to get immediately the Shapley value for all agents, since upper and lower bounds coincide for all of them. Moreover, it emerges that the randomized method is much better than the theoretical guarantee on the maximum approximation error. In the considered instances, the values obtained by using the randomized method are within the correct bounds.

We also evaluated how many computations of optimal allocations have been avoided by considering whether or not an agent has neighbours before her in the permutation at

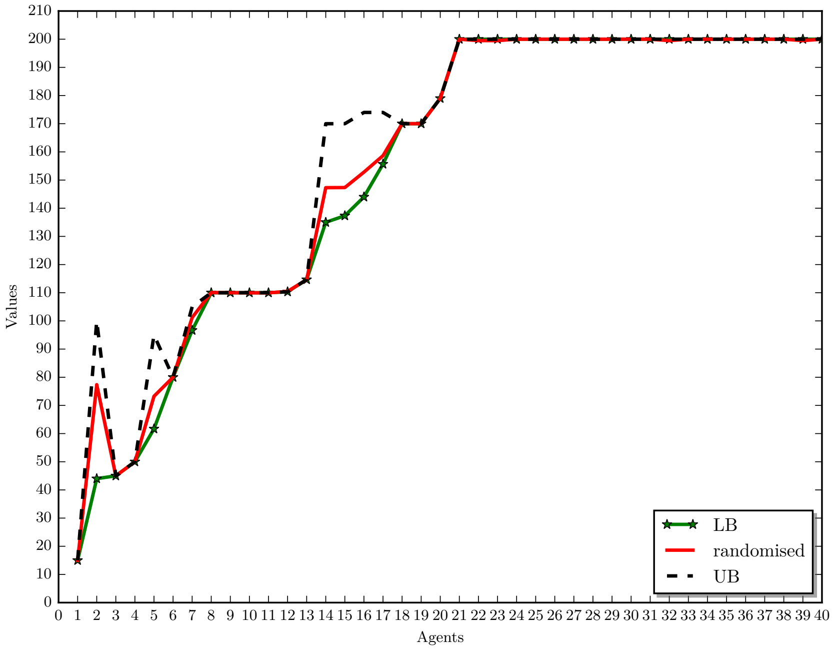

Fig. 4: Methods comparison ($n = 40$)

hands (and hence executing in the latter case Step 10 rather than Step 8 in Algorithm 1). By fixing the approximation error at $\epsilon = 0.3$, for each $n \in \{15, 23, 26, 30, 40\}$ we get the following savings: $9.65 \cdot 10^5$ out of $3.5 \cdot 10^6$ (i.e., 28%), $2.34 \cdot 10^6$ out of $1.29 \cdot 10^7$ (18%), $5.36 \cdot 10^6$ out of $1.87 \cdot 10^7$ (29%), $8.78 \cdot 10^6$ out of $2.9 \cdot 10^7$ (30%), and $1.46 \cdot 10^7$ out of $6.93 \cdot 10^7$ (21%), respectively. Finally, we also report the real maximum and average approximation errors (denoted with X and Y, respectively) of our implementation w.r.t. the exact algorithm for each $n \in \{15, 23, 26\}$. In particular, for $n = 15$ we get X = 0.01 and Y = $3 \cdot 10^{-3}$, for $n = 23$ we get X = $1.5 \cdot 10^{-3}$ and Y = $1.7 \cdot 10^{-4}$ and, for $n = 26$ we get X = $1.06 \cdot 10^{-4}$ and Y = $1.59 \cdot 10^{-5}$. In all the cases, the maximum approximation error was about 1% (or less) and therefore considerably below the theoretical guarantee (30%).

We stress that we were not able to fully analyse the component with 685 nodes, as it would simply have taken too long. However, we hope to be able to do that in a future work, possibly developing more powerful preprocessing techniques. For the time being, we report that the computation of our upper and lower bounds, when applied to this component, allowed us to approximate the Shapley value of 581 agents within a 30% error bound, of which 362 agents are within a 10% error bound, and 204 within 5%. Moreover, we were able to compute the exact Shapley value for 53 agents.

Running Times. Figures 5 and 6 report the computation time of the various algorithms In particular, Figure 5 focuses on the brute-force algorithm for computing the exact

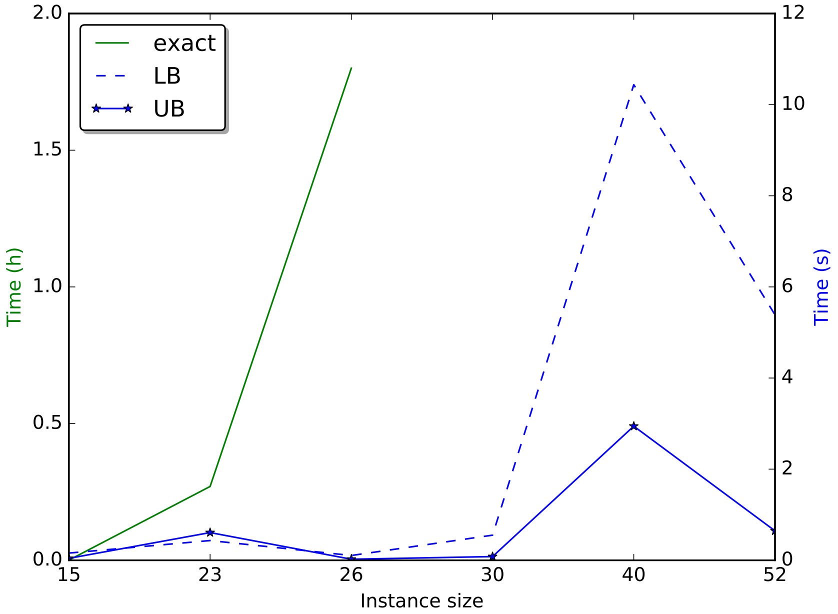

Fig. 5: Running times for the computation of the exact value by using the brute-force algorithm (green) and of the upper and lower bounds (blue) vs instances size

values, and on the algorithms for the computation of the upper and lower bounds. For the experiments, we computed separately the two bounds in order to point out that the computation of the lower bound requires in general more time, because it considers allocation over larger coalitions than those considered for the computation of the upper bound. Moreover, as discussed in Section 6, the running times for computing the bounds heavily depends on the cardinality of agents' neighbours. This explains why the running times for the case $n = 52$ are smaller than those for the case $n = 40$.

Figure 6 shows the running time of the randomized parallel algorithm using 24 threads, for different values of ϵ. In particular, for $\delta = 0.01$ we performed five trials over the different (sub)games described above, and averaged measures are reported. We can see that for games of reasonable size we can achieve high theoretical approximation error guarantee. For instance, for the biggest considered game ($n = 52$) we were able to compute the approximate Shapley value with $\epsilon = 0.1$ (and $\delta = 0.01$) in less than 90 minutes. There is a big gap between the performances of the randomized algorithm when using the extreme values for the approximation error that we considered. However, as already pointed out, even when we used a poor theoretical guarantee on the approximation error, we still got a quite reasonable accuracy.

Notably, for the case $n = 15$ the exact algorithm is faster than the sequential randomized algorithm, while it is definitely slower than the lower and upper bound algorithms.

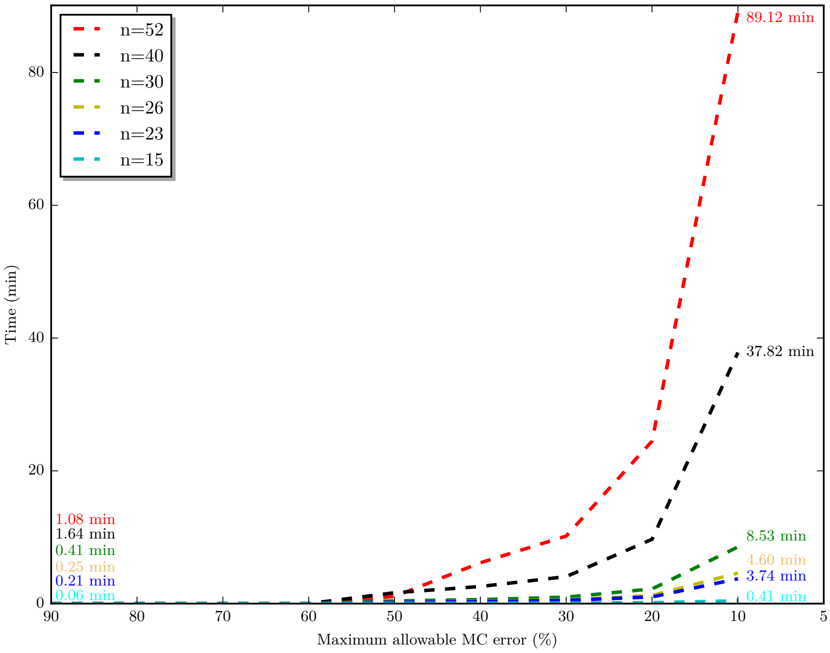

Fig. 6: Parallel implementation: running times vs ϵ

8 Future work

We are working on further theoretical properties of the Shapley value in allocation problems. The goal is to get some more tools for simplifying the VQR real-world large instances.

Moreover, we would like to extend the structure-based technique described in [1] to the more general class of games where more than one good can be allocated to each agent (as it is the case in VQR allocations). This way, we could compute efficiently the exact Shapley value for large games, provided that the treewidth of the agents graph is small. With this respect, we note that, by applying the simplifications described in this paper, all the connected components of the big VQR instance relative to Sapienza University of Rome have a small treewidth (at most 8) except for the largest one (comprising 685 agents), which has treewidth at most 64. For instance, the component having 52 agents has treewidth 5.

Finally, we would like to obtain tighter lower and upper bounds, with a computational effort that can be tuned to meet given time constraints.

References

1. G. Greco, F. Lupia, and F. Scarcello. Structural tractability of shapley and banzhaf values in allocation games. In *Proc. of IJCAI'15*, pages 547–553, 2015.
2. G. Greco and F. Scarcello. Fair division rules for funds distribution: The case of the italian research assessment program (vqr 2004-2010). *Intelligenza Artificiale*, 7(1):45–56, 2013.
3. G. Greco and F. Scarcello. Counting solutions to conjunctive queries: structural and hybrid tractability. In *Proc. of PODS'14*, pages 132–143, 2014.
4. G. Greco and F. Scarcello. Mechanisms for fair allocation problems: No-punishment payment rules in verifiable settings. *J. Artif. Intell. Res. (JAIR)*, 49:403–449, 2014.
5. H. Aziz and B. de Keijzer. Shapley meets shapley. In *Proc. of STACS'14*, pp. 99–111.
6. Y. Bachrach and Y.S. Rosenschein. Power in threshold network flow games, pp. 106–132, 2009.
7. X. Deng and C.H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19:257–266, May 1994.
8. D. Liben-Nowell, A. Sharp, T. Wexler, and K. Woods. Computing Shapley value in supermodular coalitional games. In *Proc. of COCOON'12*, pages 568–579. 2012.
9. F. Maniquet. A characterization of the Shapley value in queueing problems. *Journal of Economic Theory*, 109(1):90–103, 2003.
10. D. Mishra and B. Rangarajan. Cost sharing in a job scheduling problem. *Social Choice and Welfare*, 29(3):369–382, October 2007.
11. H. Moulin. An application of the Shapley value to fair division with money. *Econometrica*, 60(6):1331–49, November 1992.
12. H. Nagamochi and Dao-Zhi Zeng and Naohisa Kabutoya and Toshihide Ibaraki. Complexity of the Minimum Base Game on Matroids *Math. Oper. Res.*, 22(1):146–164, 1997.
13. A. Iera and L. Militano and L. Romeo and F. Scarcello. Fair Cost Allocation in Cellular-Bluetooth Cooperation Scenarios *EEE Transactions on Wireless Communications.*, 10(8):2566–2576, 2011.
14. R. Pichler and S. Skritek. Tractable counting of the answers to conjunctive queries. *Journal of Computer and System Sciences*, 79(6):984–1001, 2013.
15. N. Robertson and P. Seymour. Graph minors iii: Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, February 1984.
16. L. S. Shapley. A value for n-person games. *Contributions to the theory of games*, 2, 1953.