A Multiagent Path Search Algorithm for Large-Scale Coalition Structure Generation

Extended Abstract

Redha Taguelmimt
Univ Lyon, UCBL, CNRS, INSA Lyon, Centrale Lyon, Univ Lyon 2, LIRIS, UMR5205
F-69622 Villeurbanne, France
redha.taguelmimt@gmail.com

Samir Aknine
Univ Lyon, UCBL, CNRS, INSA Lyon, Centrale Lyon, Univ Lyon 2, LIRIS, UMR5205
F-69622 Villeurbanne, France
samir.aknine@univ-lyon1.fr

Djamila Boukredera
University of Bejaia, Faculty of Exact Sciences, Laboratory of Applied Mathematics
06000 Bejaia, Algeria
djamila.boukredera@univ-bejaia.dz

Tuomas Sandholm
Carnegie Mellon University
Strategic Machine, Inc.
Strategy Robot, Inc.
Optimized Markets, Inc.
Pittsburgh, USA
sandholm@cs.cmu.edu

Narayan Changder
TCG Centres for Research and Education in Science and Technology
Kolkata, India
narayan.changder@tcgcrest.org

ABSTRACT

Coalition structure generation (CSG) is a critical problem in multiagent systems, involving the optimal partitioning of agents into disjoint coalitions to maximize social welfare. This paper introduces SALDAE, a novel multiagent path finding algorithm for CSG on a coalition structure graph. SALDAE employs various heuristics and strategies for efficient search, making it an anytime algorithm suitable for handling large-scale problems.

KEYWORDS
Coalition formation; Coalition structure generation; Multiagent path finding

ACM Reference Format:

1 INTRODUCTION

Coalition structure generation is a major problem in artificial intelligence that is central to many practical applications, including transportation [9], disaster response [15], distributed sensor networks [4], and e-commerce [14]. Several algorithms have been proposed for this problem, including optimal and approximate solutions [1–3, 5–8, 10, 12, 13, 15, 16]. In this paper we present SALDAE, a new algorithm to search the coalition structure graph with multiagent path finding. It is anytime and scales to hundreds and thousands of agents. A CSG problem defined on a set of $n$ agents $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ is a problem of size $n$. A coalition $C$ is any non-empty subset of $\mathcal{A}$. In CSG, a characteristic function $\nu$ assigns a real value to each coalition. This value determines the efficiency of the coalition. A coalition structure $CS$ is a partition of the set of agents $\mathcal{A}$ into disjoint coalitions. Formally, given a set of non-empty coalitions $\{C_1, C_2, \ldots, C_k\}, \text{CS} = \{C_1, C_2, \ldots, C_k\}$, where $k = |CS|$ and $CS$ satisfies the following constraints: $\bigcup_{j=1}^{k} C_j = \mathcal{A}$ and for all $i, j \in \{1, 2, \ldots, k\}$ where $i \neq j$, $C_i \cap C_j = \emptyset$. The value of $CS$ is assessed as: $\nu(\text{CS}) = \sum_{C \in \text{CS}} \nu(C)$. The goal in CSG is to find the optimal solution $CS^* = \text{arg max}_{CS \in \Pi(\mathcal{A})} \nu(\text{CS})$. The coalition structure graph (see Figure 1) divides the solutions into levels and each level $l$ contains coalition structures that are formed of exactly $l$ coalitions.

2 THE SALDAE ALGORITHM

In this paper, we introduce a novel path finding variant where each node represents a solution to the coalition structure generation problem—that is, a coalition structure. The objective is to identify the optimal or near-optimal coalition structure by navigating a graph with a search agent. The agent, starting at a designated node, traverses adjacent nodes, maintaining a sorted list based on coalition structure values.

Multiagent Path Finding has several variants, and our goal is to find the optimal solution using multiple search agents in the CSG context. Conflicts may arise when multiple agents evaluate the same coalition structure. While the paths taken by agents are crucial for finding high-quality solutions quickly, the optimal solution is the highest-valued coalition structure found among the nodes.

Our proposed path search algorithm uses a gradually constructed search graph (refer to Figure 1), starting from a designated start node representing a coalition structure. Each parent node in the graph connects to child nodes generated through coalition splitting or merging operations. The algorithm iteratively builds the search graph using a greedy approach.
Iteration value, becoming the new start node. Subsequent iterations involve generating child nodes, with selection based on coalition structure values. SUBSTITUTE holds nodes visited during the path exploration. Figure 1 illustrates the three phases of our algorithm. The child nodes are represented by the nodes that are directly connected to the bottom node. The coalition structure \{ (a₁, a₂), (a₂, a₃) \} in level 2 is assumed to have the highest value, becoming the new start node. Subsequent iterations involve generating child nodes, with selection based on coalition structure values. Blue edges represent the generation of coalition structure by merging two coalitions.

1. Generation (Step 1): Child coalition structures are generated from the current start node by either splitting a coalition or joining two coalitions.
2. Selection (Step 2): A start node selection procedure considers the most promising child coalition structure among a set of candidates generated in the previous step. The start node is chosen as the highest-valued child coalition structure.
3. Comparison to Incumbent (Step 3): The selected coalition structure is evaluated, and if superior to the current best solution, it becomes the new best solution.

The construction of the search graph continues iteratively, with each node representing a potential solution. The algorithm’s search process is illustrated through a 4-agent example (Figure 1), highlighting the generation, selection, and comparison steps.

The algorithm dynamically explores solutions by maintaining a list of nodes and iteratively expanding them based on the three fundamental steps. In addition to these steps, whenever a better solution is found, the algorithm constructs a path of nodes connecting the previous best solution to the new one. This path exploration aims to uncover potentially better solutions by evaluating intermediate solutions between the two. The rationale lies in the correlation between the distribution of coalitions and agents in current best solutions, as slight changes in these structures could lead to improved solutions.

SALDAE, the proposed algorithm, also employs memory management strategies that enhance its efficiency in seeking global optima. It maintains three lists of nodes: OPEN, SUBSTITUTE, and RESERVE. OPEN, constructed through the search process with the 3 steps, is sorted in descending order based on the values of coalition structures. SUBSTITUTE holds nodes visited during the path search between current best solutions, only expanded if they can improve the current solution, and RESERVE stores less promising nodes. The key strategies for memory management are:

1. OPEN Replacement:
   - If OPEN becomes empty, SUBSTITUTE replaces OPEN.
2. OPEN to RESERVE Transition:
   - If OPEN and SUBSTITUTE are empty, RESERVE replaces OPEN.
3. Conflict Resolution: Conflicts arise when agents evaluate the same coalition structure. Two conflict resolution techniques:
   - Bypassing Conflicts: Prevents using conflicting child nodes found in other agents’ lists and ensures agents explore distinct areas of the solution space.
   - Managing Conflicts: Compares rankings of conflicting child nodes in respective lists and allows the agent with higher ranking to keep and expand the node.

3 EMPIRICAL EVALUATION
A comprehensive evaluation of SALDAE’s performance is conducted, covering small-scale and large-scale benchmarks. SALDAE consistently outperforms other algorithms, such as PICS [12], FACS [11], and CSG-UCT [15], in terms of solution quality (Figure 2.a) and gain rate (Figure 2.b). It provides optimal solutions more frequently and achieves higher solution quality compared to the other algorithms. The evaluation also shows that SALDAE can handle large problems with hundreds and thousands of agents.

ACKNOWLEDGMENTS
Tuomas Sandholm is supported by the Vannevar Bush Faculty Fellowship ONR N00014-23-1-2876, National Science Foundation grants RI-2312342 and RI-1901403, and ARO award W911NF2210266.
REFERENCES


