Coalition Structure Generation and CS-core: Results on the Tractability Frontier for games represented by MC-nets

Julien Lesca
Université Paris-Dauphine
Place du Mal de Lattre de Tassigny, Paris, France
CNRS, LAMSADE UMR 7243
julien.lesca@dauphine.fr

Patrice Perny
Sorbonne Universités, UPMC
CNRS, LIP6 UMR 7606
4 Place Jussieu, Paris, France
patrice.perny@lip6.fr

Makoto Yokoo
Kyushu University
Motooka 744, Fukuoka, Japan
yokoo@inf.kyushu-u.ac.jp

ABSTRACT

The coalition structure generation (CSG) problem consists in partitioning a group of agents into coalitions to maximize the sum of their values. We consider here the case of coalitional games whose characteristic function is compactly represented by a set of weighted conjunctive formulae (an MC-net). In this context the CSG problem is known to be computationally hard in general.

In this paper, we first study some key parameters of MC-nets that complicate solving make the CSG problem. Then we consider a specific class of MC-nets, called bipolar MC-nets, and prove that the CSG problem is polynomial for this class. Finally, we show that the CS-core of a game represented by a bipolar MC-net is never empty, and that an imputation belonging to the CS-core can be computed in polynomial time.

1. INTRODUCTION

Cooperative game theory develops a set of mathematic tools for modeling various situations involving rational agents and analyzing how groups of agents may adopt cooperative behaviors. These tools provide important foundations for the design and analysis of multiagent systems. One important issue in cooperative games is to analyze coalition structures, i.e. partitions of the set of agents into subsets of agents who perform coordinate actions.

Formally a cooperative game is characterized by a set function that provides the value of every possible coalition of agents. Nevertheless the explicit representation of this set function becomes exponentially large as the number of agents increases. An important issue is therefore to describe the game as succinctly as possible. Several schemes have been proposed for the compact representation of games see e.g. [14, 12, 1]. Among them, MC-nets describing games using a set of weighted logic formulae [22] are presented as a simple and intuitive language. They have received much attention in the last few years [25, 16, 23].

For a given game, an important concern is to find the optimal organization of cooperations among agents. One way of addressing this problem is to look for a coalition structure that maximizes the social welfare (i.e., the total amount of value produced by the set of coalitions). This problem, which is called coalition structure generation (CSG), has been widely investigated in the literature (e.g. [30, 33, 24], see [10] and rahwan15 for a survey on that topic). The CSG problem is highly combinatorial due to the high number[1] of possible coalition structures. However, for some subclasses of games (e.g., super-additive games) the CSG problem is easy to solve since the *grand coalition* is known to be optimal. The CSG problem for games represented by MC-nets has been investigated (e.g. [33, 21, 32]) and proven to be NP-hard in [25]. Various polynomial-time solvable classes of games have also been studied for it [3, 34, 6, 4]. It is worth continuing the study of easy and hard cases for other classes of MC-nets to further specify the frontier of tractability for this problem. One contribution of this paper is to study games represented by a specific class of MC-nets and identify both hard and easy cases for the CSG problem.

Once coalitions are formed we need to define how the welfare (or value) obtained by each one is shared among participants. In an environment with money, a frequent assumption consists of considering transferable utilities which supposes that the overall gain or welfare obtained by a coalition can be freely divided among agents (for example using money transfers). In that case, we need to define how important each agent is to the overall cooperation, and what payoff she can reasonably expect. Knowing that agents may collude to obtain more money, we have to impose some stability properties to reach an equilibrium. One simple way of defining this stability is to consider the core concept [17]. Informally, a solution belongs to the core if no subset of agents has an incentive to deviate. The concept of the core has been generalized [2], under the name of CS-core, to cases where agents are not constrained to form the grand coalition. Unfortunately there are situations where this stability cannot be attained because the game's CS-core is empty. On the other hand, many families of games have a non-empty CS-core, e.g. balanced games [29, 20], and looking for a solution in the CS-core makes sense to ensure stability. The complexity of computing solutions belonging to the core has been widely studied in AI (e.g. [12, 15, 1, 18, 23]) and polynomial cases have been identified (e.g. [7, 8, 9, 5]). However, most of these works focused on the classical notion of core and very few considered the more general definition of CS-core. Testing the emptiness of the CS-core is Δ_2^P-complete [19]. In the second part of this paper, we define a new class of

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

[1] The number of possible coalition structures is in $\Omega(n^{\frac{n}{2}})$ [27], where n is the number of agents

MC-nets and prove that the games representable in it have a non-empty CS-core, and computing a solution that belongs to the CS-core can be performed in polynomial time. Some polynomial cases for CS-cores have been recently identified [11], under some restrictions about possible coalition structures. The polynomial cases presented in this paper do not require such restrictions.

2. PRELIMINARIES

For any $i \in \mathbb{N}$, we denote by $[i]$ set $\{1, \ldots, i\}$ and consider a set of n agents $\mathcal{N} = [n]$ who are willing to cooperate to improve their incomes. The income for any subset of agents is defined by a set function $v : 2^{\mathcal{N}} \to \mathbb{R}_+$ such that $v(\emptyset) = 0$, which is called a *game*. For any $S \subseteq \mathcal{N}$ value $v(S)$ represents the total amount of money that agents of S can share if they cooperate. In the following, let us call *coalition* any subset of cooperating agents.

A *coalition structure* $\pi = \{S_1, \ldots, S_k\}$ is a partition of $\mathcal{N}$ into coalitions. Let $\Pi_{\mathcal{N}}$ be the set of all possible coalition structures on $\mathcal{N}$. We assume that agents are free to form any possible coalition structure. The value of a coalition structure is defined additively as follows.

Definition 1. *The value of a coalition structure π is defined by $v(\pi) = \sum_{S \in \pi} v(S)$.*

A coalition structure $\pi^* \in \Pi_{\mathcal{N}}$ is said to be *optimal* whenever $v(\pi^*) = \max\{v(\pi) : \pi \in \Pi_{\mathcal{N}}\}$. The *CSG problem* consists in finding such an optimal coalition structure π^*.

Describing a game by its characteristic form, i.e. the set of values $v(A)$ for all A in $\mathcal{N}$, may become unfeasible as n growths, since 2^n values are needed. In this paper we assume that games are represented by basic MC-nets [22]. A basic MC-net is a finite set of rules $\{\phi_i : (P_i, N_i) \to v_i\}_{i \in K}$, where $P_i \subseteq \mathcal{N}$, $N_i \subset \mathcal{N}$, $P_i \cap N_i = \emptyset$, $v_i \in \mathbb{R} \setminus \{0\}$, and K is the set of indices for these rules. A rule ϕ_i is said to be *applicable* to coalition S if $P_i \subseteq S$ and $N_i \cap S = \emptyset$, i.e. where S contains all the agents in P_i and no agent in N_i. By extension, ϕ_i is said to be applicable to a coalition structure π if it is applicable to a coalition S belonging to π. The set of indices of rules applicable to S is denoted as $\mathcal{A}(S)$. Hence, the value of any coalition S is given by the sum of the weights of all rules applicable to S. Formally, $v(S) = \sum_{i \in \mathcal{A}(S)} v_i$.

Example 1. *Consider MC-net $\{\phi_1 : (\{1,3\}, \emptyset) \to 3, \phi_2 : (\{1,2\}, \emptyset) \to 3, \phi_3 : (\{1\}, \{2\}) \to 2\}$. Let $S = \{1, 2, 3\}$. We have $\mathcal{A}(S) = \{1, 2\}$, corresponding to rules ϕ_1 and ϕ_2. Hence $v(S) = 3 + 3 = 6$. It can easily be checked that $\{S\}$ is the optimal coalition structure.*

One can interpret any pair (P_i, N_i) as a conjunctive formula including the positive literals of P_i and the negative literals of N_i. Hence, basic MC-nets form a subclass of more general MC-nets that may include other forms of weighted formulae [22]. Yet, they are still fully expressive for representing games.

3. COMPUTATIONAL ISSUES IN CSG

Given an MC net, we distinguish two types of rules. Let $P = \{i \in K : P_i \neq \emptyset\}$, and let $\bar{P} = \{i \in K : P_i = \emptyset\}$ be its complement. By definition, the rules in P are applicable at most to one coalition within a given coalition structure, whereas the rules in $\bar{P}$ may be applicable to several coalitions. In this section, we assume that $\bar{P}$ is empty, i.e. where no rule containing only negative literals is used. Furthermore, analogously to previous work [25], we focus here on basic MC-nets with positive weights, i.e. any weight v_i in a rule of P must be strictly positive. Such a restriction on the representation does not harm the expressivity because it is well known that any game with positive values can be represented by a basic MC-net with positive weights (e.g. [31]). Even under such restrictions on MC-nets, the following result (Theorem 2 of [25]) shows that CSG is computationally hard to solve.

Theorem 1. *Finding an optimal coalition structure is strongly[2] NP-hard. Moreover, unless $P = NP$, there exists no polynomial-time $O(|P|^{1-\epsilon})$-approximation algorithm for any $\epsilon > 0$.*

In the following subsections, we describe a graphical representation of the CSG problem for basic MC-nets with positive weights and study the impact of some parameters of this representation on the problem's complexity.

3.1 CSG-graph of an MC-net

Each rule ϕ_i in P supports cooperation between agents in the sense that the agents in P_i have to belong to the same coalition to obtain reward v_i. A set of rules $R \subseteq P$ is said to be *feasible* if there exists a coalition structure π where each rule in R is applicable to π. As highlighted by [25], the CSG problem can be reformulated to find a feasible subset of rules in P that maximizes the sum of their weights.

As an illustration, let us return to Example 1. The feasible sets of rules are $\{\phi_1, \phi_2\}$, $\{\phi_1, \phi_3\}$ and any subset of them, including the empty set. For example, both ϕ_1 and ϕ_3 are applicable to coalition structure $\pi = \{\{1, 3\}, \{2\}\}$ which makes $\{\phi_1, \phi_3\}$ a feasible set. Here the optimal feasible set is $\{\phi_1, \phi_2\}$ with a value 6 corresponding to the grand coalition.

The feasibility of a set of rules results from the interactions between its rules. These interactions are complex but can be modeled by a colored multigraph $G = (V, E)$, called a *CSG-graph*[3], defined as follows:

- set of vertices V is P (vertex i represents rule ϕ_i);
- green edge $\{i, j\}$ in E is associated to any pair of rules $(i, j) \in P^2$ such that $P_i \cap P_j \neq \emptyset$;
- red edge $\{i, j\}$ in E is associated to any pair of rules $(i, j) \in P^2$ such that $P_i \cap N_j \neq \emptyset$.

An instance of a CSG-graph is given in Fig. 1 in the case of Example 1 (red edges are represented as dotted lines).

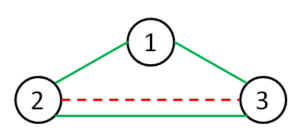

Figure 1: CSG-graph of Example 1

Let G_g denote the subgraph of G that is restricted to the green edges, and let G_r be the subgraph of G that is restricted to the red edges. Furthermore, let $G_g[R]$ denote the

[2]The strong NP-hardness was not previously stated but this result is implied by the proof of Theorem 2 in [25].

[3]Note that a CSG-graph is a simplified version of the graphical representation described in [25].

vertex-induced subgraph of G_g restricted to the vertices of $R \subseteq P$. For given coalition structure π, if two rules ϕ_i and ϕ_j are connected by a green edge $\{i, j\}$, then they are applicable to the same coalition $S \in \pi$ such that $P_i \cup P_j \subseteq S$. On the other hand, if i and j are connected by a red edge then rules ϕ_i and ϕ_j are not applicable to the same coalition of π due to condition $P_i \cap N_j \neq \emptyset$. This implies that for any coalition structure π and any pair of rules $\{i, j\}$ that are connected by a red edge, if i and j are also connected by a path containing only green edges, at least one rule visited by this path is not applicable to π. Hence, as shown[4] in [25] (Theorem 1), set of rules R is feasible if and only if no connected component of $G_g[R]$ contains a pair $\{i, j\}$ connected by a red edge in G. Identifying a feasible set of rule with maximum weight (i.e. for solving the CSG problem) can be achieved by solving the weighted vertex multicut problem:

Weighted Vertex Multicut (WVMC)
Input: Undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, collection $\mathcal{H} \subseteq \mathcal{V}^2$ of pairs of vertices, and weights $w(x) \in \mathbb{R}_{\geq 0}$, for $x \in \mathcal{V}$.
Question: Find $A \subseteq \mathcal{V}$ whose removal separates each pair in $\mathcal{H}$ and such that $w(A) = \sum_{x \in A} w(x)$ is minimum.

To solve the CSG problem, we just have to solve the above WVMC problem on graph G_g with $\mathcal{H}$ defined as the set of red edges and $w(i) = v_i$ for any vertex i of G_g. The optimal set of rules R we are looking for is given by the complement of A in V. By minimizing the weight of A we indeed maximize the weight of its complement.

Let us illustrate this point on the graph of Example 1 (Fig. 1). Pair $\{2, 3\}$ is connected with a red edge but also by two green paths (one direct and one including vertex 1). Hence to cancel both paths, we need to remove either vertex 2 or 3. Here the optimal cut is $A = \{3\}$ because $w(2) > w(3)$ and the optimal feasible set is $R = \{1, 2\}$.

3.2 Complexity issues and CSG-graph

First, note that the WVMC problem can be solved independently in the different connected components of $\mathcal{G}$. This implies that the CSG-problem can also be solved independently in the different connected components of G_g. In the following, we assume that a CSG-graph is such that G_g is composed by a unique connected component.

Even though the WVMC problem is known to be NP-hard [13] for $|\mathcal{H}| \geq 3$, it is polynomial for $|\mathcal{H}| \leq 1$ (e.g., for $|\mathcal{H}| = 1$ it boils down to a mincut problem). The latter property suggests a polynomial algorithm to solve instances when the CSG-graph includes at most one red edge. It is indeed sufficient to find a vertex mincut that separates the two vertices linked by a red edge in G (vertex mincut is a variant of min cut where cuts are made of vertices instead of edges). On the other hand we have:

Proposition 1. *The CSG problem is NP-hard even if the CSG-graph includes at most three red edges.*

Proof. The proof is based on a reduction from WVMC with parameters $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{H}$. We introduce one agent a_i per vertex $i \in \mathcal{V}$ and one agent a_{ij} per pair $\{i, j\} \in \mathcal{H}$. Then, we define an MC-net with exactly one rule ϕ_i by vertex $i \in \mathcal{V}$, with $P_i = \{a_i\} \cup \{a_j : j \in \mathcal{V}, \{i, j\} \in \mathcal{E}\} \cup \{a_{ij} : j \in \mathcal{V}, \{i, j\} \in \mathcal{H}, i < j\}$, $N_i = \{a_{ij} : i \in \mathcal{V}, \{i, j\} \in \mathcal{H}, i > j\}$ and $v_i = w(i)$. By construction, optimal solution A of WVMC has value $w(\mathcal{V}) - v(\pi^*)$, where π^* is the optimal coalition structure of the game defined by the MC-net. □

[4]Note that the statement of Theorem 1 in [25] is slightly different due to the difference between the two graphical representations. However, the implication of this theorem on CSG-graph representation is straightforward.

This proposition provides evidence that, for games represented by basic MC-nets with positive weights, the complexity of the CSG problem is related to the number of red edges in CSG-graphs. Unfortunately, even if the number of red edges of a CSG-graph is a fixed parameter, there is no hope to find a polynomial algorithm. In other words, the CSG problem does not belong to the parameterized complexity class XP, when the fixed parameter under consideration is the number of red edges in the CSG-graph.

Another complexity issue is the approximability of the CSG problem for games represented by basic MC-nets with positive weights. Actually, a $|P|$-approximation can easily be obtained by choosing the rule ϕ_i of P with the highest weight v_i, and by constructing a coalition structure where the agents of P_i form a coalition and the other agents form singleton coalitions. Theorem 1 shows that this is essentially the best approximation ratio achievable, up to a multiplicative constant factor. Now the question is whether we can achieve a better approximation ratio by fixing some parameters of the problem. The following proposition shows that a small number of red edges in the CSG-graph enables a better approximation ratio than $O(|P|)$.

Proposition 2. *Whenever the CSG-graph contains no more than k red edges, it is possible to compute in polynomial time a k-approximation of the optimal coalition structure.*

Proof. Let $C_1, C_2, \ldots, C_t$ be the connected components of G_r that contain at least two vertices. Any vertex of these connected components has a degree greater than or equal to one. For any $i \in \{1, ..., t\}$, and $j \in \{1, \ldots, k\}$, we define C_i^j as follows. If C_i contains at least j vertices of degree greater than or equal to 2 then C_i^j is a singleton containing the j^{th} vertex of a degree greater than or equal to 2 in C_i, where the order is taken arbitrarily. Note that at most k vertices can have a degree greater than or equal to 2 in a graph containing no more than k edges. If C_i contains exactly 2 vertices then C_i^1 is a singleton that contains one of these vertices taken arbitrarily. For other values of i and j, C_i^j is an empty set.

Consider the following k sets of rules:

- $R_j = \bigcup_{i=1}^{t} C_i^j$, for $j = 1, \ldots, k-1$
- $R_k = P \setminus \bigcup_{j=1}^{k-1} R_j$

For any $j \in [k-1]$, R_j contains at most one rule per connected component of G_r. Therefore, these sets are feasible since they cannot contain a pair of rules connected by a red edge. Furthermore, if R_k does not contain a vertex of degree higher than 1 in G_r then it cannot contain two rules connected by a red edge, except if those two rules are the two members of a connected component of G_r that contains exactly two vertices. But in the later case, we know that one rule is part of R_1 and the other is part of R_k. On the other hand, if R_k contains a vertex of degree more than two then there is only one connected component containing at least two vertices in G_r i.e., $t = 1$. This connected component C_1 is a cycle containing exactly k vertices, and R_k contains only one of them. Therefore R_k is also feasible.

Among such k feasible sets, let R_{j^*} be a set that maximizes $\sum_{i\in R_j} v_i$. It is easy to verify that the value of R_{j^*} is at least equal to $\sum_{i\in P} v_i/k$. Hence, the coalition structure associated with R_{j^*} has a value greater than or equal to $1/k$ times the value of the optimal coalition structure. $\square$

Note that the proof of Proposition 2 remains valid if k is the size of the largest connected component of G_r. Therefore the following result also holds:

Corollary 1. *Whenever the size of the largest connected component of the red edges in the CS-graph equals k, it is possible to compute in polynomial time a k-approximation of the optimal coalition structure.*

To summarize this section, we provided evidence that the number of red edges in the CSG-graph contributes to the complexity of the CSG problem.

4. TOWARDS MORE GENERAL RULES

In this section we consider a more general case where $\bar{P}$ is nonempty and propose an algorithm to solve the CSG problem for that specific case. In $\bar{P}$ all rules are assumed to have positive weights, as in P. Note that, by definition, all the rules of $\bar{P}$ apply to the empty set. However, since the value of the empty coalition equals 0, we assume that no rule of $\bar{P}$ applies to the empty set. This assumption can be made without loss of generality since the empty set does not play a role in the CSG problem. We denote by $\hat{v}$ the restriction of v to the rules of $\bar{P}$, i.e. where for any coalition S, $\hat{v}(S) = \sum_{i\in\bar{P}:S\cap N_i=\emptyset} v_i$, and for any coalition structure π, $\hat{v}(\pi) = \sum_{S\in P} \hat{v}(S)$.

Example 2. *Modify Example 1 by adding rule $\phi_4 : (\emptyset, \{1\}) \rightarrow 2$. With this new set of rules, the value of the grand coalition $\mathcal{N}$ remains unchanged because ϕ_4 does not apply to this coalition. On the other hand, the value of coalition structure $\pi = \{\{1, 3\}, \{2\}\}$ becomes $3 + 2 + 2$. It is easy to check that π is now the optimal coalition structure for this new set of rules.*

As highlighted in [25], a rule of $\bar{P}$ can be simulated by a polynomial number of rules from P. For example, rule $(\emptyset, \mathcal{N}\setminus[k]) \rightarrow v$ can be simulated using k rules $\phi_1 : (\{1\}, \mathcal{N}\setminus[1]) \rightarrow v$, $\phi_2 : (\{2\}, \mathcal{N}\setminus[2]) \rightarrow v$, $\ldots$, $\phi_k : (\{k\}, \mathcal{N}\setminus[k]) \rightarrow v$. However, these rules introduce multiple red edges in the CSG-graph, and we know from Section 3.2 that they are a source of complexity. In this example, there is a red edge between any pair of new rules $\{\phi_i, \phi_j\}$, and G_r contains a clique of size k. Moreover, ϕ_1 contains almost all of the agents in N_1. Such a rule will be connected in G_r to almost all the rules and create a large connected component in G_r. This section propose another way of dealing with the rules of $\bar{P}$.

Let us extend our discussion about the expressivity of the rules belonging to $\bar{P}$. For a given set of agents S, rule $(\emptyset, \mathcal{N}\setminus S) \rightarrow v$ assigns weight v to any subset of S. The particularity of this rule, compared to the rules of P, is that it can be applied to several coalitions of a coalition structure. For example, rule ϕ_4 of Example 2 can be applied to coalitions $\{1\}$ and $\{3\}$. Hence ϕ_4 can be applied twice to coalition structure $\{\{1\}, \{2\}, \{3\}\}$. Actually, a rule $(\emptyset, \mathcal{N}\setminus S) \rightarrow v$ can be applied as many times as there are subsets of S in a coalition structure. Therefore whenever v is positive, it is always beneficial to split a coalition into subcoalitions to collect several times the value v (once for each subset of S).

4.1 New approach to solve CSG problem

Similar to the case where $\bar{P}$ is empty, we solve the CSG problem by computing the feasible subset of rules in P with the highest weight. But now, the weight of feasible set of rules R not only depends on the rules that are part of R, but also on the rules of $\bar{P}$. To compute the weight provided by those rules to a feasible subset of rules R, we need first to define coalition structure π where all rules of R are applicable. Then we can compute $\hat{v}(\pi)$, i.e. the weight of π provided by the rules of $\bar{P}$.

For each feasible subset of rules R, there exists a whole set $\Pi(R)$ of coalition structures for which all the rules of R are applicable. For example, for feasible subset of rules $\{\phi_2\}$ of Example 2, both $\pi_1 = \{\{1, 2\}, \{3\}\}$ and $\pi_2 = \{\{1, 2, 3\}\}$ belong to $\Pi(\{\phi_2\})$. Among them, π_1 seems most promising since $\hat{v}(\pi_1) = 2 > \hat{v}(\pi_2) = 0$. Within $\Pi(R)$, let $\pi(R)$ denote the most refined coalition structure where the notion of refinement is defined as follows. Coalition structure π' is a refinement of π when any coalition of π is the union of the coalitions of π'. By construction, $\pi(R)$ is a coalition structure maximizing $\hat{v}$ over $\Pi(R)$. Therefore $\pi(R)$ is a good candidate to represent a feasible set of rules R. Based on this observation, we define the weight of a subset of rules R as follows:

$$w(R) = \begin{cases} \sum_{i\in R} v_i + \hat{v}(\pi(R)) & \text{if } R \text{ feasible} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Note that for a given feasible set of rules R, $w(R)$ may differ from v applied to $\pi(R)$. Indeed, some rules of $P \setminus R$ may apply to $\pi(R)$. However, the following proposition shows that the CSG problem can be solved by finding the feasible subset of rules that maximizes w:

Proposition 3. *If $R^* \subseteq P$ maximizes w, then $\pi(R^*)$ is an optimal coalition structure of game v.*

PROOF. Note that $\pi(\emptyset)$ is a coalition structure that includes one coalition per agent. Furthermore, $\bar{P}$ includes at least one rule i with strictly positive weight v_i. This rule applies to at least one singleton in $\pi(\emptyset)$ and implies that $\hat{v}(\pi(\emptyset)) \geq v_i > 0$. Therefore $w(\pi(R^*)) > 0$ and R^* is necessarily feasible.

For any feasible set of rules $R \subseteq P$ there exists at least one feasible set of rules $R' \subseteq P$ such that $R' \supseteq R, \pi(R) = \pi(R')$ and $w(R') = v(\pi(R'))$. Set R' contains all of the formulae of P that are true for at least one coalition of $\pi(R)$. Since $\forall R, R' \subseteq P$ such that $R \subseteq R'$ and $\pi(R) = \pi(R')$ we have $w(R) \leq w(R')$, and then $\pi(R^*)$ is the best coalition structure in $\{\pi(R)|R \subseteq P, R \text{ feasible}\}$.

Let us prove now that the optimal coalition structure necessarily belongs to $\{\pi(R)|R \subseteq P, R \text{ feasible}\}$. Let $\pi \in \Pi_{\mathcal{N}}\setminus\{\pi(R)|R \subseteq P, R \text{ feasible}\}$. Let $R \subseteq P$ be the maximal subset of rules in P that apply to π. Note that R is feasible. For any coalition $S \in \pi$, by definition of $\pi(R)$, there exists a subset of coalitions $\pi' \subseteq \pi(R)$ such that π' forms a partition of S. If we divide S according to π', then the rules in R remain applicable to $\pi(R)$. Let C be the subset of rules of $\bar{P}$ that apply to S. For any $S' \in \pi'$, we have $S' \subseteq S$ and therefore all the formulae in C are also applicable to S' (since for any $i \in \bar{P}, S \cap N_i = \emptyset \Rightarrow S' \cap N_i = \emptyset$). Since the weights are strictly positive, then applying this subdivision to any coalition of π implies $v(\pi) \leq v(\pi(R))$. So an optimal coalition structure necessarily belongs to $\{\pi(R) | R \subseteq P\}$, which concludes the proof. □

4.2 Computing optimal coalition structure

To compute the value of w for a given subset of rules $R \subseteq P$, we need first to define whether R is feasible and then compute $\pi(R)$. Deciding whether R is feasible can be performed in polynomial time. We only have to inspect all of the connected components of $G_g[R]$ and check that none includes two vertices connected by a red edge in G. Moreover, $\pi(R)$ can be computed by regrouping agents by the rules of R as described by the following polynomial algorithm.

Algorithm 1: Algorithm to compute $\pi(R)$

```
Data: Feasible set of rules R and CSG-graph G
Result: Coalition structure π ∈ Π_N
π ← {{}};
A ← N;
foreach connected component C of G_g[R] do
    S ← ⋃_{i∈C} P_i;
    π ← π ∪ {S};
    A ← A \ S;
end
foreach i ∈ A do
    π ← π ∪ {i};
end
return π;
```

The correctness of Algorithm 1 is established as follows.

Proposition 4. *For any feasible set of rules R, the coalition structure returned by Algorithm 1 corresponds to $\pi(R)$.*

Proof. To establish the result we prove that all of the rules of R are applicable to coalition structure π returned by Algorithm 1, and no coalition of π can be split without loosing this property. Let us first show that all rules of R are applicable to π. By contradiction, we assume that there exists a rule in R that is not applicable to π. Without loss of generality, let ϕ_1 be this rule. By construction of π, there exists a coalition S in π such that $P_1 \subseteq S$. If ϕ_1 does not apply to S then $N_1 \cap S \neq \emptyset$. This implies by the construction of π that there is a sequence of rules $\phi_1, \phi_2, \dots \phi_k$ belonging to R such that $P_i \cap P_{i+1} \neq \emptyset$ for any $i < k$, and $N_1 \cap P_k \neq \emptyset$. But this sequence implies that there is a path of green edges from ϕ_1 to ϕ_k in $G_g[R]$, and ϕ_1 and ϕ_k are linked by a red edge in G. This leads to a contradiction with R being feasible.

Let us show now that no coalition of π can be split while preserving the applicability of all the rules in R. Assume, by contradiction, that there exists refinement π' of π in $\Pi(R)$, resulting from π by splitting at least one of its coalitions S. Note that S cannot be a singleton. This implies that there exists a subset of rules $R_S \subseteq R$ connected in $G_g[R]$ and such that $S = \bigcup_{i \in R_S} P_i$. Let i and j be two agents belonging to S who are not in the same coalition in π'. Let ϕ_1 and ϕ_k be two rules of R_S such that $i \in P_1$ and $j \in P_k$. Since ϕ_1 and ϕ_k are part of the same connected component of $G_g[R]$ which contains the rules of R_S, there must be a sequence of rules $\phi_1, \phi_2, \dots \phi_k$ belonging to R_S such that $P_i \cap P_{i+1} \neq \emptyset$ for any $i < k$. Because i and j are not part of the same coalition in π', there must be at least one rule of this sequence that does not apply to π'. This leads to a contradiction. □

To summarize, the computation of $w(R)$ for any $R \subseteq P$ can be performed in polynomial time. One possible way of solving the CSG problem i.e., finding the feasible subset of rules that maximizes w (see Proposition 3), is to use a brute-force algorithm to test every subsets of rules of P. This is an FPT algorithm when the fixed parameter is the size of P. However, if $|P|$ is large then such a brute-force algorithm will be inefficient. In that case, it would be interesting to investigate the properties of set function w. Once the set of rules R^* maximizing w is computed, the optimal coalition structure can be derived in polynomial time from R^* by determining $\pi(R^*)$.

Before discussing polynomial cases, note that the proofs of Propositions 3 and 4 can be extended to any set function $\hat{v}$ where splitting coalitions is always beneficial i.e., to any set function such that $\hat{v}(S) \leq \hat{v}(\pi)$ for all $S \subseteq \mathcal{N}$ and $\pi \in \Pi_S$. Hence the algorithm remains valid even if we enrich the class of basic MC-nets with positive weights by allowing extra rules of the following types[5]:

- conjunction of positive literals with negative weight i.e., rules of type $\{\bigwedge_{j \in P_i} x_j\} \to v_i$, with $v_i < 0$;
- disjunction of positive literals with positive weight i.e., rules of type $\{\bigvee_{j \in P_i} x_j\} \to v_i$, with $v_i > 0$;
- disjunction of conjunction of negative literals with positive weight i.e. rules of type $\{\bigvee_{N_\ell \in \{N_1,\dots,N_k\}} \bigwedge_{j \in N_\ell} \neg x_j\} \to v_i$, with $v_i > 0$.

Therefore this algorithm can be applied to a broad class of MC-nets. However, since the polynomial case of Section 4.3 does not include these rules, they have been deliberately omitted for simplicity and clarity.

4.3 Polynomial cases for CSG problem

In general, it is impossible to compute the maximum value of a set function without checking all of its values. However, the problem is known to be solvable in polynomial time for supermodular set functions, which are defined as follows:

Definition 2. *Set function w over 2^P is supermodular whenever for every, $A, B \subseteq P$ such that $B \subseteq A$, and for every $j \in P \setminus A$, $w(A \cup \{j\}) - w(A) \geq w(B \cup \{j\}) - w(B)$.*

If we were to identify a restriction on basic MC-nets, and more precisely on the set of rules P, such that the set function described by (1) is supermodular, then the CSG problem would be solvable in polynomial time for this restricted language. We know that using rules of P does not simplify solving the CSG problem. Therefore according to Proposition 1, there is no hope of finding a polynomial case without restricting the number of red edges of the CSG-graph to a value lower than 3. The following proposition shows that when the CSG-graph does not contain any red edges, set function w is supermodular.

[5] The rules are described by logic formulae over the set of boolean variables $\{x_1, \dots, x_n\}$. For each agent i, x_i is true if i is part of the coalition, and false otherwise. The weight of a rule is attributed to a coalition if the logic formula is true. This description is consistent with MC-nets [22]

Proposition 5. *If P is feasible then set function w defined by (1) on 2^P is supermodular.*

PROOF. Let $A, B \subseteq P$ such that $B \subseteq A$ and $j \in P \setminus \{A\}$. Since P is feasible, any subset $R \subseteq P$ is also feasible. Therefore $w(R) = \sum_{i \in R} v_i + \hat{v}(\pi(R))$ by (1). Now, observing that $\sum_{i \in A \cup \{j\}} v_i - \sum_{i \in A} v_i = \sum_{i \in B \cup \{j\}} v_i - \sum_{i \in B} v_i = v_j$, we just need to show the following:

$$\hat{v}(\pi(A \cup \{j\})) - \hat{v}(\pi(A)) \geq \hat{v}(\pi(B \cup \{j\})) - \hat{v}(\pi(B)) \quad (2)$$

This property is proved in Lemma 1 of Section Appendix. □

The following corollary summarizes the main implication of Proposition 5 for the CSG problem:

Corollary 2. *If the CSG-graph does not contain any red edges, then the CSG problem can be solved in polynomial time.*

PROOF. By definition, we know that if the CSG-graph does not contain any red edges, then P is feasible. Furthermore, by Proposition 5, w is supermodular because P is feasible. We can resort to a polynomial time algorithm (e.g. [26]) to find $R^* \subseteq P$ that maximizes w. By Proposition 3, $\pi(R^*)$ is an optimal coalition structure that can be computed in polynomial time from R^* (see Proposition 4). □

Thus we have identified a polynomial subcase for the CSG problem characterized by basic MC-nets with positive weights. Since the NP-hardness of the CSG problem starts at the 3 red edges, the cases with 1 and 2 edges remain open. Note that without the rules of $\bar{P}$, the resolution is trivial for the case without red edges since feasible set P would always be optimal. However, this is not the case whenever $\bar{P}$ is not empty.

5. BIPOLAR MC-NETS

We introduce now a new subclass of basic MC-nets with positive weights, called *bipolar MC-nets*, defined as follows.

Definition 3. *A bipolar MC-net is a basic MC-net characterized by a set of rules $\{\phi_i : (P_i, N_i) \to v_i\}_{i \in K}$ such that $P_i = \emptyset$ or $N_i = \emptyset$, $P_i \cup N_i \neq \emptyset$ and $v_i > 0$, for all $i \in K$.*

The rules of bipolar MC-net are therefore partitioned into sets P and $\bar{P}$, where P contains all the preconditions of type $(P_i, \emptyset)$, and $\bar{P}$ contains all the preconditions of type $(\emptyset, N_j)$. Note that the CSG-graph contains by definition no red edge since $N_j = \emptyset$ for all $j \in P$. This implies by Corollary 2 that the CSG problem on a game represented by a bipolar MC-net is solvable in polynomial time.

5.1 Expressivity of bipolar MC-nets

In a bipolar MC-net, the descriptive powers of the rules of P and $\bar{P}$ are antagonist. On one hand, a rule $\phi_i : (P_i, \emptyset) \to v_i$ assigns a positive weight v_i to any superset of P_i. This descriptive power is well known in the literature of decision theory under the name of *belief functions* [28]. On the other hand, a rule $\phi_j : (\emptyset, N_j) \to v_j$ assigns a positive weight v_j to any subset of $\mathcal{N} \setminus N_j$. This language is obviously not fully expressive, but the following example illustrates the expressivity of bipolar MC-nets.

Example 3. *Consider a group of agents which must be partitioned into teams, to collectively complete a project. To define the team's value or efficiency, we may add the individual values resulting from rules of type $\phi_i : (\{i\}, \emptyset) \to v_i$ where v_i is the intrinsic value of agent i. However, some subsets of agents have an inclination to work efficiently together which will provide to their coalition an additional value. For such subset of agents S, this can be modeled by inserting a rule $\phi_S : (S, \emptyset) \to v_S$, where v_S corresponds to the added value due to positive synergy.*

On the other hand, every agent i works at a different maximal speed s_i, depending on her skill, and it may happen that the group's working speed is constrained by the speed of its slowest member. Considering just this aspect, the value of coalition S can be defined as $\min_{i \in S} s_i$. Such negative synergies can be represented by a linear number of rules of $\bar{P}$. Consider indeed n agents such that $s_1 \geq s_2 \geq \ldots \geq s_n$. For $i = 1, \ldots, n$, we insert in $\bar{P}$ rule $\phi_i : (\emptyset, \{1, \ldots, i\}) \to s_i - s_{i+1}$, assuming $s_{n+1} = 0$. With such rules, it is easy to check that the value of any coalition S induced by the rules of $\bar{P}$ is $\hat{v}(S) = \min_{i \in S} s_i$.

Finally, positive and negative synergies among agents can be considered together by merging in the same bipolar MC-net the two sets of rules introduced above, to define the game as the combination of the two phenomena.

More generally, analogously to the fact that rules of P describe belief functions, the descriptive power of the rules of $\bar{P}$ can be analyzed through *sum-min games*[6] defined as follows. Game $\hat{v}$ is a sum-min game if there exists integer $m \in \mathbb{N}$ and $m \times n$ matrix $\mathbf{s}$ of non-negative values, such that $\hat{v}(S) = \sum_{i=1}^{m} \min_{j \in S} s_{ij}$, where s_{ij} denotes the value of $\mathbf{s}$ at row i and column j. The illustrative example provided in this section uses matrix $\mathbf{s}$ of size $1 \times n$. The following proposition shows that the set of games representable by bipolar MC-nets is essentially equivalent to the set of games which are additive combinations of belief functions and sum-min games.

Proposition 6. *Game $\hat{v}$ is sum-min if and only if it can be described by a bipolar MC-net with no rule in P.*

PROOF. The description provided in Example 3 shows how value $\min_{j \in S} s_{ij}$ can be expressed through a set of rules of $\bar{P}$. It remains for us to show that any rule $\phi_i : (\emptyset, N_i) \to v_i$ defines a sum-min game. Let $\mathbf{s} = (s_1, \ldots, s_n)$ be a vector such that $s_j = v_i$ for any $j \in \mathcal{N} \setminus N_i$, and $s_j = 0$ otherwise. It is easy to check that $\min_{j \in S} s_j$ equals v_i if $S \cap N_i = \emptyset$, and 0 otherwise. Therefore the game described by ϕ_i is a sum-min game. □

The proof of Proposition 6 also shows that the transformation from one representation to another cannot increase the size of the description by a factor higher than n.

5.2 Computation of CS-core

Once coalitions are formed we need to share the payoff among the participants of any coalition. This is formalized by the notion of imputation:

Definition 4. *An imputation is a pair $\langle \pi, x \rangle$ with $\pi \in \Pi_{\mathcal{N}}$ and $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, and a vector of payoffs such that for all $i \in \mathcal{N}, x_i \geq v(\{i\})$ and for all $S \in \pi$, $x(S) = v(S)$, where $x(S) = \sum_{i \in S} x_i$.*

[6]The sum-min games introduced in this paper are defined in a similar fashion as the max-game [9].

There are many possible imputations that can be chosen and we focus on some of them that have a good stability property:

Definition 5. *[2] CS-core is the set of imputations $\langle \pi, x\rangle$ such that for any $S \subseteq \mathcal{N}$, $x(S) \geq v(S)$.*

For any imputation in the CS-core, no subset of agents S can outperform $x(S)$. Therefore no subset of agents S has an incentive to abandon its respective coalitions in π to create a new coalition. Indeed, even if the agent of a coalition share their new outcomes, then at least one of them will not benefit from this situation.

It is well known that imputation $\langle \pi, x\rangle$ belongs to the CS-core only if π is the optimal coalition structure. For bipolar MC-nets, optimal coalition structure π^* can be computed in polynomial time. However it remains for us to compute x to find imputation $\langle \pi^*, x\rangle$ that belongs to the core. The following proposition shows that we can focus on the computation of the core for each coalition of π^*.

Proposition 7. *For game v representable by bipolar MC-net, if $\pi^* = \{S_1, \dots, S_k\}$ is the optimal coalition structure then $\langle \pi^*, x\rangle$ belongs to the CS-core for any x such that $x(T_i) \geq v(T_i)$ holds for any $i \in [k]$ and any $T_i \subseteq S_i$.*

Proof. We need to show that for any $T \subseteq \mathcal{N}$, $x(T) \geq v(T)$ holds. For any $i \in [k]$, let T_i be the subset of agents in T belonging to S_i i.e., $T_i = T \cap S_i$. We know by Lemma 2 (see Appendix) that $v(\bigcup_{j=1}^{i} S_j \cup \bigcup_{j=i+1}^{k} T_j) + v(T_i) \geq v(S_i) + v(\bigcup_{j=1}^{i-1} S_j \cup \bigcup_{j=i}^{k} T_j)$ holds, for any $i \in [k]$. By summing up those inequalities, we obtain $v(\bigcup_{j=1}^{k} S_j) + \sum_{j=1}^{k} v(T_j) \geq \sum_{j=1}^{k} v(S_j) + v(\bigcup_{j=1}^{k} T_j)$. Furthermore, the optimality of π^* implies that $\sum_{j=1}^{k} v(S_j) \geq v(\bigcup_{j=1}^{k} S_j)$. By summing up these two inequalities, we obtain $\sum_{j=1}^{k} v(T_j) \geq v(\bigcup_{j=1}^{k} T_j) = v(T)$. Finally, this last inequality implies $x(T) = \sum_{j=1}^{k} x(T_j) \geq \sum_{j=1}^{k} v(T_j) \geq v(T)$. □

According to Proposition 7, we can assume without loss of generality that $\pi^* = \{\mathcal{N}\}$, because otherwise the CS-core can be computed as the concatenation of imputations belonging to the cores of each coalition of π^*. The following proposition provides a formula to compute one particular imputation belonging to the CS-core:

Proposition 8. *For any game v representable by a bipolar MC-net, if $x_i = \max_{T \subseteq [i-1]}\{v(T \cup \{i\}) - \sum_{j\in T} x_j\}$ for any $i \in \mathcal{N}$, then $\langle \pi^*, x\rangle$ belongs to the CS-core.*

Proof. Let S be a subset of $\mathcal{N}$, and let i be the highest index in S. By construction, we know that $x_i \geq v(S) - \sum_{j\in S\setminus\{i\}} x_j$. This implies that $x(S) \geq v(S)$ holds for any $S \subseteq \mathcal{N}$. It remains for us to check that $\langle \pi^*, x\rangle$ is an imputation, i.e. $x(\mathcal{N}) = v(\mathcal{N})$ holds. We show that there exists partition π_n of $\mathcal{N}$ such that $x(\mathcal{N}) \leq v(\pi_n)$. Then the optimality of coalition structure $\{\mathcal{N}\}$ implies $x(\mathcal{N}) \leq v(\pi_n) \leq v(\mathcal{N}) \leq x(\mathcal{N})$, and the result follows.

We prove the existence of π_n by induction on the number of agents. For one agent this statement holds trivially. Assume now that the statement is true for $n-1$ agents, and let us show that it is also true for n agents. By definition of x, we know that there exists $T \subseteq [n-1]$ such that $x_n = v(T \cup \{n\}) - \sum_{j\in T} x_j$, which implies that $\sum_{j=1}^{n} x_j = v(T \cup \{n\}) + \sum_{j\in[n-1]\setminus T} x_j$. We denote by s the highest index in $[n-1] \setminus T$. By induction hypothesis, we know that there exists partition $\pi_s = \{S_1, \dots, S_k\}$ of the agents in $[s]$ such that $\sum_{j=1}^{s} x_j \leq v(\pi_s)$. This implies that $\sum_{j\in[n-1]\setminus T} x_j \leq \sum_{j=1}^{k} v(S_j) - \sum_{j\in T\cap[s]} x_j$ holds. We denote by $B = \{b_1, \dots, b_t\}$ the subset of indices in $[k]$ such that $S_{b_r} \cap T \neq \emptyset$. We know that $\sum_{j\in T\cap S_{b_r}} x_j \geq v(T \cap S_{b_r})$ holds for any $r \in [t]$. This implies that $\sum_{j\in T\cap[s]} x_j = \sum_{r=1}^{t} \sum_{j\in T\cap S_{b_r}} x_j \geq \sum_{r=1}^{t} v(T \cap S_{b_r})$. Altogether, we have $\sum_{j=1}^{n} x_j \leq v(T \cup \{n\}) + \sum_{j=1}^{k} v(S_j) - \sum_{r=1}^{t} v(T \cap S_{b_r})$.

On the other hand, we know by Lemma 2 that $v(S_{b_i}) + v(\bigcup_{r=1}^{i-1} S_{b_r} \cup T \cup \{n\}) \leq v(T \cap S_{b_i}) + v(\bigcup_{r=1}^{i} S_{b_r} \cup T \cup \{n\})$ holds for any $i \in [t]$ (see Appendix). We obtain $v(T \cup \{n\}) + \sum_{r=1}^{t} v(S_{b_r}) \leq v(\bigcup_{r=1}^{t} S_{b_r} \cup T \cup \{n\}) + \sum_{r=1}^{t} v(T \cap S_{b_r})$ by summing all these inequalities over i. Finally, by combining this inequality with the inequality shown in the previous paragraph, we obtain $\sum_{j=1}^{n} x_j \leq v(\bigcup_{r=1}^{t} S_{b_r} \cup T \cup \{n\}) + \sum_{j\in[k]\setminus B} v(S_j)$. Therefore, for π_n defined as the coalition structure containing coalitions $\{\bigcup_{r=1}^{t} S_{b_r} \cup T \cup \{n\}\}$ and $\{S_j\}$, for all $j \in [k] \setminus B$, the requirement is fulfilled. □

Proposition 8 also implies that the CS-core is non empty for any game represented by a bipolar MC-net. We must show how vector x can be computed in polynomial time. The values of x are computed iteratively, starting from x_1 and ending with x_n. To compute x_i, given the values of $x_1, \dots, x_{i-1}$, we define set function $q_i : 2^{[i-1]} \to \mathbb{R}$ such that $q_i(T) = v(T \cup \{i\}) - x(T)$, for any $T \subseteq [i-1]$. x_i should clearly equal $\max_{T\subseteq[i-1]} q_i(T)$. The following proposition shows that these set functions are supermodular, which justifies resorting to a polynomial time algorithm (e.g., [26]) to compute vector x.

Proposition 9. *For any $i \in \mathcal{N}$, q_i is supermodular.*

Proof. For any $A, B \subseteq [i-1]$, such that $B \subseteq A$, and for any $j \in [i-1] \setminus A$, we have $q_i(A) + q_i(B \cup \{j\}) = v(A\cup\{i\}) - x(A) + v(B\cup\{j\}\cup\{i\}) - x(B\cup\{j\}) \leq v(B\cup\{i\}) - x(B) + v(A \cup \{j\} \cup \{i\}) - x(A \cup \{j\}) = q_i(B) + q_i(A \cup \{j\})$, where the inequality is due to Lemma 2 (see Appendix). □

6. CONCLUSION

This paper provides a partial mapping of the hard and easy cases for the CSG problem for games represented by basic MC-nets with positive weights. Some parameters of MC-net representation, such as the number of red edges in the CSG-graph, may be a source of complexity in the CSG problem. The paper also provides an algorithm that solves the CSG problem for a broader class of MC-nets than previous work [25], and studies the polynomial time cases that arise from this new type of optimization. Finally it introduces a subclass of MC-nets, called bipolar MC-nets, which represent games with a non-empty CS-core. The CSG problem as well as the determination of an imputation of the CS-core are problems solvable in polynomial time.

This paper raises several open questions. The CSG problem's complexity remains open for basic MC-nets with positive weights when the number of red edges in the CSG-graph is restricted to one[7] or two. Another question which de-

[7]Whenever the basic MC-net does not contain any rule without positive literals, this case is solvable in polynomial time.

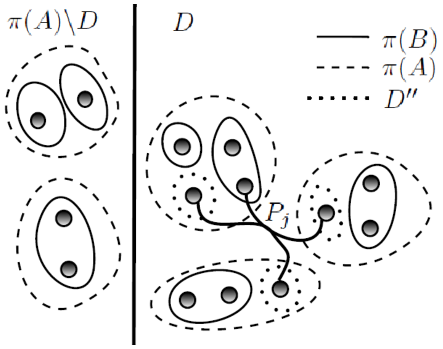

Figure 2: Example of $\pi(A)$ and $\pi(B)$

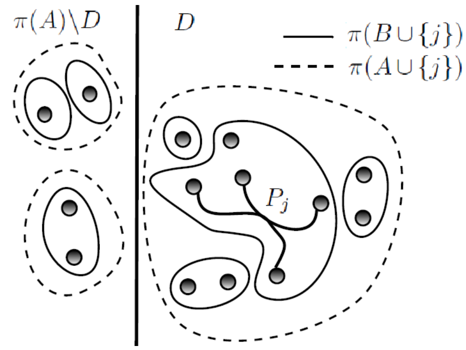

Figure 3: example after adding j to A and B

serves investigation is the possibility of designing an $o(\sqrt{k})$-approximation algorithm, where k is the number of red edges in the CSG-graph. Finally, we know that characteristic functions are fully expressive and that the CSG problem can be solved in polynomial time in the size of this representation. However this representation is not compact. It would be interesting to design a fully expressive language that is as compact as possible and where the CSG problem can be solved in polynomial time. Bipolar MC-nets may be a good starting point to explore this direction.

Appendix

Lemma 1. *Let $j \in P$ and $A, B \subseteq P \backslash \{j\}$ such that $B \subseteq A$. Then the following inequality holds:*

$$\hat{v}(\pi(A \cup \{j\})) - \hat{v}(\pi(A)) \geq \hat{v}(\pi(B \cup \{j\})) - \hat{v}(\pi(B)) \quad (3)$$

Proof. Let $D \subseteq \pi(A)$ be the set of coalitions such that $\forall S \in D, S \cap P_j \neq \emptyset$ and $\forall S' \in \pi(A)\backslash D, S' \cap P_j = \emptyset$. For any $S' \in \pi(A)\backslash D$ we have $S' \in \pi(A \cup \{j\})$, therefore:

$$\hat{v}(\pi(A \cup \{j\})) - \hat{v}(\pi(A)) = \hat{v}(\bigcup_{S \in D} S) - \sum_{S \in D} \hat{v}(S) \quad (4)$$

Figures 2 and 3 can help to figure out where come from the r.h.s. of (4). The coalition belonging to $\pi(A)\backslash D$ does not change by adding rule j to the set of rules that has to hold (left part of Figs. 2 and 3). The only coalition left for $\pi(A \cup \{j\})$ is the one containing P_j which is $\bigcup_{S \in D} S$ (right part of Fig. 3). Let $D' \subseteq \pi(B)$ be the set of coalitions such that $\forall S \in D', S \cap P_j \neq \emptyset$ and $\forall S' \in \pi(B)\backslash D', S' \cap P_j = \emptyset$. For any $S' \in \pi(B)\backslash D'$ we have $S' \in \pi(B \cup \{j\})$, therefore:

$$\hat{v}(\pi(B \cup \{j\})) - \hat{v}(\pi(B)) = \hat{v}(\bigcup_{S \in D'} S) - \sum_{S \in D'} \hat{v}(S) \quad (5)$$

The reasoning that justifies the r.h.s. of (5) is the same as for (4) (in that case we need to isolate in Figs. 2 and 3 the elements of D' instead of D). Select now $D'' \subseteq D'$ such that $\forall S \in D, |\{S'' | S'' \in D'', S'' \subseteq S\}| = 1$ (there are various possible sets D'' and the proof holds for any of them). Fig. 2 illustrates one possible way of choosing D'' within D'. We have $D'' \subseteq D'$, $\bigcup_{S \in D''} S \neq \emptyset$ and $\forall S \in D', \hat{v}(S) \geq 0$, therefore:

$$\hat{v}(\bigcup_{S \in D'} S) - \sum_{S \in D'} \hat{v}(S) \leq \hat{v}(\bigcup_{S \in D''} S) - \sum_{S \in D''} \hat{v}(S) \quad (6)$$

Consider now the difference between the r.h.s. of (4) and (6) (the latter exceeding (5)):

$$\hat{v}(\bigcup_{S \in D} S) - \sum_{S \in D} \hat{v}(S) - \hat{v}(\bigcup_{S \in D''} S) + \sum_{S \in D''} \hat{v}(S) \quad (7)$$

For any $S \in D''$ let $\alpha(S) \in D$ be the coalition of D such that $S \subseteq \alpha(S)$. By the definition of D'' for any $S \in D''$ exists a unique $\alpha(S) \in D$ (this fact is illustrated in Fig. 2) so by the definition of $\hat{v}$, Eq. (7) reads:

$$\sum_{S \in D''} \sum_{\substack{i \in \bar{P} \\ S \cap N_i = \emptyset \\ \alpha(S) \cap N_i \neq \emptyset}} v_i - \sum_{\substack{i \in \bar{P} \\ (\bigcup_{S \in D''} S) \cap N_i = \emptyset \\ (\bigcup_{S \in D} S) \cap N_i \neq \emptyset}} v_i \quad (8)$$

This quantity is non-negative because for any $i \in \bar{P}$ such that $(\bigcup_{S \in D''} S) \cap N_i = \emptyset$ and $(\bigcup_{S \in D} S) \cap N_i \neq \emptyset$ there exists at least one $S \in D''$ such that $S \cap N_i = \emptyset$ and $\alpha(S) \cap N_i \neq \emptyset$. Let us prove this by contradiction. Assume that $\nexists S \in D''$ such that $S \cap N_i = \emptyset$ and $\alpha(S) \cap N_i \neq \emptyset$. $(\bigcup_{S \in D''} S) \cap N_i = \emptyset$ implies that $S \cap N_i = \emptyset$ holds $\forall S \in D''$. So $\alpha(S) \cap N_i = \emptyset$ necessarily holds for any $S \in D''$, but this yields a contradiction since $\bigcup_{S \in D''} \alpha(S) = \bigcup_{S \in D} S$ implies $\bigcup_{S \in D} S \cap N_i \neq \emptyset$. Thus, by combining (7) and (8) we obtain:

$$\hat{v}(\bigcup_{S \in D} S) - \sum_{S \in D} \hat{v}(S) \geq \hat{v}(\bigcup_{S \in D''} S) - \sum_{S \in D''} \hat{v}(S)$$

Finally, combining this inequality with (6), and using (4) and (5), we obtain (3) that concludes the proof. $\square$

Lemma 2. *For any bipolar game $\forall A, B \subseteq \mathcal{N}, A \cap B \neq \emptyset \Rightarrow v(A) + v(B) \leq v(A \cap B) + v(A \cup B)$*

Proof. $\forall A, B \subseteq \mathcal{N}$ such that $A \cap B \neq \emptyset$ we have:

$$\sum_{\substack{i \in P \\ P_i \subseteq A}} v_i + \sum_{\substack{i \in P \\ P_i \subseteq B}} v_i \leq \sum_{\substack{i \in P \\ P_i \subseteq A \cap B}} v_i + \sum_{\substack{i \in P \\ P_i \subseteq A \cup B}} v_i$$

$\forall i \in P, (P_i \subseteq A) \wedge (P_i \subseteq B) \Rightarrow P_i \subseteq A \cap B$ and $(P_i \subseteq A) \vee (P_i \subseteq B) \Rightarrow P_i \subseteq A \cup B$. Furthermore we have $\hat{v}(A) + \hat{v}(B) \leq \hat{v}(A \cap B) + \hat{v}(A \cup B)$. $\forall i \in \bar{P}, (\bar{P}_i \cap A = \emptyset) \wedge (\bar{P}_i \cap B = \emptyset) \Rightarrow N_i \cap (A \cup B) = \emptyset$ and $(N_i \cap S = \emptyset) \vee (N_i \cap B = \emptyset) \Rightarrow N_i \cap (A \cap B) = \emptyset$ (no negative v_i is involved since $A \cap B \neq \emptyset$). By adding these two inequalities we obtain $v(A) + v(B) \leq v(A \cup B) + v(A \cap B)$ which concludes the proof. $\square$

REFERENCES

[1] K. Aadithya, T. Michalak, and N. Jennings. Representation of coalitional games with algebraic decision diagrams. In *Proc. of the 10th AAMAS*, pages 1121–1122, 2011.

[2] R. Aumann and J. Dreze. Cooperative games with coalition structures. *International Journal of game theory*, 3(4):217–237, 1974.

[3] H. Aziz and B. de Keijzer. Complexity of coalition structure generation. In *Proc. of the 10th AAMAS*, pages 191–198, 2011.

[4] Y. Bachrach, P. Kohli, V. Kolmogorov, and M. Zadimoghaddam. Optimal coalition structure generation in cooperative graph games. In *Proc. of the 27th AAAI*, 2013.

[5] Y. Bachrach, O. Lev, S. Lovett, J. Rosenschein, and M. Zadimoghaddam. Cooperative weakest link games. In *Proc. of the 13th AAMAS*, pages 589–596, 2014.

[6] Y. Bachrach, D. Parkes, and J. Rosenschein. Computing cooperative solution concepts in coalitional skill games. *Artificial Intelligence*, 204:1–21, 2013.

[7] Y. Bachrach and E. Porat. Path disruption games. In *Proc. of the 9th AAMAS*, pages 1123–1130, 2010.

[8] Y. Bachrach, E. Porat, and J. Rosenschein. Sharing rewards in cooperative connectivity games. *Journal of Artificial Intelligence Research*, 47:281–311, 2013.

[9] Y. Bachrach, R. Savani, and N. Shah. Cooperative max games and agent failures. In *Proc. of the 13th AAMAS*, pages 29–36, 2014.

[10] G. Chalkiadakis, E. Elkind, and M. Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011.

[11] G. Chalkiadakis, G. Greco, and E. Markakis. Characteristic function games with restricted agent interactions: Core-stability and coalition structures. *Artificial Intelligence*, 232:76–113, 2016.

[12] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6):607–619, 2006.

[13] W. Cunningham. The optimal multiterminal cut problem. In *Reliability of Computer and Communication Networks*, pages 105–120, 1991.

[14] X. Deng and C. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

[15] E. Elkind, G. Chalkiadakis, and N. Jennings. Coalition structures in weighted voting games. In *Proc. of the 18th ECAI*, volume 8, pages 393–397, 2008.

[16] E. Elkind, L. A. Goldberg, P. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. *Mathematical Logic Quarterly*, 55(4):362–376, 2009.

[17] D. B. Gillies. Solutions to general non-zero-sum games. *Contributions to the Theory of Games*, 4(40):47–85, 1959.

[18] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of core, kernel, and bargaining set. *Artificial Intelligence*, 175(12):1877–1910, 2011.

[19] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of the core over coalition structures. In *Proc. of the 22th IJCAI*, pages 216–221, 2011.

[20] L. Guardiola, A. Meca, and J. Puerto. Production-inventory games: A new class of totally balanced combinatorial optimization games. *Games and Economic Behavior*, 65(1):205–219, 2009.

[21] R. Ichimura, T. Hasegawa, S. Ueda, A. Iwasaki, and M. Yokoo. Extension of mc-net-based coalition structure generation: handling negative rules and externalities. In *Proc. of the 10th AAMAS*, pages 1173–1174, 2011.

[22] S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *Proc. of the 6th EC*, pages 193–202, 2005.

[23] Y. Li and V. Conitzer. Complexity of stability-based solution concepts in multi-issue and mc-net cooperative games. In *Proc. of the 13th AAMAS*, pages 581–588, 2014.

[24] T. Michalak, T. Rahwan, E. Elkind, M. Wooldridge, and N. Jennings. A hybrid exact algorithm for complete set partitioning. *Artificial Intelligence*, 230:14–50, 2016.

[25] N. Ohta, V. Conitzer, R. Ichimura, Y. Sakurai, A. Iwasaki, and M. Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *Proc. of the 15th CP*, pages 623–638. 2009.

[26] J. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118:237–251, 2009.

[27] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1):209–238, 1999.

[28] G. Shafer. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.

[29] L. Shapley and M. Shubik. Competitive outcomes in the cores of market games. *International Journal of Game Theory*, 4(4):229–237, 1975.

[30] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1):165–200, 1998.

[31] J. Uckelman, Y. Chevaleyre, U. Endriss, and J. Lang. Representing utility functions via weighted goals. *Mathematical Logic Quarterly*, 55(4):341–361, 2009.

[32] S. Ueda, T. Hasegawa, N. Hashimoto, N. Ohta, A. Iwasaki, and M. Yokoo. Handling negative value rules in mc-net-based coalition structure generation. In *Proc. of the 11th AAMAS*, pages 795–804, 2012.

[33] S. Ueda, A. Iwasaki, M. Yokoo, M. Silaghi, K. Hirayama, and T. Matsui. Coalition structure generation based on distributed constraint optimization. In *Proc. of the 24th AAAI*, pages 197–203, 2010.

[34] S. Ueda, M. Kitaki, A. Iwasaki, and M. Yokoo. Concise characteristic function representations in coalitional games based on agent types. In *Proc. of the 22th IJCAI*, pages 393–399, 2011.