ICIC Express Letters
Part B: Applications
Volume **7**, Number **1**, January **2016**

 ISSN 2185-2766
pp. 203–208

SOLVING A COALITION STRUCTURE GENERATION PROBLEM CONSIDERING ROBUSTNESS BY USING A COALITION LATTICE

Katsuya Nakano, Shun Shiramatsu, Tadachika Ozono and Toramatsu Shintani

Department of Computer Science and Engineering
Graduate School of Engineering
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan
{ nkatsu; siramatu; ozono; tora }@toralab.org

Received August 2015; accepted November 2015

Abstract. *Forming effective coalitions is a major issue in multiagent systems. Coalition structure generation (CSG) involves partitioning a set of agents into teams, i.e., coalitions. The goal of CSG is forming the coalition structure that maximizes the social surplus, i.e., the sum of utilities obtained by forming coalitions. Recalculating the optimal coalition structure should be avoided when agents leave the coalition structure because CSG is NP-hard. The target of this research is to form a robust coalition structure considering agents leaving. The robustness of a coalition structure is a new important aspect. In this paper, we propose a new CSG problem considering robustness ($RCSG^-$), which has the property that the utility of any coalitions is non-negative whenever any agents leave the coalitions. We describe a coalition lattice, a data structure for calculating the robustness of a coalition, and moreover we present its evaluations. We found that the solver using the coalition lattice can solve $RCSG^-$ faster than CSG even if considering robustness and the quality of the $RCSG^-$ solution is semi-optimal.*
Keywords: Robust coalition structure generation, Robustness of coalitions, Coalition lattice

1. **Introduction.** Forming effective coalitions is a major issue in multiagent systems. Coalition structure generation (CSG) involves partitioning a set of agents into teams, i.e., coalitions [1, 2]. A coalition structure is a set of coalitions. The goal of CSG is forming the coalition structure that maximizes the social surplus, the sum of utilities obtained by forming coalitions.

Algorithms to form the optimal coalition structure for CSG have been proposed. Rahwan and Jennings proposed an algorithm based on improved dynamic programming (IDP) [3]. Rahwan et al. proposed Integer Partition (IP) algorithm that is one example of anytime algorithms [4]. Rahwan and Jennings proposed an algorithm that consists of IP and IDP [5]. Michalak et al. proposed a decentralized algorithm for CSG [6].

The robustness of coalition structures is a new important problem. The robustness of a coalition structure is the property that the social surplus is kept at the maximum when any agents leave the coalition structure [7]. Robust coalition structure generation (RCSG) is CSG focused on the robustness of coalition structures. RCSG is tight to find feasible solutions. Okimoto et al. proposed the framework for robust team formation problem (RTFP). RTFP is the problem of forming the best possible team to accomplish some tasks of interest, given some limited resources [8]. Okimoto et al. presented the computational complexity of RTFP that the order of computational complexity is not increased even if we consider the robustness of teams.

We focus on the new robustness of coalitions to solve CSG considering robustness ($RCSG^-$). The goal of $RCSG^-$ is forming a semi-optimal coalition structure that consists of robust coalitions, we call such a coalition structure *a robust solution*. A robust coalition in a robust solution is non-negative whenever any agents leave the coalition. In our prior study, we developed a coalition lattice that is a novel data structure to find robust coalitions [9]. In this paper, we demonstrate that the coalition lattice can efficiently find a robust and semi-optimal solution of RCSG$^-$.

2. **RCSG$^-$: Coalition Structure Generation Considering Robustness.** The coalition structure generation considering robustness (RCSG$^-$) is one of coalition structure generation (CSG) considering the robustness of the solution. For $A = \{a_1, a_2, \ldots, a_n\}$ as a set of n agents, a subset of agents, i.e., a coalition, is denoted by $S \subseteq A$. If a coalition structure CS is a partition of A, CS satisfies (1).

$$\forall i, j (i \neq j), \quad S_i \cap S_j = \emptyset, \quad \bigcup_{S_i \in CS} S_i = A \tag{1}$$

Each agent belongs to only one coalition at a time. A characteristic function $v : 2^A \rightarrow \mathbb{R}$ is provided, where $\mathbb{R}$ is the cooperation utility among agents in coalition S, denoted by $v(S)$. The characteristic function v is assumed to be calculated in polynomial time. The CS utility, which corresponds to the social surplus, is denoted by $V(CS)$. The value of $V(CS)$ is calculated by (2).

$$V(CS) = \sum_{S_i \in CS} v(S_i) \tag{2}$$

The optimal coalition structure CS^* satisfies (3).

$$\forall CS : V(CS) \leq V(CS^*) \tag{3}$$

Let k be a non-negative integer and A' be a subset of A. Then, CS_k is the k-robust coalition structure, if any coalition structure CS' of $A \setminus A'$ does not satisfy (4), where $k \leq |A'|$ $(0 \leq k \leq |A| - 2)$ [7]. In CSG, any coalition structure could be the 0-robust coalition structure, and $0 \leq k \leq |A| - 2$ because all coalition structures are $(|A| - 1)$-robust coalition structures.

$$V(CS_k \setminus A') < V(CS') \tag{4}$$

${}_{|A|}C_k$ patterns must be considered to determine whether the coalition structure is the k-robust coalition structure CS_k. Figure 1 presents an example evaluation of a 1-robust

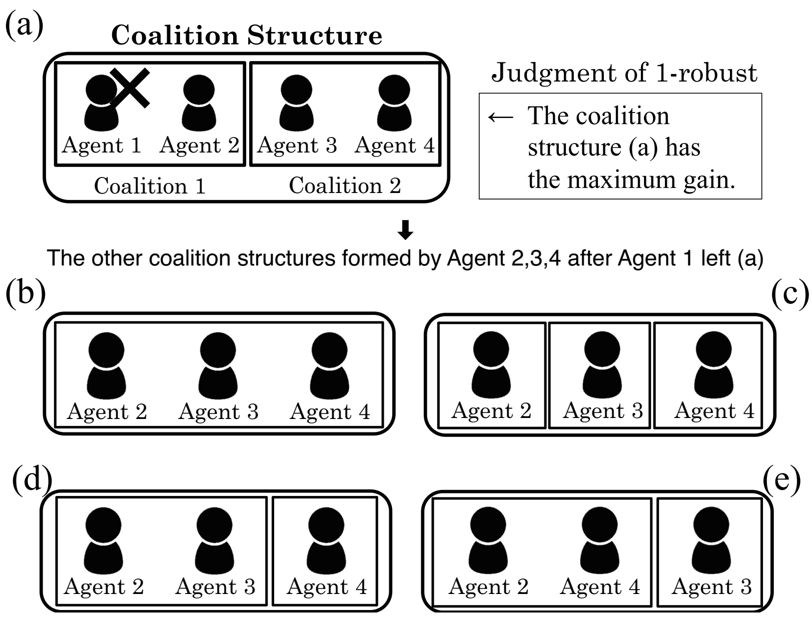

FIGURE 1. The example of the judgment of 1-robust

coalition structure where the number of agents is four, i.e., $|A| = 4$. The coalition structure (a) results from "Agents 1 to 4" (Figure 1). For (a) to be a 1-robust coalition structure, the social surplus of (a) should be maximum, even if any agents leave (a). If, for example (Figure 1), "Agent 1" leaves (a), the social surplus would result from the coalition structures formed by "Agents 2 to 4." In Figure 1, the coalition structures (b), (c), (d), and (e) are coalition structures of "Agents 2 to 4." The social surplus of (a) without "Agent 1" should be more than the social surplus of (b), (c), (d), and (e). Moreover, the social surplus of (a) after any other agent's withdrawal should be the greatest.

The goal of RCSG$^-$ is forming a semi-optimal coalition structure CS_k^+ that consists of robust coalitions. The coalition structure CS_k^+ is semi-optimal whenever any k agents leave from the coalition structure. We define the goal of RCSG$^-$ as follows:

$$\arg\max_{CS_k^+} V(CS_k^+) \text{ where } \forall S \in (CS_k^+ \setminus A'),\ v(S) - \sum_{a_i \in S} v(\{a_i\}) > 0,\ k \geq |A'|$$

3. **Robustness of Coalitions.** Let a be an agent. The utility of coalition $\{a\}$ is denoted by $v(\{a\})$. In this study, a coalition S_P is positive if satisfying (5), and a coalition S_N is negative if satisfying (6). A coalition of a single agent is positive exceptionally.

$$v(S_P) - \sum_{a_i \in S_P} v(\{a_i\}) > 0 \quad (5)$$

$$v(S_N) - \sum_{a_i \in S_N} v(\{a_i\}) < 0 \quad (6)$$

CSG and RCSG$^-$ are the same with respect to forming the coalition structure that increases the social surplus. For example, let CS_1 be a coalition structure including negative coalition $S_{N1} = \{a_1, a_4\}$, and let CS_2 be a coalition structure including $\{\{a_1\}, \{a_4\}\}$ instead of S_{N1}. Then $V(CS_1)$ is less than $V(CS_2)$. A coalition structure including one or more negative coalitions would not be optimal. In CSG, we can find the optimal coalition structure efficiently by considering only positive coalitions. In RCSG$^-$, an additional element is needed, i.e., we need to consider the robustness of coalition structures. Computational complexity becomes clearly more enormous in RCSG$^-$ than in CSG. Because we need to consider all cases that any k agents leave the coalition structure CS to distinguish whether CS is the robust solution or not.

At most, k agents leave one of the coalitions in CS. Let S' be a subset of S, where S' is formed by remaining agents except k agents in S. We should consider the utility of S'. In this study, we focus on the robustness of coalitions to find the robust solution efficiently. Now, we define that the robustness of S is the property that S' is not a negative coalition if any agents leave S. The value of k, i.e., the robustness of coalitions, is calculated by (7). In (7), $S_N^* \subset S_P$ satisfies $\forall S_N \subset S_P : |S_N^*| \geq |S_N|$. Then we call S_P a k-robust coalition. When S is a k-robust coalition, S' ($\subset S$) is not a negative coalition if any k agents leave S. We can find the robust solution in RCSG$^-$ by using k-robust coalitions.

$$k = |S_P| - |S_N^*| - 1 \quad (7)$$

4. **Coalition Lattice.** To calculate the robustness of coalitions, it is necessary to clarify the inclusive relation of each coalition. We proposed the coalition lattice that is a new data structure for the robustness of coalitions [9]. The coalition lattice is similar to the Hasse diagram. In the coalition lattice, a node represents a coalition described by a characteristic function. The coalition lattice is the data structure that connects two nodes through an inclusive relationship represented by an arc.

S	$v(S)$	P/N
$\{a_1\}$	1	P
$\{a_2\}$	5	P
$\{a_3\}$	3	P
$\{a_4\}$	4	P
$\{a_1, a_2\}$	8	P
$\{a_1, a_3\}$	5	P
$\{a_1, a_4\}$	3	N
$\{a_2, a_3\}$	5	N
$\{a_2, a_4\}$	4	N
$\{a_3, a_4\}$	9	P
$\{a_1, a_2, a_3\}$	10	P
$\{a_1, a_2, a_4\}$	7	N
$\{a_1, a_3, a_4\}$	10	P
$\{a_2, a_3, a_4\}$	8	N
$\{a_1, a_2, a_3, a_4\}$	19	P

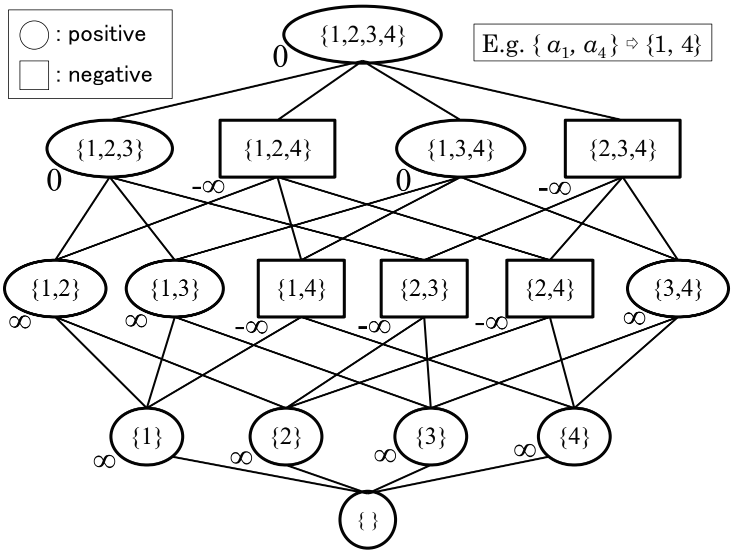

FIGURE 2. Left: a characteristic function, Right: a coalition lattice

4.1. **Data structure.** The left side of Figure 2 offers a summary of the characteristic functions when $|A| = 4$ ($A = \{a_1, a_2, a_3, a_4\}$), as an example. The corresponding coalition lattice is presented in the right side of Figure 2. In the coalition lattice, the minimum upper bound is assumed the whole coalition, whereas the maximum lower bound would be the empty set. The coalition lattice divides the agents in four levels with the number of agents, as in the right side of Figure 2. In the right side of Figure 2, a circular node represents a positive coalition, and a rectangle node shows a negative coalition. The left lower value of each node represents the robustness of the coalition. For example, if the value of a coalition is zero, then the coalition allows no agents to leave. The robustness of each coalition is calculated easily by constructing a coalition lattice, as in the right side of Figure 2, which provides the inclusive relationship of each individual coalition.

We describe a method to calculate the robustness of each coalition on the coalition lattice. Let CS_1^* be the optimal coalition structure where CS_1^* includes the coalition $\{a_4\}$. When a_4 leaves CS_1^*, $V(CS_1^* \setminus \{a_4\})$ is kept maximum because other coalitions in CS_1^* are not affected by the secession of a_4 at all. Therefore, the robustness of the coalition formed by a single agent is infinity, i.e., $k = \infty$. In addition, the robustness of the negative coalition is $k = -\infty$ as a matter of convenience. When a negative coalition exists in the child node of the positive coalition S_P on the coalition lattice, the robustness of S_P is calculated by (7). If no negative coalitions exist in the child nodes of S_P, $k_{\min}$ is the minimum value of k in the child nodes of S_P, and $S_{\min}$ is the coalition with $k_{\min}$, then the difference in the number of agents between S_P and $S_{\min}$ is M, where $M = |S_P| - |S_{\min}|$. Even if M agents leave S_P, the S_P without M agents is not a negative coalition. Therefore, the robustness of S_P is calculated by (8), when there are no negative coalitions in the child nodes of S_P.

$$k = k_{\min} + |S_P| - |S_{\min}| \tag{8}$$

4.2. **Algorithm.** We describe an algorithm to construct the coalition lattice in the left side of Figure 3. In a coalition lattice CL, A represents the set of agents, S the coalition, and l the robustness value. CL is a set of $\langle i, S, l\rangle$. $\forall S \subset A$ are sorted, in ascending order, by the number of agents in S. The index i identifies the coalitions.

CL is initialized in lines 1 to 4 in the left side of Figure 3. In line 3, the function $GetS(i)$ returns S corresponding to i. The value of i from 1 to $|A|$ represents the case where S is formed by a single agent. In a coalition lattice, the robustness of the coalitions

Require: $k \geq 0$
Ensure: Necessary parts of CL
1: $CL \leftarrow \{\}$
2: **for** $i = 0$ to $|A|$ **do**
3: $\quad CL \leftarrow CL \cup \{\langle i, GetS(i), +\infty \rangle\}$
4: **end for**
5: **for** $i = |A| + 1$ to $2^{|A|} - 1$ **do**
6: $\quad S \leftarrow GetS(i)$
7: $\quad$ **if** S is positive **then**
8: $\quad\quad CL \leftarrow CL \cup \{\langle i, S, Calculate(S) \rangle\}$
9: $\quad$ **else if** S is negative **then**
10: $\quad\quad CL \leftarrow CL \cup \{\langle i, S, -\infty \rangle\}$
11: $\quad$ **end if**
12: **end for**
13: $CL \leftarrow \{\langle i, S, l \rangle\} | k \leq l, \langle i, S, l \rangle \in CL\}$

Require: $|S| > 1$
Ensure: Value of robustness of S
1: $l_{\min} \leftarrow +\infty$
2: **for all** $\langle S_{child}, l_{child} \rangle \in Child(S)$ **do**
3: $\quad$ **if** S_{child} is negative **then**
4: $\quad\quad l' \leftarrow |S| - |S_{child}| - 1$
5: $\quad$ **else**
6: $\quad\quad l' \leftarrow l_{child} + |S| - |S_{child}|$
7: $\quad$ **end if**
8: $\quad$ **if** $l_{\min} > l'$ **then**
9: $\quad\quad l_{\min} \leftarrow l'$
10: $\quad$ **end if**
11: **end for**

FIGURE 3. An algorithm to construct a coalition lattice

that are formed by a single agent is $+\infty$. For the coalition with more than two agents, a coalition lattice is constructed in bottom up (as described in lines 5 to 12 in the left side of Figure 3). For all coalitions, the algorithm distinguishes S whether it is a positive or negative coalition, and adds each node to CL in lines 7 to 11. If S is a positive coalition, the algorithm adds the node to CL after calculating the robustness of S by the function $Calculate(S)$ (described in the right side of Figure 3). If S is a negative coalition, the algorithm adds the node to CL and the robustness of S is $-\infty$ in line 10. The function $Calculate(S)$ checks all child nodes of S and calculates the robustness of S by (7) or (8). Finally, the function $Calculate(S)$ returns the minimum value of the robustness as the robustness of S.

The required coalitions to find the robust solution are not all coalitions in a coalition lattice. To find the robust solution, we should use only the coalitions that the robustness is greater than k. In line 13, the algorithm prunes the extra part of the coalition lattice by using the value of k.

Now, we describe an RCSG$^-$ solver by using the coalition lattice that consists of three steps. First, the RCSG$^-$ solver generates the coalition lattice from a given problem. The coalition lattice can be used to find robust coalitions. Second, the solver makes a new CSG problem from the only robust coalitions, it reduces the search space dramatically. Finally, the solver can solve the new CSG problem by using an ordinary CSG solver. The optimal solution of the new CSG problem is the robust solution of RCSG$^-$, i.e., a semi-optimal solution.

5. **Evaluation.** We compared the execution time and the solution quality of RCSG$^-$ with CSG. We implemented a CSG solver based on Branch and Bound algorithm to find CS^* and an RCSG$^-$ solver based on the coalition lattice and the same CSG solver.

We examined the execution time for solving CSG and RCSG$^-$ where $|A| = 15, 16, \ldots, 22$ and the robustness $k = 3$. The environment was 3.5GHz 6-Core Intel Xeon E5, 32GB memory, Java 1.8.0_31, and OS X 10.10.3. The result is shown in Figure 4. The vertical axis indicates the average execution time of 100 iterations and the horizontal axis indicates $|A|$. Notably, the execution time of both solvers is very high, when there are a lot of coalitions. As $|A|$ increases, the difference between the execution times of CSG and RCSG$^-$ solvers is decreasing because the RCSG$^-$ solver converts a problem into the smaller problem by pruning non-robust coalitions on the coalition lattice. For example, the proposed method, i.e., the RCSG$^-$ solver, was 3.64 times faster than the normal CSG

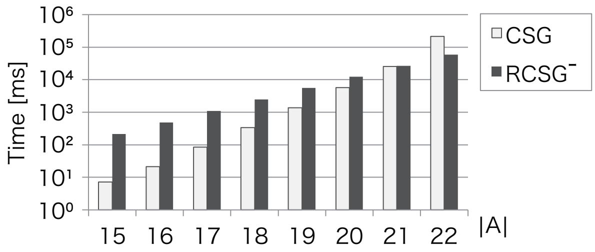

FIGURE 4. The comparison of execution times

solver even if considering robustness where $|A| = 22$. Therefore, we suppose that the RCSG$^-$ solver outperforms the CSG solver where $|A| \geq 23$.

We also evaluated the quality of RCSG$^-$ solutions defined as $V(CS_k^+)/V(CS^*)$. We generated 5,000 problems, $|A| = 15$ and the robustness $k = 3$, and compared between the solutions of CSG and RCSG$^-$. The average quality was 0.913. Therefore, the quality of the RCSG$^-$ solution is semi-optimal.

6. **Conclusions.** We found that the RCSG$^-$ solver using the coalition lattice can solve RCSG$^-$ faster than CSG even if considering robustness and the RCSG$^-$ solution is semi-optimal. We described the coalition lattice, a data structure that enables the calculation of the robustness of each coalition in RCSG$^-$. The coalition lattice represents our new robustness for CSG, a robust coalition is non-negative whenever any agents leave from the coalition. A k-robust coalition consists of at least $(k-1)$-robust coalitions in the lattice. The coalition lattice can be built from the bottom-up.

Our experiments demonstrate that the proposed method, the RCSG$^-$ solver using the coalition lattice, can efficiently find semi-optimal solution. The proposed method was 3.64 times faster than a normal CSG solver even if considering robustness where $|A| = 22$. Moreover, the quality of solutions for RCSG$^-$ was 0.913 where $|A| = 15$. The coalition lattice has the potential to overcome weakness of CSG.

REFERENCES

[1] T. Sandholm, K. Larson, M. Andersson, O. Shehory and F. Tohmé, Coalition structure generation with worst case guarantees, *Artificial Intelligence*, vol.111, nos.1-2, pp.209-238, 1999.
[2] Y. Bachrach, P. Kohli, V. Kolmogorov and M. Zadimoghaddam, Optimal coalition structure generation in cooperative graph games, *AAAI-13*, pp.81-87, 2013.
[3] T. Rahwan and N. R. Jennings, An improved dynamic programming algorithm for coalition structure generation, *AAMAS 2008*, pp.1417-1420, 2008.
[4] T. Rahwan, S. D. Ramchurn, V. D. Dang, A. Giovannucci and N. R. Jennings, Anytime optimal coalition structure generation, *AAAI-07*, pp.1184-1190, 2007.
[5] T. Rahwan and N. R. Jennings, Coalition structure generation: Dynamic programming meets anytime optimization, *AAAI-08*, pp.156-161, 2008.
[6] T. Michalak, J. Sroka, T. Rahwan, M. Wooldridge, P. McBurney and N. R. Jennings, A distributed algorithm for any coalition structure generation, *AAMAS 2010*, pp.1007-1014, 2010.
[7] T. Okimoto, N. Schwind and K. Inoue, A study for robust coalition structure generation problem, *JSAI 2014*, 2014 (in Japanese).
[8] T. Okimoto, N. Schwind, M. Clement, T. Ribeiro, K. Inoue and P. Marquis, How to form a task-oriented robust team, *AAMAS 2015*, pp.395-403, 2015.
[9] K. Nakano, S. Shiramatsu, T. Ozono and T. Shintani, Coalition lattice: A data structure considering robustness for robust coalition structure generation problem, *ICIAE2015*, pp.24-29, 2015.