Proceedings of the Twelfth International
Symposium on Combinatorial Search (SoCS 2019)

An Improved Algorithm for Optimal Coalition Structure Generation

Narayan Changder, Samir Aknine, Animesh Dutta
Department of Computer Science and Engineering, NIT Durgapur, India, narayan.changder@gmail.com
LIRIS Laboratory Claude Bernard University, Lyon 1, France, samir.aknine@univ-lyon1.fr
Department of Computer Science and Engineering, NIT Durgapur, India,animeshnit@gmail.com

Abstract

The Coalition Structure Generation (CSG) problem is a partitioning of a set of agents into exhaustive and disjoint coalitions to maximize social welfare. The fastest exact algorithm to solve the CSG problem is ODP-IP (Michalak et al. 2016). In this paper, we propose a modified version of IDP (Rahwan and Jennings 2008) (named MIDP) and an improved version of IP (Rahwan et al. 2007) (named IIP). Based on these two improved algorithms, we develop a hybrid version (MIDP-IIP) to solve the CSG problem. After a description of the new algorithm MIDP-IIP, the results of the experimental comparison against ODP-IP are provided. Our analysis shows that MIDP-IIP performs fewer operations than ODP-IP. In addition, MIDP-IIP reduced significantly many problem instances running times (11% to 37%).

The optimal CSG problem formulation

Coalition formation can be applied to many real-world problems such as task allocation, airport slot allocation, and social network analysis. ODP-IP (Michalak et al. 2016) algorithm is the fastest exact algorithm for the CSG to date in practice. Given a set of n agents $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, a coalition C_i is a non-empty subset of $\mathcal{A}$. A coalition structure ($\mathcal{CS}$) over $\mathcal{A}$ is a partitioning of $\mathcal{A}$ into a set of disjoint coalitions $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$, where $k \in \{1, \ldots, n\}$ is called the size of the coalition structure i.e. $k = |\mathcal{CS}|$. In other words, $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ satisfies the following constraints:1) $\mathcal{C}_i$, $\mathcal{C}_j \neq \emptyset$, $i, j \in \{1, 2, \ldots, k\}$. 2) $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$, for all $i \neq j$. and 3) $\bigcup_{i=1}^{k} \mathcal{C}_i = \mathcal{A}$. The value of any coalition structure $\mathcal{CS}$ is defined by $v(\mathcal{CS}) = \sum_{\mathcal{C}_i \in \mathcal{CS}} v(\mathcal{C}_i)$. The optimal solution of CSG is an optimal coalition structure $\mathcal{CS}^* \in \Pi^{A}$. The set of all coalition structures over $\mathcal{A}$ is denoted as $\Pi^{\mathcal{A}}$. Thus, $\mathcal{CS}^* = \arg\max_{\mathcal{CS} \in \Pi^{\mathcal{A}}} v(\mathcal{CS})$.

MIDP-IIP algorithm

The MIDP-IIP algorithm is a hybrid version of MIDP and IIP algorithms. Before further discussion on MIDP-IIP, we explain MIDP algorithm. Let, P_t be the partition table,

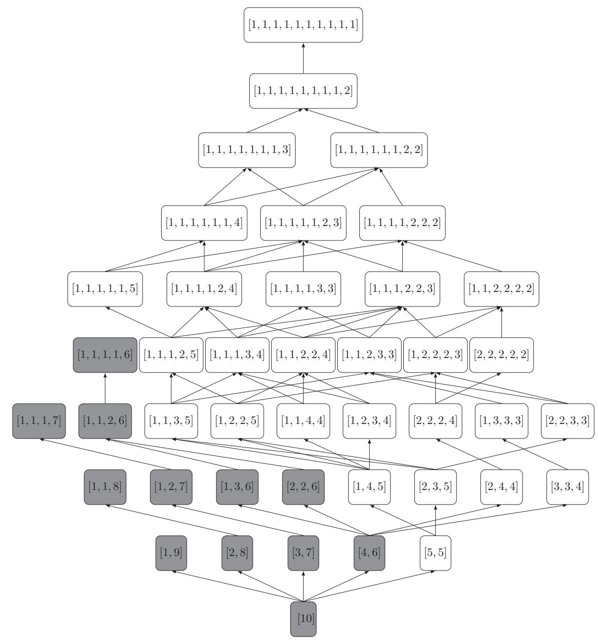

Figure 1: Searched subspaces after evaluation of all coalitions of size 4, given 10 agents. Gray colored subspaces are fully searched by MIDP, whereas white colored subspaces are not yet searched.

$P_t(\mathcal{C})$ stores one optimal partition of each coalition $\mathcal{C}$ and V_t be the optimal value table, $V_t(\mathcal{C})$ stores the optimal value of the coalition $\mathcal{C}$. MIDP produces two tables P_t and V_t using the below recursion (cf. equation 1).

Let $\mathcal{C}'' = \left\{\mathcal{C}' | \mathcal{C}' \subset \mathcal{C} \text{ and } 0 \leq |\mathcal{C}'| \leq \frac{|\mathcal{C}|}{2}\right\}$, table V_t for each coalition $\mathcal{C}$ is constructed as follows:

$$V_t(\mathcal{C}) = \begin{cases} v(\mathcal{C}) & \text{if } |\mathcal{C}| = 1 \\ \arg\max_{\mathcal{C}' \in \mathcal{C}''} \{V_t(\mathcal{C}') + V_t(\mathcal{C} \setminus \mathcal{C}')\} & \text{otherwise} \end{cases} \tag{1}$$

Now, we explain how the IIP algorithm works. The IIP algorithm divides the whole search space of the CSG problem into different subspaces (cf. Figure 1). It is possible to compute an upper bound and a lower bound on the values of all the coalition structures in each subspace. IIP sorts all these subspaces according to the upper bound values and starts searching them one by one until all the subspaces are searched. IIP algorithm searches any subspace in depth-first manner. Let's say IIP is now searching a subspace $[i_1, i_2, \ldots, i_k]$. IIP algorithm first iterates over all coalitions $\mathcal{C}^{i_1}$ of size i_1. Next for each coalition $\mathcal{C}_{x_1} \in \mathcal{C}^{i_1}$, IIP iterates over all coalitions $\mathcal{C}_{x_2} \in \mathcal{C}^{i_2}$ of size i_2 that does not overlap with $\mathcal{C}_{x_1}$. Similarly, IIP iterates over all coalitions $\mathcal{C}_{x_3} \in \mathcal{C}^{i_3}$ of size i_3 that does not overlap with the coalition $\mathcal{C}_{x_1} \cup \mathcal{C}_{x_2}$, and so on. This process is repeated until the last coalition of size i_k is picked. Using this process all the coalition structures in the subspace $[i_1, i_2, \ldots, i_k]$ are searched. After generating d coalitions $\mathcal{C}_{x_1} \in \mathcal{C}^{i_1}, \ldots, \mathcal{C}_{x_d} \in \mathcal{C}^{i_d}$, and before iterating over the next feasible coalitions of size $d+1, \ldots, k$, IIP checks the inequality 2.

$$P_t(\{\mathcal{C}_d\}) \neq \{\mathcal{C}_d\} \tag{2}$$

If the inequality 2 holds, then all the coalition structures composed of the coalitions $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_d$ can be skipped during IIP's search, because the coalition $\mathcal{C}_d$ cannot be part of the optimal coalition structure as the coalition $\mathcal{C}_d$ is stored in the optimal partition table P_t in two disjoint coalitions. Otherwise IIP applies the inequalities 3, 4, and 5.

Let $V(\mathcal{CS}^{**})$ denotes the best coalition structure found by the IIP algorithm at any point in time. If the inequality 3 holds, then all coalition structures composed of coalitions $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_d$ can be skipped because the coalition structures containing coalitions $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_d$ in the subspace $[i_1, i_2, \ldots, i_k]$ will always generate a coalition structure value less than $V(\mathcal{CS}^{**})$ and cannot be part of the optimal coalition structure.

$$\sum_{i=1}^{d} v(\mathcal{C}_i) + \sum_{i=d+1}^{k} Max_i < V(\mathcal{CS}^{**}) \tag{3}$$

$$\sum_{i=1}^{d} w(\mathcal{C}_i) > \sum_{j=1}^{d} v(\mathcal{C}_j) \tag{4}$$

$$w(\mathcal{C}_d) > v(\mathcal{C}_d) \tag{5}$$

If the inequality 4 holds then any coalition structure containing the coalition $\{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ cannot be the optimal coalition structure in the subspace $[i_1, i_2, \ldots, i_k]$ and all such coalition structures can be skipped. Similarly, if the inequality 5 holds then the coalition $\{\mathcal{C}_d\}$ is not part of any optimal coalition structure in the subspace $[i_1, i_2, \ldots, i_k]$. Hence, every coalition structure containing the coalition $\{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ can be skipped during IIP's search. To use the strength of MIDP in IIP's search, we use the same table w (more details are provided in (Michalak et al. 2016).) used by IIP and MIDP.

Experimental evaluation

We empirically evaluated the MIDP-IIP algorithm and benchmarked it against ODP-IP. We compared the performances of both algorithms given different numbers of agents (5 to 27). For ODP-IP, we used the code provided by the authors of ODP-IP (Michalak et al. 2016). In total, we performed 12375 tests over 11 data distributions. The experimental results show that MIDP-IIP algorithm performs well for many problem instances. In particular, we observe the following:

- Given 27 agents, with agent-based uniform, agent-based normal, gamma, exponential, NDCS, and Chi-square distributions, running time is reduced significantly by 10.88%, 11.17%, 21.95%, 26.65%, 36.17%, and 36.64% respectively when compared with ODP-IP algorithm. In these class of problems, the inequality 2 works well and IIP does not use inequalities 3, 4, and 5 frequently.
- With beta, modified-uniform, normal, uniform, and modified-normal distributions, MIDP-IIP and ODP-IP performance are almost the same. When compared MIDP-IIP with ODP-IP, we found that in this class of problems sometimes inequality 2 works and sometimes inequalities 3, 4, and 5 work.

	Time in seconds		
Distribution	ODP-IP	MIDP-IIP	Difference
	time (t_1)	time (t_2)	$t_1 - t_2$
Agent-based uniform	4126	3677	449
Agent-based normal	3269	2904	365
Chi-square	1030	632	398
NDCS	470	300	170
Exponential	409	300	109
Gamma	401	313	88

Figure 2: Effectiveness of ODP-IP and MIDP-IIP. The table shows runtime (in seconds) for 27 agents, taken for each coalition value distribution as an average over 50 runs.

The results in Figure 2 show that there are data distributions in which MIDP-IIP gives more positive synergies. Out of 11 distributions, MIDP-IIP outperforms ODP-IP on 6 distributions. Given, 27 agents, in the case of agent-based uniform distribution, MIDP-IIP took 449 seconds less time as compared to ODP-IP.

Acknowledgments

The research presented in this article is funded by "Visvesvaraya PhD Scheme for Electronics & IT", grant no: PhD-MLA/4(29)/2015-16.

References

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

Rahwan, T., and Jennings, N. R. 2008. An improved dynamic programming algorithm for coalition structure generation. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, 1417–1420. International Foundation for Autonomous Agents and Multiagent Systems.

Rahwan, T.; Ramchurn, S. D.; Dang, V. D.; Giovannucci, A.; and Jennings, N. R. 2007. Anytime optimal coalition structure generation. In *AAAI*, volume 7, 1184–1190.