Allocating Spatially Distributed Tasks in Large, Dynamic Robot Teams

(Extended Abstract)

Steven Okamoto, Nathan Brooks, Sean Owens, Katia Sycara, Paul Scerri
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{sokamoto, nbb, owens, katia, pscerri}@cs.cmu.edu

ABSTRACT

For an interesting class of emerging applications, a large robot team will need to distributedly allocate many more tasks than there are robots, with dynamically appearing tasks and a limited ability to communicate. The LA-DCOP algorithm can conceptually handle both large-scale problems and multiple tasks per robot, but has key limitations when allocating spatially distributed tasks. In this paper, we extend LA-DCOP with several alternative acceptance rules for robots to determine whether to take on an additional task, given the interaction with the tasks it has already committed to. We show that these acceptance rules dramatically outperform a naive LA-DCOP implementation. In addition, we developed a technique that lets the robots use completely local knowledge to adjust their task acceptance criteria to get the best possible performance at a given communication bandwidth level.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Algorithms

Keywords

task allocation, LA-DCOP

1. OVERVIEW

A key problem for coordinated robot team is to allocate tasks for best overall performance. For many domains, the primary feature that distinguishes which robot should be allocated which task is the location of the task, since overall performance will be dominated by the time taken to reach the task. For example, in a surveillance scenario where a robot is simply taking images, the key is to get any robot to the location. In an interesting class of emerging applications, a large robot team will need to distributedly allocate many more tasks than there are robots, with dynamically appearing tasks and a limited ability to communicate. Examples of such tasks include exploration, item delivery and environment monitoring [1].

Task allocation when robots take on multiple tasks and need to plan paths between those tasks is computationally hard [2]. Most existing solutions require centralization, handle only very simple tasks, or do not scale to large numbers of robots [3, 4, 5]. The LA-DCOP algorithm [6] can conceptually handle both large-scale problems and multiple tasks per robot, getting good allocations with low computational and communications costs, but is not effective when allocating spatially distributed tasks. The key to LA-DCOP is that tasks are passed around the team on tokens, with robots deciding to accept or reject responsibility for tasks based on resource constraints and a *threshold* on a scalar capability value that is assumed to be independent of other tasks. This assumption is violated for spatially distributed tasks where capability is primarily the time to get to a task, because that time depends on the path the robot traverses between tasks.

In this paper, we generalize LA-DCOP's simple threshold rule into different acceptance rules. **LILO** is the naive LA-DCOP implementation that appends a new task to the path if the length of the resulting path is less than an *absolute threshold*. Once accepted, tasks are never removed from a robot's path. The other acceptance rules also use absolute thresholds, but applies them to all tasks if a new task is accepted (because the path to old tasks can change). **Marginal cost** minimizes the change in path length by inserting a new task into the path where the resulting total path length is minimized (*optimal insertion*), and accepting only if the increase in length is less than a *marginal cost threshold*. **Myopic** greedily replans paths for each new task, with the maximum number of tasks limited by a *path count threshold*. **T-over-t** tries to directly maximize task completion rate by optimally inserting and accepting only if the number of tasks divided by the path length increases.

We evaluated the acceptance rules using an abstracted two-dimensional simulation where robots and tasks were situated in a 100-by-100 planar region without obstacles. Robots communicated using a fully-connected multihop network. In order to complete a task, a robot was required to move to the location of a task and intentionally perform it; task execution was instantaneous and all robots were assumed to move a constant speed. When a task was completed, new tasks

Cite as: Allocating Spatially Distributed Tasks in Large, Dynamic Robot Teams (Extended Abstract), Steven Okamoto, Nathan Brooks, Sean Owens, Katia Sycara, and Paul Scerri, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1245-1246.

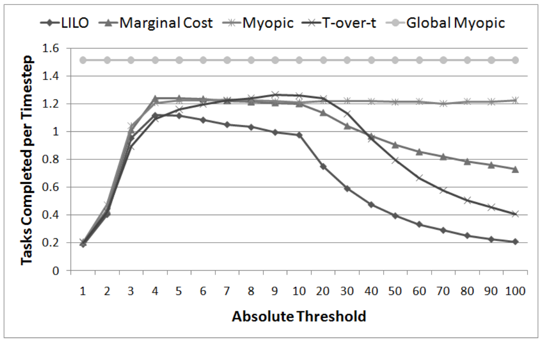

Figure 1: Task completion rate for varying absolute thresholds.

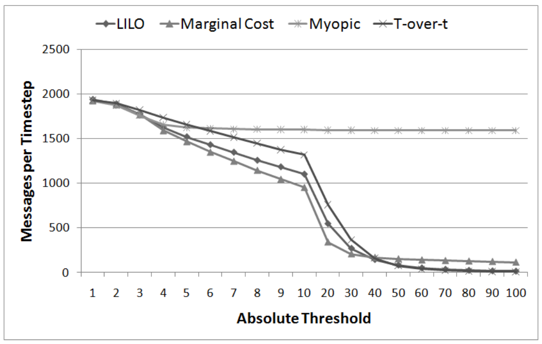

Figure 2: Communication rate for varying absolute thresholds.

were randomly created nearby. There were 200 robots and initially 2000 tasks. We measured two performance metrics: task completion rate and communication rate. We compared task completion rate to a baseline (global myopic) that was an all-knowing greedy algorithm that allocated tasks to the nearest robot without a task. (Because it is all-knowing, it does not make sense to compare on communication rate.)

Figure 1 shows the task completion rate for varying absolute thresholds. For very low thresholds, task completion rate was low because robots had difficulty finding sufficiently close tasks, but at moderate thresholds, naive LA-DCOP (LILO) was dramatically outperformed by the other acceptance rules. Myopic (with a path count threshold of 2) plateaus to a good allocation because the path count threshold becomes the limiting factor but robots are still able to find nearby tasks. This good allocation comes at the price of high communication, as shown in Figure 2, while the other acceptance rules decrease communication as tasks are "locked up" in robots' paths.

LA-DCOP assumes that an appropriate threshold can be set globally at the beginning of some mission and will be appropriate for the entire mission. However, a single, global threshold does not perform well when task creation frequency and density varies. We developed a technique that lets the robots use completely local knowledge to locally adjust the path count threshold for the myopic acceptance rule to get best possible global performance at a given level communication bandwidth usage. Typically, lower thresholds lead to higher quality allocations at the expense of more communication. By monitoring their local communication over time, robots estimate the likelihood of the desired global, aggregate communication rate being met, and stochastically update their local path count threshold. Figure 3 shows the message rate and task completion rate over time, when the desired communication rate is changed twice: from an initial value of 4 to 8 at timestep 30000, and then from 8 to 2 at timestep 60000. The team reacts quickly and accurately to the adjust to the initial value and the first change, but has difficulty with the final change.

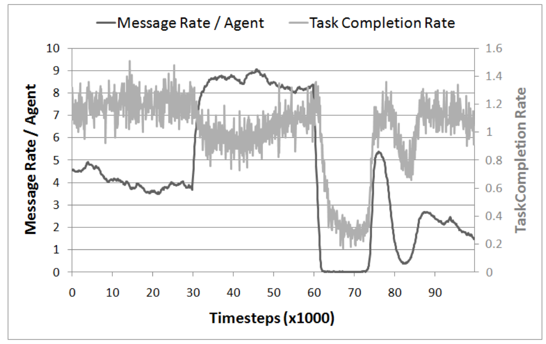

Figure 3: Communication rate and task completion rate with dynamically adjusted thresholds.

2. CONCLUSIONS

While we were able to realize dramatic performance gains over a naive implementation of LA-DCOP, none of the acceptance rules dominated the others across all parameters. A deeper understanding of what properties favor each rule is a key area for future work, as is searching for alternative rules that perform better under a wider set of circumstances. In immediate future work, we will look at how other types of information might be used by the agents. Examples include, noisy information about the locations of other robots, knowledge that tasks are clustered around some areas, or knowledge that the number of tasks to be performed is going to increase. We will also investigate network types where communication is localized.

3. REFERENCES

[1] N. Agmon, S. Kraus, and G.A. Kaminka. Multi-robot perimeter patrol in adversarial settings. 2008.

[2] T. Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209 – 219, 2006.

[3] K. Lerman, C. Jones, A. Galstyan, and M.J. Mataric. Analysis of Dynamic Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research*, 25(3):225–241, 2006.

[4] P.J. Modi, W. Shen, M. Tambe, and M. Yokoo. An asynchronous complete method for distributed constraint optimization. In *AAMAS'03*, 2003.

[5] R. Nair, T. Ito, M. Tambe, and S. Marsella. Task allocation in robocup rescue simulation domain. In *RoboCup'02*, 2002.

[6] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe. Allocating tasks in extreme teams. In *AAMAS '05*, 2005.