Organization-Based Coalition Formation

Sherief Abdallah Victor Lesser
University of Massachusetts, Amherst
MAS Laboratory
{shario,lesser}@cs.umass.edu

Abstract

The coalition formation problem has received a considerable amount of attention in recent years. In this work we present a novel distributed algorithm that returns a solution in polynomial time and the quality of the returned solution increases as agents gain more experience. Our solution utilizes an underlying organization to guide the coalition formation process. We use reinforcement learning techniques to optimize decisions made locally by agents in the organization. Experimental results are presented, showing the potential of our approach.

1. Introduction

The coalition formation problem has received considerable attention in multiagent systems community [3, 2]. The input to the coalition formation problem is a set of agents, each controlling some amount of resources, and a set of tasks, each requiring some amount of resources and each worth some utility. The solution assigns subsets of agents to a subset of tasks, such that each task's requirements are satisfied and total utility is maximized.

This situation is common in domains where a task requires more than one agent and there are more than one task competing for resources. Computational grids and distributed sensor networks are examples of such domains. In computational grids a large number of computing systems are connected via a high-speed network. The goal of the grid is to meet the demands of new applications (tasks) that require large amounts of resources and reasonable responsiveness. Such requirements cannot be met by an individual computing system. Only subset of the available computing systems (aka a coalition) has enough resources to accomplish an incoming task. This paper proposes a scalable, distributed solution to the coalition formation problem.

AAMAS'04, July 19-23, 2004, New York, New York, USA.

2. Proposed Solution

We organize agents into a *hierarchy* (Figure 1), which is both distributed and scalable. This organization is then used to guide the coalition formation process more efficiently as we describe later.

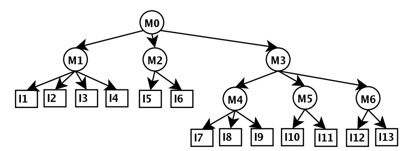

Figure 1. An Organization Hierarchy

The algorithm for local decision works as follows. M evaluates its current state s. M then selects an action a based on its policy. Each action corresponds to a child $M_i \in children(M)$. Once a child is selected, a subtask T_i of T is dynamically created based on M_i's state. M then asks M_i to form a subcoalition capable of accomplishing T_i. M_i forms a subcoalition C_{T_i} and sends a commitment back to M. M updates the overall coalition C_T and learns about this action. M updates its state, including the amount of resources to be allocated (UR) and the corresponding utility to be gained (uu).

M selects the next best action and the process continues as long as the following conditions hold: T requires more resources than currently allocated AND M still controls some unallocated resources that are required by T. At the end if enough resources are allocated to T, M adds the formed coalition C_T to its list of commitments LOC and returns C_T. Otherwise T is passed up the hierarchy.

Since managers control exponentially more individuals as we ascend in the organization, abstraction of state information is necessary to achieve scalability (otherwise we are effectively centralizing the problem). In our solution, each manager M abstracts the state of its organization, $organization(M)$ (the subtree beneath M). The price of this abstraction is loss of information (a manager higher in the hierarchy "sees" fewer details about its organization).

When a manager M selects a child M_i to ask for contribution regarding task T, M decomposes T heuristically to T_i. A manager M only sees the abstract state of its child M_i. Using this information, M needs to find T_i that is more suitable to $Organization(M_i)$. We use a heuristic-based algorithm for this.

A key factor in the performance of our system is how a manager selects its actions. In particular, in what order should a manager ask each child for its contribution? We used the Q-learning algorithm [4] with neural nets to approximate action values. We used a separate neural net for each action. This uses more memory space, but provides better approximation. Intermediate rewards are small negative rewards to reflect the communication and the processing costs of each additional step spent forming the coalition. Once a manager M successfully allocates a coalition to task T, it gains a reward equal to T's utility.

3. Experiments and Results

3.1. Setup

We compared our approach to centralized (a single manager controlling all individuals) random (CRP) and greedy (CGP) policies. We also investigated the effect of learning in an organization by comparing three distributed policies: distributed learned policy (DLP), distributed random policy (DRP), and distributed greedy policy (DGP). Finally to measure the effect of the organization structure on system performance, we collected results using different organizations, all constructed from the same population of individual agents as shown in figure below. More details can be found in [1].

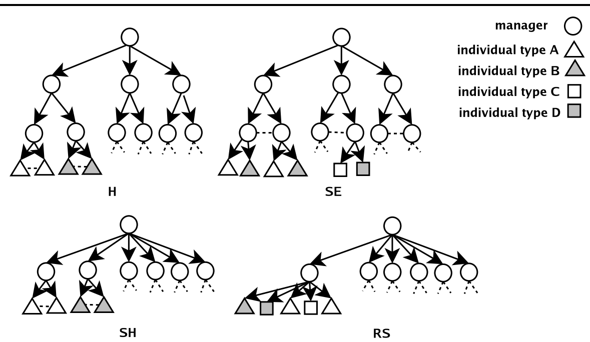

Figure 2. Different Organizations.

3.2. Results

Figure 3 illustrates how the performance of our system, DLP, improves as agents gain more experience, using different underlying organizations.

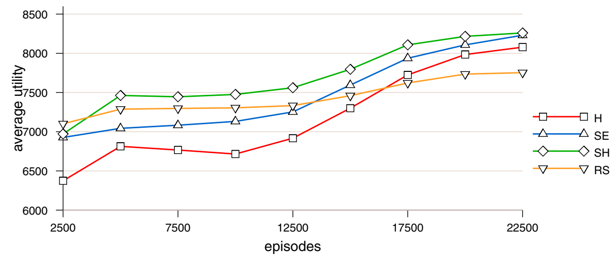

Figure 3. Learning curve.

We measured average utility for all policies. CRP achieved least average utility. DRP performed better than CRP. CGP is better than both. Our approach, DLP, outperformed all other policies for all organization structures, except when using a random organization structure. More importantly, DLP is more stable than other approaches. The standard deviation (of achieved utility) using CGP was 70% worse than DLP with SE organization. CRP was 30% worse than DLP. We had similar results with the larger agent population. DLP had the least standard deviation, which was around one third that of DGP.

4. Conclusions and Future work

Our initial results show that our approach outperformed both random and greedy policies for most of the organizations we studied. It achieved higher utility and more stability with a smaller percentage of wasted resources and fewer exchanged messages. The results also verify the scalability of our approach as it still outperforms the other approaches we studied for larger systems.

In future, we aim to study a wider variety of organizations for different types of environments. We will also investigate further our abstraction and decomposition schemes, as we believe better schemes can considerably improve the learned policy performance. We also plan to study the optimization of the underlying organization and how this interacts with optimizing the hierarchical policy.

References

[1] S. Abdallah and V. Lesser. Organization-based coalition formation. Technical report, UMASS, 2004.

[2] T. Sandholm and et al. Coalition structure generation with worst case guarantee. *Proceedings of the 3rd Internation Conference on Autonomous Agents*, 1999.

[3] O. Shehory. Methods for task allocation via agent coalition formation. *Artificial Intelligence Journal*, 101(1–2):165–200, 1998.

[4] R. Sutton and A. Barto. *Reinforcment Learning: An Introduction*. MIT Press, 1999.