Concise Characteristic Function Representations in Coalitional Games Based on Agent Types

Suguru Ueda, Makoto Kitaki, Atsushi Iwasaki, and Makoto Yokoo

Department of ISEE, Kyushu University

Fukuoka, Japan

{ueda@agent., kitaki@agent., iwasaki@, yokoo@}inf.kyushu-u.ac.jp

Abstract

Forming effective coalitions is a major research challenge in AI and multi-agent systems (MAS). Thus, coalitional games, including Coalition Structure Generation (CSG), have been attracting considerable attention from the AI research community. Traditionally, the input of a coalitional game is a black-box function called a characteristic function. A range of previous studies have found that many problems in coalitional games tend to be computationally intractable when the input is a black-box function. Recently, several concise representation schemes for a characteristic function have been proposed. Although these schemes are effective for reducing the representation size, most problems remain computationally intractable.

In this paper, we develop a new concise representation scheme based on the idea of *agent types*. Intuitively, a type represents a set of agents, which are recognized as having the same contribution. This representation can be exponentially more concise than existing concise representation schemes. Furthermore, this idea can be used in conjunction with existing schemes to further reduce the representation size. Moreover, we show that most of the problems in coalitional games, including CSG, can be solved in polynomial time in the number of agents, assuming the number of possible types is fixed.

1 Introduction

Forming effective coalitions is a major research challenge in AI and multi-agent systems (MAS). A coalition of agents can sometimes accomplish things that individual agents cannot or can do things more efficiently. There are two major research topics in coalitional games. The first topic involves partitioning a set of agents into coalitions so that the sum of the rewards of all coalitions is maximized. This problem is called the Coalition Structure Generation problem (CSG) [Sandholm *et al.*, 1999; Rahwan and Jennings, 2008]. The second topic involves how to divide the value of the coalition among agents. The theory of coalitional games provides a number of solution concepts, such as the core, the Shapley value, and the nucleolus.

A range of previous studies have found that many problems in coalitional games, including CSG, tend to be computationally intractable. Traditionally, the input of a coalitional game is a black-box function called a characteristic function, which takes a coalition as an input and returns the value of the coalition (or a coalition structure as a whole). Recently, several concise representation schemes for a characteristic function have been proposed, e.g., synergy coalition group (SCG) [Conitzer and Sandholm, 2006] and marginal contribution nets (MC-nets) [Ieong and Shoham, 2005]. These schemes represent a characteristic function as a set of rules rather than as a single black-box function and can effectively reduce the representation size. However, most problems are still computationally intractable (Table 1).

In this paper, we develop a new concise representation scheme for a characteristic function, which is based on the idea of *agent types*. Intuitively, a type represents a set of agents, which are recognized as having the same contribution. Most of the hardness results in Table 1 are obtained by assuming that all agents are different types. In practice, however, in many MAS application problems, while the number of agents grows, the number of different types of agents remains small. This type-based representation can be exponentially more concise than existing concise representation schemes. Furthermore, this idea can be used in conjunction with existing schemes, i.e., SCG and MC-nets, for further reducing the representation size. We show that most of the problems in coalitional games, including CSG, can be solved in polynomial time in the number of participating agents, assuming the number of possible types t is fixed (Table 2).

Let us introduce a motivating example that can be formalized as coalitional games with agent types. Consider a problem of forming rescue groups in a disaster area. There exists a set of participating agents and each agent has following different capabilities: providing medical treatment, driving a vehicle, acting as a firefighter, etc. We can regard the capability of an agent as the agent type. In the problem, while the number of agents grows, we can assume that the number of agent types is fixed. Thus, we can use type-based characteristic function representations to compactly represent this problem.

We can also consider our type-based representation provides an approximation method. Even if the capabilities of agents are not exactly the same, as long as they are similar,

Table 1: Computational complexities of coalition formation problems and CSG using conventional representations [1][Conitzer and Sandholm, 2006] [2][Ieong and Shoham, 2005] [3][Rahwan and Jennings, 2008] [4][Ohta et al., 2009]

	Conventional representation schemes		
	Characteristic function	SCG	MC-nets
Core non-empty	exponential	NP-complete[1]	co-NP-hard[2]
Core membership	exponential	linear[1,5]	co-NP-complete[2]
The Shapley value	exponential	$O(2^{2n})$[6]	linear[2,5]
CSG	$O(3^n)$[3]	NP-hard[4]	NP-hard[4]

Table 2: Computational complexities of coalition formation problems and CSG using type-based representations

	Type-based representation schemes		
	Characteristic function	SCG	MC-nets
Core non-empty	polynomial (Theorem 2)	polynomial (Theorem 6)	polynomial (Theorem 10)
Core membership	$O(n^t)$ (Theorem 2)	$O(n^{2t})$ (Theorem 6)	$O(n^{2t})$ (Theorem 10)
The Shapley value	$O(n^t)$ (Theorem 3)	$O(n^{2t})$ (Theorem 7)	$O(|R| \cdot n^{2t})$ (Theorem 11)
CSG	$O(n^{2t})$ (Theorem 4)	$O(n^{2t})$ (Theorem 8)	$O(n^{2t})$ (Theorem 12)

we can consider that their types are the same and assume each member of that type has the same capability (e.g., the average of members). Then, by using our type-based representation, we can obtain an approximate solution efficiently.

Our idea of using agent types is inspired by the recent innovative work of [Shrot *et al.*, 2010]. They assume that a game is already represented in some concise representation, e.g., SCG. More precisely, they consider the cases where a game is represented as a graphical representation [Deng and Papadimitriou, 1994], a coalition resource game [Wooldridge and Dunne, 2006], or SCG. The goal of their work is first to identify agent types and then to efficiently solve problems in coalitional games by utilizing the knowledge of agent types. For example, they show that when the characteristic function is represented as an SCG and types of agents are determined, computing the Shapley value and determining whether the core is non-empty can be done in polynomial time in the number of agents. However, this approach becomes infeasible when a standard characteristic function representation is used, since there exists no efficient way for identifying agent types.

In contrast to their study, we assume that agent types are explicitly used for describing a characteristic function in the first place. Also, we consider a wider range of problems including CSG. As a result, the overlap between our work and [Shrot *et al.*, 2010] is very small. In Table 2, only two entries, i.e., Core non-empty and the Shapley value for SCG, might be considered as somewhat overlapping, while other topics are not discussed in [Shrot *et al.*, 2010].

1.1 Related Works

Several other works than [Shrot *et al.*, 2010] have also examined the concept of agent types to represent agent capabilities. Bachrach and Rosenschein [2008] introduce *coalitional skill games*, where the capability of an agent is characterized by its skills. We can consider such skills correspond to agent types. However, they do not assume that the possible types/skills of an agent are fixed (even if the number of skills is fixed,

the combinations of skills are exponential). Thus, their algorithms and complexity results are quite different from ours. Chalkiadakis *et al.* [2009] consider another representation scheme that can represent any characteristic function, as well as ours. However, they examine the complexity of coalition formation problems only in *simple games*, where the value of a characteristic function is either 0 or 1, while we consider general games, where the value can be arbitrary determined.

Furthermore, the literature of *weighted voting games* [Bachrach and Elkind, 2008; Elkind and Pasechnik, 2009] assumes that the possible types of an agent are bounded. In this game, each agents has a weight and the value of a coalition is determined by the sum of their weights. The value of a coalition is 1 if the total weights exceeds a certain quota, and 0 otherwise. Their works are very similar to our works if we regard each agent's weight as the agent type. They showed that the core-related coalition formation problems and computing the Shapley value and the nucleolus become more tractable when the number of weights is bounded. However, as well as [Chalkiadakis *et al.*, 2009], they concentrate on only simple games and their type-based representation by using a weight is a subclass of our proposed one where all types are classified by a cardinal utility. Thus, we believe that this paper has significant new contributions since it can handle general characteristic functions.

2 Model

2.1 Coalitional Games and Coalition Structure Generation

Let $A = \{1, 2, \dots, n\}$ be a set of all agents. The value of a coalition S is given by a characteristic function v. A characteristic function $v : 2^A \to \mathbb{R}$ assigns a value to each set of

---

[5]These problems can be solved in linear time in the input size.

[6]This bound is not tight. Examining a tight bound is an open problem.

agents (coalition) $S \subseteq A$. We assume that each coalition's value is non-negative.

Let $x = (x_1, x_2, \ldots, x_n)$ be a payoff vector. A *solution concept* assigns to each coalitional game a set of reasonable payoff vectors. Two of the best-known ones are the core and the Shapley value.

Definition 1. *The core is the set of all payoff vectors x, which satisfy the feasibility condition:* $\sum_{i \in A} x_i = v(A)$, *and the non-blocking condition:* $\forall S \subseteq A, \sum_{i \in S} x_i \geq v(S)$.

If there exists a blocking coalition S such that $\sum_{i \in S} x_i < v(S)$ holds, then the agents in S have an incentive to collectively deviate from the grand coalition and divide $v(S)$ themselves. In general, the core can be empty or contain a large set of payoff vectors.

Definition 2. *The Shapley value of agent i, ϕ_i, is defined as:* $\phi_i = \sum_{S \subseteq A \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!}(v(S \cup \{i\}) - v(S))$.

One intuitive interpretation of the Shapley value is that it averages an agent's marginal contribution over all possible orders in which the agent may join the coalition.

A coalition structure CS is a partition of A, into disjoint, exhaustive coalitions. More precisely, $CS = \{S_1, S_2, \ldots\}$ satisfies the following conditions: $\forall i, j \ (i \neq j), \ S_i \cap S_j = \phi, \ \bigcup_{S_i \in CS} S_i = A$. In other words, in CS, each agent belongs to exactly one coalition, and some agents may be alone in their coalitions.

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 is optimal. In this paper, we assume a characteristic function can be non-super-additive.

Example 1. *Let there be four agents a, b, c, and d. The characteristic function is given as follows:*

$$v(\{a\}) = 3, \quad v(\{b\}) = 3, \quad v(\{c\}) = 2,$$
$$v(\{d\}) = 2, \quad v(\{a,b\}) = 6, \quad v(\{a,c\}) = 5,$$
$$v(\{a,d\}) = 5, \quad v(\{b,c\}) = 5, \quad v(\{b,d\}) = 5,$$
$$v(\{c,d\}) = 2, \quad v(\{a,b,c\}) = 8, \quad v(\{a,b,d\}) = 8,$$
$$v(\{a,c,d\}) = 5, \quad v(\{b,c,d\}) = 5, \quad v(\{a,b,c,d\}) = 5.$$

In this case, there exist multiple optimal CSs. For example, $\{\{a,b,c\},\{d\}\}$ and $\{\{a,b,d\},\{c\}\}$ are optimal CSs, and the value of these CSs is 10.

2.2 Agent Types

Shrot *et al.* [2010] introduced the idea of using *agent types* to reduce the computational complexity of coalition formation problems. If two agents have the same type, their marginal contributions are the same. They introduced two different notions of agent types, i.e., *strategic types* and *representational types*. The former defines types based on the strategic power of the agents, and the latter defines them based on the representation of the game.

Strategic types are defined based on the marginal contribution of each agent, i.e., if two agents are strategically equivalent, they belong to the same (strategic) type.

Definition 3 (Definition 2.1 in [Shrot *et al.*, 2010]). *Agents $i, j \in A$ are* strategically equivalent *if for any coalition S, such that $i, j \notin S : v(S \cup \{i\}) = v(S \cup \{j\})$.*

The notion called *representational type* is introduced to check the equivalence of agents more conveniently based on a concise representation. Agents are representationally equivalent if they only differ in their identifier. If two agents are representationally equivalent, they are also strategically equivalent, but not vice versa.

Shrot *et al.* [2010] examined the computational complexity for determining strategic or representational types for several concise representations [Deng and Papadimitriou, 1994; Wooldridge and Dunne, 2006; Conitzer and Sandholm, 2006]. They further showed that if the number of types is fixed in these representations, most intractable problems in coalitional games become polynomial.

3 Type-based Characteristic Function Representation

Shrot *et al.* assume that a game is already represented in some concise representation. The goal of their work is to identify agent types and to solve problems in coalitional games by efficiently utilizing agent types.

In this paper, we propose an alternate approach. We assume the person who is describing a game has some prior information about the equivalence of agents. Then the person will describe the game by explicitly using the information of the agent types of which he/she is aware. We need another notion of agent types. This is because (i) the information of the person can be partial and he/she is not necessarily aware of all strategic equivalence, and (ii) the equivalence that he/she is aware of is representation-independent. Therefore, we introduce another notion called *recognizable types*.

Definition 4. *Agents $i, j \in A$ are recognizably equivalent if the person who is describing the game (either by a characteristic function or by a concise representation) knows that for any coalition S, such that $i, j \notin S : v(S \cup \{i\}) = v(S \cup \{j\})$.*

From this definition, if two agents are recognizably equivalent, they are also strategically equivalent, but not vice versa. Furthermore, assuming appropriate representation is chosen, if two agents are recognizably equivalent, they are very likely to be representationally equivalent.

Let $T = \{1, 2, \ldots, t\}$ be the set of all recognizable types and n_A^i be the number of agents of type $i \in T$ in the set of all agents A. Also, $n_A = \langle n_A^1, n_A^2, \ldots, n_A^t \rangle$ denotes a vector, where each element represents the number of agents of each type in A.

We represent a characteristic function as follows:

Definition 5. *For a coalition S, the coalition type of S is a vector $n_S = \langle n_S^1, n_S^2, \ldots, n_S^t \rangle$, where each n_S^i is the number of type i agents in S. We denote the set of all possible coalition types as $A^t = \{\langle n^1, n^2, \ldots, n^t \rangle \mid 0 \leq n^i \leq n_A^i\}$. A type-based characteristic function is defined as $v_t : A^t \to \mathbb{R}$.*

From the definition of recognizable equivalence, $\forall S$ and its type n_S, $v(S) = v_t(n_S)$ holds.

Theorem 1. *A type-based characteristic function requires $O(n^t)$ space.*

Proof. It is clear from the fact that $|A^t| = (n_A^1 + 1) \times \ldots \times (n_A^t + 1) < n^t$. □

Example 2. *Let agents a, b be type 1 and agents c, d be type 2 in Example 1. A type-based characteristic function representation for Example 1 is given as follows:*

$$v_t(\langle 1, 0 \rangle) = 3, \quad v_t(\langle 0, 1 \rangle) = 2, \quad v_t(\langle 1, 1 \rangle) = 5,$$
$$v_t(\langle 2, 0 \rangle) = 6, \quad v_t(\langle 0, 2 \rangle) = 2, \quad v_t(\langle 2, 1 \rangle) = 8,$$
$$v_t(\langle 1, 2 \rangle) = 5, \quad v_t(\langle 2, 2 \rangle) = 5.$$

For example, the type of coalition $S = \{a, b, c\}$ is $\langle 2, 1 \rangle$ because S contains two agents of type 1 and one agent of type 2. Thus, $v(S) = 8$. Here, the type-based representation defines the value for each of eight possible coalition types, while the standard representation needs to specify the value for each of fifteen possible coalitions. In general, a type-based representation is exponentially more concise than a standard representation.

We say a payoff vector is *symmetric* if all agents with the same type receive an identical amount. We can restrict our attention to *symmetric* payoff vectors without loss of generality (Lemma 3.2 in [Shrot *et al.*, 2010]). From these definitions, we obtain the following theorems about the coalition formation problems of the core and the Shapley value.

Theorem 2. *If the number of agent types t is fixed, by using a type-based characteristic function, determining whether the core is non-empty can be done in polynomial time in the number of agents n. Also, determining whether a symmetric payoff vector x is in the core can be done in $O(n^t)$ time.*

Theorem 3. *By using a type-based characteristic function representation, computing the Shapley value of any agent can be done in $O(n^t)$ time.*

For space reasons, we omit the proofs. These results come from the fact that the number of possible coalition types $|A^t|$ is at most $O(n^t)$.

4 Coalition Structure Generation with Agent Types

In this section, we develop an algorithm for the CSG problem based on knapsack problems [Kellerer *et al.*, 2004]. A multidimensional unbounded knapsack problem (MUKP) is the knapsack problem, where the knapsack has multidimensional constraint and multiple copies exist for each item. For each item j, we denote the profit as p_j, the weight of the i-th constraint as w_{ij}, and the number of copies packed in the knapsack as q_j. A MUKP with m items and t constraints of knapsack $c_1, \ldots, c_t$ is formalized as follows:

$$\text{maximize} \quad \sum_j p_j q_j$$
$$\text{subject to} \quad \sum_j w_{ij} q_j \leq c_i, \ i = 1, \ldots, t$$
$$q_j \geq 0, \ j = 1, \ldots, m.$$

Theorem 4. *By using a type-based characteristic function representation, finding an optimal coalition structure can be done in $O(n^{2t})$ time.*

Proof. We show that a CSG problem with $m = |A^t|$ coalition types and t possible agent types can be formalized as a

MUKP with m items and t constraints. Let us assume that one possible coalition type $n_{S_j} \in A^t$ corresponds to item j, where its value p_j is equal to $v_t(n_{S_j})$ and its weight for the i-th constraint is equal to $n_{S_j}^i$. The capacity constraint of knapsack c_i is determined by n_A^i.

Let $z_j[d_1] \ldots [d_t]$ be the optimal solution value for the knapsack problem (CSG) with j coalition types $\{n_{S_1}, \ldots, n_{S_j}\}$ and a capacity constraint of knapsack $c_i = d_i, \forall i \in T$. If $z_{j-1}[d_1] \ldots [d_t]$ is known for all capacity values $0 \leq d_i \leq n_A^i, \forall i \in T$, then we can include another coalition type n_{S_j} and compute the corresponding solutions $z_j[d_1] \ldots [d_t]$ using the following recursive formula:

$$z_j[d_1][d_2] \ldots [d_t] =$$
$$\max \begin{cases} z_j[d_1 - n_{S_j}^1][d_2 - n_{S_j}^2] \ldots [d_t - n_{S_j}^t] + v_t(n_{S_j}) \\ (\text{if } \forall i \in T, d_i \geq n_{S_j}^i) \\ z_{j-1}[d_1][d_2] \ldots [d_t]. \end{cases}$$

We can construct a dynamic programming based algorithm from this recursive formula, which takes $O(n^t \times |A^t|) = O(n^{2t})$ steps (see Section 9.3.2 in [Kellerer *et al.*, 2004]). Thus, for any fixed t, finding an optimal coalition structure can be done in $O(n^{2t})$ time. □

A similar argument has been done for the winner determination problem in combinatorial auctions with a fixed number of types of items [Tennenholtz, 2000]. In fact, a CSG problem can be mapped into that problem by assuming each coalition type corresponds to a bid and that each type of agent corresponds to a type of item.

5 Combining with Concise Representation Schemes

5.1 Type-based SCG

We first show the original definition of SCG [Conitzer and Sandholm, 2006].

Definition 6. *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 S_i are disjoint and $\cup_{S_i \in p_S} S_i = S$, 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.*

Using this original definition, we can represent only super-additive characteristic functions. To allow for characteristic functions that are not super-additive, Ohta *et al.* [2009] slightly modify the definition, i.e., they add the following requirement for partition p_S: $\forall p_S' \subseteq p_S$, where $|p_S'| \geq 2$, $(\cup_{S_i \in p_S'} S_i, v(\cup_{S_i \in p_S'} S_i))$ is not an element of SCG. We refer to this modified definition as a standard SCG.

Next, we introduce the definition of a type-based SCG.

Definition 7. *A type-based SCG consists of a set of pairs of the form: $(n_S, v_t(n_S))$. For any coalition type n_S, the value of the characteristic function is defined in a similar way as a standard SCG.*

Theorem 5. *A type-based SCG can represent any characteristic function represented in a standard SCG using at most the same amount of space and is exponentially more concise than a standard SCG for certain games.*

We omit the proofs due to space limitations. Intuitively, the worst case occurs when the recognizable types of all agents are different. Also, when all agents have an identical type, the value of a characteristic function is determined only by the number of agents in a coalition. Then the required size of a type-based SCG becomes $O(n)$, but the size of a standard SCG can be exponential.

Example 3. *A type-based SCG for Example 1 is given as follows:*

$$(\langle 1,0 \rangle, 3), \quad (\langle 0,1 \rangle, 2), \quad (\langle 0,2 \rangle, 2), \quad (\langle 2,2 \rangle, 5).$$

In this case, $v_t(\langle 2,1 \rangle) = v_t(\langle 1,0 \rangle) + v_t(\langle 1,0 \rangle) + v_t(\langle 0,1 \rangle) = 8$. Here, the type-based SCG defines the value of four coalition types, while the type-based characteristic function representation needs to describe eight possible coalition types.

As discussed in [Conitzer and Sandholm, 2006], core-related coalition formation problems remain hard in a standard SCG, because determining the value of the grand coalition is hard. If the value of the grand coalition is given explicitly, these problems become tractable. By using a type-based SCG, we can obtain the value of the grand coalition using a method similar to the DP-based algorithm described in Theorem 4. As a result, the following theorem holds.

Theorem 6. *If the number of agent types t is fixed, by using a type-based SCG representation, determining whether the core is non-empty can be done in polynomial time in the number of agents n. Also, determining whether a symmetric payoff vector x is in the core can be done in $O(n^{2t})$ time.*

Unfortunately, as far as the authors are aware, there is no efficient way to compute Shapley values using SCG-based representations. However, we can use a naive translation approach, which can be done in polynomial time.

Theorem 7. *If the number of agent types t is fixed, by using a type-based SCG representation, computing the Shapley value can be done in $O(n^{2t})$ time.*

We can use the DP-based algorithm described in Theorem 4 for finding an optimal coalition structure. The only difference is that we need to consider coalition types that are explicitly represented in a type-based SCG.

Theorem 8. *By using a type-based SCG representation, finding an optimal coalition structure can be done in $O(n^{2t})$ time.*

5.2 Type-based MC-nets

We first show the original definition of MC-nets [Ieong and Shoham, 2005].

Definition 8. *An MC-net consists of a set of rules R. Each rule $r \in R$ is of the form: $(P_r, N_r) \rightarrow v_r$, where $P_r \subseteq A$, $N_r \subseteq A$, $P_r \cap N_r = \emptyset$, $v_r \in \mathbb{R}$. v_r can be either positive or negative. 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) but 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.*

Next, we introduce the definition of type-based MC-nets.

Definition 9. *A type-based MC-net consists of a set of rules R. Each rule $r \in R$ is of the form: $(L_r, U_r) \rightarrow v_r$, where $L_r = \langle l_r^1, l_r^2, \ldots, l_r^t \rangle$ and $U_r = \langle u_r^1, u_r^2, \ldots, u_r^t \rangle$. Each l_r^i (and u_r^i) represents the lower (upper) bound of the number of i-th type agents in a coalition so that this rule becomes effective. We say that rule r is applicable to coalition S if $\forall i \in T, l_r^i \leq n_S^i \leq u_r^i$. 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.*

Theorem 9. *A type-based MC-net can represent any characteristic function represented in a standard MC-net using at most the same amount of space and is exponentially more concise than a standard MC-net for certain games.*

For space reasons, we omit the proofs. We can prove these theorems using a similar argument described in Theorem 5.

Example 4. *A type-based MC-net for Example 1 is given as follows:*

$$r_1 : (\langle 1,0 \rangle, \langle 1,2 \rangle) \rightarrow 3, \quad r_2 : (\langle 2,0 \rangle, \langle 2,1 \rangle) \rightarrow 6,$$
$$r_3 : (\langle 0,1 \rangle, \langle 2,2 \rangle) \rightarrow 2, \quad r_4 : (\langle 2,2 \rangle, \langle 2,2 \rangle) \rightarrow 3.$$

In this case, r_2 and r_3 are applicable to coalition type $\langle 2,1 \rangle$, but r_1 and r_4 are not. Thus, $v_t(\langle 2,1 \rangle)$ is equal to $6 + 2 = 8$. Here, the type-based MC-net consists of four rules, while the type-based characteristic function representation needs to specify the value of eight possible coalition types.

Unfortunately, there is no efficient way to solve core-related coalition formation problems using MC-net-based representations. However, we can use a naive translation approach, which can be done in polynomial time.

Theorem 10. *If the number of agent types t is fixed, by using a type-based MC-net representation, determining whether the core is non-empty can be done in polynomial time in the number of agents n. Also, determining whether a symmetric payoff vector x is in the core can be done in $O(n^{2t})$ time.*

In contrast to core-related coalition formation problems, the standard MC-net representation is suitable for computing Shapley values. This is also true for our type-based MC-net representation, i.e., the following theorem holds.

Theorem 11. *If the number of agent types t is fixed, by using a type-based MC-net representation, computing the Shapley value of any agent can be done in $O(|R| \cdot n^{2t})$ time.*

Proof. To compute the Shapley value of an agent, we can compute its Shapley value for each rule and use the summation of these values (see Proposition 5 in [Ieong and Shoham, 2005]). Furthermore, we can decompose a rule into multiple rules, where each decomposed rule has a form: $(\langle y^1, \ldots, y^t \rangle, \langle y^1, \ldots, y^t \rangle) \rightarrow v$, i.e., the rule is applicable to exactly one coalition type. The Shapley value of type i agent for rule r

(denoted as $\phi_{i,r}$) is computed by the following procedure:

$$\phi_{i,r} = \frac{v}{n!}(f_i^+(y_1,\ldots,y_t) - f_i^-(y_1,\ldots,y_t)),$$

$$f_i^+(y_1,\ldots,y_t) =$$

$$\begin{cases} 0 & \text{if } y_i = 0, \\ \Pi_{j \neq i} \; _{n_A^j}C_{y_j} \cdot \; _{n_A^i-1}C_{y_i-1} \cdot (s_y-1)!(n-s_y)! & \text{otherwise.} \end{cases}$$

$$f_i^-(y_1,\ldots,y_t) =$$

$$\begin{cases} 0 & \text{if } y_i = n_A^i, \\ \Pi_{j \neq i} \; _{n_A^j}C_{y_j} \cdot \; _{n_A^i-1}C_{y_i} \cdot (s_y)!(n-s_y-1)! & \text{otherwise.} \end{cases}$$

$$\text{where } s_y = \sum_{j \in T} y_j, \; n = \sum_{j \in T} n_A^j.$$

Here, $f_i^+(y_1,\ldots,y_t)$ represents the number of orderings where the marginal contribution of one type i agent is v, and $f_i^-(y_1,\ldots,y_t)$ represents the number of orderings where the marginal contribution of one type i agent is $-v$. Thus, $\frac{v}{n!}(f_i^+(y_1,\ldots,y_t) - f_i^-(y_1,\ldots,y_t))$ represents the Shapley value of type i agent for this rule. Using this procedure, the required time for computing the Shapley value of an agent becomes $O(|R| \cdot n^{2t})$. $\square$

Although Ohta *et al.* [2009] proposed an efficient method for solving CSG problems based on the standard MC-net, we cannot apply this method straightforwardly. Nevertheless, we can still rely on a naive approach that translates a type-based MC-net representation into the corresponding type-based characteristic function representation.

Theorem 12. *If the number of agent types t is fixed, by using a type-based MC-net representation, finding an optimal coalition structure can be done in $O(n^{2t})$ time.*

6 Experimental Evaluations

In this section, we experimentally evaluate the performance of our proposed methods. We concentrate on methods for type-based SCG, since we can control the input size of a problem instance. In addition, the DP-based algorithm is also used in other representations. All tests were run on a Core 2 Quad Q9650 3GHz processor with 16GB RAM. The test machine runs Windows 7 Enterprise x64 Edition.

Let us consider a type-based SCG problem instance, where n agents have one of five different types ($t = 5$). We vary n from 10 to 100 and set the number of elements in a type-based SCG to n, (i.e., equal to the number of agents). We generate each element using a decay distribution as follows. Initially, the required number of agents in each type is set to zero. First, we randomly choose one type and increment the required number of agents in the type by one. Then, we repeatedly choose a type randomly and increment its required number of agents with probability α, until a type is not chosen or the required number of agents exceeds the limit. We choose the value of that coalition between 1 and $10 \times n$ uniformly at random and use $\alpha = 0.55$. In this way, we generated 50 problem instances for each n.

We translate each generated problem instance represented by type-based SCG into an equivalent problem instance represented by standard SCG. The number of elements in the
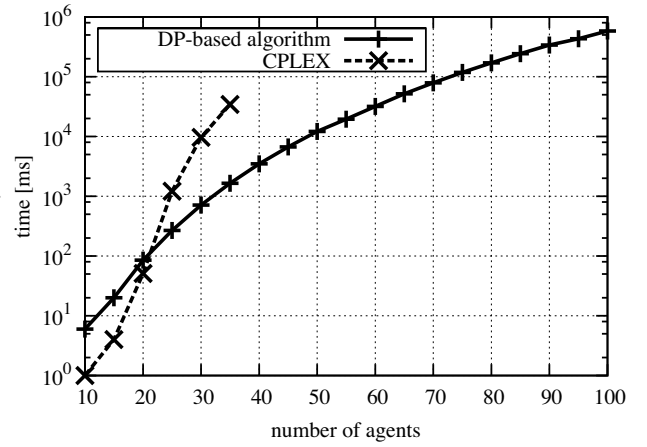

Figure 1: Computation time

standard SCG grows exponentially compared to that in the type-based SCG. When the number of elements in the type-based SCG representation exceeds 40, we cannot translate the problem instances due to insufficient memory. This result illustrates that the type-based representation is exponentially more concise than the standard representation.

We investigate the computation time of our DP-based algorithm. For comparison, we show the results of the MIP formulation in Ohta *et al.* [2009], which uses a standard SCG representation. To obtain this result, we used CPLEX version 12.1, a general-purpose mixed integer programming package.

Figure 1 illustrates the average computation times for solving the generated problem instances by our DP-based algorithm using the type-based SCG (DP) and by CPLEX in the MIP formulation using the standard SCG (CPLEX). The x-axis indicates the number of agents, and the y-axis shows the average computation times. When $n \leq 20$, CPLEX is faster than DP, while DP eventually outperforms CPLEX for $n > 20$. CPLEX can reduce the search space efficiently when the input size is relatively small. However, the input size for CPLEX grows exponentially. Thus, its computation time increases very rapidly. When $n > 40$, even generating problem instances becomes infeasible. On the other hand, the computation time for DP grows more slowly than the exponential rate. This result corresponds to the theoretical complexity presented in Theorem 8, i.e., finding an optimal coalition structure can be done in $O(n^{2t})$ time. As shown in this result, the type-based SCG enables us to solve a CSG problem instance with up to 100 agents in a reasonable amount of time.

7 Conclusion

In this paper, we developed a new concise representation scheme for a characteristic function, which is based on the idea of *agent types*. The type-based representation can be exponentially more concise than existing concise representation schemes. Furthermore, this idea can be used in conjunction with existing schemes, i.e., MC-nets and SCG, for further reducing the representation size. We showed that most problems in coalitional games, including CSG, can be solved in

polynomial time in the number of agents, assuming the number of types t is fixed. We also experimentally showed that a type-based SCG enables us to solve a CSG problem instance with up to 100 agents in a reasonable amount of time. Our idea of using agent types is inspired by the recent work of [Shrot *et al.*, 2010]. However, in contrast to their study, our work introduced the idea of describing a characteristic function explicitly using agent types in the first place, and considered a wider range of problems in coalitional games including CSG.

In future works, we first would like to examine the complexity of other solution concepts in coalitional games with or without coalition structures, such as the nucleolus, kernel, and core with coalition structures. Second, we would like to investigate our representation in conjunction with a graph-based concise representation scheme, e.g. [Deng and Papadimitriou, 1994; Kalai and Zemel, 1982; Bachrach *et al.*, 2008; Aziz *et al.*, 2009; Ueda *et al.*, 2010]. For example, Shrot *et al.* showed that utilizing agent types makes various problems in coalitional games tractable when a characteristic function is represented graphically by the formalization used in [Deng and Papadimitriou, 1994]. Thus, we guess that our scheme also would work well, by assuming that a graphical structure describes a relationship between agent types, rather than a relationship between agents themselves.

Acknowledgements

References

[Aziz *et al.*, 2009] H. Aziz, O. Lachish, M. Paterson, and R. Savani. Power indices in spanning connectivity games. In *Proceedings of the 5th International Conference on Algorithmic Aspects in Information and Management (AAIM)*, pages 55–67, 2009.

[Bachrach and Elkind, 2008] Y. Bachrach and E. Elkind. Divide and conquer: false-name manipulations in weighted voting games. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 975–982, 2008.

[Bachrach and Rosenschein, 2008] Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1023–1030, 2008.

[Bachrach *et al.*, 2008] Y. Bachrach, J. S. Rosenschein, and E. Porat. Power and stability in connectivity games. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 999–1006, 2008.

[Chalkiadakis *et al.*, 2009] G. Chalkiadakis, E. Elkind, and N. R. Jennings. Simple coalitional games with beliefs. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 85–90, 2009.

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

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

[Elkind and Pasechnik, 2009] E. Elkind and D. V. Pasechnik. Computing the nucleolus of weighted voting games. In *Proceedings of the 20th Annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 327–335, 2009.

[Ieong and Shoham, 2005] 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.

[Kalai and Zemel, 1982] E. Kalai and E. Zemel. Totally balanced games and games of flow. *Mathematics of Operations Research*, 7(3):476–478, 1982.

[Kellerer *et al.*, 2004] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer, 2004.

[Ohta *et al.*, 2009] N. Ohta, V. Conitzer, R. Ichimura, Y. Sakurai, A. Iwasaki, and M. Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *Proceedeings of the 15th International Conference on Principles and Practice of Constraint Programming (CP)*, pages 623–638, 2009.

[Rahwan and Jennings, 2008] T. Rahwan and N. R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1417–1420, 2008.

[Sandholm *et al.*, 1999] 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.

[Shrot *et al.*, 2010] T. Shrot, Y. Aumann, and S. Kraus. On agent types in coalition formation problems. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 757–764, 2010.

[Tennenholtz, 2000] M. Tennenholtz. Some tractable combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 98–103, 2000.

[Ueda *et al.*, 2010] S. Ueda, A. Iwasaki, M. Yokoo, M.-C. Silaghi, K. Hirayama, and T. Matsui. Coalition structure generation based on distributed constraint optimization. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*, pages 197–203, 2010.

[Wooldridge and Dunne, 2006] M. Wooldridge and P. E. Dunne. On the computational complexity of coalitional resource games. *Artificial Intelligence*, 170(10):853–871, 2006.