Rescue-ISI: Confronting Team Formation in RoboCupRescue

Ranjit Nair, Milind Tambe, and Stacy Marsella

University of Southern California
Computer Science Department and Information Sciences Institute
941 W. 37th Place, Los Angeles, CA 90007
{nair,tambe,marsella}@ISI.edu
Web Page: http://www.isi.edu/teamcore/Rescue/

1 Introduction

The RoboCupRescue Simulation Environment [2] is a challenging multiagent domain where tasks need to be done collaboratively by heterogeneous agents. Our goal is to enable ambulances, fire brigades, police forces, etc. to work as teams in this domain. A key prerequisite to teamwork is *Team Formation*, i.e., the allocation of agents to roles within tasks. Teams once formed, especially in a hostile and dynamic environment like an urban disaster, need to be *reformed* upon arrival of new tasks, team member failure, etc. If every team formation is done by trying to anticipate future reformations, there can be substancial savings in reformation costs and time, thus leading to more lives being saved. We refer to this technique as "Team formation for Reformation"[5].

In last year's team's description[4] we describe various team formation algorithms that we tried out. However, as we noted there, these algorithms are not globally optimal in that they do not form teams anticipating future reformations that may be required. This myopic reasoning lead to many situations where the agents had committed themselves to tasks to early.

The "Team Formation for Reformation" approach uses a theoretical model called R-COM-MTDPs [5], based on decentralized communicating POMDPs. Here the team formation policies are computed off-line by probabilistically reasoning how the current scenario could unfold. In that work we show why this policy computation is intractable (NEXP-complete), thus emphasizing the need for approximations. In this year's team we abstract the state of each agent and limit the amount of look ahead to make the search for a team formation policy more tractable. We employ the same communication and role execution strategy as last year's team and concentrate on coming up with good team formation policies that will improve on our last year's team which finished in third place at RoboCupRescue 2001.

2 Previous Approaches

Our previous approaches for team formation in RoboCupRescue are myopic in the sense that how future events will unfold is not considered while forming teams . The tasks were thought of as civilians who need to be rescued, buildings which are on fire and roads that are blocked. We described a centralized combinatorial auction mechanism demonstrated at Agents-2001 and a distributed method based on localized reasoning.

In our auction mechanism, the fire station, ambulance center and the police office took on the role of auctioneers, and the ambulances, fire brigades and police forces take on the roles of bidders . The items being bid for are the tasks. At the beginning of each cycle, each free agent makes several bids - each bid consists of a different combination of tasks and an estimate of the cost of performing sequentially the tasks in this combination. This approach had the following 2 mainshortcomings:

- Intractable Computational Complexity: The nature of the algorithm mandated that the computation be done in real-time. Owing to the comlexity of the algorithms and further restrictions imposed by the domain, this method was not very feasible.
- Not globally optimal: The allocation found is not optimal if we consider all time steps since we didn't consider future reformations.

The distributed method, based on our agents described in [3], relied on each agent deciding for herself as to which task to perform. This localized reasoning allowed agents to evaluate the seriousness of a task before committing to that task. This localized reasoning is an estimation how the task would unfold in the future. The strength of this approach lay in the low number of messages that it required. A major shortcoming of this approach was that the agents relied on local information and don't concern themselves much with what tasks the other agents were performing. Thus, this allocation scheme is clearly sub-optimal.

3 Team Formation for Reformation in RoboCupRescue

Team Formation for Reformation relies on a theoretical model of teamwork called R-COM-MTDP [5]. R-COM-MTDP is based on decentralized communicating Partially Observable Markov Models. For a detailed description of how the notation can be applied to RoboCup Rescue please see [5].

The goal is to come up with team formation policies for each individual mobile agent. That is for each ambulance, police force and fire brigade. Injured civilians, buildings on fire and blocked roads can be grouped together to form tasks. We specify sub-plans which consists of roles for each task type. Each agent, needs to maintain a belief state of what it believes the true world state is. Based on the agent's believe state is, it chooses whether to continue it current role or to take on another role. The agent's belief state depends on its observations about the objects within its visible range and on the communication it receives from other agents. Note, that there may be parts of the world that are not observable because there are no agents there. Thus, highlighting the importance of communication.

We also define a reward function that specifies the immediate reward for performing an action, be it a role changing, execution or communication action. We assume that communication and reformation always ahev negative reward but they can result in future postive gains. Thus we can now come up with offline policies for role-taking, role execution and communication that maximize the expected utility. In [5], we show that this problem is NEXP-Complete and hence we need to limit the complexity of our algorithms by doing suitable approximations.

Since we are interested in "Team Formation" and "Reformation", we fix the communication and role-execution policies to be almost the same as that in our last year's team and focus on coming up with role-taking policies. We further simplify our problem by considering only an abstraction of the state space. Thus we discard several features as not being relevant. A thorough investigation of which features can be discarded needs to be done. Further, we can limit the amount of look-ahead that is done in computing the policy. This drastically reduces the amount of time it takes to come up with policies for team formation for reformation.

4 Conclusion

In this paper we emphasize the importance of *Team Formation for Reformation* in the RoboCupRescue Simulation Environment. We pointed out the flaws in our previous agents and described how the "Team Formation for Reformation" approach will address the problems. We showed how this technique can be applied to RoboCupRescue for coming up with optimal policies for role changing, role execution and communication. Finding such policies is NEXP-Complete for RoboCupRescue emphasizing the need for approximations like abstracted state space and limited lookahead. We fix the communication and role execution policy to that of our last year's team that finished in 3rd place and concentrate on coming up with an improved team formation policy.

References

1. Hunsberger, L., Grosz, B.: A Combinatorial Auction for Collaborative Planning. Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000), 2000.
2. Kitano, H. et al: RoboCup-Rescue: Search and Rescue for Large Scale Disasters as a Domain for Multi-Agent Research. Proceedings of IEEE Conference, **SMC-99**, October 1999.
3. Nair, R., Ito, T., Tambe, M., Marsella, S.: RoboCup-Rescue: A Proposal and Preliminary Experiences. Proceedings of RoboCup-Rescue Workshop, Fourth International Conference on Multi-Agent Systems(ICMAS-2000), July 2000.
4. Nair, R., Ito, T., Tambe, M., Marsella, S.: Task Allocation in the Rescue Simulation Domain: A Short Note. Proceedings of the RoboCup Symposium, 2001
5. Nair, R., Tambe, M., Marsella, S.: Team Formation for Reformation in Multiagent Domains like RoboCupRescue. Proceedings of RoboCup Symposium, 2002