Available online at www.sciencedirect.com

SCIENCE DIRECT®

Artificial Intelligence 170 (2006) 607–619

Artificial Intelligence

www.elsevier.com/locate/artint

Complexity of constructing solutions in the core based on synergies among coalitions ☆

Vincent Conitzer *, Tuomas Sandholm

Carnegie Mellon University, Computer Science Department, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Received 5 December 2004; received in revised form 10 January 2006; accepted 16 January 2006

Available online 17 February 2006

Abstract

Coalition formation is a key problem in automated negotiation among self-interested agents, and other multiagent applications. A coalition of agents can sometimes accomplish things that the individual agents cannot, or can accomplish them more efficiently. Motivating the agents to abide by a solution requires careful analysis: only some of the solutions are stable in the sense that no group of agents is motivated to break off and form a new coalition. This constraint has been studied extensively in cooperative game theory: the set of solutions that satisfy it is known as the *core*. The computational questions around the core have received less attention. When it comes to coalition formation among software agents (that represent real-world parties), these questions become increasingly explicit.

In this paper we define a concise, natural, general representation for games in characteristic form that relies on superadditivity. In our representation, individual agents' values are given as well as values for those coalitions that introduce synergies. We show that this representation allows for efficient checking of whether a given outcome is in the core. We then show that determining whether the core is nonempty is NP-complete both with and without transferable utility. We demonstrate that what makes the problem hard in both cases is determining the collaborative possibilities (the set of outcomes possible for the grand coalition); we do so by showing that if these are given, the problem becomes solvable in time polynomial in the size of the representation in both cases. However, we then demonstrate that for a hybrid version of the problem, where utility transfer is possible only within the grand coalition, the problem remains NP-complete even when the collaborative possibilities are given. Finally, we show that for *convex* characteristic functions, a solution in the core can be computed efficiently (in $O(nl^2)$ time, where n is the number of agents and l is the number of synergies), even when the collaborative possibilities are not given in advance.

1. Introduction

Coalition formation is a key problem in automated negotiation among self-interested agents, and other multiagent applications. A coalition of agents can sometimes accomplish things that the individual agents cannot, or can accomplish them more efficiently. Motivating the agents to abide by a solution requires careful analysis: only some of the

☆ The material in this paper is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, ITR IIS-0121678, a Sloan Fellowship, and an IBM Ph.D. Fellowship. A short, early version of this paper appeared in the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03).

* Corresponding author.
E-mail addresses: conitzer@cs.cmu.edu (V. Conitzer), sandholm@cs.cmu.edu (T. Sandholm).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2006.01.005

solutions are stable in the sense that no group of agents is motivated to break off and form a new coalition. This constraint has been studied extensively in cooperative game theory: the set of solutions that satisfy it is known as the *core*. The computational questions around the core have received less attention. When it comes to coalition formation among software agents (that represent real-world parties), these questions become increasingly explicit.

While the driving motivation of this work is coalition formation among software agents, the computation of stable solutions may have applications beyond automated negotiation as well, for instance in electronic commerce. As an example, consider a large number of companies, some subsets of which could form profitable virtual organizations that can respond to larger or more diverse orders than the individual companies can. Determining stable value divisions allows one to see which potential virtual organizations would be viable in the sense that the companies in the virtual organization would naturally stay together. As another example, consider a future online service that determines how much each employee of a company should be paid so that the company does not collapse as a result of employees being bought away by other companies. The input to this service would be how much subsets of the company's employees would be paid if they left collectively (for instance, a whole department could be bought away). This input could come from salary databases or a manager's estimate. The computational problem of determining a stable renumeration would be crucial for such a service. Both of these example problems fit exactly under the model that we study in this paper.

The rest of the paper is organized as follows. In Section 2, we review the required concepts from cooperative game theory. In Section 3, we define a concise general representation for games in characteristic form that relies on superadditivity, and show that it allows for efficient (relative to the size of the representation) checking of whether a given outcome is in the core. In Section 4, we show that determining whether the core is nonempty is NP-complete both with and without transferable utility. In Section 5, we demonstrate that what makes the problem hard in both cases is determining the collaborative possibilities (the set of outcomes possible for the grand coalition); we do so by showing that if these are given, the problem becomes solvable in time polynomial in the size of the representation in both cases. In Section 6, we show that for a hybrid version of the problem, where utility transfer is possible only within the grand coalition, the problem remains NP-complete even when the collaborative possibilities are given. Finally, in Section 7, we show that if the characteristic function is convex, a solution in the core can be constructed efficiently (in $O(nl^2)$ time, where n is the number of agents and l is the number of synergies), even when the collaborative possibilities are not given in advance.

2. Definitions from cooperative game theory

In this section we review standard definitions from cooperative game theory, which we will use throughout the paper. In the definitions, we follow Mas-Colell et al. [21].

In general, how well agents in a coalition do may depend on what nonmembers of the coalition do (for example, see [2,6,10,22–24,28]). However, in cooperative game theory, coalition formation is usually studied in the context of *characteristic function games* where the utilities of the coalition members do not depend on the nonmembers' actions [5,16,31,35–37]. (One way to interpret this is to consider the coalition members' utilities to be the utilities they can *guarantee* themselves no matter what the nonmembers do [1].)

Definition 1. Given a set of agents A, a *utility possibility vector* u^B for $B = \{b_1, \ldots, b_{n_B}\} \subseteq A$ is a vector $(u_{b_1}, \ldots, u_{b_{n_B}})$ representing utilities that the agents in B can guarantee themselves by cooperating with each other. A *utility possibility set* is a set of utility possibility vectors for a given set B.

Definition 2. A *game in characteristic form* consists of a set of agents A and a utility possibility set $V(B)$ for each $B \subseteq A$.

Sometimes games in characteristic form have *transferable utility*, which means agents in a coalition can transfer utility among themselves. (Usually, this is done using monetary transfers.)

Definition 3. A game in characteristic form is said to have *transferable utility* if for every $B \subseteq A$ there is a number $v(B)$ (the *value* of B) such that $V(B) = \{u^B = (u^B_{b_1}, \ldots, u^B_{b_{n_B}})\colon \sum_{b \in B} u^B_b \leqslant v(B)\}$.

It is commonly assumed that the joining of two coalitions does not prevent them from acting as well as they could have acted separately. In other words, the composite coalition can coordinate by choosing not to coordinate. This assumption is known as *superadditivity*. When superadditivity holds, it is always best for the grand coalition of all agents to form.[1] We will assume superadditivity throughout the paper. This makes our hardness results *stronger* because even a restricted version of the problem is hard.

Definition 4. A game in characteristic form is said to be *superadditive* if, for any two disjoint subsets $B, C \subseteq A$, and for any $u^B \in V(B)$ and $u^C \in V(C)$, we have $(u^B, u^C) \in V(B \cup C)$. (In the case of transferable utility, this is equivalent to saying that for any $B, C \subseteq A$ with B and C disjoint, $v(B \cup C) \geqslant v(B) + v(C)$.)

We now need a solution concept. In this paper, we study only what is arguably the best known solution concept, namely the *core* (see, for example, [16,21,35]). It was first introduced by Gillies [13]. An outcome is said to be in the core if there is no subset of agents that can break off and form their own coalition in such a way that all of the agents in that coalition are better off. The following definition makes this precise.

Definition 5. An *outcome* $u^A = (u_1^A, \dots, u_n^A) \in V(A)$ is *blocked* by coalition $B \subseteq A$ if there exists $u^B = (u_{b_1}^B, \dots, u_{b_{n_B}}^B) \in V(B)$ such that for all $b \in B$, $u_b^B > u_b^A$. (In the case of transferable utility, this is equivalent to saying that the outcome is blocked by B if $v(B) > \sum_{b \in B} u_b^A$.) An outcome is in the *core* if it is blocked by no coalition.

In general, the core can be empty. If the core is empty, the game is inherently unstable because no matter what outcome is chosen, some subset of agents is motivated to pull out and form their own coalition. In other words, requiring that no subset of agents is motivated to break off into a coalition of its own overconstrains the system. An example of a game with an empty core is the one with agents $\{x, y, z\}$, where we have the utility possibility vectors $u^{\{x,y\}} = (2, 1)$, $u^{\{y,z\}} = (2, 1)$, and $u^{\{x,z\}} = (1, 2)$ (and the ones that can be derived from this through superadditivity). The same example with transferable utility also has an empty core.

In the rest of this paper, we will study the question of how complex it is to determine whether the core is nonempty, that is, whether there is a solution or the problem is overconstrained. First, as a basis for comparison, we mention the following straightforward, general algorithms for checking whether there exists a solution in the core. For a game without transferable utility, for every element $u^A \in V(A)$, check, for every coalition $B \subseteq A$, whether there exists an element $u^B \in V(B)$ such that u^B blocks u^A. For a game with transferable utility, we can use linear programming: we must decide on each agent's utility u_i in such a way that the total utility does not exceed $v(A)$, and such that for every coalition $B \subseteq A$, $v(B)$ does not exceed the total utility of the agents in B [25]. There are at least two reasons why these algorithms are not always practical. First, both algorithms are exponential in the number of agents, because they require a check or constraint for every subset of the agents.[2] Second, the first algorithm requires the computation of every utility possibility vector, and the second algorithm requires the computation of every value of a coalition. The computation of a single utility possibility vector or coalition value is not necessarily trivial, because it may require that the agents solve a complex collaborative planning problem.

Both of these two problems are intimately related to the *representation* of the characteristic form game. We cannot hope to compute solutions in the core in time that is less than exponential in the number of agents unless the representation is concise (that is, it does not simply list all the utility possibility vectors or all the coalition values). Moreover, we cannot address the question of how hard it is to compute individual utility possibility vectors or coalition values

[1] On the other hand, without superadditivity, even finding the optimal coalition structure (partition of agents into coalitions) can be hard [18,19, 29,32,33]. A third potential source of computational complexity (in addition to payoff/utility division as studied in this paper, and coalition structure generation) stems from each potential coalition having to solve a hard optimization problem in order to determine the value of the coalition. For example, when the agents are carrier companies with their own trucks and delivery tasks, they can save costs by forming a coalition (pooling their trucks and tasks), but each potential coalition faces a hard optimization problem: a vehicle routing problem defined by the coalition's trucks and tasks. The effect of such hard optimization problems on coalition formation has been studied by Sandholm and Lesser [30] and Tohmé and Sandholm [34].

[2] In the case of the first algorithm, there may also be exponentially or even infinitely many utility possibility vectors per coalition, making the algorithm even less satisfactory.

unless we know the representation from which they must be derived.[3] Because of this, in the next section, we introduce a concise representation of characteristic form games that is based on the synergies among coalitions.

3. Representing characteristic form games concisely

In our representation of games in characteristic form, we distinguish between games without transferable utility, where we specify some utility possibility vectors for some coalitions, and games with transferable utility, where we specify the values of some coalitions.

If the representation of the game specifies $V(B)$ or $v(B)$ explicitly for each coalition $B \subseteq A$, then the length of the representation is exponential in the number of agents. In that case, any algorithm for evaluating nonemptiness of the core (as long as it reads all the input) requires time exponential in the number of agents. However, that runtime is polynomial in the size of the input (this can be accomplished, for example, using the algorithms that we introduce in Section 5).

Most characteristic form games that represent real-world settings have some special structure. This usually allows for a game representation that is significantly more concise. The complexity of characterizing the core has already been studied in certain specific concisely expressible families of games. For example, Deng and Papadimitriou [9] study a game in which the agents are vertices of a graph with weights on the edges, and the value of a coalition is determined by the total weight of the edges contained in the subgraph induced by that coalition. Faigle et al. [12] study a game in which the agents are vertices on a graph, and the cost of a coalition is the weight of a *minimum spanning tree* for the subgraph induced by that coalition. They show that determining whether a solution is in the core is NP-complete in this game. In other work [11], the same authors study a modification of this game where the cost of a coalition is the minimum length of a traveling salesman tour for the subgraph induced by that coalition, and show how to use linear programming to give a solution that is *approximately* in the core (in the sense that every coalition pays at most $4/3$ times their own cost). Markakis and Saberi [20] study how to construct a solution in the core for a *multicommodity flow game* on graphs (defined by Papadimitriou [27]). In this game, the agents correspond to the vertices of the graph, and each agent's utility is the total flow originated or terminated at that agent's vertex. A coalition can obtain any utility vector corresponding to a feasible flow for the subgraph induced by that coalition. Markakis and Saberi show that the core is always nonempty in such games, and they show that for the case of transferable utility, a solution in the core can be constructed in polynomial time in such games. Goemans and Skutella [14] study how to construct a solution in the core for a *facility location game*. In this game, several facilities must be opened (at a cost), and the agents must be connected to them (at a cost). Here, a solution is in the core if no subset of agents has an incentive to deviate and build the facilities and connections on their own. Goemans and Skutella show that there is a solution in the core if and only if there is no integrality gap for a particular linear programming relaxation (which is NP-complete to verify). Pal and Tardos [26] and Gupta et al. [15] study a game in which the agents need to be connected to a common source in a graph, and give a polynomial-time algorithm for constructing a solution that is approximately in the core. Finally, Deng et al. [8] study an integer programming formulation which captures many games on graphs, and (as in the result by Goemans and Skutella [14]) show that the core is nonempty if and only if the linear programming relaxation has an integer optimal solution. All of those results depend on concise game representations which are specific to the game families under study. Typically, such a family of games is played on a combinatorial structure such as a graph. Cooperative games on combinatorial structures have been systematically studied [3].

As a point of deviation, we study a natural representation that can capture *any* characteristic form game.[4] Conciseness in our representation stems only from the fact that in many settings, the synergies among coalitions are sparse. When a coalition introduces no new synergy, its utility possibility vectors can be *derived* using superadditivity. Therefore, the input needs to include only the utility possibility vectors of coalitions that introduce synergy. The definitions below make this precise. We first present our representation of games without transferable utility.

[3] It should be remarked that deriving utility possibility vectors or coalition values from a given representation may be only part of the problem: it is possible that this representation itself must be constructed from another, even more basic representation, and this construction may itself be computationally nontrivial.

[4] Our hardness results are not implied by the earlier hardness results for specific game families, because the games must be concisely representable in our input language to prove our hardness results.

Definition 6. We represent a game in characteristic form without transferable utility by a set of agents A, and a set of utility possibility vectors $W = \{(B, u^{B,k})\}$. (Here there may be multiple vectors for the same B, distinguished by different k indices.) The utility possibility set for a given $B \subseteq A$ is then given by $V(B) = \{u^B\colon\ u^B = (u^{B_1}, \ldots, u^{B_r}), \bigcup_{1 \leqslant i \leqslant r} B_i = B$, all the B_i are disjoint, and for all the B_i, $(B_i, u^{B_i}) \in W\}$. To avoid senseless cases that have no outcomes, we also require that $(\{a\}, (0)) \in W$ for all $a \in A$.[5]

Listing all the utility possibility vectors is of course not always feasible, and when it is not feasible, another representation should be used. For example, in a game with transferable utility, there are infinitely many (actually, a continuum of) utility possibility vectors. Therefore, we give a more appropriate representation of games with transferable utility below. Nevertheless, there do exist settings in which all the utility possibility vectors can be listed explicitly. For example, we may have historical data on which coalitions have formed in the past and what the agents' utilities were in those coalitions. At the minimum, we should ensure that no subcoalition will collectively yearn for times past (and perhaps seek to recreate them by breaking off from the present-day coalition). In this case, we should try to find a solution that is in the core of the game given by the historical characteristic function.

We now present our representation of games with transferable utility.

Definition 7. We represent a game in characteristic form with transferable utility by a set of agents A, and a set of values $W = \{(B, v(B))\}$. The value for a given $B \subseteq A$ is then given by $v(B) = \max\{\sum_{1 \leqslant i \leqslant r} v(B_i)\colon \bigcup_{1 \leqslant i \leqslant r} B_i = B$, all the B_i are disjoint, and for all the B_i, $(B_i, v(B_i)) \in W\}$. To avoid senseless cases that have no outcomes, we also require that $(\{a\}, 0) \in W$ whenever $\{a\}$ does not receive a value elsewhere in W.

Thus, we only need to specify a basis of utility possibilities, from which we can then derive the others. (This derivation can, however, be computationally hard, as we will discuss shortly.) This representation integrates rather nicely with real-world problems where determining any coalition's value is complex. For example, in the multiagent vehicle routing problem, we solve the routing problem for every coalition that might introduce new synergies. When it is clear that there is no synergy between two coalitions (for example, if they operate in different cities and each one only has deliveries within its city), there is no need to solve the routing problem of the coalition that would result if the two coalitions were to merge.

The following lemmas indicate that we can also use this representation effectively for checking whether an outcome is in the core, that is, whether it satisfies the strategic constraints.

Lemma 1. *Without transferable utility, an outcome $u^A = (u_1^A, \ldots, u_n^A) \in V(A)$ is blocked by some coalition if and only if it is blocked by some coalition B through some utility vector u^B, where $(B, u^B) \in W$.*

Proof. The "if" part is trivial. For "only if", suppose u^A is blocked by coalition C through some u^C, so that for every $c \in C$, $u_c^C > u_c^A$. We know $u^C = (u^{C_1}, \ldots, u^{C_r})$ where $(C_i, u^{C_i}) \in W$. But then, C_1 blocks u^A through u^{C_1}. □

The proof for the same lemma in the case of transferable utility is only slightly more intricate.

Lemma 2. *With transferable utility, an outcome $u^A = (u_1^A, \ldots, u_n^A) \in V(A)$ is blocked by some coalition if and only if it is blocked by some coalition B through its value $v(B)$, where $(B, v(B)) \in W$.*

Proof. The "if" part is trivial. For "only if", suppose u^A is blocked by coalition C through $v(C)$, so that $v(C) > \sum_{c \in C} u_c^A$. We know that $v(C) = \sum_{1 \leqslant i \leqslant r} v(C_i)$ where $(C_i, v(C_i)) \in W$. It follows that $\sum_{1 \leqslant i \leqslant r} v(C_i) > \sum_{1 \leqslant i \leqslant r} \sum_{c \in C_i} u_c^A$, and hence for at least one C_i, we have $v(C_i) > \sum_{c \in C_i} u_c^A$. But then C_i blocks u^A through $v(C_i)$. □

One drawback is that under this representation, it is, in general, NP-complete to compute a given coalition's value. Thus, the representation still leaves part of the collaborative optimization problem of determining a coalition's value to

[5] Setting the utility to 0 in this case is without loss of generality, as we can simply normalize the utility function to obtain this.

be solved (which is not entirely unnatural given that in many practical settings, the collaborative optimization problem is very complex). Rather than prove this NP-hardness here directly, we note that it follows from results in the coming sections, as we will show.

4. Checking whether the core is nonempty is hard

We now show that with this representation, it is hard to check whether the core is nonempty. This holds both for the nontransferable utility setting and for the transferable utility setting.

Definition 8 *(CORE-NONEMPTY).* We are given a superadditive game in characteristic form (with or without transferable utility) in our representation language. We are asked whether the core is nonempty.

We will demonstrate NP-hardness of this problem by reducing from the NP-complete EXACT-COVER-BY-3-SETS problem [17].

Definition 9 *(EXACT-COVER-BY-3-SETS).* We are given a set S of size $3m$ and a collection of subsets $\{S_i\}_{1\leqslant i\leqslant r}$ of S, each of size 3. We are asked if there is a cover of S consisting of m of the subsets.

The intuition behind both NP-hardness reductions from EXACT-COVER-BY-3-SETS presented in this section is the following. We let the elements of S correspond to some of the agents, and we let the subsets S_i correspond to some of the elements of W, in such a way that the grand coalition can achieve a "good" outcome if and only if we can cover all these agents with a subset of disjoint elements of W—that is, if and only if an exact cover exists. Then, we add some additional agents and elements of W such that only such a good outcome would be strategically stable (this is the nontrivial part of both proofs, which is why these results do not follow trivially from the hardness of solving a coalition's collaborative optimization problem). The formal proofs follow.

Theorem 1. *CORE-NONEMPTY without transferable utility is NP-complete.*

Proof. To show that the problem is in NP, nondeterministically choose a subset of W, and check if the corresponding coalitions constitute a partition of A. If so, check if the outcome corresponding to this partition is blocked by any element of W.

To show NP-hardness, we reduce an arbitrary EXACT-COVER-BY-3-SETS instance to the following CORE-NONEMPTY instance. Let the set of agents be $A = S \cup \{w, x, y, z\}$. For each S_i, let (S_i, u^{S_i}) be an element of W, with $u^{S_i} = (2, 2, 2)$. Also, for each $s \in S$, let $(\{s, w\}, u^{\{s,w\}})$ be an element of W, with $u^{\{s,w\}} = (1, 4)$. Also, let $(\{w, x, y, z\}, u^{\{w,x,y,z\}})$ be an element of W, with $u^{\{w,x,y,z\}} = (3, 3, 3, 3)$. Finally, let $(\{x, y\}, u^{\{x,y\}})$ with $u^{\{x,y\}} = (2, 1)$, $(\{y, z\}, u^{\{y,z\}})$ with $u^{\{y,z\}} = (2, 1)$, $(\{x, z\}, u^{\{x,z\}})$ with $u^{\{x,z\}} = (1, 2)$ be elements of W. The only other elements of W are the required ones giving utility 0 to singleton coalitions. We claim the two instances are equivalent.

First suppose there is an exact cover by 3-sets consisting of $S_{c_1}, \ldots, S_{c_m}$. Then the following outcome is possible: $(u^{S_{c_1}}, \ldots, u^{S_{c_m}}, u^{\{w,x,y,z\}}) = (2, 2, \ldots, 2, 3, 3, 3, 3)$. It is easy to verify that this outcome is not blocked by any coalition. It follows that the core is nonempty.

Now suppose there is no exact cover by 3-sets. Suppose the core is nonempty, that is, it contains some outcome $u^A = (u^{C_1}, \ldots, u^{C_r})$ with each (C_i, u^{C_i}) an element of W, and the C_i disjoint. Then one of the C_i must be $\{s, w\}$ for some $s \in S$: for if this were not the case, there must be some $s \in S$ with $u^A_s = 0$, because the C_i that are equal to S_i cannot cover S; but then $\{s, w\}$ would block the outcome. Thus, none of the C_i can be equal to $\{w, x, y, z\}$. Then one of the C_i must be one of $\{x, y\}$, $\{y, z\}$, $\{x, z\}$, or else two of $\{x, y, z\}$ would block the outcome. By symmetry, we can without loss of generality assume it is $\{x, y\}$. But then $\{y, z\}$ will block the outcome. (Contradiction.) It follows that the core is empty. □

We might hope that the convexity introduced by transferable utility makes the problem tractable through, for example, linear programming. This turns out not to be the case.

Theorem 2. *CORE-NONEMPTY with transferable utility is NP-complete.*

Proof. To show that the problem is in NP, nondeterministically choose a subset of W, and check if the corresponding coalitions constitute a partition of A. If so, nondeterministically divide the sum of the coalitions' values over the agents, and check if this outcome is blocked by any element of W.

To show NP-hardness, we reduce an arbitrary EXACT-COVER-BY-3-SETS instance to the following CORE-NONEMPTY instance. Let the set of agents be $A = S \cup \{x, y\}$. For each S_i, let $(S_i, 3)$ be an element of W. Additionally, let $(S \cup \{x\}, 6m)$, $(S \cup \{y\}, 6m)$, and $(\{x, y\}, 6m)$ be elements of W. The only other elements of W are the required ones giving value 0 to singleton coalitions. We claim the two instances are equivalent.

First suppose there is an exact cover by 3-sets consisting of $S_{c_1}, \ldots, S_{c_m}$. Then the value of coalition S is at least $\sum_{1\leqslant i\leqslant m} v(S_{c_i}) = 3m$. Combining this with the coalition $\{x, y\}$, which has value $6m$, we conclude that the grand coalition A has value at least $9m$. Hence, the outcome $(1, 1, \ldots, 1, 3m, 3m)$ is possible. It is easy to verify that this outcome is not blocked by any coalition. It follows that the core is nonempty.

Now suppose there is no exact cover by 3-sets. Then the coalition S has value less than $3m$ (since there are no m disjoint S_i), and as a result the value of the grand coalition is less than $9m$. It follows that in any outcome, the total utility of at least one of $S \cup \{x\}$, $S \cup \{y\}$, and $\{x, y\}$ is less than $6m$. Hence, this coalition will block. It follows that the core is empty. □

Our results imply that in the general case, it is computationally hard to make any strategic assessment of a game in characteristic form when it is concisely represented.

5. Specifying information about the grand coalition makes the problem tractable

Our proofs that CORE-NONEMPTY is hard relied on constructing instances where it is difficult to determine what the grand coalition can accomplish. In effect, the hardness derived from the fact that even collaborative optimization is hard in these instances. While this is indeed a real difficulty that occurs in the analysis of characteristic form games, we may nevertheless wonder to what extent computational complexity issues are introduced by the purely strategic aspect of the games. To analyze this, we investigate the computational complexity of CORE-NONEMPTY when $V(A)$ (or $v(A)$) is *explicitly* provided as input, so that determining what the grand coalition can accomplish can no longer be the source of any complexity.[6] It indeed turns out that the problem becomes easy both with and without transferable utility.

Theorem 3. *When $V(A)$ is explicitly provided, CORE-NONEMPTY without transferable utility is in P.*

Proof. The following simple algorithm accomplishes this efficiently. For each element of $V(A)$, check whether it is blocked by any element of W. The algorithm is correct due to Lemma 1. □

For the transferable utility case, we make use of linear programming.

Theorem 4. *When $v(A)$ is explicitly provided, CORE-NONEMPTY with transferable utility is in P.*

Proof. We decide how to allocate the $v(A)$ among the agents by solving a linear program. Because of Lemma 2, the core is nonempty if and only if the following linear program has a solution:

$$\sum_{1\leqslant i\leqslant n} u_i \leqslant v(A); \quad \forall \big(B, v(B)\big) \in W, \quad \sum_{b\in B} u_b \geqslant v(B). \tag{1}$$

Because the size of the program is polynomial, this constitutes a polynomial-time algorithm. □

[6] Bilbao et al. [4] have studied the complexity of the core in characteristic form games with transferable utility when there is an oracle that can provide the value $v(B)$ of any coalition B. Our amended input corresponds to asking one such query in addition to obtaining the unamended input.

The algorithms in the proofs also construct a solution that is in the core, if the core is nonempty. We note that we have now also established that computing $v(A)$ is NP-hard, because, apart from the computation of $v(A)$, we have an efficient algorithm for computing a solution in the core—which we have already shown is NP-hard.

6. Games with limited utility transfer remain hard

Not all complexity issues disappear through having the collaborative optimization problem solution available. It turns out that if we allow for games *with limited utility transfer*, where only *some* coalitions can transfer utility among themselves, the hardness returns. In particular, we show hardness in the case where only the grand coalition can transfer utility. This is a natural model for example in settings where there is a market institution that enforces payments, but if a subset of the agents breaks off, the institution collapses and payments cannot be enforced. Another example application is the setting described in Section 3 in which we wish to ensure that there is no subcoalition such that all of its agents were happier in a past time period when they formed a coalition by themselves.

Definition 10. We represent a game in characteristic form in which only the grand coalition can transfer utility by a set of agents A, a set of utility possibility vectors $W = \{(B, u^{B,k})\}$, and a value $v(A)$ for the grand coalition. The utility possibility set for a given $B \subseteq A$, $B \neq A$ is then given by $V(B) = \{u^B\colon u^B = (u^{B_1}, \ldots, u^{B_r}), \bigcup_{1 \leqslant i \leqslant r} B_i = B$, all the B_i are disjoint, and for all the B_i, $(B_i, u^{B_i}) \in W\}$. To avoid senseless cases that have no outcomes, we also require that $(\{a\}, (0)) \in W$ for all $a \in A$. For the grand coalition A, any vector of utilities that sum to at most $v(A)$ is possible.

We demonstrate that the CORE-NONEMPTY problem is NP-hard in this setting by reducing from the NP-complete VERTEX-COVER problem [17].

Definition 11 *(VERTEX-COVER).* We are given a graph $G = (V, E)$, and a number k. We are asked whether there is a subset of V of size k such that each edge has at least one of its endpoints in the subset.

We are now ready to state our result.

Theorem 5. *When only the grand coalition can transfer utility, CORE-NONEMPTY is NP-complete* (*even when* $v(A)$ *is explicitly provided as input*).

Proof. To show that the problem is in NP, nondeterministically divide $v(A)$ over the agents, and check if this outcome is blocked by any element of W.

To show NP-hardness, we reduce an arbitrary VERTEX-COVER instance to the following CORE-NONEMPTY instance. Let $A = V \cup \{x, y, z\}$, and let $v(A) = 6|V| + k$. Furthermore, for each edge (v_i, v_j), let $(\{v_i, v_j\}, u^{\{v_i,v_j\}})$ be an element of W, with $u^{\{v_i,v_j\}} = (1, 1)$. Finally, for any $a, b \in \{x, y, z\}$ $(a \neq b)$, let $(\{a, b\}, u^{\{a,b\}})$ be an element of W, with $u^{\{a,b\}} = (3|V|, 2|V|)$. The only other elements of W are the required ones giving utility 0 to singleton coalitions. This game does not violate the superadditivity assumption, since without the explicit specification of $v(A)$, superadditivity can at most imply that $v(A) = 6|V| \leqslant 6|V| + k$. We claim the two instances are equivalent.

First suppose there is a vertex cover of size k. Consider the following outcome: all the vertices in the vertex cover receive utility 1, all the other vertices receive utility 0, and each of x, y, and z receives utility $2|V|$. Using the fact that all the edges are covered, it is easy to verify that this outcome is not blocked by any coalition. It follows that the core is nonempty.

Now suppose there is some outcome u^A in the core. In such an outcome, either each of $\{x, y, z\}$ receives at least $2|V|$, or two of them receive at least $3|V|$ each. (For if not, there is some $a \in \{x, y, z\}$ with $u^A_a < 2|V|$ and some $b \in \{x, y, z\}$ $(b \neq a)$ with $u^A_b < 3|V|$, and the coalition $\{b, a\}$ will block through $u^{\{b,a\}} = (3|V|, 2|V|)$.) It follows that the combined utility of all the elements of V is at most k. Now, for each edge (v_i, v_j), at least one of its vertices must receive utility at least 1, or this edge would block. Thus, the vertices that receive at least 1 cover the edges. But because the combined utility of all the elements of V is at most k, there can be at most k such vertices. It follows that there is a vertex cover. □

Games in which only some coalitions can transfer utility are quite likely to appear in real-world multiagent settings, for example because only some of the agents use a currency. Our result shows that for such games, even when the collaborative optimization problem has already been solved, it can be computationally hard to strategically assess the game.

7. A fast algorithm for convex characteristic functions

While we have shown that constructing a solution in the core is hard in general, we may wonder whether restrictions on the characteristic function make the problem easier. In this section, we show that if the characteristic function is *convex* (in which case it is known that the core is always nonempty), a solution in the core can be constructed efficiently (in $O(nl^2)$ time, where n is the number of agents and l is the number of synergies) under our representation. We do not require that the value of the grand coalition is known, nor do we need to use techniques from linear programming.

We will restrict our attention to games with transferable utility in this section. A characteristic function v is *convex* if for any coalitions B and C, we have $v(B \cup C) - v(C) \geqslant v(B) - v(B \cap C)$ [25]. That is, the marginal contribution of a subcoalition to a larger set is always larger (or at least as large). One natural setting in which the characteristic function is convex is the following. Suppose that every agent i has a set of *skills* S_i, and suppose that every agent's skills are unique, that is, $S_i \cap S_j = \{\}$ for all $i \neq j$. Furthermore, suppose that there is a set of *tasks* T that the agents can accomplish, and that each task $t \in T$ has a value $v(t) \geqslant 0$, as well as a set of skills $S(t)$ that are required to accomplish t. Then, the value of a coalition B is the sum of the values of the tasks that they can accomplish, that is, $v(B) = \sum_{t \in T:\ S(t) \subseteq \bigcup_{i \in B} S_i} v(t)$. In this setting, the marginal set of tasks that can be accomplished by adding a given subset of the agents to an existing coalition of agents is increasing in the existing coalition (as the existing coalition gets larger, the marginal set of tasks that can be accomplished by adding the given subset of agents can get no smaller), and therefore the characteristic function is convex. One example application is the following: suppose each agent holds some medical patents (these are the agents' "skills"), and there is a set of drugs (the "tasks"), each of which has a market value and requires a subset of the patents to be manufactured.

When the characteristic function is convex, one method of constructing a solution in the core is the following: impose an arbitrary order on the agents, add them to the coalition one at a time, and give each agent its marginal value (see, for example, Osborne and Rubinstein [25]). Unfortunately, this scheme cannot be applied directly under our representation, as it requires knowing the values of the relevant coalitions. As we have seen earlier, computing these values is hard for general characteristic functions. However, in this section we will show that when convexity holds, this is no longer the case, and solutions in the core can be constructed efficiently.

As it turns out, it is easier to think about the implications of convexity for our representation if we first assume that there are no explicitly specified pairs $(B, v(B))$ that are redundant. That is, if $v(B)$ could have been inferred from the values of other coalitions by superadditivity, we assume that it is not explicitly specified. Thus, only the *synergetic* coalitions' values are explicitly specified, where a coalition is synergetic if its value could not have been derived from the values of its proper subcoalitions. We will later show how to remove this assumption. Formally:

Definition 12. A coalition B is *synergetic* if $v(B) > \max\{\sum_{1 \leqslant i \leqslant r} v(B_i)$: $\bigcup_{1 \leqslant i \leqslant r} B_i = B$, each B_i is a proper subset of B, all the B_i are disjoint, and for all the B_i, $(B_i, v(B_i)) \in W\}$.

First we show that the union of two intersecting synergetic coalitions must be synergetic when convexity holds.

Lemma 3. *Let v be a convex characteristic function. Suppose B and C are both synergetic coalitions (for v), with a nonempty intersection ($B \cap C \neq \{\}$). Then $B \cup C$ is a synergetic coalition.*

Proof. If one of B and C is contained in the other, the lemma is trivially true. Hence, we can assume that this is not the case, that is, $B - C \neq \{\}$ and $C - B \neq \{\}$.

First suppose that $v(B \cup C) = v(B) + v(C - B)$. Then, $v(B \cup C) - v(C) = v(B) + v(C - B) - v(C) < v(B) + v(C - B) - v(C - B) - v(B \cap C) = v(B) - v(B \cap C)$ (where the inequality follows from C being a synergetic coalition), contradicting convexity. Hence, $v(B \cup C) > v(B) + v(C - B)$. By symmetry, $v(B \cup C) > v(C) + v(B - C)$.

We will now prove the lemma by induction on $|B \cup C|$. For $|B \cup C| \leqslant 2$ the lemma is vacuously true. Now suppose that it is always true for $|B \cup C| < k$, but it fails for $|B \cup C| = k$. That is, we have some B and C (both synergetic, with a nonempty intersection and neither contained in the other), where $B \cup C$ is not synergetic. Without loss of generality, we assume that B is maximal, that is, there is no strict superset of B in $B \cup C$ that is also synergetic. Let $D_1, D_2, \ldots, D_n$ be a partition of $B \cup C$ (with $n > 1$) such that each D_i is synergetic and $\sum_{i=1}^{n} v(D_i) = v(B \cup C)$ (such a partition must exist because we are supposing $B \cup C$ is not synergetic). If one of the D_i were equal to either B or C, then we would have $v(B \cup C) = v(B) + v(C - B)$ or $v(B \cup C) = v(C) + v(B - C)$, which we showed impossible at the beginning of the proof. Moreover, it is not possible that all the D_i are either contained in B, or have an empty intersection with B. For if this were the case, then $v(B \cup C) = \sum_{i=1}^{n} v(D_i) = \sum_{D_i \subseteq B} v(D_i) + \sum_{D_i \subseteq C-B} v(D_i) < v(B) + \sum_{D_i \subseteq C-B} v(D_i) \leqslant v(B \cup C)$ (where the first inequality derives from the facts that B is synergetic and no D_i is equal to B). Thus, there is some D_i^* with $D_i^* \cap B \neq \{\}$ and $D_i^* \cap (C - B) \neq \{\}$. Suppose that this D_i^* does not contain all of $C - B$. Then, we can apply the induction assumption to B and D_i^* to conclude that $B \cup D_i^*$ is synergetic. But this contradicts the maximality of B. On the other hand, suppose that D_i^* does contain all of $C - B$. Then $B \cup D_i^* = B \cup C$, and $v(B \cup D_i^*) = v(D_i^*) + v(B - D_i^*)$, which we showed impossible at the beginning of the proof. □

Next, we show that any partition of B into maximal synergetic coalitions will give us the value of B when convexity holds.

Lemma 4. *If v is convex, for any coalition B, for any partition $D_1, D_2, \ldots, D_n$ of B where each D_i is synergetic and maximal in B (there is no superset of D_i in B which is also synergetic), $v(B) = \sum_{i=1}^{n} v(D_i)$.*

Proof. Suppose not. Let $E_1, E_2, \ldots, E_m$ be a partition of B where each E_j is synergetic and $\sum_{j=1}^{m} v(E_j) = v(B)$. No E_j can be a strict superset of a D_i, because the D_i are maximal. Also, it cannot be the case that every E_j is a subset of some D_i, because then we would have $\sum_{j=1}^{m} v(E_j) < \sum_{i=1}^{n} v(D_i) < v(B) = \sum_{j=1}^{m} v(E_j)$. It follows that there is some E_j^* that is not contained in any D_i. Some D_i^* must intersect with E_j^*. But then by Lemma 3, $D_i^* \cup E_j^*$ must be synergetic, contradicting that D_i^* is maximal. □

We now show how to quickly compute the value of any coalition when convexity holds, presuming that there are no redundant specifications of coalition values. (That is, all the coalitions with explicitly specified values are synergetic. We will show how to remove this assumption shortly.)

Lemma 5. *Suppose v is convex. For any coalition B, if all its subcoalitions $D \subseteq B$ for which $v(D)$ is explicitly specified are synergetic (that is, there are no redundant specifications), the explicitly specified $(D, v(D))$ pairs are sorted by $|D|$, and each subset D is represented as a sorted list of agents, then $v(B)$ can be computed in $O(nl)$ time by computing a partition of maximal synergetic subcoalitions. (Here, n is the number of agents and l is the number of explicitly specified $(D, v(D))$ pairs.)*

Proof. Because all synergetic subcoalitions must be explicitly specified, we know that the set of subcoalitions D for which $(D, v(D))$ is explicitly specified is exactly the set of synergetic subcoalitions. Thus, by Lemma 4, we merely need to find a partition of B consisting of maximal subcoalitions whose value is explicitly specified. To do so, we will add such maximal subcoalitions to our partition one at a time. Because the explicitly specified $(D, v(D))$ pairs are sorted by $|D|$, we can read through them starting with those with the highest $|D|$ and work our way downwards to the ones with smallest $|D|$. When we arrive at any given D, we add it to the partition if and only if no members of D were previously added to the partition (as a member of another subcoalition). (We can perform each such check in $O(n)$ time, by keeping track of the set of agents that have already been added to the partition as a sorted list. To see if this set intersects with D, which is also stored as a sorted list, we go through the two sorted lists in parallel (as in, for instance, the well-known MergeSort algorithm), which requires $O(n)$ time. Updating the list of all agents that have already been added to a coalition when a given set D is added to the partition can be done in $O(n)$ time as well, again going through the two lists in parallel, and merging them.) Every synergetic D that we add to the partition must be maximal: for, if D is not maximal, when we arrive at D, the maximal synergetic D' containing it must already have been added, for the following reason. By Lemma 3 a maximal synergetic D' cannot intersect with another synergetic subcoalition of equal or larger size. Hence none of the members of D' were already in some element of the partition when we were considering adding D', and thus we must have added D'. Moreover, by the same reasoning, any member of B

must eventually be in some element of the partition (because the maximal synergetic subcoalition containing it must be added). Thus we will get a valid partition. □

Of course, for generality, we would like to allow for redundant specifications. Next, we show that we can indeed allow for them, because we can quickly remove any redundant specifications when convexity holds.

Lemma 6. *If v is convex and each subset $B \subseteq A$ is represented as a sorted list of agents, then the set of explicitly specified $(B, v(B))$ pairs can be reduced to the subset of only the synergetic ones in $O(nl^2)$ time. (Here, n is the number of agents and l is the number of explicitly specified $(B, v(B))$ pairs.)*

Proof. After sorting the $(B, v(B))$ pairs by $|B|$ (which takes $O(l \log(l))$ time), we will go through them starting at those with the smallest $|B|$ and working our way up to the larger ones, eliminating each nonsynergetic one, as follows. We observe that when we arrive at the pair $(B, v(B))$, all the remaining explicitly specified $(D, v(D))$ with $D \subseteq B$ will be synergetic (apart from possibly B itself). Thus, we can temporarily remove $(B, v(B))$, then according to Lemma 5 compute the new value of B (say, $v'(B)$) in $O(nl)$ time, and remove the pair $(B, v(B))$ if and only if $v'(B) = v(B)$. (We observe that, if B is indeed synergetic, removing the pair $(B, v(B))$ may break the convexity, in which case the algorithm from Lemma 5 may select the wrong maximal subcoalitions and compute too small a value for $v'(B)$. Of course, in this case the computed $v'(B)$ is still smaller than $v(B)$ and we will not remove the pair $(B, v(B))$, as required.) □

Finally, we show how to quickly compute a solution in the core when convexity holds. To distribute the value, we make use of the following known lemma mentioned at the beginning of this section.

Lemma 7. (Osborne and Rubinstein [25].) *For any order imposed on the agents, if we add the agents to the coalition one at a time, and give each agent its marginal value, then, if the characteristic function is convex, we obtain a solution in the core.*

This leads to the following result.

Theorem 6. *If v is convex, a solution in the core can be constructed in $O(nl^2)$ time. (Here, n is the number of agents and l is the number of explicitly specified $(B, v(B))$ pairs.)*

Proof. If the subsets $B \subseteq A$ are not yet represented as sorted lists, we can correct this in $O(nl \log(n))$ time. Then, we remove all the redundant $(B, v(B))$ pairs using Lemma 6, and sort the remaining ones by $|B|$, in $O(nl^2)$ time. Then (using the scheme from Lemma 7) we order the members of the grand coalition in an arbitrary way, add them in one by one, and give each of them their marginal value. This requires the computation of the value of one additional coalition for each agent (namely, the coalition that results from adding that agent), which takes $O(nl)$ time per agent using Lemma 5, for a total of $O(n^2 l)$ time. Because every agent appears in at least one synergetic subcoalition, and hence $n \leqslant l$, the whole algorithm runs in $O(nl^2)$ time. □

Another polynomial-time approach for constructing a solution in the core that does not rely on Lemma 7 is to compute the value of the grand coalition (using Lemmas 6 and 5), and then solve the linear programming problem as in Section 5.

8. Conclusions and future research

As we have argued, coalition formation is a key problem in automated negotiation among self-interested agents, and other multiagent applications. A coalition of agents can sometimes accomplish things that the individual agents cannot, or can accomplish them more efficiently. Motivating the agents to abide by a solution requires careful analysis: only some of the solutions are stable in the sense that no group of agents is motivated to break off and form a new coalition. This constraint has been studied extensively in cooperative game theory, but the computational questions around this constraint have received less attention. When it comes to coalition formation among software agents (that represent real-world parties), these questions become increasingly explicit.

In this paper we defined a concise general representation for games in characteristic form that relies on superadditivity, and showed that it allows for efficient checking of whether a given outcome is in the core. We then showed that determining whether the core is nonempty is NP-complete both with and without transferable utility. We demonstrated that what makes the problem hard in both cases is determining the collaborative possibilities (the set of outcomes possible for the grand coalition); we did so by showing that if these are given, the problem becomes solvable in time polynomial in the size of the representation in both cases. However, we then demonstrated that for a hybrid version of the problem, where utility transfer is possible only within the grand coalition, the problem remains NP-complete even when the collaborative possibilities are given. Finally, we showed that if the characteristic function is convex, a solution in the core can be constructed efficiently (in $O(nl^2)$ time, where n is the number of agents and l is the number of synergies), even when the collaborative possibilities are not given in advance.

Future research can take a number of different directions. One such direction is to investigate restricted families of games in characteristic form (such as the family of convex games, which we analyzed in this paper). Another direction is to evaluate other solution concepts in cooperative game theory from the perspective of computational complexity under our input representation. A long-term goal is to extend the framework for finding a strategically stable solution to take into account issues of computational complexity in determining the synergies among coalitions [30,34]. For example, nontrivial routing problems may need to be solved (potentially only approximately) in order to determine the synergies. How to efficiently construct a stable solution when the cost of determining a synergy is significant (and possibly varying across coalitions) is an important open problem. Finally, we can investigate the same questions under different representations. These representations can be for restricted domains only (we discussed some results along this line in Section 3); they can also be general, so that the representation is concise only on some instances (ideally those that are likely to occur in practice). In the latter category, we have recently introduced and studied another representation that allows for the decomposition of the characteristic function over distinct *issues*, each of which concerns only a small number of the agents [7].

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] R. Aumann, Acceptable points in general cooperative n-person games, in: Contributions to the Theory of Games, vol. IV, in: Annals of Mathematics Study, vol. 40, Princeton University Press, Priceton, NJ, 1959, pp. 287–324.
[2] B.D. Bernheim, B. Peleg, M.D. Whinston, Coalition-proof Nash equilibria: I. Concepts, J. Econ. Theory 42 (1) (1987) 1–12.
[3] J.M. Bilbao, Cooperative Games on Combinatorial Structures, Kluwer Academic, Dordrecht, 2000.
[4] J.M. Bilbao, J.R. Fernandez, J.J. Lopez, Complexity in cooperative game theory, 2000.
[5] A. Charnes, K.O. Kortanek, On balanced sets, cores, and linear programming, Technical Report 12, Cornell Univ., Dept. of Industrial Eng. and Operations Res., Ithaca, NY, 1966.
[6] K. Chatterjee, B. Dutta, D. Ray, K. Sengupta, A noncooperative theory of coalitional bargaining, Rev. Econ. Stud. 60 (1993) 463–477.
[7] V. Conitzer, T. Sandholm, Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains, in: Proceedings of the National Conference on Artificial Intelligence (AAAI), San Jose, CA, 2004, pp. 219–225.
[8] X. Deng, T. Ibaraki, H. Nagamochi, Algorithms and complexity in combinatorial optimization games, in: Proceedings of the Eighth Annual ACM–SIAM Symposium on Discrete Algorithms, 1997, pp. 720–729.
[9] X. Deng, C.H. Papadimitriou, On the complexity of cooperative solution concepts, Math. Oper. Res. (1994) 257–266.
[10] R. Evans, Coalitional bargaining with competition to make offers, Games Econ. Behav. 19 (1997) 211–220.
[11] U. Faigle, S. Fekete, W. Hochstattler, W. Kern, Approximating the core of Euclidean TSP games, Oper. Res. (1993) 152–156.
[12] U. Faigle, S. Fekete, W. Hochstattler, W. Kern, On the complexity of testing membership in the core of min-cost spanning trees, Internat. J. Game Theory 26 (1997) 361–366.
[13] D. Gillies, Some theorems on n-person games, Ph.D. thesis, Princeton University, Department of Mathematics, 1953.
[14] M. Goemans, M. Skutella, Cooperative facility location games, J. Algorithms 50 (2004) 194–214, early version: SODA 2000, pp. 76–85.
[15] A. Gupta, A. Srinivasan, E. Tardos, Cost-sharing mechanisms for network design, in: Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Cambridge, MA, 2004, pp. 139–150.
[16] J.P. Kahan, A. Rapoport, Theories of Coalition Formation, Lawrence Erlbaum Associates, 1984.
[17] R. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.

[18] S. Ketchpel, Forming coalitions in the face of uncertain rewards, in: Proceedings of the National Conference on Artificial Intelligence (AAAI), Seattle, WA, 1994, pp. 414–419.
[19] K. Larson, T. Sandholm, Anytime coalition structure generation: An average case study, J. Experiment. Theoret. AI 11 (2000) 1–20, early version: AGENTS 1999, pp. 40–47.
[20] E. Markakis, A. Saberi, On the core of the multicommodity flow game, in: Proceedings of the ACM Conference on Electronic Commerce (ACM-EC), San Diego, CA, 2003, pp. 93–97.
[21] A. Mas-Colell, M. Whinston, J.R. Green, Microeconomic Theory, Oxford University Press, Oxford, 1995.
[22] P. Milgrom, J. Roberts, Coalition-proofness and correlation with arbitrary communication possibilities, Games Econ. Behav. 17 (1996) 113–128.
[23] D. Moreno, J. Wooders, Coalition-proof equilibrium, Games Econ. Behav. 17 (1996) 80–112.
[24] A. Okada, A noncooperative coalitional bargaining game with random proposers, Games Econ. Behav. 16 (1996) 97–108.
[25] M.J. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, Cambridge, MA, 1994.
[26] M. Pal, E. Tardos, Group strategyproof mechanisms via primal-dual algorithms, in: Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS), 2003, pp. 584–593.
[27] C. Papadimitriou, Algorithms, games and the Internet, in: Proceedings of the Annual Symposium on Theory of Computing (STOC), 2001, pp. 749–753.
[28] I. Ray, Coalition-proof correlated equilibrium: A definition, Games Econ. Behav. 17 (1996) 56–79.
[29] T. Sandholm, K. Larson, M. Andersson, O. Shehory, F. Tohmé, Coalition structure generation with worst case guarantees, Artificial Intelligence 111 (1–2) (1999) 209–238, early version: AAAI 1998, pp. 46–53.
[30] T. Sandholm, V.R. Lesser, Coalitions among computationally bounded agents, Artificial Intelligence 94 (1) (1997) 99–137, special issue on Economic Principles of Multiagent Systems. Early version: IJCAI 1995, pp. 662–669.
[31] L.S. Shapley, On balanced sets and cores, Naval Research Logistics Quarterly 14 (1967) 453–460.
[32] O. Shehory, S. Kraus, A kernel-oriented model for coalition-formation in general environments: Implementation and results, in: Proceedings of the National Conference on Artificial Intelligence (AAAI), Portland, OR, 1996, pp. 134–140.
[33] O. Shehory, S. Kraus, Methods for task allocation via agent coalition formation, Artificial Intelligence 101 (1–2) (1998) 165–200.
[34] F. Tohmé, T. Sandholm, Coalition formation processes with belief revision among bounded rational self-interested agents, J. Logic Comput. 9 (97–63) (1999) 1–23.
[35] W.J. van der Linden, A. Verbeek, Coalition formation: A game-theoretic approach, in: H.A.M. Wilke (Ed.), Coalition Formation, in: Advances in Psychology, vol. 24, North-Holland, Amsterdam, 1985, pp. 29–114.
[36] L.S. Wu, A dynamic theory for the class of games with nonempty cores, SIAM J. Appl. Math. 32 (1977) 328–338.
[37] G. Zlotkin, J.S. Rosenschein, Coalition, cryptography and stability: Mechanisms for coalition formation in task oriented domains, in: Proceedings of the National Conference on Artificial Intelligence (AAAI), Seattle, WA, 1994, pp. 432–437.