BOSS: A Bi-directional Search Technique for Optimal Coalition Structure Generation with Minimal Overlapping (Student Abstract)

Narayan Changder[1], Samir Aknine [2], Sarvapali Ramchurn [3], Animesh Dutta[1]

[1] National Institute of Technology Durgapur, India.
[2] LIRIS, Lyon 1 University, France.
[3] University of Southampton,UK.
narayan.changder@gmail.com, samir.aknine@univ-lyon1.fr, sdr1@soton.ac.uk, animesh@cse.nitdgp.ac.in

Abstract

In this paper, we focus on the Coalition Structure Generation (CSG) problem, which involves finding exhaustive and disjoint partitions of agents such that the efficiency of the entire system is optimized. We propose an efficient hybrid algorithm for optimal coalition structure generation called BOSS[1]. When compared to the state-of-the-art, BOSS is shown to perform better by up to 33.63% on benchmark inputs. The maximum time gain by BOSS is 3392 seconds for 27 agents.

The BOSS Algorithm

The ODP-IP (Michalak et al. 2016) and the ODSS (Changder et al. 2020) algorithms are the fastest exact algorithms for the CSG problem. Both ODP-IP and ODSS struggle to cope with specific types of inputs which undermine their hybridization approach. This paper proposes a novel bi-directional search algorithm for the CSG problem. Given a set of n agents $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$, $\mathcal{C} = \{a_1, a_2, \dots, a_l\}$ is a coalition of agents, where $l \leq n$. Let v be a characteristic function, v assigns a real value $v(\mathcal{C})$ to each coalition $\mathcal{C}$. A coalition structure $\mathcal{CS}$ over $\mathcal{A}$ is a partitioning of $\mathcal{A}$ into $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$, where: 1) $k = |\mathcal{CS}|$. 2) $\mathcal{C}_i \neq \emptyset$, $i \in \{1, 2, \dots, k\}$. 3) $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$, for all $i \neq j$. 4) $\bigcup_{i=1}^{k} \mathcal{C}_i = \mathcal{A}$. The value of any $\mathcal{CS}$ is defined by $v(\mathcal{CS}) = \sum_{\mathcal{C}_i \in \mathcal{CS}} (v(\mathcal{C}_i))$. The optimal solution of the CSG problem is a coalition structure $\mathcal{CS}^* \in \Pi^{\mathcal{A}}$, where $\Pi^{\mathcal{A}}$ denotes the set of all the coalition structures over $\mathcal{A}$. Thus, $\mathcal{CS}^* = \arg\max_{\mathcal{CS} \in \Pi^{\mathcal{A}}} v(\mathcal{CS})$.

In BOSS, IDP (Rahwan and Jennings 2008) and IP (Rahwan et al. 2009) run in parallel. BOSS minimizes the duplicated operations performed by ODP-IP and ODSS by dividing the whole search space of CSG into $\lfloor \frac{2n}{3} \rfloor$ disjoint sets D_i, $\forall i \in \{1, 2, \dots, \lfloor \frac{2n}{3} \rfloor\}$ of subspaces using the properties P_1, P_2, P_3 observed on the integer partitions of an integer n. A partition of n is an increasing sequence of positive integers $p_1, p_2, \dots, p_k$ the sum of which is n. Each p_i is called a part of the partition.

- P_1: the highest part of a partition can be greater than $\lceil \frac{n}{2} \rceil$.
- P_2: the highest part or the subset-sum of an integer partition can be equal to $\lceil \frac{n}{2} \rceil$. Given an integer partition, the subset-sum problem is to find a subset of parts that are selected from the parts of the given integer partition, the sum of which is a given number.
- P_3: the subset-sum of a partition is not equal to $\lceil \frac{n}{2} \rceil$ and the highest part of the integer partition is less than $\lceil \frac{n}{2} \rceil$.

The subspaces following the properties P_1 and P_2 are disjoint with the subspaces following P_3. All the subspaces in the sets D_i, $\forall i \in \{1, 2, \dots \lceil \frac{n}{2} \rceil - 1\}$ follow the property P_1. The set $D_{\lceil \frac{n}{2} \rceil}$ follows P_2, while the sets D_i, $\forall i \in \{\lceil \frac{n}{2} \rceil + 1, \dots, \lfloor \frac{2n}{3} \rfloor\}$ follow P_3. First, BOSS considers each subspace $\mathcal{SP}$ in the CSG. If the largest integer in $\mathcal{SP}$ is $k > \lceil \frac{n}{2} \rceil$, then $\mathcal{SP}$ is added to the set D_{n-k}. In our example (cf. Figure 1), the subspaces $[1, 9]$, $[1, 2, 7]$ are added to D_1, and D_3 respectively. On the other hand, if $k = \lceil \frac{n}{2} \rceil$ or $k < \lceil \frac{n}{2} \rceil$ and subset-sum($\mathcal{SP}$) $= \lceil \frac{n}{2} \rceil$, then $\mathcal{SP}$ is added to $D_{\lceil \frac{n}{2} \rceil}$, where subset-sum checks whether there is a subset X of the given set $\mathcal{SP}, X \subseteq \mathcal{SP}$ and the parts of X sum to $\lceil \frac{n}{2} \rceil$. For example, given ten agents, the subspaces $[1, 1, 4, 4]$ and $[1, 1, 3, 5]$ are added to D_5 because subset-sum $([1, 1, 4, 4]) = 1 + 4 = 5$, and $\lceil \frac{n}{2} \rceil = 5$ is in $[1, 1, 3, 5]$. Next, BOSS processes the rest of the subspaces in the integer partition graph and creates the set $D_{\lceil \frac{n}{2} \rceil + 1}$, then the set $D_{\lceil \frac{n}{2} \rceil + 2}$ and so on up to the set $D_{\lfloor \frac{2n}{3} \rfloor}$ as follows. For each $\lceil \frac{n}{2} \rceil < i <= \lfloor \frac{2n}{3} \rfloor$, if subset-sum($\mathcal{SP}$) $= i$, then $\mathcal{SP}$ is added to the set D_i.

BOSS deals with the two following cases:

Case 1: IP finishes searching all the subspaces in the set D_i, and IDP finishes evaluating all the coalitions of size $i - 1$, where $i \leq \lfloor \frac{2n}{3} \rfloor$. Hence, all the subspaces have been explored. BOSS stops and returns the optimal solution.

Case 2: Both IP and IDP are searching the subspaces in a set D_i, where $i \leq \lfloor \frac{2n}{3} \rfloor$. In this case, there are still duplicated operations in the set D_i carried out by IP and IDP. As soon as one of them (IP or IDP) finishes, BOSS returns the optimal solution.

Empirical Evaluation

We empirically evaluated the BOSS algorithm and benchmarked it against the state-of-the-art (Michalak et al. 2016;

[1]BOSS stands for **B**i-directional, **O**verlapping minimization and **S**ubspace **S**hrinking

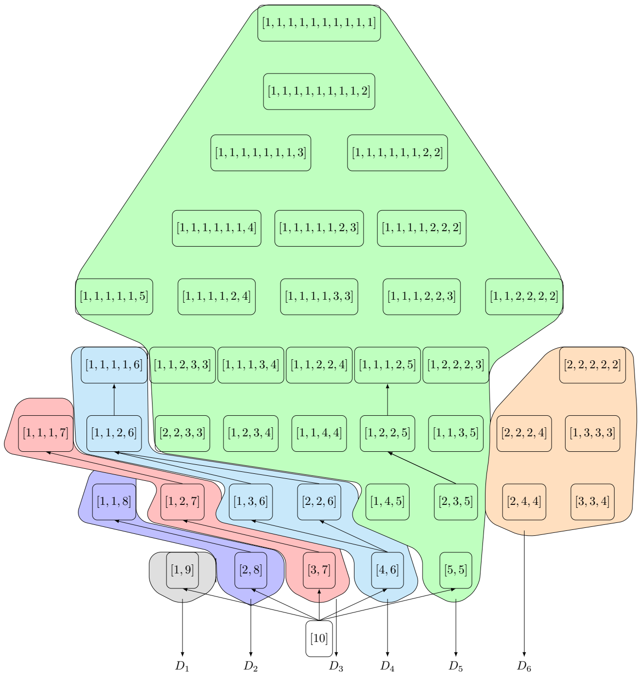

Figure 1: Subspaces of CSG given $n = 10$ agents. Each shaded area indicates the subspaces in the set D_i, $\forall i \in \{1, 2, \ldots \lfloor \frac{2n}{3} \rfloor\}$. In BOSS, IDP searches the subspaces in an increasing order starting from D_1, then D_2 and so on, whereas IP searches in a decreasing order starting from $D_{\lfloor \frac{2n}{3} \rfloor} = D_6$, then $D_{\lfloor \frac{2n}{3} \rfloor - 1} = D_5$, and so on.

Changder et al. 2020) on several value distributions. Given 27 agents, with SVA-β, F, agent-based uniform, raleigh, agent-based normal, and modified uniform distributions, the time gain with BOSS is 3392, 37, 33, 7, 6, and 6 seconds, respectively, compared to the minimum time taken by ODP-IP and ODSS. With these distributions, BOSS performs well (cf. Table 1). In the case of beta, and weibull distributions, the performance of ODP-IP, ODSS, and BOSS is the same. It is clear that the performance of BOSS is better for the most challenging input distributions. Using the subspace division technique in BOSS, IDP and IP always work on disjoint subspaces except for the meeting point. To determine whether there are any statistically significant differences between the means of the runtime of ODP-IP, ODSS, and BOSS, we perform one-way analysis of variance (ANOVA) (Green and Salkind 2011). Our null hypothesis H_0: There is no statistically significant difference between the runtimes of ODP-IP, ODSS, and BOSS. Table 2 shows the ANOVA information for each distribution. We only report three values obtained from the ANOVA test: P, F, and F_c. $P < 0.05$ and $F > F_c$ indicate that there is a very strong evidence against H_0. (1) For the first four distributions (cf. Table 2), there is a significant difference among the means of the runtime of the algorithms at the α error level 0.05. The results suggest us to reject the null hypothesis (H_0). (2) For other distributions, we failed to reject the null hypothesis H_0. This means that for these distributions, we have no evidence to suggest that the runtime means are different. As a conclusion, we observe that the improvement obtained on ODP-IP and ODSS is of the order of 33.63% for some distributions.

	Time in seconds			
Distribution	ODP-IP	ODSS	BOSS	Δt
SVA-β	10700	10087	6695	3392
F	682	227	190	37
ABU	2780	1455	1422	33
R	407	355	348	7
ABN	3075	1573	1567	6
MU	1244	102	96	6
Beta	1	1	1	0
Weibull	3	3	3	0

Table 1: Evaluating the effectiveness of ODP-IP, ODSS, and BOSS. The table shows the runtime (in seconds) for 27 agents. Δt represents min(ODP-IP time, ODSS time)-BOSS time in seconds.

	F	P	$P < \alpha$	$F > F_c$
ABU	61.79	4.8e−07	Yes	Yes
ABN	175.32	1.31e−09	Yes	Yes
SVA-β	82.29	9.8e−08	Yes	Yes
MU	5.05	0.02	Yes	Yes
Beta	0.90	0.431	No	No
Weibull	0.04	0.96	No	No
R	0.28	0.75	No	No
F	2.55	0.11	No	No

Table 2: Important ANOVA information for 8 different data distributions with α error level set to 0.05.

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

Changder, N.; Aknine, S.; Ramchurn, S. D.; and Dutta, A. 2020. ODSS: Efficient Hybridization for Optimal Coalition Structure Generation. In *AAAI*, 7079–7086.

Green, S. B.; and Salkind, N. J. 2011. *Using SPSS for the MacIntosh and Windows: Analyzing and understanding data*. Boston, MA: Prentice Hall.

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 *AAMAS*, 1417–1420.

Rahwan, T.; Ramchurn, S. D.; Jennings, N. R.; and Giovannucci, A. 2009. An anytime algorithm for optimal coalition structure generation. *JAIR* 34: 521–567.