Towards Multiagent Meta-Level Control

Shanjun Cheng and **Anita Raja**
Department of Software and Information Systems
The University of North Carolina at Charlotte
9201 University City Blvd, Charlotte, NC 28223
cheng6, anraja@uncc.edu Tel:704-763-1943
http://webpages.uncc.edu/∼scheng6/

Victor Lesser
Department of Computer Science
University of Massachusetts Amherst
140 Governor's Dr, Box 9264, Amherst, MA 01003
lesser@cs.umass.edu

Abstract

Embedded systems consisting of collaborating agents capable of interacting with their environment are becoming ubiquitous. It is crucial for these systems to be able to adapt to the dynamic and uncertain characteristics of an open environment. In this paper, we argue that multiagent meta-level control (MMLC) is an effective way to determine when this adaptation process should be done and how much effort should be invested in adaptation as opposed to continuing with the current action plan. We describe a reinforcement learning based approach to learn decentralized meta-control policies offline. We then propose to use the learned reward model as input to a global optimization algorithm to avoid conflicting meta-level decisions between coordinating agents. Our initial experiments in the context of NetRads, a multiagent tornado tracking application show that MMLC significantly improves performance in a 3-agent network.

Introduction

Meta-level control (Cox and Raja 2008) in an agent involves making decisions about whether to deliberate, how many resources to dedicate to this deliberation and what specific deliberative control to perform in the current context. Multiagent meta-level control (MMLC) facilitates agents to have a decentralized meta-level multiagent policy, where the progression of what deliberations the agents should do, and when, is choreographed carefully and includes branches to account for what could happen as deliberation plays out.

In this research, we study the role of MMLC on NetRads, a real application. NetRads (Krainin, An, and Lesser 2007) is a fielded, next generation distributed sensor network system developed by the University of Massachusetts NSF Engineering Research Center for Collaborative Adapting Sensing of the Atmosphere (CASA). It is modeled as a network of adaptive radars controlled by a collection of Meteorological Command and Control (MCC) agents that instruct where to scan based on emerging weather conditions. The NetRads radar is designed to quickly detect low-lying meteorological phenomena such as tornadoes. The time allotted to the radar and its control systems for data gathering and analysis is known as a heartbeat.

Our intent for this work is to design and develop a framework for MMLC. At the highest level, the question we plan to address is the following: How does the meta-level control component of each agent learn policies so that it can efficiently support agent interactions and reorganize the underlying network when needed? Specifically in NetRads domain, reorganizing the network involves addressing the following questions:

1. What triggers a radar to be handed off to another MCC and how do we determine which MCC to hand off the radar to?
2. How to assign different heartbeats to sub networks of agents in order to adapt to changing weather conditions?

Our Approach

Each MCC's heartbeat (30 seconds or 60 seconds long) is split up into deliberative-level actions and meta-level actions. The current deployed version of the MCC (Krainin, An, and Lesser 2007) handles the deliberative-level actions that give radars instructions as to where to scan based on emerging weather conditions.

MMLC will use a learned meta-level policy to handle the coordination of MCCs and guide the deliberative-level actions in other phases in NetRads. This involves reassigning radars and adjusting the heartbeats of MCCs in a decentralized fashion.

We design and develop a MMLC approach that involves coordination of decentralized Markov Decision Processes (DEC-MDPs) (Bernstein, Zilberstein, and Immerman 2000) using the Weighted Policy Learning (WPL) algorithm (Abdallah and Lesser 2007). WPL is a reinforcement learning algorithm that achieves convergence using an intuitive idea: slow down learning when moving away from a stable policy and speed up learning when moving towards the stable policy. WPL is used to learn the policies for the meta-level DEC-MDPs belonging to individual agents. Online learning on a very large MDP that captures all possible weather scenarios during the MMLC phase can be very time expensive. To overcome this challenge, sets of weather scenarios are grouped into abstract meta-level scenarios based on number of tasks and types of tasks and MCCs learn the policies for each abstract scenario offline. The MDPs of these abstract scenarios and their policies will be stored in a library that is

available to each MCC. At real time, each MCC will adopt the scenario-appropriate policy.

We map the NetRads meta-level control problem to a DEC-MDP model in the following way. The model is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, where

- $\mathcal{S}$ is a finite set of world states, with a distinguished initial state s^0. In NetRads domain, the state of each MCC is the *meta-level state*, defined as the abstract representation of the state which captures the important qualitative state information relevant to the meta-level control decision making process.
- $\mathcal{A}$ is a finite set of actions. In NetRads domain, the actions for each MCC are radar handoffs or heartbeat changing.
- $\mathcal{P}$ is a transition function. $\mathcal{P}(s' \mid s, a_i)$ is the probability of the outcome state s' when the action a_i is taken in state s. In NetRads domain, the transition function is based on the time/quality distribution for the actions MCC_i chooses to execute.
- $\mathcal{R}$ is a reward function. $\mathcal{R}(s, a_i, s')$ is the reward obtained from taking action a_i in state s and transitioning to state s'. In NetRads domain, the reward is only received in a terminal state, and it represents the average of *qualities* of all tasks collected by MCC_i from last heartbeat. The *quality* of a task from a single radar is the priority of the task multiplied by a factor meant to represent the quality of the data that would result from the scan (specified by experts in the field e.g. meteorologists) (Krainin, An, and Lesser 2007).

Experiments and Future Work

We use the NetRads radar simulator system (Krainin, An, and Lesser 2007) to conduct some initial experiments to study the effectiveness of the WPL algorithm for MMLC on a small set of MCCs (3 MCCs supervises 3 radars each). We test the results in three different weather scenarios. They are defined as: High Rotation Low Storm (HRLS), Low Rotation High Storm (LRHS), and Medium Rotation Medium Storm (MRMS). For example, HRLS denotes the scenario in which the number of rotations overwhelms the number of storms in a series of heartbeats. We compare the results of three methods: No-MLC, Adaptive Heuristic Heartbeat (AHH) and MMLC. No-MLC is the method that without meta-level control module. AHH is the method where we incorporate simple heuristics in meta-level control to adaptively change the heartbeat of each MCC. In our MMLC approach, we used 50 training cases and each has a long sequence of training data (500 heartbeat periods) to learn the policies for all the abstract scenarios offline. Using each method mentioned above, we ran 30 test cases for each of three weather scenarios. *Average Quality* and *Negotiation Time* are the parameters to compare the scanning performance. *Average Quality* rates the performance of tasks achieved. *Negotiation Time* denotes the total time (seconds) MCCs spend in negotiation with other MCCs. In figure 1, MMLC spends least *Negotiation Time* among the three. Figure 2 shows that our MMLC approach performs significantly ($p < 0.05$) better than No-MLC (p values in the t-tests are 0.038, 0.014 and 0.00043) and AHH (p values are 0.029, 0.0033 and 0.005) on *Average Quality*.

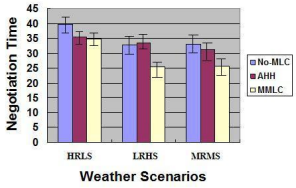

Figure 1: *Negotiation Time* of No-MLC, AHH and MMLC in different weather scenarios.

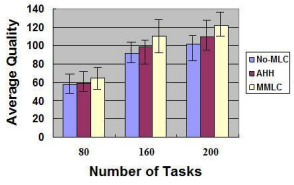

Figure 2: *Average Quality* of No-MLC, AHH and MMLC, for number of tasks to be 80, 160 and 200.

Our current results are encouraging and show that MMLC can be an efficient way to allocate resources and reorganize the network with the goal of improving performance in NetRads. However, our current implementation only guarantees optimal policies for each agent from a local perspective, possibly leading to conflicting action choices among agents. The Bounded Max-Sum (Farinelli et al. 2008) algorithm is a decentralized coordination algorithm that provides bounded approximate (within 95% of the optimum) solutions for general constraint networks while requiring very limited communication overhead and computation. We plan to extend this algorithm to achieve the global optimization required in NetRads when we scale up the problem to hundreds of radars.

References

Abdallah, S., and Lesser, V. 2007. Multiagent Reinforcement Learning and Self-Organization in a Network of Agents. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 172–179. Honolulu: IFAAMAS.

Bernstein, D.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of markov decision processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence(UAI)*, 32–37.

Cox, M., and Raja, A. 2008. Metareasoning: A Manifesto. In *Proceedings of AAAI 2008 Workshop on Metareasoning: Thinking about Thinking*, 1–4.

Farinelli, A.; Rogers, A.; Petcu, A.; and Jennings, N. 2008. Decentralised Coordination of Low-power Embedded Devices Using the Max-sum Algorithm. In *The Seventh International Conference on Autonomous Agents and Multiagent Systems*, 639–646.

Krainin, M.; An, B.; and Lesser, V. 2007. An Application of Automated Negotiation to Distributed Task Allocation. In *2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2007)*, 138–145. Fremont, California: IEEE Computer Society Press.