Compact Representation Scheme of Coalitional Games Based on Multi-terminal Zero-suppressed Binary Decision Diagrams

Ryo Ichimura, Yuko Sakurai, Suguru Ueda, Atsushi Iwasaki, Makoto Yokoo
Kyushu University
{ichimura@agent., ysakurai@, ueda@agent., iwasaki@, yokoo@}inf.kyushu-u.ac.jp
Shin-Ichi Minato
Hokkaido University
minato@ist.hokudai.ac.jp

Abstract

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. Previous studies have found that many problems in coalitional games tend to be computationally intractable in this black-box function representation. Recently, several concise representation schemes for a characteristic function have been proposed. Among them, a synergy coalition group (SCG) has several good characteristics, but its representation size tends to be large compared to other representation schemes.

We propose a new concise representation scheme for a characteristic function based on a Zero-suppressed Binary Decision Diagram (ZDD) and a SCG. We show our scheme (i) is fully expressive, (ii) can be exponentially more concise than the SCG representation, (iii) can solve core-related problems in polynomial time in the number of nodes, and (iv) can solve a CSG problem reasonably well by utilizing a MIP formulation. A Binary Decision Diagram (BDD) has been used as unified infrastructure for representing/manipulating discrete structures in such various domains in AI as data mining and knowledge discovery. Adapting this common infrastructure brings up the opportunity of utilizing abundant BDD resources and cross-fertilization with these fields.

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 involves partitioning a set of agents into coalitions so that the sum of the rewards of all coalitions is maximized. This is called the Coalition Structure Generation problem (CSG) [Sandholm *et al.*, 1999]. 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.

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 that takes a coalition as an input and returns its value. Representing an arbitrary characteristic function explicitly requires $\Theta(2^n)$ numbers, which is prohibitive for large n.

Recently, several concise representation schemes for a characteristic function have been proposed [Conitzer and Sandholm, 2006; Elkind *et al.*, 2008; Ieong and Shoham, 2005; Shrot *et al.*, 2010]. Among them, the synergy coalition group (SCG) [Conitzer and Sandholm, 2006] has several good characteristics. However, a SCG tends to require more space than other representation schemes such as marginal contribution networks [Ieong and Shoham, 2005].

In this paper, we propose a new concise representation scheme for a characteristic function, based on the idea of *Binary Decision Diagram* (BDD) [Akers, 1978]. A BDD is graphical representations that can compactly represent a boolean function. We use ae variant of BDD called a Zero-suppressed BDD (ZDD) [Minato, 1993] that can compactly represent a set of combinations. More specifically, we use a Multi-Terminal ZDD (MTZDD), which can compactly represent a SCG. This representation preserves the good characteristics of a SCG. The following are the features of our scheme: (i) it is fully expressive, (ii) it can be exponentially more concise than a SCG, (iii) such core-related problems as core-non-emptiness, core-membership, and finding a minimal non-blocking payoff vector (cost of stability) can be solved in polynomial time in the number of nodes in a MTZDD, and (iv) although solving a CSG is NP-hard, it can be solved reasonably well by utilizing a MIP formulation.

A BDD was originally developed for VLSI logic circuit design. Recently, A BDD has been applied to various domains in AI, including data mining and knowledge discovery . In these domains, we need to handle logic functions or combination sets efficiently. A BDD has been used as unified infrastructures for representing/manipulating such *discrete structures*. A vast amount of algorithms, software, and tools related to a BDD already exist, e.g., an arithmetic boolean expression manipulator based on a BDD, and a programs for calculating combination sets based on a ZDD [Minato, 1993]. Adapting this common infrastructure for coalitional game theory brings up the opportunity to utilize these abundant resources and for cross-fertilization with other related fields in AI.

2 Preliminaries

2.1 Coalitional Games

Let $A = \{1, 2, \ldots, n\}$ be the set of agents. Since we assume a characteristic function game, the value of coalition S is given by characteristic function v, which assigns a value to each set of agents (coalition) $S \subseteq A$. We assume that each coalition's value is non-negative.

Coalition structure CS is a partition of A into disjoint and exhaustive coalitions. To be more precise, $CS = \{S_1, S_2, \ldots\}$ satisfies the following conditions: $\forall i, j\ (i \neq j),\ S_i \cap S_j = \emptyset, \bigcup_{S_i \in CS} S_i = A$. The value of coalition structure CS, denoted as $V(CS)$, is given by: $V(CS) = \sum_{S_i \in CS} v(S_i)$. Optimal coalition structure CS^* is a coalition structure that satisfies $\forall CS, V(CS^*) \geq V(CS)$.

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; the grand coalition is optimal. We assume a characteristic function can be non-super-additive.

The core is a prominent solution concept focusing on stability. When a characteristic function is not necessarily super-additive, creating a grand coalition does not make sense. As discussed in [Aumann and Dreze, 1974], we need to consider the stability of a coalition structure. The concept of the core can be extended to the case where agents create an optimal coalition structure. Assume $\pi = (\pi_1, \ldots, \pi_n)$ describes how to divide the obtained reward among agents. We call π a *payoff vector*.

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

If for some set of agents S, the non-blocking condition does not hold, then the agents in S have an incentive to collectively deviate from CS^* and divide $v(S)$ between themselves. As discussed in [Airiau and Sen, 2010], there are two alternative definitions of the feasibility condition: (i) $\sum_{i \in A} \pi_i = V(CS^*)$, and (ii) $\forall S \in CS^*, \sum_{i \in S} \pi_i = v(S)$. If (ii) holds, then (i) holds, but not vice versa. Condition (ii) requires that no monetary transfer (side payment) exists across different coalitions. However, as shown in [Aumann and Dreze, 1974], if a payoff vector satisfies both condition (i) and the non-blocking condition, it also satisfies condition (ii). Thus, we use condition (i) as the feasibility condition.

In general, the core can be empty. The ϵ-core can be obtained by relaxing the non-blocking condition as follows: $\forall S \subseteq A, \sum_{i \in S} \pi_i + \epsilon \geq v(S)$. When ϵ is large enough, the ϵ-core is guaranteed to be non-empty. The smallest non-empty ϵ-core is called the least core.

Alternatively, we can relax the feasibility condition as follows: $\sum_{i \in A} \pi_i = V(CS^*) + \Delta$. This means that an external party is willing to pay amount Δ as a subsidy to stabilize the coalition structure. The minimal amount of Δ is called the *cost of stability* [Bachrach *et al.*, 2009].

2.2 SCG

Conitzer and Sandholm [2006] introduced 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 some *positive* synergy exists.

Definition 2 *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: $v(S) = \max_{p_S} \{\sum_{S_i \in p_S} v(S_i)\}$, where p_S is a partition of S; all S_is 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 without feasible partitions, we require that $(\{a\}, 0) \in SCG$ whenever $\{a\}$ does not receive a value elsewhere in SCG.*

If the value of 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. 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.

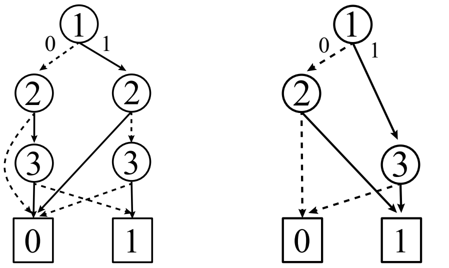
This additional condition requires that if the value of a coalition is explicitly given in SCG, then we cannot further divide it into smaller subcoalitions to calculate values. In this way, we can represent *negative* synergies.

2.3 BDD and ZDD

A BDD represents boolean functions as a rooted, directed acyclic graph of internal nodes and two 0/1-terminal nodes. Each internal node represents a variable and has two outgoing edges: a high-edge and a low-edge. The high-/low-edge means that the value of the variable is true/false. A path from the root node to the 1-terminal node represents that the corresponding value assignment to the path makes the boolean function true. A ZDD is a variant of BDD that can efficiently represent a set of combination. The high-/low-edge means the presence/absence of an element in a combination. In a ZDD, a path from the root node to the 1-terminal node represents that the corresponding value assignment to the path is included in the set.

Consider boolean function $((x_1 \bar{x}_2 x_3) \vee (\bar{x}_1 x_2 \bar{x}_3))$, which can be equivalently represented by using a set of combinations $(\{\{1, 3\}, \{2\}\})$. Figure 1 shows the BDD/ZDD representation for this function/set of combinations. In a tree, a node with x_i represents i. A ZDD is more concise than a BDD. If a variable never appears within any elements in a set of combinations, a node that represents the variable is removed from the ZDD. If the sum of elements contained in all combinations in a set is k, the number of nodes in a ZDD is at most $O(k)$.

Quite recently, two different BDD-based representation schemes for a characteristic function have been developed independently from our work [Aadithya *et al.*, 2011; Berghammer and Bolus, 2010]. While Berghammer and Bolus [2010] deals with simple games, Aadithya *et al.* [2011] considers



(a) BDD (b) ZDD
Figure 1: Examples of a BDD and a ZDD

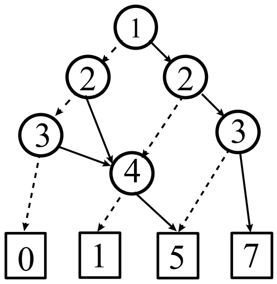


Figure 2: MTZDD representation in Example 1

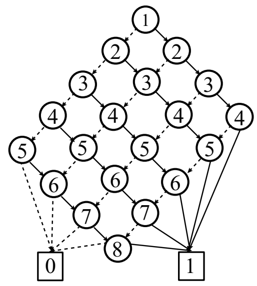


Figure 3: MTZDD representation in Theorem 2

general games. Both of schemes try to represent a characteristic function directly, while our scheme represents SCGs.

3 New Concise Representation Scheme

We propose our new representation scheme for a characteristic function based on a SCG and a ZDD. Although a ZDD can only represent whether a combination exists in a set, a SCG is not just a set of coalitions, because each coalition S in a SCG is associated with its value $v(S)$. Thus, we use a multi-terminal ZDD (MTZDD) representation.

3.1 MTZDD representation based on SCG

A MTZDD G is defined by (V, T, H, L), where V is a set of internal (non-terminal) nodes, T is a set of terminal nodes, H is a set of high-edges, and L is a set of low-edges. Each internal node $u \in V$ is associated with one agent, which we denote as $agent(u)$. u has exactly two outgoing edges, $h(u) = (u, u')$ and $l(u) = (u, u'')$, where $h(u) \in H$ and $l(u) \in L$. Each terminal node $t \in T$ is associated with a non-negative value, which we denote as $r(t)$. Root node u_0 has no incoming edges. For each node $u \in V \setminus \{u_0\} \cup T$, at least one incoming edge exists. We denote the parents of u as $Pa(u)$, $Pa(u) = \{u' \mid (u', u) \in H \cup L\}$.

Path p from root node u_0 to terminal node t is represented by a sequence of edges on path $p = ((u_0, u_1), (u_1, u_2), \ldots, (u_k, t))$. For p, we denote $S(p) = \{agent(u_i) \mid h(u_i) \in p\}$, because $S(p)$ denotes a coalition represented by path p. Also, we denote the value of path p as $r(p)$, which equals $r(t)$: $v(S(p)) = r(t)$. In a MTZDD, a particular ordering among agents is preserved. In path p from root node u_0 to terminal node t, agents associated with nodes in p appear in the same order. More specifically, if node u appears before node u' in p, then $agent(u) \neq agent(u')$. Also, there exists no path p', in which node u appears before node u', where $agent(u) = agent(u')$. For each agent $i \in A$, $nodes(i)$ denotes a set of nodes that are associated with agent i, i.e., $nodes(i) = \{u \mid u \in V \wedge agent(u) = i\}$.

Example 1 *Let there be four agents:* 1, 2, 3, *and* 4. *Let* $SCG = \{ (\{1\}, 1), (\{2\}, 1), (\{3\}, 1), (\{4\}, 0), (\{1,2\}, 5), (\{1,4\}, 5), (\{2,4\}, 5), (\{3,4\}, 5), (\{1,2,3\}, 7)\}$. *This MTZDD representation is described in Figure 2. For example, the rightmost path of the tree represents a coalition* $\{1, 2, 3\}$ *and its value* 7.

3.2 Conciseness of MTZDD Representation

Theorem 1 *MTZDD can represent any characteristic function represented in a SCG using at most $O(n|SCG|)$ nodes, where n is the number of agents and $|SCG|$ is the number of elements in a SCG.*

Proof *In a MTZDD, for each agent i, $|nodes(i)|$ is at most $|SCG|$ because $|nodes(i)|$ represents the number of different contexts that result in different outcomes. This number is bounded by the number of different combinations of agents, which appear before i in the ordering among agents. Clearly, this number is at most $|SCG|$. Thus, the number of non-terminal nodes, i.e., $\sum_{i \in A} |nodes(i)|$, is at most $n|SCG|$. Also, the number of terminal nodes is at most $|SCG| + 1$. As a result, the total number of nodes is $O(n|SCG|)$.* □

Theorem 2 *A MTZDD representation is exponentially more concise than a SCG for certain games.*

Proof *Consider a coalitional game with $2m$ agents, where the value of characteristic function $v(S)$ is 1 if $|S| \geq m$, and 0 otherwise. A SCG must include each coalition with size m. The number of such coalitions is given as $\binom{2m}{m}$, which is $O(2^n)$ using Stirling's approximation.*

On the other hand, we can create a MTZDD that counts the number of agents in a coalition and returns 1 when the number reaches m. Such MTZDD requires $m(m+1)$ nodes, i.e., $O(n^2)$. □

As shown in the proof of Theorem 2, when some agents are *symmetric*, the MTZDD representation can be much more concise than a SCG. Figure 3 shows a MTZDD when we set $m = 4$. The number of nodes is 20, but a SCG requires 70 coalitions.

Instead of representing a SCG with a MTZDD, we can directly represent a characteristic function using a MTBDD (such an approach is considered in [Aadithya *et al.*, 2011; Berghammer and Bolus, 2010]). In a MTBDD, an agent that does not appear in a path is considered irrelevant; if $v(S \cup \{i\}) = v(S)$, we only need to describe S in a MTBDD[1]. Thus, we can reduce the representation size to a certain extent by using a MTBDD. However, this MTBDD representation for a characteristic function is not as concise as the MTZDD representation. The following theorem holds.

[1]Note that such an irrelevant agent is not included in a SCG.

Theorem 3 *A MTZDD representation of a SCG is always as concise as a MTBDD representation of a characteristic function. Also, it is exponentially more concise than a MTBDD representation for certain games.*

Proof *The worst case occurs when a SCG contains all possible coalitions. In this case, the representation sizes of the MTZDD and MTBDD are the same.*

Then, we show the case where the MTZDD representation is exponentially more concise. Consider a coalitional game with agents $1, 2, \ldots, n$, where $v(\{i\}) = 2^i$, and $v(S) = \sum_{i \in S} v(\{i\})$. $v(S)$ can take any integer value from 1 to $2^{n+1} - 1$. Thus, the number of terminal nodes in the MTBDD becomes $O(2^n)$. On the other hand, the number of elements in a SCG is n, the number of internal nodes in the MTZDD is n, and the number of terminal nodes is $n + 1$. Thus, the total number of nodes is $O(n)$. □

3.3 Procedure of constructing a MTZDD representation

Let us consider how a person, who has knowledge of a coalitional game, can describe our MTZDD representation. We assume the person is aware of symmetry among agents. Then, the person first describe several partial MTZDDs considering the symmetry among agents. For example, if a person is describing the characteristic function used in the proof of Theorem 2, we can assume she describes multiple partial MTZDDs, each of which corresponds to coalitions of k agents (where k varies from m to $2m$). Note that each partial MTZDD can correspond to multiple (possibly exponentially many) items in a SCG. Then, these partial MTZDDs are integrated into a single MTZDD by applying a Union operation [Minato, 1993] and reduction rules described in Section 2.3.

4 Coalition Structure Generation

We propose a new mixed integer programming formulation for solving a CSG problem in the MTZDD representation. In our MTZDD representation, a path from the root node to a terminal node represents a coalition that is included in a SCG. We define a condition where a set of paths, i.e., a set of coalitions, is *compatible*.

Definition 3 *Two paths, p and p', are compatible if $S(p) \cap S(p') = \emptyset$. Also, set of paths P is compatible if $\forall p, p' \in P$, where $p \neq p'$, p, and p' are compatible.*

Finding optimal coalition structure CS^* is equivalent to finding set of paths P^*, which is compatible, and $\sum_{p \in P^*} r(p)$ is maximized. We show that P^* is NP-complete and inapproximable.

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

Proof *The maximum independent set problem is to choose $V' \subseteq V$ for a graph $G = (V, E)$ such that no edge exists between vertices in V', and $|V'|$ is maximized under this constraint. It is NP-hard, and unless $\mathcal{P} = \mathcal{NP}$, there exists no*

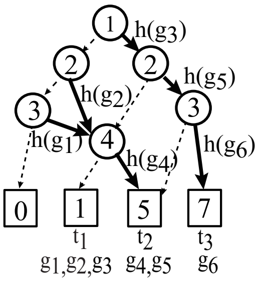


Figure 4: GS in Example 2

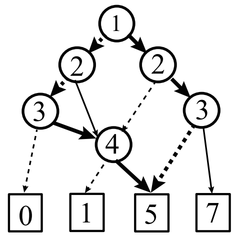


Figure 5: P^* in Example 2

polynomial-time $O(|V|^{1-\epsilon})$ approximation algorithm for any $\epsilon > 0$ [Håstad, 1999]. We reduce an arbitrary maximum independent set instance to a CSG problem instance as follows. For each $e \in E$, let there be agent a_e. For each $v \in V$, we create an element of SCG, where the coalition is $\{a_e \mid e \ni v\}$ and its value is 1. Thus, two coalitions have a common element only if they correspond to neighboring vertices. Coalition structures correspond exactly to independent sets of vertices. Furthermore, we transform this SCG representation to a MTZDD representation in polynomial time [Minato, 1993]. As a result, the number of internal nodes in a MTZDD is at least $|E|$ and at most $2|E|$, since an agent appears in exactly two coalitions. □

Ohta *et al.* [2009] developed a MIP formulation for a CSG problem when a characteristic function is represented by a SCG. If we enumerate paths, we can use their results. However, the number of paths can be exponential to the number of nodes in a MTZDD. Thus, we need to find P^* without explicitly enumerating all possible paths. We first identify the maximal number of paths within P^*, which leads to one terminal node $r(t)$, using a concept called *minimal required high-edge set* that is concisely described *minimal set*.

Definition 4 *For each terminal node $t \in T$, where $r(t) > 0$, $E \subseteq H$ is a required high-edge set if for all paths p, where t is p's terminal node, there exists $h \in E$ such that h is included in p. E is a minimal set, if E is a required high-edge set, and there exists no proper subset of E that is a required high-edge set.*

There can be multiple minimal sets. We can find one minimal set using backtrack search starting from the terminal node. The complexity of this procedure is $O(|V|)$. We denote one minimal set of t as $min(t)$. It is clear that the number of paths within P^*, which leads to terminal node $r(t)$, is at most $|min(t)|$.

A MIP formulation of finding P^* is defined as follows. We define some terms and notations. For each terminal node t, where $r(t) > 0$, we create one goal for each element in $min(t)$ and denote the set of goals created from t as $goals(t)$. For each goal $g \in goals(t)$, we denote the corresponding element in $min(t)$ as $h(g)$ and the value of g as $r(g)$, which equals $r(t)$. Let $GS = \bigcup_{t \in T | r(t) > 0} goals(t)$. For each $g \in GS$, $x(g)$ is a 0/1 decision variable that denotes whether g is active ($x(g) = 1$ means g is active). For each goal $g \in GS$ and for each edge (u, u'), $x(g, (u, u'))$ is a 0/1 decision variable that denotes that the edge (u, u') is used for goal g.

Definition 5 *The problem of finding P^* can be modeled as follows.*

$$
\begin{aligned}
&\max \textstyle\sum_{g \in GS} x(g) \cdot r(g) \\
&\quad \textit{s.t.} \quad \forall g \in GS, x(g) = x(g, h(g)), \text{— (i)} \\
&\qquad \forall t \in T, \text{ where } r(t) > 0, \forall g \in goals(t), \\
&\qquad\quad x(g) = \textstyle\sum_{u \in Pa(t)} x(g, (u, t)), \text{— (ii)} \\
&\qquad \forall u \in V \setminus \{u_0\}, \forall g \in GS, \\
&\qquad\quad x(g, h(u)) + x(g, l(u)) \\
&\qquad\quad = \textstyle\sum_{u' \in Pa(u)} x(g, (u', u)), \text{— (iii)} \\
&\qquad \forall i \in A, \textstyle\sum_{u \in nodes(i)} \sum_{g \in GS} x(g, h(u)) \leq 1, \text{— (iv)} \\
&\qquad x(\cdot), x(\cdot, \cdot) \in \{0, 1\}.
\end{aligned}
$$

Constraint (i) ensures that if goal g is selected, its required high-edge must be selected. Constraint (ii) ensures if one of its goal g is selected for terminal node t, then an edge must exist that is included in a path for g. Constraint (iii) ensures that for each non-terminal, non-root node, correct paths are created (the numbers of inputs and outputs must be the same). Constraint (iv) ensures that one agent can be included in at most one path. In this MIP formulation, the number of constraints is linear to the number of nodes in a MTZDD.

Example 2 *We consider a MIP problem of a MTZDD representation in Example 1.*

First, we create a minimal set for a non-zero-terminal node. As shown in Figure 4, we denote each non-zero terminal node as t_1, t_2, and t_3 from the left. No high-edge directly points to t_1, but using backtracking search, we find three high-edges labeled $h(g_1)$, $h(g_2)$, and $h(g_3)$ as elements of $min(t_1)$. t_2 has both incoming high-edge and low-edge, and so we obtain $min(t_2) = \{h(g_4), h(g_5)\}$. t_3 only has an incoming high-edge, i.e., $min(t_3) = \{h(g_6)\}$. Thus, we obtain $\{g_1, \ldots, g_6\}$ as GS.

Next, we solve a MIP defined by Definition 5 and obtain optimal set of paths P^* that consists of two paths that represent coalitions $\{1, 2\}$ and $\{3, 4\}$ (Figure 5). The value of P^* is calculated as 10.

5 Core-related Problems

5.1 Core-Non-Emptiness

By assuming that the value of an optimal coalition structure $V(CS^*)$ is given, checking the core-non-emptiness for CS^* can be done in a polynomial time in the number of nodes in a MTZDD. We represent the payoff of an agent as the distance of its high edge. For terminal node t, its shortest distance to the root node represents the minimal total reward of coalition S, where $v(S) = r(t)$. The non-blocking condition requires that, for each terminal node t, its shortest distance to the root node is at least $r(t)$. Let $dis(u)$ represent the shortest distance from root node u_0 to node u.

Definition 6 *The following LP formulation gives an element in the ϵ-core:*

$$
\begin{aligned}
&\min \ \epsilon \\
&\quad \textit{s.t.} \quad dis(u_0) = 0, \\
&\qquad \textstyle\sum_{i \in A} \pi_i = V(CS^*), \\
&\qquad \forall u \in V \setminus \{u_0\} \cup T, \forall u' \in Pa(u), \\
&\qquad\quad dis(u) \leq dis(u') + \pi_{agent(u')} \text{— if } (u', u) \in H, \\
&\qquad\quad dis(u) \leq dis(u') \qquad \text{— otherwise}, \\
&\qquad \forall t \in T, dis(t) + \epsilon \geq r(t).
\end{aligned}
$$

Theorem 5 *By using a MTZDD representation, determining whether the core is non-empty can be done in polynomial time in the number of nodes in a MTZDD, assuming that the value of an optimal coalition structure $V(CS^*)$ is given.*

Proof *To examine whether the core is non-empty, it is sufficient to check whether a solution of the above LP problem is 0 or less. The LP can be solved in polynomial time in the number of its constraints, which is given as $2|V| + |T|$.* □

5.2 Core-Membership

For given payoff vector π, we need to examine whether π is in the core. Assuming the value of an optimal coalition $V(CS^*)$ is given, checking the feasibility condition is easy. For each terminal node $t \in T$, where $r(t) > 0$, similar to checking the core-non-emptiness, the non-blocking condition holds if the shortest path $dis(t)$ from the root node to terminal node t is the value of path $r(t)$ or more.

Theorem 6 *By using a MTZDD representation, determining whether a payoff vector π is in the core can be done in $O(|V|)$ time, assuming the value of an optimal coalition structure $V(CS^*)$ is given.*

Proof *A MTZDD is a single-source directed acyclic graph (DAG). Thus, for each terminal node, we can find the distance from the root node using the DAG-shortest paths algorithm, which requires $O(|V| + |H| + |L|)$ time. In a MTZDD, since each internal node has one high-edge and one low-edge, $|V| = |H| = |L|$ holds. It requires $O(|V|)$ time.* □

5.3 The Cost of Stability

Definition 7 *The following LP formulation gives the cost of stability Δ:*

$$
\begin{aligned}
&\min \ \Delta, \\
&\quad \textit{s.t.} \quad dis(u_0) = 0, \\
&\qquad \textstyle\sum_{i \in A} \pi_i = V(CS^*) + \Delta, \\
&\qquad \forall u \in V \setminus \{u_0\} \cup T, \forall u' \in Pa(u), \\
&\qquad\quad dis(u) \leq dis(u') + \pi_{agent(u')} \text{— if } (u', u) \in H, \\
&\qquad\quad dis(u) \leq dis(u') \qquad \text{— otherwise}, \\
&\qquad \forall t \in T, dis(t) \geq r(t).
\end{aligned}
$$

Theorem 7 *By using a MTZDD representation, the cost of stability can be obtained in polynomial time in the number of nodes in a MTZDD, assuming that the value of optimal coalition structure $V(CS^*)$ is given.*

Proof *The cost of stability can be obtained by solving the above LP formulation. The LP can be solved in polynomial time in the number of its constraints, i.e., $2|V| + |T|$.* □

6 Experimental Evaluations

In order to show that our proposed CSG algorithm is reasonably efficient and scalable, we experimentally evaluate its performance, in comparison with the MIP formulation using a SCG representation [Ohta *et al.*, 2009]. The simulations were run on a Xeon E5540 processor with 24-GB RAM. The test machine ran Windows Vista Business x64 Edition SP2. We used CPLEX 12.1, a general-purpose MIP package.

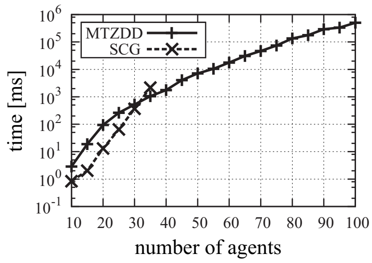


Figure 6: Computation time

We generated problem instances with 5 different groups of symmetric agents. First, we created a set of *abstract rules*. Each rule specifies the required number of agents in each group, which is generated using a decay distribution as follows. Initially, the required number of agents in each group is set to zero. First, we randomly chose one group and incremented the required number of agents in it by one. Then, we repeatedly chose a group randomly and incremented its required number of agents with probability α until a group is not chosen or the required number of agents exceeds the limit ($\alpha = 0.55$) . For each rule, we randomly chose an integer value from $[1, 10]$ as the value of the coalition. The number of abstract rules is set equal to the number of agents. Then, we translated these abstract rules into a MTZDD representation. The MIP formulation using a SCG representation is also generated from these abstract rules. Figure 6 shows the median computation times for solving the generated 50 instances.

When $n \leq 30$, a SCG representation is more efficient than a MTZDD representation for finding an optimal coalition structure, while a MTZDD representation eventually outperforms the SCG for $n > 30$. When the number of coalitions in a SCG is relatively small, the MIP formulation of a SCG representation is simple and CPLEX can reduce the search space efficiently. However, the number of coalitions in a SCG grows exponentially based on the increase of the number of agents/rules. For $n \geq 40$, generating problem instances becomes impossible due to insufficient memory. On the other hand, the number of nodes in a MTZDD grows linearly based on the increase of the number of agents/rules. As a result, the computation time for a MTZDD representation grows more slowly compared to the SCG.

7 Conclusion

We developed a new representation scheme by integrating a ZDD data structure and an existing compact representation scheme called SCG. A ZDD is an efficient data structures applied in various domains in AI. We showed that our MTZDD representation scheme (i) is fully expressive, (ii) can be exponentially more concise than SCG representation, (iii) can solve core-related problems in polynomial time in the number of nodes, and (iv) can solve a CSG problem reasonably well by utilizing a MIP formulation.

Future work includes overcoming the complexity of solving other problems including the Shapley value in coalitional games. We will also consider applying BDD/ZDD-based graphical representation for characteristic functions in non-transferable utility coalitional games.

References

[Aadithya *et al.*, 2011] K. Aadithya, T. Michalak, and N. R. Jennings. Representation of Coalitional Games with Algebraic Decision Diagrams. In *AAMAS*, pages 1121–1122, 2011.

[Airiau and Sen, 2010] S. Airiau and S. Sen. On the stability of an optimal coalition structure. In *ECAI*, pages 203–208, 2010.

[Akers, 1978] S. B. Akers. Binary decision diagrams. *IEEE Transactions on Computers*, C-27(6):509–516, 1978.

[Aumann and Dreze, 1974] R. J. Aumann and J. H. Dreze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3:217–237, 1974.

[Bacchus and Grove, 1995] B. Bacchus and A. Grove. Graphical models for preference and utility. In *UAI*, pages 3–10, 1995.

[Bachrach *et al.*, 2009] Y. Bachrach, R. Meir, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The Cost of Stability and Its Application to Weighted Voting Games. In *SAGT*, pages 122–134, 2009.

[Berghammer and Bolus, 2010] R. Berghammer and S. Bolus. Problem Solving on Simple Games via BDDs. In *COMSOC*, 2010.

[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.

[Elkind *et al.*, 2008] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. In *AAMAS*, pages 1007–1014, 2008.

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

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

[Minato, 1993] S. Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. In *Proc. of the 30th Design Automation Conference (DAC)*, pages 272–277, 1993.

[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 *CP*, pages 623–638, 2009.

[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 *AAMAS*, pages 757–764, 2010.