Coordination in Large Scale Multi-Agent Systems for Complex Environments

(Doctoral Consortium)

James Parker
Dept. of Computer Science, University of Minnesota
Minneapolis, Minnesota, USA
jparker@cs.umn.edu

ABSTRACT

Multi-agent systems allow for robust and flexible solutions, but require complex coordination to operate efficiently. Efficient task allocation is difficult even when agents and tasks are known and unchanging, yet in many circumstances agents might fail or the environment is only partially observable. Our work explores models for large multi-agent systems in partially observable environments to efficiently reduce the complexity of the problem space and uses the RoboCup Rescue Simulator as a testbed. The first model is forming coalitions for extended period of time to allow synergy between agents and the second model is estimating aggregate behavior of growing tasks to minimize the travel time of agents.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

Keywords

Multi-robot systems, Coordination, Search and Rescue

1. INTRODUCTION

Multi-agent systems solve a wide variety of problems from wide-area surveillance to assisting in search and rescue operations. Task allocation with a fixed number agents and tasks is an NP-hard problem, so practical applications must either prune the search space heavily or find an approximate solution. Centralized approaches, such as auction based methods, suffer from a single point of failure and typically higher communication costs, while decentralized algorithms often perform worse due to more limited information.

Our work focuses on partially observable environments where the number of agents in the system can suddenly change. Both the number and completion requirements of tasks can also change as time progresses and paths are not known to be actually traversable until attempted. Due to the changing environment, the task allocation needs to have a real-time solution in which all agents can easily adapt without excess communication overhead.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

2. RESEARCH CHALLENGES

The cost of producing robots has decreased rapidly, so the bottleneck in using multi-agent systems is coordinating all the agents to efficiently accomplish the goal. For partially observable environments where agents must explore to find all the tasks, agents cannot guarantee to find an optimal solution. In many cases, due to the uncertainty in exploration, the effectiveness of an algorithm is hard to gauge.

In order to determine the effectiveness of solutions the RoboCup Rescue Simulator [1] is used, because yearly competitions provide a set of cutting edge algorithms to compare against. The RoboCup Rescue Simulator is an open source project designed as a scenario for urban search and rescue with a city scale environment and a large number of heterogeneous agents. The Paris map is shown in Figure 1 with agents as circles, roads as light colored and buildings as gray and multi-colored polygons. In the simulator the building and road locations are known at the start, but agents must explore the environment to find which roads are passable and the locations of tasks.

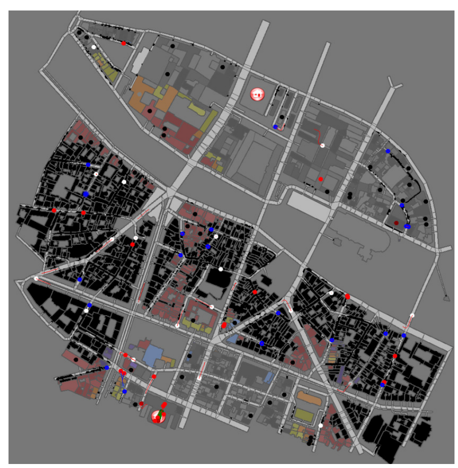

Figure 1: The RoboCup Rescue Paris map.

2.1 Forming long term teams

Many animals form long term relationships with others to create social organizations. These groups benefit from having a diversity and redundancy of skills, which allow the groups to accomplish much more than a single individual could and reduce the risk of failure. In fully observable environments, coalitions are often formed for a single task and then disbanded to more efficiently complete tasks. However, in partially observable environments the uncertainty might require longer term coalitions so when a new task is discovered the completion requirements are more easily met. With limited agents and environment, Vig et al. [4] provide a real-time solution using auctions.

Our first contribution is a more scalable approach using long term teams to both have redundancy and diversity of skills and reduce the complexity of the problem. Long term teams are a group of agents that remain in close spacial proximity with each other until all tasks are completed. This ensures sufficient resources are nearby when a new task is discovered and since agents must stay close together, each team must only consider its own agents and nearby tasks when tasks are allocated. However, the composition and size of teams are important to maintain efficiency of all agents.

Initially, teams are formed based on spacial locality using hierarchical clustering in order to reduce the amount of distance between agents. While agents must be in the proximity of other agents on the same team, all agents do not necessarily work on the same tasks. As in general task allocation, agents can form coalitions with other agents on the team for a single task and then disband, but as new tasks are found the distance agents need to travel to form these coalitions is limited due to the proximity requirement on teams. If there are too many agents in a team, some agents will not be able to complete any tasks without breaking the proximity requirements. To counteract this, we track the utilities of each agent based on how active they are on the team and if an agent has a low utility it can transfer to a different team where its utility will be higher. This ensures a real-time team composition that will correspond to the needs of tasks around each team.

The long term team formation and utility tracking components of the algorithm were tested separately using the RoboCup Rescue Simulator. Generic task allocation slightly out performed the hierarchical clustering without transferring agents on average. This shows that optimizing the initial formation of the team does not compensate for the inefficient team compositions. However, when agents track their own utilities to adjust team compositions, the results were significantly better than the previous two experiments. Furthermore, after incorporating domain knowledge from the RoboCup Rescue Simulator and applying restrictions on team membership this configuration performed competitively against the top competitors from the yearly competition.

2.2 Modeling clustered tasks

Ramchurn et al. [3] outline a task allocation algorithm that uses coalition formation along with temporal constraints on tasks. Specifically, each agent applies a specific amount of work each time step and tasks require a specific workload to be reached before a deadline in order to be completed. Our work [2] extends this situation to where the workloads of tasks can change over time in a predictable manner. Natural examples of tasks with workloads that grow include cancer, invasive species and forest fires, where these tasks are much harder to complete later rather than sooner. We cluster similar tasks together and model the aggregate behavior to allow more efficient assignments of agents.

Task allocation is often trivial if agents can instantly travel to tasks, however when completion requirements change over time this is no longer the case. With instant travel time and tasks that grow under certain assumptions[1], we show that there is an efficiently computable optimal solution when all tasks are known initially. When agents require time to travel between tasks, there is a trade off between efficient assignment of agents and the amount of time agents must spend traveling to those assignments. We provide a non-zero travel time real-time algorithm which determines assignments of agents to minimize the amount of travel needed for agents.

Clusters of tasks in the RoboCup Rescue Simulation is empirically shown have a workload that grows approximately exponential, which satisfies the assumptions of our algorithm. After using domain knowledge to overcome the partial observability to estimate the size of the task cluster, the non-zero travel time solution was applied to RoboCup Rescue and shown to out perform more naïve approaches. However, this algorithm is only run for a subproblem of the RoboCup Rescue Simulator so we were unable to directly compare against the yearly competitors.

3. FUTURE WORK

Currently the RoboCup Rescue Simulator has three types of agents, where each type has the exact same capabilities. The simulator also has three types of tasks, where each agent is able to complete only one type of tasks. The long term team formation is independent of this categorization, but the utility tracking and team transferring utilizes this fact. Namely, agents with a low utility transfer to a team where agents of the same type have a high utility. Since agents in the RoboCup Rescue Simulator can only accomplish one type of tasks, agents on the same team can have drastically different utilities based on the type of available tasks nearby. If these categories did not exist, more research would need to be done in how to estimate an agents effectiveness on a different team.

The model for clustering tasks only applies to a subproblem of the RoboCup Rescue Simulator since other parts do not satisfy the necessary growth assumptions. The long term team approach also cannot incorporate this clustering since there is no apparent mapping between an individual's utility with the team transfer criteria and the clustering model's assignments. More work must be done to find a model which allows these approaches to be integrated.

4. REFERENCES

[1] H. Kitano and S. Tadokoro. RoboCup rescue: A grand challenge for multiagent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.

[2] J. Parker and M. Gini. Tasks with cost growing over time and agent reallocation delays. In *Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, 2014.

[3] S. D. Ramchurn, M. Polukarov, A. Farinelli, C. Truong, and N. R. Jennings. Coalition formation with spatial and temporal constraints. In *Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1181–1188, 2010.

[4] L. Vig and J. A. Adams. Coalition formation: From software agents to robots. *Journal of Intelligent Robotic Systems*, 50(1):85–118, 2007.

[1]if $g(t)$ is the workload cost at time t then 1) task clusters can have different initial values but $g(t)$ identical for all clusters, 2)$\lim_{t \to 0^+} g(t) = 0$ and 3) $\frac{\partial^2}{\partial t^2} g(t) \geq 0$.