Coalition Structure Generation Utilizing Compact Characteristic Function Representations

Naoki Ohta[1], Vincent Conitzer[2], Ryo Ichimura[1], Yuko Sakurai[1], Atsushi Iwasaki[1], and Makoto Yokoo[1]

[1] Department of ISEE, Kyushu University, Fukuoka 819-0395, Japan,
{ohta@agent, ichimura@agent, sakurai@agent, iwasaki@, yokoo@}is.kyushu-u.ac.jp

[2] Department of Computer Science, Duke University, Durham, NC 27708, USA,
conitzer@cs.duke.edu

Abstract. This paper presents a new way of formalizing the Coalition Structure Generation problem (CSG), so that we can apply constraint optimization techniques to it. Forming effective coalitions is a major research challenge in AI and multi-agent systems. CSG involves partitioning a set of agents into coalitions so that social surplus is maximized. Traditionally, the input of the CSG problem is a black-box function called a *characteristic function*, which takes a coalition as an input and returns the value of the coalition. As a result, applying constraint optimization techniques to this problem has been infeasible. However, characteristic functions that appear in practice often can be represented concisely by a set of rules, rather than a single black-box function. Then, we can solve the CSG problem more efficiently by applying constraint optimization techniques to the compact representation directly.

We present new formalizations of the CSG problem by utilizing recently developed compact representation schemes for characteristic functions. We first characterize the complexity of the CSG under these representation schemes. In this context, the complexity is driven more by the number of rules rather than by the number of agents. Furthermore, as an initial step towards developing efficient constraint optimization algorithms for solving the CSG problem, we develop mixed integer programming formulations and show that an off-the-shelf optimization package can perform reasonably well, i.e., it can solve instances with a few hundred agents, while the state-of-the-art algorithm (which does not make use of compact representations) can solve instances with up to 27 agents.

Keywords: multiagent systems, coalition structure generation, constraint optimization

1 Introduction

Coalition formation is an important capability in automated negotiation among self-interested agents. Coalition structure generation (CSG) involves partitioning a set of agents into coalitions so that social surplus is maximized. This problem has become a popular research topic in AI and multi-agent systems. Possible applications of CSG include distributed vehicle routing (Sandholm and Lesser, 1997), multi-sensor networks (Dang et al., 2006), etc. The CSG problem is equivalent to a *complete set partition problem* (Yeh, 1986), and various algorithms for solving the CSG problem have been developed. Sandholm et al. (1999) propose an anytime algorithm with worst-case guarantees. However, to obtain an optimal coalition structure, this algorithm must check all coalition structures. Thus, the worst-case time complexity is $O(n^n)$, where n is the number of agents. On the other hand, Dynamic Programming (DP) based algorithms (Yeh, 1986; Rothkopf et al., 1998; Rahwan and Jennings, 2008b) are guaranteed to find an optimal solution in $O(3^n)$. Shehory and Kraus (1998) propose a greedy algorithm that puts constraints on the possible size of the coalitions.

Arguably, the state-of-the-art algorithm is the IP (integer partition) algorithm (Rahwan et al., 2007). This is an anytime algorithm, which divides the search space into partitions based on integer partition, and performs branch & bound search. Although the worst-case time complexity for obtaining an optimal solution is $O(n^n)$, in practice, IP is much faster than DP based algorithms. Furthermore, Rahwan and Jennings (2008a) introduce an extension of the IP algorithm that utilizes DP for preprocessing.

As far as we are aware, all existing works on CSG assume that the characteristic function is represented implicitly, and we have oracle access to the function—that is, the value of a coalition (or a coalition structure as a whole) can be obtained using some procedure. This is because representing an *arbitrary* characteristic function explicitly requires $\Theta(2^n)$ numbers, which is prohibitive for large n. When a characteristic function is represented by a black-box function, there is no room for applying constraint optimization techniques. Thus, this problem has been irrelevant to the CP community.

However, characteristic functions that appear in practice often display significant structure, and it is likely that such characteristic functions can be represented much more concisely. Indeed, recently, several new methods for representing characteristic functions have been developed (Ieong and Shoham, 2005; Conitzer and Sandholm, 2004, 2006). These representation schemes capture characteristics of interactions among agents in a natural and concise manner, and can reduce the representation size significantly. Surprisingly, to our knowledge, these representation schemes have not yet been used for CSG; this is what we set out to do in this paper. Using these compact representation schemes, a characteristic function is represented by a set of rules, rather than a single black-box function. It is likely that we can solve the CSG problem more efficiently by applying constraint optimization techniques to the compact representation directly.

We examine three representative compact representation schemes: (i) marginal contribution nets (MC-nets) (Ieong and Shoham, 2005), (ii) synergy coalition groups (SCGs) (Conitzer and Sandholm, 2006), and (iii) SCGs in multi-issue domains (Conitzer and Sandholm, 2004). The optimal choice of a representation scheme depends on the application.

There exist several other compact representation schemes, e.g., logic-based approaches (Wooldridge and Dunne, 2004, 2006) and skill-based approaches (Yokoo et al., 2005; Bachrach and Rosenschein, 2008). In this paper, we restrict our attention to the schemes mentioned earlier, since they are more closely related to the traditional CSG problem.

Quite interestingly, we find that there exists some common structure among these cases: in essence, the problem is to find a subset of rules that maximizes the sum of rule values under certain constraints. For each case, we show that solving the CSG problem is NP-hard, and the size of a problem instance is naturally measured by the number of rules rather than the number of agents.

Furthermore, as an initial step towards developing efficient constraint optimization algorithms for solving the CSG problem, we give a mixed integer programming (MIP) formulation that captures the above mentioned structure. We show that an off-the-shelf optimization package (CPLEX) can solve the resulting MIP problem instances reasonably well, i.e., it can solve instances with a few hundred agents, while the state-of-the-art algorithm (which does not make use of compact representations) can solve instances up to 27 agents.

The rest of this paper is organized as follows. First, we review the model of coalition structure generation (Section 2). Next, we introduce solution algorithms when the characteristic function is represented by MC-nets (Section 3), SCGs (Section 4) and SCGs in multi-issue domains (Section 5). Finally, we show the evaluation results and discussions (Section 6).

2 Model

Let $A = \{1, 2, \ldots, n\}$ be the set of agents. We assume a characteristic function game, i.e., the value of a coalition S is given by a characteristic function v. A characteristic function $v : 2^A \to \Re$ assigns a value to each set of agents (coalition) $S \subseteq A$. We assume that each coalition's value is nonnegative. This is not an unreasonable assumption (Sandholm et al., 1999); even if some coalition's values are negative, as long as each coalition's value is bounded (i.e., not infinitely negative), we can normalize the coalition values so that all values are non-negative. This rescaled game is strategically equivalent to the original game.

A coalition structure CS is a partition of A, into disjoint, exhaustive coalitions. To be more precise, $CS = \{S_1, S_2, \ldots\}$ satisfies the following conditions:

$$\forall i, j \ (i \neq j), \ S_i \cap S_j = \emptyset, \bigcup_{S_i \in CS} S_i = A.$$

In other words, in CS, each agent belongs to exactly one coalition, and some agents may be alone in their coalitions.

For example, in a game with three agents a, b, and c, there are seven possible coalitions: $\{a\}$, $\{b\}$, $\{c\}$, $\{a, b\}$,$\{b, c\}$, $\{a, c\}$, $\{a, b, c\}$, and five possible coalition structures: $\{\{a\}, \{b\}, \{c\}\}$, $\{\{a, b\}, \{c\}\}$, $\{\{a\}, \{b, c\}\}$, $\{\{b\}, \{a, c\}\}$, $\{\{a, b, c\}\}$.

The value of a coalition structure CS, denoted as $V(CS)$, is given by:

$$V(CS) = \sum_{S_i \in CS} v(S_i).$$

An optimal coalition structure CS^* is a coalition structure that satisfies the following condition:

$$\forall CS, V(CS^*) \geq V(CS).$$

We say a characteristic function is super-additive, if for any disjoint sets S_i, S_j, $v(S_i \cup S_j) \geq v(S_i) + v(S_j)$ holds. If the characteristic function is super-additive, solving CSG becomes trivial, i.e., the grand coalition (the coalition of all agents) is optimal.

Super-additivity means that any pair of coalitions is better off by merging into one. One might think that super-additivity holds in most of the cases since the agents in the composite coalition can work separately and perform at least as well as the case that they were in different coalitions. However, organizing a large coalition can be costly, e.g., there might be coordination overhead like communication costs, or possible anti-trust penalties. Also, if time is limited, the agents may not have time to carry out the communications and computations required to coordinate effectively within the composite coalition, so component coalitions may be more advantageous. Thus, we assume a characteristic function can be non-super-additive.

3 CSG using MC-nets

Ieong and Shoham (2005) develop a concise representation of a characteristic function called *marginal contribution networks (MC-nets)*.

Definition 1 (MC-nets). *An* MC-net *consists of a set of* rules R. *Each rule* $r \in R$ *is of the form:* $(P_r, N_r) \to v_r$, *where* $P_r \subseteq A$, $N_r \subseteq A$, $P_r \cap N_r = \emptyset$, $v_r \in \Re$. *We say that rule* r *is* applicable *to coalition* S *if* $P_r \subseteq S$ *and* $N_r \cap S = \emptyset$, *i.e.,* S *contains all agents in* P_r *(positive literals), and it contains no agent in* N_r *(negative literals). For a coalition* S, $v(S)$ *is given as* $\sum_{r \in R_S} v_r$, *where* R_S *is the set of rules applicable to* S. *Thus, for a coalition structure* CS, $V(CS)$ *is given as* $\sum_{S \in CS} \sum_{r \in R_S} v_r$.

Example 1. Let there be five agents a, b, c, d, e and four rules: $r_1 : (\{b, e\}, \{\}) \to 3$, $r_2 : (\{a, b, c\}, \{d\}) \to 2$, $r_3 : (\{a, d\}, \{\}) \to 1$, and $r_4 : (\{c\}, \{e\}) \to 1$. In this case, r_1 and r_2 are applicable to coalition $\{a, b, c, e\}$, but r_3 and r_4 are not. Thus, $v(\{a, b, c, e\})$ is equal to $3 + 2 = 5$.

In the original definition from (Ieong and Shoham, 2005), a rule may have a negative value. In this paper, we assume all rules have positive values. Furthermore, we assume each rule has at least one positive literal. Under these restrictions, we can guarantee that having more applicable rules never hurts, and each rule is applicable to only one coalition. Even under these restrictions, MC-nets can represent any characteristic function. This is because, in the worst case, for each coalition $S \subseteq A$, we can create a rule $(S, A \setminus S) \rightarrow v(S)$, i.e., each rule is applicable only to S.

Definition 2 (Feasible rule set). *We say a set of rules $R' \subseteq R$ is* feasible *if there exists CS where each rule $r \in R'$ is applicable to some $S \in CS$.*

In Example 1, $\{r_2, r_4\}$ is feasible because each rule is applicable to $CS = \{\{a, b, c\}, \{d, e\}\}$. On the other hand, $\{r_1, r_2, r_4\}$ and $\{r_2, r_3\}$ are infeasible. The problem of finding CS^* is equivalent to finding a feasible rule set R', so that $\sum_{r \in R'} v_r$ is maximized.

Definition 3 (Relations between rules). *The possible relations between two rules r and r' can be classified into the following four nonoverlapping and exhaustive cases:*

Compatible on the same coalition: *$P_r \cap P_{r'} \neq \emptyset$ and $P_r \cap N_{r'} = P_{r'} \cap N_r = \emptyset$. For example, in Example 1, r_1 and r_2 are compatible on the same coalition: if r_1 and r_2 are applicable at the same time, there must be a coalition S with $S \supseteq \{a, b, c, e\}$ and $d \notin S$.*

Incompatible: *$P_r \cap P_{r'} \neq \emptyset$, and ($P_r \cap N_{r'} \neq \emptyset$ or $P_{r'} \cap N_r \neq \emptyset$). For example, r_2 and r_3 are incompatible: these two rules are not applicable at the same time.*

Compatible on different coalitions: *$P_r \cap P_{r'} = \emptyset$, and ($P_r \cap N_{r'} \neq \emptyset$ or $P_{r'} \cap N_r \neq \emptyset$). For example, r_1 and r_4 are compatible on different coalitions: if r_1 and r_4 are applicable at the same time, there must be two different coalitions S_1 and S_2, where $S_1 \supseteq \{b, e\}$ and $S_2 \supseteq \{c\}$.*

Independent: *$P_r \cap P_{r'} = \emptyset$, and $P_r \cap N_{r'} = P_{r'} \cap N_r = \emptyset$. For example, r_1 and r_3 are independent. These two rules can be applied to the same coalition or to different coalitions.*

Let us consider a graphical representation of an MC-net in which each vertex is a rule, and between any two vertices, there exists an edge whose type is one of the four cases described above. Figure 1 shows the graphical representation of Example 1 ("independent" edges are not shown).

The following conditions characterize whether a rule set is feasible.

Theorem 1. *A set of rules R' is feasible if and only if it satisfies the following conditions.*

(a) R' includes no pair of rules/vertices connected by an "incompatible" edge, and
(b) if two rules/vertices in R' are connected by a "compatible on different coalitions" edge, then they are not reachable via "compatible on the same coalition" edges within R'.

Proof. First, we prove the "if" part. From (a), there exists no incompatible edge within R'. From (b), R' can be divided into groups $G_1, G_2, \ldots, G_k$ where the rules within G_i are reachable from each other by "compatible on the same coalition"

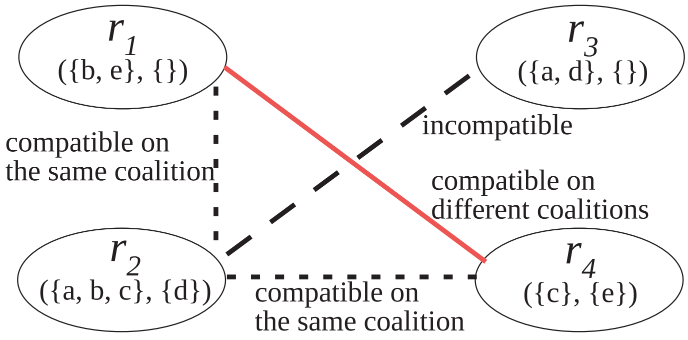

Fig. 1. Graphical representation of Example 1.

edges, there exists no "compatible on different coalitions" edge between rules in G_i, and there exists no "compatible on the same coalition" edge between rules that belong to different groups.

Let us choose $CS = \{S_1, S_2, \ldots, S_k\}$ so that S_i is the union of all positive literals of $r \in G_i$. Then, for $i \neq j$, $S_i \cap S_j = \emptyset$ holds. This is because $S_i \cap S_j \neq \emptyset$ would imply that there exists at least one pair $r \in G_i, r' \in G_j$ for which r and r' are connected by a "compatible on the same coalition" edge (since there cannot be an "incompatible" edge between them)—but this is in contradiction with the way in which $G_1, \ldots, G_k$ are chosen. Thus, $\{S_1, \ldots, S_k\}$ is a valid coalition structure.[3]

Now, we show that for any $r \in G_i$, r is applicable to coalition S_i. Clearly, S_i contains all the positive literals of r. It remains to show that S_i does not contain any negative literal of r. For the sake of contradiction, assume S_i contains agent a, where a is a negative literal of r. Then, there exists another rule $r' \in G_i$ for which a is a positive literal. There must be a "compatible on different coalitions" or an "incompatible" edge between r and r'. Either case leads to a contradiction. Hence, R' is feasible.

Next, we prove the "only if" part. We show that if R' does not satisfy the above conditions, then there exists no coalition structure where R' is applicable. Clearly, if (a) is not satisfied, i.e., some $r, r' \in R'$ are connected by an "incompatible" edge, then there exists no coalition structure where r and r' are applicable at the same time.

Now, let us assume (b) is not satisfied, i.e., there exist $r_i, r_j \in R'$ such that r_i and r_j are connected by a "compatible on different coalitions" edge, and they are reachable by "compatible on the same coalition" edges within R'. Assume r_i is applicable to coalition S_i and r_j is applicable to coalition S_j. Since r_i and r_j are connected by a "compatible on different coalitions" edge, S_i and S_j must be different. However, S_i must contain all positive literals of rules reachable from r_i

[3] If some agent is not included in any S_i, we can assume the agent forms its own coalition.

via "compatible on the same coalition" edges: otherwise, some rule in R' is not applicable. Similarly, S_j must contain all positive literals of rules reachable from r_j via "compatible on the same coalition" edges. Since r_i and r_j are reachable from each other via "compatible on the same coalition" edges, S_i and S_j must be the same—but this contradicts the fact that they must be different. □

Theorem 2. *When the characteristic function is represented as an MC-net, finding an optimal coalition structure is NP-hard. Moreover, unless $\mathcal{P} = \mathcal{NP}$, there exists no polynomial-time $O(|R|^{1-\epsilon})$ approximation algorithm for any $\epsilon > 0$, where $|R|$ is the number of rules.*

Proof. The maximum independent set problem is to choose $V' \subseteq V$ for a graph $G = (V, E)$ such that there exists no edge between vertices in V', and $|V'|$ is maximized under this constraint. It is NP-hard and, unless $\mathcal{P} = \mathcal{NP}$, there exists no polynomial-time $O(|V|^{1-\epsilon})$ approximation algorithm for any $\epsilon > 0$ (Håstad, 1999; Zuckerman, 2007). We reduce an arbitrary maximal independent set instance to a CSG problem instance, as follows. For each $v \in V$, let there be an agent a_v; also, for each $e \in E$, let there be an agent a_e. For each $v \in V$, we create a rule r_v where $P_{r_v} = \{a_v\} \cup \{a_e : v \in e\}$, $N_{r_v} = \{a_w : (v, w) \in E\}$, and $v_{r_v} = 1$. Thus, rules are "incompatible" if they correspond to neighboring vertices, and "independent" otherwise. It follows that feasible rule sets correspond exactly to independent sets of vertices. □

The reduction in Theorem 2 relies heavily on "incompatibilities" between rules. If there are no "incompatibilities" then the problem is equivalent to the multi-cut problem (Vazirani, 2001), which is a generalization of the min-cut problem.

Definition 4 (MIP formulation of CSG for MC-nets). *The problem of finding a feasible rule set R' that maximizes $\sum_{r \in R'} v_r$ can be modeled as follows.*

$$
\begin{aligned}
&\mathit{max} \ \ \textstyle\sum_{r \in R} v_r \cdot x(r) \\
&\quad s.t. \forall e = (r, r'), \textit{ where } e \textit{ is an "incompatible" edge,} \\
&\qquad\quad x(r) + x(r') \le 1, \ \text{— (i)} \\
&\qquad \forall e = (r_i, r_j), \textit{ where } e \textit{ is} \\
&\qquad\quad \textit{a "compatible on different coalitions" edge and } i < j, \\
&\qquad\quad dis(e, r_i) = 0,\ dis(e, r_j) \ge 1, \ \text{— (ii)} \\
&\qquad\quad \forall e' = (r_1, r_2), \textit{ where } e' \textit{ is} \\
&\qquad\qquad \textit{a "compatible on the same coalition" edge,} \\
&\qquad\qquad dis(e, r_1) \le dis(e, r_2) + (1 - x(r_1)) + (1 - x(r_2)), \ \text{— (iii)} \\
&\qquad\qquad dis(e, r_2) \le dis(e, r_1) + (1 - x(r_1)) + (1 - x(r_2)), \ \text{— (iv)} \\
&\qquad \forall r \in R,\ x(r) \in \{0, 1\}.
\end{aligned}
$$

$x(r) = 1$ means that rule r is selected. The constraint (i) ensures that two rules connected by an "incompatible" edge will not be selected at the same time. Also, for each "compatible on different coalitions" edge $e = (r_i, r_j)$, we define a distance/potential for e, so that $dis(e, r_i) = 0$ and $dis(e, r_j) \ge 1$ (ii). The constraints (iii) and (iv) ensure that if both of r_1 and r_2 are selected, where r_1

and r_2 are connected by a "compatible on the same coalition" edge, then the distance/potential of these two rules for the aforementioned e must be equal. Then, the facts that $dis(e, r_i) = 0$ and $dis(e, r_j) \geq 1$ ensure that r_i and r_j are not reachable from each other via "compatible on the same coalition" edges. Using such a distance/potential is a standard method for representing connectivity constraints in MIP formalization without enumerating possible paths.

In this formulation, the number of binary variables is equal to the number of rules. The number of constraints is $d_{in} + d_{cd}(2d_{cs} + 1)$, where d_{in}, d_{cd}, d_{cs} are the number of edges with types "incompatible", "compatible on different coalitions", and "compatible on the same coalition", respectively.

4 CSG using Synergy Coalition Groups

Conitzer and Sandholm (2006) introduce a concise representation of a characteristic function called a *synergy coalition group (SCG)*. The main idea is to explicitly represent the value of a coalition only when there exists some *positive* synergy.

Definition 5 (SCG). *An SCG consists of a set of pairs of the form:* $(S, v(S))$. *For any coalition* S*, the value of the characteristic function is:*

$$v(S) = \max\{ \sum_{S_i \in p_S} v(S_i)\},$$

where p_S *is a partition of* S*, i.e., all the* S_i *are disjoint and* $\bigcup_{S_i \in p_S} S_i = S$*, and for all the* S_i*,* $(S_i, v(S_i)) \in SCG$*. To avoid senseless cases that have no feasible partitions, we require that* $(\{a\}, 0) \in SCG$ *whenever* $\{a\}$ *does not receive a value elsewhere in* SCG*.*

Thus, if the value of a coalition S is not given explicitly in SCG, it is calculated from the possible partitions of S. Using this original definition, we can represent only super-additive characteristic functions, i.e., for any disjoint sets S_i, S_j, $v(S_i \cup S_j) \geq v(S_i) + v(S_j)$ holds. But, as mentioned in Section 2, if the characteristic function is super-additive, solving CSG becomes trivial: the grand coalition is optimal. To allow for characteristic functions that are not super-additive, we add the following requirement on the partition p_S.

- $\forall p'_S \subseteq p_S$, where $|p'_S| \geq 2$, $(\bigcup_{S_i \in p'_S} S_i, v(\bigcup_{S_i \in p'_S} S_i))$ is not an element of SCG.

This additional condition requires that if the value of a coalition is explicitly given in SCG, then we cannot further divide it into smaller subcoalitions to calculate values. In this way, we can represent *negative* synergies.

Example 2. Let there be five agents a, b, c, d, e and let $SCG = \{(\{a\}, 0), (\{b\}, 0), (\{c\}, 1), (\{d\}, 2), (\{e\}, 3), (\{a, b\}, 3), (\{a, b, c\}, 3)\}$. In this case, $v(\{d, e\}) = v(\{d\}) + v(\{e\}) = 5$, and $v(\{a, b, c, d, e\}) = v(\{a, b, c\}) + v(\{d\}) + v(\{e\}) = 8$. For $v(\{a, b, c, d, e\})$, we cannot use $v(\{a, b\}) + v(\{c\}) + v(\{d\}) + v(\{e\}) = 9$, because $\{a, b\} \cup \{c\} = \{a, b, c\}$ appears in SCG.

The (modified) SCG can represent any characteristic function, including characteristic functions that are non-super-additive, or even non-monotone. This is because in the worst case, we can explicitly give the value of every coalition. Due to the additional condition, only these explicit values can then be used to calculate the characteristic function.

We show that when searching for CS^*, we need to consider only the coalitions that are explicitly described in SCG.

Theorem 3. *There exists a coalition structure* CS *for which* $V(CS) = V(CS^*)$ *and* $\forall S \in CS, (S, v(S)) \in SCG$.

Proof. For the sake of contradiction, let us assume there exists some CS^* so that $V(CS^*)$ is strictly larger than any CS that consists of only elements of SCG. Let us examine some coalition $S \in CS^*$ that is not an element of SCG. From the definition of SCG, there exists a partition of S (denoted as p_S) such that $v(S) = \sum_{S_i \in p_S} v(S_i)$, and each S_i is an element of SCG. Then, by replacing each such S by p_S, we obtain a new coalition structure CS that consists of only elements of SCG, and $V(CS) = V(CS^*)$ holds—so we have the desired contradiction. □

Due to Theorem 3, finding CS^* is equivalent to a weighted set packing problem—equivalently, to the winner determination problem in combinatorial auctions (Sandholm, 2002), where each agent is an item and each coalition described in SCG is a bid.

Theorem 4. *When the characteristic function is represented as an SCG, finding an optimal coalition structure is NP-hard. Moreover, unless* $\mathcal{P} = \mathcal{NP}$*, there exists no polynomial-time* $O(|SCG|^{1-\epsilon})$ *approximation algorithm for any* $\epsilon > 0$.

Proof. This follows directly from the corresponding inapproximability for the winner determination problem (Sandholm, 2002) and the maximum independent set problem (Zuckerman, 2007).

Definition 6 (MIP formulation of CSG for SCG). *The problem of finding* CS^* *can be modeled as follows.*

$$
\begin{aligned}
max \quad & \sum_{(S,v(S)) \in SCG} v(S) \cdot x(S) \\
s.t. \quad & \forall a \in A, \sum_{S \ni a} x(S) = 1, \\
& x(S) \in \{0, 1\}.
\end{aligned}
$$

$x(S)$ *is 1 if* S *is included in* CS^**, 0 otherwise.*

In this formulation (which corresponds to a standard winner determination formulation), the number of binary variables is equal to $|SCG|$, and the number of constraints is equal to the number of agents.

5 CSG in Multi-issue Domain

Conitzer and Sandholm (2004) introduce the concept of a *multi-issue domain*. In a multi-issue domain, there are k independent issues. The overall value of a coalition is the sum of the values of the coalition for the individual issues. More specifically, we assume there are k characteristic functions $v_1, v_2, \ldots, v_k$ such that for any $S \subseteq A$, $v(S) = \sum_{i=1}^{k} v_i(S)$. If each v_i can be represented concisely, then this leads to a concise representation for v. In this paper, we assume that v_i is represented by SCG_i.

Definition 7 (SCGs in multi-issue domains). *We represent the characteristic function by a vector of SCGs* $(SCG_1, \ldots, SCG_k)$. *For any* $S \subseteq A$, $v(S) = \sum_{i=1}^{k} v_i(S)$, *where* v_i *is calculated using* SCG_i. *Also, for a coalition structure* CS, *we denote* $V_i(CS) = \sum_{S \in CS} v_i(S)$. *Thus,* $V(CS) = \sum_{i=1}^{k} V_i(CS)$.

Example 3. Let there be four agents a, b, c, d and two $SCGs$: $SCG_1 = \{(\{a\}, 0), (\{b\}, 0), (\{c\}, 1), (\{d\}, 0), (\{a, b\}, 2), (\{a, b, c\}, 2)\}$, $SCG_2 = \{(\{a\}, 0), (\{b\}, 0), (\{c\}, 0), (\{d\}, 1), (\{a, b, c\}, 2)\}$.

In this case, $v(\{a, b, c\})$ is $v_1(\{a, b, c\}) + v_2(\{a, b, c\}) = 2 + 2 = 4$.

When there are multiple issues, an optimal coalition structure CS^* may need to contain a coalition S that is not explicitly described in any SCG_i. For example, assume that in issue i, a and b have a strong positive synergy. Also, in issue j, b and c have a strong positive synergy. Then, coalition $\{a, b, c\}$ may need to be included in CS^*, even though $\{a, b, c\}$ appears in neither SCG_i nor SCG_j.

Definition 8 (Value-producing subset). *Given a coalition structure* CS, *we say that* SCG'_i *(where* $SCG'_i \subseteq SCG_i$*) is a* value-producing subset *of* SCG_i *for* CS, *if* SCG'_i *consists exactly of elements of* SCG_i *that are used to calculate* $V_i(CS)$. *Thus,* $V_i(CS) = \sum_{(S, v_i(S)) \in SCG'_i} v_i(S)$.

In Example 3, $SCG'_1 = \{(\{a, b, c\}, 2), (\{d\}, 0)\}$ and $SCG'_2 = \{(\{a, b, c\}, 2), (\{d\}, 1)\}$ are value-producing subsets for $CS = \{\{a, b, c\}, \{d\}\}$. From this definition, a value-producing subset SCG'_i must contain all agents, and elements of SCG'_i must be disjoint. We call a subset that satisfies these conditions a *valid subset.*

Definition 9 (Valid subset). $SCG'_i \subseteq SCG_i$ *is a valid subset if* $\bigcup_{(S, v_i(S)) \in SCG'_i} S = A$, *and* $\forall (S, v_i(S))$, $(S', v_i(S')) \in SCG'_i$ *where* $S \neq S'$, $S \cap S' = \emptyset$ *holds.*

Theorem 5. *A valid subset* $SCG'_i \subseteq SCG_i$ *is a value-producing subset of* SCG_i *for* CS *if and only if for each* $S \in CS$, *either one of the following conditions holds:*

1. $(S, v_i(S)) \in SCG'_i$,
2. $\exists p_S$, *where* p_S *is a partition of* S, *such that* $|p_S| \geq 2$, $\forall S' \in p_S, (S', v_i(S')) \in SCG'_i$, *and* $\forall p'_S \subseteq p_S$, *where* $|p'_S| \geq 2$, $(\bigcup_{S'' \in p'_S} S'', v_i(\bigcup_{S'' \in p'_S} S'')) \notin SCG_i$.

We omit the proof since it is straightforward from the (modified) definition of the SCG representation. Quite interestingly, we can define the possible relations between elements in SCGs in the same way as we did for MC-nets.

Definition 10 (Relations between coalitions). *The possible relations between two coalitions $(S, v_i(S)) \in SCG_i$ and $(S', v_j(S')) \in SCG_j$ can be classified into the following four cases, which are nonoverlapping and exhaustive:*

Compatible on the same coalition: *$i \neq j$ and $S \cap S' \neq \emptyset$. For example, in Example 3, $(\{a, b\}, 2) \in SCG_1$ and $(\{a, b, c\}, 2) \in SCG_2$ are compatible on the same coalition. If these two elements are a part of value-producing subsets at the same time, there must be a coalition S with $S \supseteq \{a, b, c\}$.*

Incompatible: *$i = j$ and $S \cap S' \neq \emptyset$. For example, $(\{a, b\}, 2) \in SCG_1$ and $(\{a, b, c\}, 2) \in SCG_1$ are incompatible. They cannot be used simultaneously.*

Compatible on different coalitions: *$i = j$, and there exists $(S \cup S', v_i(S \cup S')) \in SCG_i$. For example, $(\{a, b\}, 2) \in SCG_1$ and $(\{c\}, 1) \in SCG_1$ are compatible on different coalitions. If these two elements are included in value-producing subsets at the same time, there must be two coalitions S_1, S_2, where $S_1 \supseteq \{a, b\}$ and $S_2 \supseteq \{c\}$, since if there exists $S \supseteq \{a, b, c\}$, then, we need to use $(\{a, b, c\}, 2)$ for calculating v_1. To be more precise, this relation must be extended to a hyper-edge. If there exists $(S, v_i(S)) \in SCG_i$, such that $\forall S' \in p_S, (S', v_i(S')) \in SCG_i$ holds, where p_S is a partition of S, then, we create a hyper-edge connecting the elements in p_S. Note that we need to add hyper-edges only for sub-additive cases.*

Independent: *otherwise. For example, $(\{a, b\}, 2) \in SCG_1$ and $(\{d\}, 0) \in SCG_1$ are independent. They can be used in both cases.*

The following conditions characterize whether coalitions are value-producing.

Theorem 6. *$(SCG'_1, \ldots, SCG'_k)$, where each SCG'_i is a valid subset of SCG_i, is a vector of value-producing subsets for some CS if and only if the following conditions hold:*

(a) $(SCG'_1, \ldots, SCG'_k)$ include no pair of coalitions connected by an "incompatible" edge, and
(b) if a set of coalitions in $(SCG'_1, \ldots, SCG'_k)$ is connected by a "compatible on different coalitions" hyper-edge, then there exists at least one element that is not reachable from other elements via "compatible on the same coalition" edges.

We omit the proof since it is basically the same as that of Theorem 1.

Definition 11 (MIP formulation in multi-issue domains). *The problem of finding value-producing subsets that maximize the summation of values can be modeled as follows.*

$max \ \sum_{p=(S,v_*(S)) \in \bigcup_{i=1}^{k} SCG_i} v_*(S) \cdot x(p)$
$s.t. \forall e = (p, p')$, where e is an "incompatible" edge,

$$
\begin{array}{l}
x(p) + x(p') \le 1, \\
\forall e = (p_1, p_2, \ldots, p_l),\ \textit{where } e \textit{ is} \\
\quad \textit{a "compatible on different coalitions" hyper-edge,} \\
\quad dis(e, p_1) = 0,\ dis(e, p_2) + \ldots + dis(e, p_l) \ge 1, \text{ — } (i) \\
\quad \forall e' = (p_i, p_j),\ \textit{where } e' \textit{ is} \\
\quad\quad \textit{a "compatible on the same coalition" edge,} \\
\quad dis(e, p_i) \le dis(e, p_j) + (1 - x(p_i)) + (1 - x(p_j)), \\
\quad dis(e, p_j) \le dis(e, p_i) + (1 - x(p_i)) + (1 - x(p_j)), \\
\forall p \in \bigcup_{i=1}^{k} SCG_i,\ x(p) \in \{0, 1\}.
\end{array}
$$

$x(p) = 1$ means the element p in $\bigcup_{i=1}^{k} SCG_i$ is selected. This formulation is basically the same as Definition 4, except for the constraint (i). This constraint means that for a hyper-edge e that connects nodes $p_1, p_2, \ldots, p_l$, at least one element must be unreachable. The number of variables and constraints are basically the same as MC-nets.

Theorem 7. *When the characteristic function is represented as SCGs in a multi-issue domain, finding an optimal coalition structure is NP-hard. Moreover, unless $\mathcal{P} = \mathcal{NP}$, there exists no polynomial-time $O(m^{1-\epsilon})$ approximation algorithm for any $\epsilon > 0$, where m is the number of elements in SCGs.*

Proof. We can use the same proof as Theorem 2. □

6 Evaluation and Discussion

We experimentally evaluated the performance of our proposed methods. All the tests were run on a Core 2 Duo E6850 3GHz processor with 8GB RAM. The test machine runs WindowsXP Professional x64 Edition SP2. We used CPLEX version 11.2, a general-purpose mixed integer programming package.

We show results for the following cases: (i) MC-nets, (ii) SCGs, and (iii) SCGs in multi-issue domains. The problem instances are generated slightly differently in each case. For case (ii), we use a decay distribution (Sandholm, 2002) described as follows. Create a coalition with one random agent. Then repeatedly add a new random agent with probability α until an agent is not added or the coalition includes all agents. Choose the value of the coalition between 0 and the number of agents in the coalition uniformly at random. We use $\alpha = 0.55$. For case (i), we first create a rule $(S, \{\}) \to v(S)$ for each SCG in case (ii). Then, we modify each rule by randomly moving an agent from the positive to the negative literals with probability p. We use $p = 0.2$. For case (iii), the way of generating problem instances is basically identical to case (ii), but we create five issues and each issue has the same number of rules. For each issue, we assume 30% of agents are involved.

In Figure 2 (a), we set #rules=#agents[4], and vary #agents from 10 to 120. In Figure 2 (b), (c), (d), we set #rules to 60, 90, and 120, respectively, and vary #agents. Each data point is the median of 50 problem instances.

[4] By #rules, we mean the number of elements in $SCG/SCGs$ in cases (ii) and (iii).

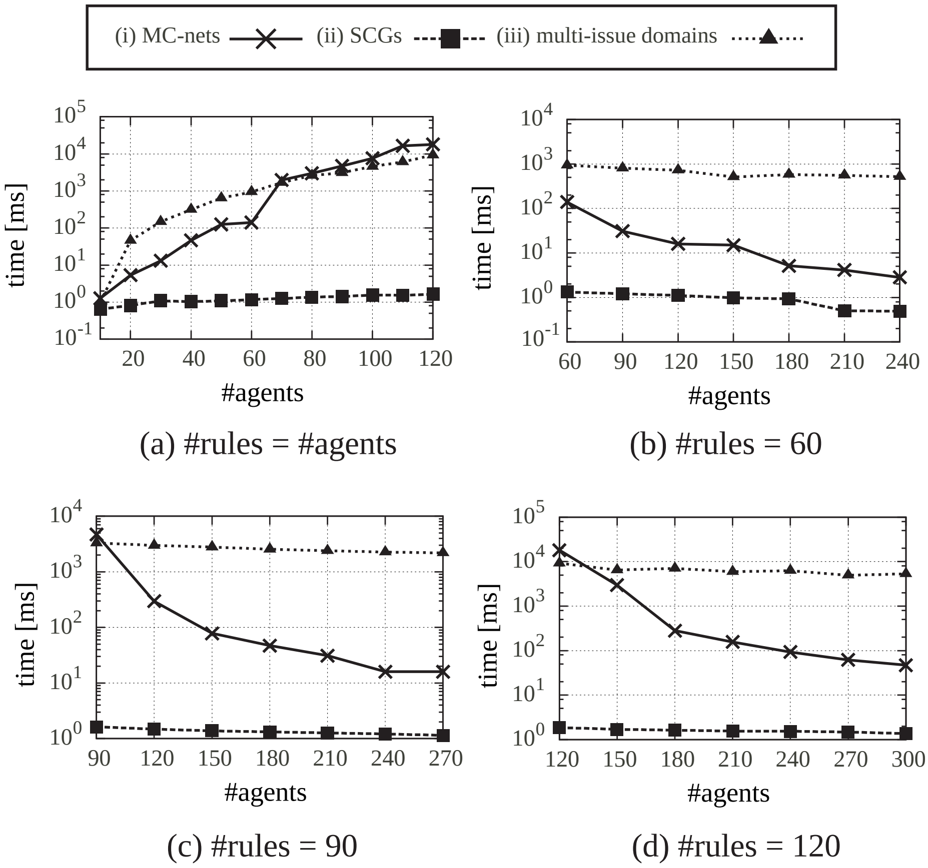

Fig. 2. Computation time for MC-nets, SCGs, SCGs in multi-issue domains

In Figure 2 (b), (c), (d), the CSG problem actually becomes easier when #agents increases. Since #rules, i.e., the number of vertices in the graph, is constant, the graph becomes more sparse by increasing #agents. As long as #rules is the same, case (ii) solves much faster than (i) and (iii). Also, as long as #rules is the same, case (i) and (iii) are about the same, except for the instances where #agents is large (Figure 2 (b), (c), (d)). This is because case (iii) has more constraints since it tends to have more "compatible on different coalitions" hyper-edges. We have tried several different settings and confirmed that the trends are basically similar.

The SCG representation has an advantage since the MIP formulation is simple and the resulting problem instances can be solved quite efficiently. Furthermore, we can leverage existing mature techniques for winner determination problems, including constraint-based approaches such as (Hoos and Boutilier, 2000). When using the MC-net or multi-issue representation, the limiting factor would be the number of edges (excluding "independent" edges) between rules, since we need to use auxiliary variables for representing connectivity constraints. However, in many cases, we can represent a characteristic function much more concisely by using MC-nets or multi-issue domains than by using SCGs.

As discussed in (Sandholm, 2002), specialized algorithms are usually more efficient than CPLEX for solving the winner determination problem. Thus, we can expect that specialized algorithms would be more efficient that CPLEX for solving the CSG problem. Here, we discuss several directions how specialized algorithms can be constructed. Certainly, we can use a depth-first branch and bound procedure. If we relax integer variables to continuous ones in the MIP formalization, we can obtain an admissible estimation. Of course, CPLEX performs a similar procedure, but we can use specialized graph-based heuristics for selecting nodes/rules. Furthermore, it would be possible that we have an efficient algorithm when a graph has some special structure (e.g., the graph is tree, or the graph can be divided into independent subgraphs by removing a small number of nodes). Also, if there exists no "incompatible" edge, then the problem is equivalent to the multi-cut problem (Vazirani, 2001). There exists an efficient approximation algorithm for the multi-cut problem (Vazirani, 2001). We can construct an algorithm that interleaves the selection among incompatible rules and the application of the approximate algorithm for the multi-cut problem.

Rahwan et al. (2007) reports that their IP algorithm can solve problem instances with 27 agents in less than 90 minutes. Also, they report that an extension of the IP algorithm that utilizes DP for preprocessing (Rahwan and Jennings, 2008a) can obtain four-fold speed-up compared to IP. We cannot directly compare our results with these results, since the formalizations of the CSG are different. Here, we are not comparing the efficiency of particular algorithms, but checking the scalability of different formalizations. Their algorithms inevitably evaluate all possible (2^n) coalitions. Thus, it is very unlikely that their approaches can scale up to $n = 100$. On the other hand, the advantage of these approaches is that they do not rely on particular representations.

7 Conclusion

We showed that coalition structure generation can scale up significantly when the characteristic function is represented using recently developed compact representation schemes: MC-nets, SCGs, and SCGs in multi-issue domains, even though we use an off-the-shelf optimization package. For each case, we proved that the problem is NP-hard and inapproximable and developed MIP formulations. Experimental results illustrated that while the state-of-the-art algorithm, which does not make use of compact representations, requires around 90 minutes to solve a problem with 27 agents, our methods can solve a problem with 120 agents and 120 rules in less than 20 seconds. Future work includes developing new algorithms (i) that can find an optimal solution more efficiently, (ii) that can return a suboptimal solution in any time, and (iii) that can find an approximate solution quickly, utilizing constraint optimization techniques.

Acknowledgments

The authors would like to thank Takayoshi Shoudai and Hirotaka Ono for their helpful comments on the earlier version of this paper. This work is partially supported by Japan Society for the Promotion of Science with Grant-in-Aid for Scientific Research (A) 20240015 and 20240003. Conitzer is supported by NSF award number IIS-0812113, a Research Fellowship from the Alfred P. Sloan Foundation, and a Yahoo! Faculty Research Grant.

Bibliography

Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 1023–1030, 2008.

V. Conitzer and T. Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, pages 219–225, 2004.

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

V. D. Dang, R. K. Dash, A. Rogers, and N. R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 635–640, 2006.

H. H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pages 22–29, 2000.

J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182: 105–142, 1999.

S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *Proceedings of the 6th ACM Conference on Electronic Commerce (ACM EC)*, pages 193–202, 2005.

T. Rahwan and N. R. Jennings. Coalition structure generation: dynamic programming meets anytime optimisation. In *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI)*, pages 156–161, 2008a.

T. Rahwan and N. R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *Proceedings of the 7th International joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 1417–1420, 2008b.

T. Rahwan, S. D. Ramchurn, V. D. Dang, A. Giovannucci, and N. R. Jennings. Anytime optimal coalition structure generation. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI)*, pages 1184–1190, 2007.

M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002.

T. Sandholm and V. R. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94(1-2):99–137, 1997.

T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111 (1-2):209–238, 1999.

O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, 1998.

V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

M. Wooldridge and P. E. Dunne. On the computational complexity of qualitative coalitional games. *Artificial Intelligence*, 158(1):27–73, 2004.

M. Wooldridge and P. E. Dunne. On the computational complexity of coalitional resource games. *Artificial Intelligence*, 170(10):835–871, 2006.

D. Y. Yeh. A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics*, 26(4):467–474, 1986.

M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, and A. Iwasaki. Coalitional games in open anonymous environments. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 509–515, 2005.

D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103–128, 2007.