

Allocating Roles in Extreme Teams

Paul Scerri*, Alessandro Farinelli⁺, Stephen Okamoto[#] and Milind Tambe[#]

* Carnegie Mellon University

⁺ University of Rome

[#] University of Southern California

pscerri@cs.cmu.edu, Alessandro.Farinelli@dis.uniroma1.it, sokamoto@usc.edu, tambe@usc.edu

New domains are emerging that impose new requirements for teamwork, where current teamwork infrastructure is inadequate. One such large class of application require *extreme teams*, which are large teams that need (soft) real-time response given dynamic tasks, and where many *resource limited* agents have similar functionality, but possibly varied capability. For instance, when responding to a disaster, fire fighters and paramedics comprise an extreme team as they must respond rapidly to dynamic tasks; and fire fighters can all extinguish fires although their capability to extinguish a particular fire quickly will depend on their initial distance from that fire.

This paper focuses on the critical challenge of role allocation in extreme teams. In general, role allocation is problem of assigning roles to agents so as to maximize overall team utility[5]. Extreme teams emphasize key additional requirements in role allocation: (i) rapid role allocation as domain dynamics may cause tasks to disappear; (ii) agents may perform one or more roles, but within resource limits; (iii) many agents can fulfill the same role; (iv) inter-role constraints may be present. This role allocation challenge in extreme teams will be referred to as E-GAP, as it subsumes the generalized assignment problem (GAP), which is NP-complete[4].

This paper focuses on Distributed Constraint Optimization (DCOP)[2] for role allocation, as DCOP offers the key advantages of distributedness and a rich representational language which can consider costs/utilities of tasks. Despite these advantages, DCOP approaches to role allocation suffer from three weaknesses. First, complete DCOP algorithms[2] have exponential runtime complexity and, thus, fail to meet the response requirements of extreme teams. One reason for this is that the purely local view of the team that each agent has, forces the search to explore many potential solutions that are clearly sub-optimal. However, teams of agents will often have reasonably accurate estimates of both the situation and the state of the team which can be used to accurately estimate likely solution characteristics. While relying on such estimates prevents

guarantees of optimality, they can dramatically reduce the search space. Second, similar agent functionality within extreme teams results in dense constraint graphs increasing communication within a DCOP algorithm. Third, DCOP algorithms do not address the additional complications of constraints *between* roles.

To address these limitations in addressing E-GAP, we propose a novel DCOP algorithm called LA-DCOP (Low communication Approximate DCOP). LA-DCOP uses a representation where agents are variables that can take on values from a common pool, i.e., the pool of roles to be assigned. The mechanism for allocating values to variables encapsulates two novel ideas. First, LA-DCOP improves efficiency by not focusing on an exact optimal reward; instead by exploiting the likely characteristics of optimal allocations, given the available probabilistic information, it focuses on maximizing the team's expected total reward. In particular, the agents *compute* a minimum threshold on the expected capability of the agent that would maximize expected team performance. If the agent's capability to perform a role is less than the threshold capability, it does not consider taking on the role, channeling the role towards more capable agents. Second, to reduce the significant communication overheads due to constraint graph denseness, *tokens* are used to regulate access to values. Only the agent currently holding the token for a particular value can consider assigning that value to its variable. The use of tokens removes the possibility of several agents taking on the same role, thus dramatically reducing the need to communicate about and repair conflicts.

Our first experiments tests LA-DCOP against three competitors in an abstract simulator. The first competitor is DSA, which is shown to outperform other approximate DCOP algorithms in a range of settings [2]; we choose optimal parameters for DSA [6]. DSA does not easily allow multiple roles to be assigned to a single agent so our comparison focuses on the case where each agent can take only one role. As a baseline we also compare against a centralized algorithm

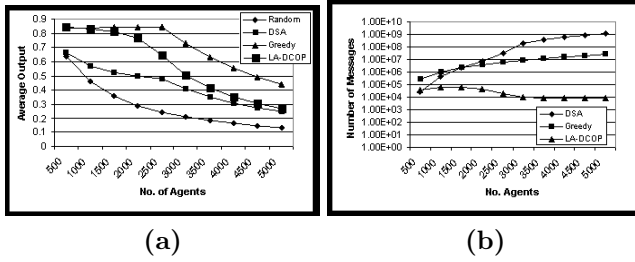


Figure 1. (a) comparing the average output per agent versus the number of agents. (b) the number of messages sent versus the number of agents. Each point represents the average of 20 runs.

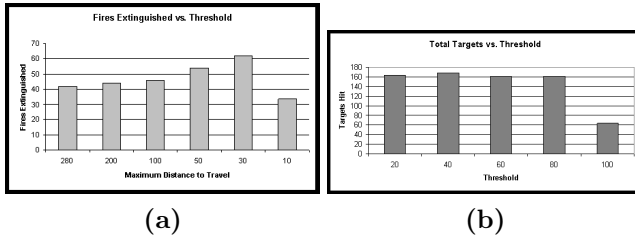


Figure 2. (a) shows the number of fires extinguished by 200 fire trucks versus threshold (b) shows the number of targets hit by UAVs versus threshold.

that uses a “greedy” assignment[1] and against a random assignment. In the simulator, agents are randomly given capabilities for each type of role with some percentage being given zero capability. For each time step that the agent has the role, the team receives ongoing reward based on the agent’s capability. Figure 1(a) shows the relative performance of each algorithm. The experiment used 2000 roles over 1000 time steps. The y-axis shows the total reward per agent, while the x-axis shows the number of agents. Not surprisingly, the centralized algorithm performs best and the random algorithm performs worst. LA-DCOP is statistically significantly better than DSA. However, the key is the amount of communication used, as shown in Figure 1(b). Notice that the y-axis is a logarithmic scale, thus LA-DCOP uses approximately three orders of magnitude fewer messages than the greedy algorithm and four orders of magnitude less messages than DSA. Thus, LA-DCOP performs better than DSA despite using far less communication and only marginally worse than a centralized approach, despite using only a tiny fraction of the number of messages.

In our second set of experiments, we used 200 LA-DCOP enhanced versions of Machinetta proxies[3], distributed over a network, executing plans in two sim-

ple simulation environments. To the best of our knowledge, this is larger than any published report on complex multiagent teams, certainly an order of magnitude jump over the last published reports of teamwork based on proxies [3]. Previous published techniques for role allocation in the proxies fail to scale up to extreme teams of 200 agents — complete DCOP fails on dense graphs, and symbolic matching ignores quantitative information. The proxies execute sophisticated teamwork algorithms as well as LA-DCOP and thus provide a realistic test of LA-DCOP. The first environment is a version of a disaster response domain where fire trucks must fight fires. Capability in this case is the distance of the truck from the fire, since this affects the time until the fire is extinguished. Hence, in this case, the threshold corresponds to the maximum distance the truck will travel to a fire. Figure 2(a) shows the number of fires extinguished by the team versus threshold. Increasing thresholds initially improves the number of fires extinguished, but too high a threshold results in a lack of trucks accepting roles and a decrease in performance. In the second domain, 200 simulated unmanned aerial vehicles (UAVs) explored a battle space, destroying targets of interest Figure 2(b). While in this domain LA-DCOP effectively allocates roles across a large team, thresholds are of no benefit. The key point of these experiments is to show that LA-DCOP can work effectively, in a fully distributed environment with realistic domains and large teams.

References

- [1] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa. Coordination among heterogeneous robotic soccer players. In *Proceedings of IROS’02*, 2002.
- [2] P. J. Modi, W. Shen, and M. Tambe. Distributed constraint optimization and its application. Technical Report ISI-TR-509, University of Southern California/Information Sciences Institute, 2002.
- [3] P. Scerri, D. V. Pynadath, L. Johnson, Rosenbloom P., N. Schurr, M Si, and M. Tambe. A prototype infrastructure for distributed robot-agent-person teams. In *The Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003.
- [4] D. Shmoys and E. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62:461–474, 1993.
- [5] G. Tidhar, A.S. Rao, and E.A. Sonenberg. Guided team selection. In *Proceedings of the Second International Conference on Multi-Agent Systems*, 1996.
- [6] W. Zhang and L. Wittenburg. Distributed breakout revisited. In *Proceedings of AAAI’02*, 2002.