The Knowledge Engineering Review, Vol. 31:1, 3–18.
doi:10.1017/S0269888915000156

Combining reward shaping and hierarchies for scaling to large multiagent systems

CHRIS HOLMESPARKER[1], ADRIAN K. AGOGINO[2] and KAGAN TUMER[1]

[1]*School of MIME, Oregon State University, 442 Rogers Hall, Corvallis, OR 97331-6001, USA;*
e-mail: holmespc@onid.orst.edu, kagan.tumer@oregonstate.edu;
[2]*UCSC at NASA Ames, Mail Stop 269-3, Moffett Field, CA 94035, USA;*
e-mail: Adrian.K.Agogino@nasa.gov

Abstract

Coordinating the actions of agents in multiagent systems presents a challenging problem, especially as the size of the system is increased and predicting the agent interactions becomes difficult. Many approaches to improving coordination within multiagent systems have been developed including organizational structures, shaped rewards, coordination graphs, heuristic methods, and learning automata. However, each of these approaches still have inherent limitations with respect to coordination and scalability. We explore the potential of synergistically combining existing coordination mechanisms such that they offset each others' limitations. More specifically, we are interested in combining existing coordination mechanisms in order to achieve improved performance, increased scalability, and reduced coordination complexity in large multiagent systems.

In this work, we discuss and demonstrate the individual limitations of two well-known coordination mechanisms. We then provide a methodology for combining the two coordination mechanisms to offset their limitations and improve performance over either method individually. In particular, we combine shaped difference rewards and hierarchical organization in the Defect Combination Problem with up to 10000 sensing agents. We show that combining hierarchical organization with difference rewards can improve both coordination and scalability by decreasing information overhead, structuring agent-to-agent connectivity and control flow, and improving the individual decision-making capabilities of agents. We show that by combining hierarchies and difference rewards, the information overheads and computational requirements of individual agents can be reduced by as much as 99% while simultaneously increasing the overall system performance. Additionally, we demonstrate the robustness of this approach to handling up to 25% agent failures under various conditions.

1 Introduction

Coordinating the behavior of agents in multiagent systems such that they collectively optimize a system-level objective is a complex control task. Problems of scaling (number of agents in the thousands to tens of thousands), information handling (agents have limited computing capabilities), and robustness (unreliable components) make methods developed for small multiagent systems comprised of reliable devices inadequate (Panait & Luke, 2005; Tumer, 2005). A number of approaches have been presented to address these issues including organizational structures, shaped rewards, learning automata, and coordination graphs (Horling & Lesser, 2005; Tambe *et al*., 2005; Xu *et al*., 2005; Kok & Vlassis, 2006; Vrancx *et al*., 2008; Barrett *et al*., 2011; Howley & Duggan, 2011). Although significant progress has been made toward

improving coordination and scalability with each of these methods individually, relatively little work has focused on leveraging the complementary benefits of these approaches. In this work, we propose combining two of these methods (hierarchical organization and reward shaping) together in order to decrease coordination complexity, improve performance, and increase scalability in large multiagent systems.

Reward shaping has been shown to drastically improve coordination, scalability, and performance in multiagent systems (Agogino & Tumer, 2008; Williamson *et al.*, 2009; Grzes & Kudenko, 2010). The specific shaped rewards studied in this work are based upon the difference reward structure, which has been shown to be robust to scaling in a number of domains including air traffic control, rover navigation, satellite coordination, and distributed sensor networks (Tumer, 2005; Agogino & Tumer, 2008; Knudson & Tumer, 2010; Agogino *et al.*, 2012; HolmesParker *et al.*, 2012, 2013). Difference rewards are designed to promote coordination and scalability by filtering the information each agent receives, extracting only information relevant to each agent specifically. However, as scaling increases, the amount of information each agent must process increases, reducing the effectiveness of the filter provided by difference rewards. We address this shortcoming by introducing hierarchical organization into the system, which reduces the amount of information each agent must receive and process.

Hierarchies have shown a lot of promise in decreasing information sharing and processing requirements, improving robustness, and increasing performance in large multiagent systems (Horling & Lesser, 2005; Mehta *et al.*, 2008). These structures focus primarily upon organizing the control flow to reduce information sharing and processing overheads, which reduces the coordination complexity between agents in the system (Horling & Lesser, 2005). However, controlling these factors alone is not always enough to achieve good system performance, as they do not dictate the underlying decision-making process for agents in the system. Just as the structure of relationships and control flow impact system performance, the underlying decision-making process of each agent also heavily impacts the system performance. To address this, we combine hierarchical organization with learning agents using shaped difference rewards which promote good agent decision making.

Although both hierarchical organization and reward shaping methods have been heavily researched, relatively little work has been done to demonstrate the complementary nature of these two approaches. Generally speaking, hierarchies establish the system control flow and reduce the amount of information that each agent must receive and process (Horling & Lesser, 2005). Shaped rewards on the other hand attempt to optimize each agent's decision making given that information (Tumer, 2005). Thus, in a learning-based system, hierarchical organization would dictate the amount of information each agent receives as well as the control flow, while shaped rewards would be used in agent decision making to optimize system performance given the information available to them. In this work, we demonstrate the complementary nature of these approaches in the Defect Combination Problem (DCP) described in Section 3.1 (Challet & Johnson, 2002).

The key contributions of combining shaped difference rewards and hierarchical organization demonstrated in this paper are as follows:

- reduced information sharing and processing requirements for agents;
- increased scalability;
- robustness to increased problem complexity;
- robustness to various agent failures.

The remainder of this paper is organized as follows. Section 2 provides background material on hierarchical systems, reward shaping, and the DCP. Section 3.1 describes the DCP. Section 4 describes the learning algorithms, rewards, and hierarchical organization used in this work. Section 5 contains experimental results, empirically demonstrating the benefits of coupling hierarchies and shaped rewards with regards to decreasing the information overheads and processing requirements for agents, as well as improving overall system performance and robustness. Finally, Section 6 provides a discussion of this work.

2 Background and related work

Previous work involving the DCP utilized statistical physics to determine the theoretical optimal performance based upon the number of sensors (Challet & Johnson, 2002). This work derived the theoretical

optimal performance of an N sensor system and the corresponding ratio of active sensors, but it did not include a non-exhaustive search method for finding the actual subset of sensors to use. Using learning agents with difference rewards in a non-hierarchical setting was proposed as a method for finding a good subset of devices in Tumer (2005). In that work, difference rewards were shown to improve system performance in the DCP in a non-hierarchical setting involving up to 1000 sensors (Tumer, 2005). However, as our work shows, difference rewards alone are not sufficient to address the increased coordination complexities and increased signal noise present when scaling increases in such large systems. To address this shortcoming, we couple difference rewards with hierarchical organization which restricts the amount of information each agent in the system receives and reduces the agent-to-agent coordination complexity. We apply a hierarchy, which structures the agent-to-agent relationships and reduces the amount of information individual agents must receive and process during the decision-making process, and difference rewards which attempt to make globally optimal decisions based upon the information that is locally available to each individual agent.

2.1 Hierarchical organization

The hierarchical organization of a multiagent system can be defined as the collection of roles, relationships, and authority structures which govern its behavior (Horling & Lesser, 2005). All hierarchies have some form of these characteristics, although they may be implicitly present and not formally developed (Horling & Lesser, 2005). The structure of a hierarchy guides how its members interact with one another, influencing authority relationships, data flow, resource allocation, coordination patterns, and other system characteristics (Hayden *et al.*, 1999). Hierarchies have been shown to improve system performance in a number of domains including distributed sensor networks, autonomous aerial vehicle coordination, and rover coordination (Horling *et al.*, 2004; Horling & Lesser, 2005; Zhang *et al.*, 2009). In many cases, hierarchical organization reduces coordination complexity and increases system-level performance by providing an explicit structure and control flow (Horling *et al.*, 2004; Horling & Lesser, 2005). Although hierarchies establish the structure and control flow, they do not directly address decision making. In this work, we utilize reinforcement learning coupled with shaped difference rewards in a two-layer hierarchy to enable decision making and decrease coordination requirements for agents.

2.2 Reward shaping

Reward shaping is the practice of altering an agent's reward function in such a way that it changes its behavior (Tumer, 2005; Devlin & Kudenko, 2011). Frequently, reward shaping is used to improve system performance or to make a problem easier to solve (Agogino & Tumer, 2008; Grzes & Kudenko, 2010). Reward shaping has been used to increase performance by speeding up convergence rates and improving coordination in problems involving reinforcement learning (Agogino & Tumer, 2008; Williamson *et al.*, 2009). In Q-learning, reward shaping can be represented by the following formula (Ng *et al.*, 1999; Devlin & Kudenko, 2011):

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + F(s, s') + \gamma \max_{a'} Q(s', a') - Q(s, a)] \tag{1}$$

where $Q(s, a)$ is the Q-value associated with the agent taking action a in state s, r the standard reward, a' an alternate action, s' an alternate state, α the learning rate, γ the discount factor, and $F(s, s')$ the general form of the shaping reward. As seen, the shaping reward $F(s, s')$ is an additional reward that is applied on top of the agents original reward r in order to encourage better learning (Ng *et al.*, 1999; Devlin & Kudenko, 2011). Reward shaping techniques (e.g. potential-based reward shaping) have been used to increase performance by speeding up convergence rates and improving coordination in problems involving reinforcement learning (Agogino & Tumer, 2008; Williamson *et al.*, 2009; Grzes & Kudenko, 2010; Devlin & Kudenko, 2011).

2.3 Difference rewards

Difference rewards are a particular type of shaped rewards that were developed to address the structural credit assignment problem within multiagent systems (Wolpert & Tumer, 2001; Tumer, 2005;

Agogino *et al.*, 2012; HolmesParker *et al.*, 2012). These rewards are of the form (Agogino & Tumer, 2008):

$$D_j \equiv G(z) - G(z_{-j} + c_j) \tag{2}$$

where G is the system objective, z the complete system state vector, and z_{-j} contains all the variables not affected by agent j. All the components of z that are affected by agent j are replaced with the fixed constant c_j (counterfactual action). Such difference rewards are aligned with the system performance regardless of the choice of c_j, because the second term does not depend on j's actions (Tumer, 2005). Furthermore, they provide a cleaner learning signal than a team reward, because the second term of D, which removes a lot of the effect of other agents (i.e. noise) from j's reward. In many situations it is possible to use a value of c_j that is equivalent to taking agent j out of the system. This causes the second term to be independent of j (i.e. the system performance without agent j), and therefore D_j evaluates the agent's contribution to the global performance. There are two key advantages to using D_j: first, because the second term removes a significant portion of the impact of other agents in the system, it provides an agent with a 'cleaner' signal than G (Tumer, 2005; Agogino & Tumer, 2008). Second, because the second term does not depend on the actions of agent j, any action by agent j that improves D, also improves G (the derivatives of D and G with respect to j are the same) (Tumer, 2005; Agogino & Tumer, 2008).

We also consider the EDR which is given by

$$EDR_j \equiv G(z) - E_{z_j}[G(z) \mid z_{-j}] \tag{3}$$

where $E_{z_j}[G(z) \mid z_{-j}]$ gives the expected value of G over the possible actions of agent j. Because this term does not depend on the immediate actions of j, this reward is still aligned with G (Tumer, 2005). Furthermore, because it removes noise from each agent's own reward, EDR yields far better learnability than does G (Tumer, 2005). This noise reduction is due to the subtraction which (to a first approximation) eliminates the impact of states that are not affected by the actions of agent j. The major difference between *EDR* and D is in how they handle z_j. EDR provides an estimate of agent j's impact by sampling all possible actions of agent j, whereas D simply removes agent j from the system.

3 Domains

3.1 The Defect Combination Problem

Many real-world sensing applications require large sets of disparate sensing devices to coordinate their actions in order to collectively optimize their network attenuation, coverage areas, and sensing schedules (Bharathidasan & Ponduru, 2003; Tham & Renaud, 2005; Farinelli *et al.*, 2008; Williamson *et al.*, 2009; Rogers *et al.*, 2010; Vinyals *et al.*, 2010). In this work, a set of up to 10000 sensing devices must coordinate their sensing schedules in order to optimize their aggregated attenuation within a sensor network. This work focuses on the DCP domain which was originally introduced in Challet and Johnson (2002) and was also used in Tumer (2005). This problem assumes that there exists a set of imperfect sensors $\mathbf{X}$ which have constant attenuations due to manufacturing defects or imperfections. Each of the sensors x_i has an associated attenuation a_i (which can be positive or negative) in its reading, such that if it is taking a measurement of A (actual value) it measures $A + a_i$ where a_i is the device's individual error. The problem then becomes how to best choose a subset of the $\mathbf{X}$ sensors that minimizes the aggregated attenuation of the combined readings:

$$G = \frac{\left| \sum_{i=1}^{N} n_i a_i \right|}{\sum_{i=1}^{N} n_i} \tag{4}$$

where G is the aggregated attenuation of the combined sensor readings, a_i the attenuation of a particular sensor i, N the number of sensors, and $n_i \epsilon \{0,1\}$ based upon whether the sensor chooses to be 'on' or 'off'.

This is an NP-complete optimization problem (Challet & Johnson, 2002; Tumer, 2005) and simply choosing the single sensor with the best attenuation is an inadequate solution, as is choosing the best K sensors ($1 \leqslant K \leqslant N$). To illustrate this, consider the case where there are six sensing devices whose

attenuations are $a_1 = -0.19$, $a_2 = 0.54$, $a_3 = 0.1$, $a_4 = -0.14$, $a_5 = -0.05$, and $a_6 = 0.21$. Choosing only the best sensor a_5 would yield an aggregated attenuation of $|0.05|$, while choosing sensors a_3, a_4, and a_5 will yield an aggregated attenuation of $|0.03|$, which is better than the single best sensing device a_5 alone. This is still not the optimal solution in this six sensor case, however, as combining sensors a_1 and a_6 results in an aggregated attenuation of $|0.01|$. In this problem, individual sensors acting independently without coordinating their actions can drastically decrease the system performance. Consider the case where sensors a_1 and a_6 are turned on in conjunction with sensor a_2, the aggregated attenuation jumps from $|0.01|$ to $|0.18|$. Finding good solutions require a great deal of coordination between sensors, as any one sensor can heavily impact the system performance.

4 Agents and coordination

In this work, we used a multiagent approach in which each agent was an ε-greedy reinforcement learner which used a standard value update $Q(a) \leftarrow Q(a) + \alpha(r - Q(a))$, where a is the agents' action selection, r the reward received for taking action a, α the learning rate, and Q the value associated with taking action a. (Sutton & Barto, 1998). At every time step, the agent chooses the action with the highest table value with probability $1 - \varepsilon$ and chooses a random action with probability ε.

4.1 Teams and Hierarchical Organization

In this work, we utilized two types of organization: no teams and hierarchically coordinated teams. This section includes descriptions of each type of organization used.

4.1.1 No teams

Throughout this work, agents receive learning signals via three different reward structures: global, difference, and EDR. Global rewards provide agents with a learning signal that is equivalent to the system performance. Global rewards are in-line with the system objective, meaning that if agents maximize their own rewards they concurrently optimize the system performance. Unfortunately, global rewards provide agents with a noisy learning signal as each agent's reward depends directly upon the actions of all agents in the system. Here, all agents receiving a global reward signal get the same feedback regardless of their actions, meaning that they may receive a good reward for taking a poor action, or a bad reward for taking a good action (their rewards are highly impacted by the actions of other agents). Difference rewards address this shortcoming by filtering the noise off of the global reward signal and providing agents with specific feedback on how their actions impacted the system performance (Section 2.3).

Here, we derive the difference reward for the DCP problem when no hierarchies or teams are present. When no teams are present, each agent is required to coordinate directly with all other agents in the system. In this setting, the difference reward D_j for agent j is derived by combining Equations (2) and (4):

$$D_j = \begin{cases} \dfrac{\left|\sum_{i=1}^{N} n_i a_i\right|}{\sum_{i=1}^{N} n_i} - \dfrac{\left|\sum_{i \neq j}^{N} n_i a_i\right|}{\sum_{i \neq j}^{N} n_i} & \text{if } n_j = 1 \\ 0, & \text{if } n_j = 0 \end{cases}$$

In the DCP, D_j only provides agents with a clear learning signal if $n_j \neq 0$. If an agent chooses to be turned 'off' ($n_j = 0$), it receives a reward of 0 (it had no impact on the system) whether that action was good or bad for the system performance. This means half of the actions an agent takes will effectively be random as far as the system performance is concerned.

Next, we derive the EDR (Section 3.1) for the DCP problem by combining Equations (3) and (4). Consider the case where the probabilities are equivalent for each action 'on' and 'off', $P_{n_j=0} = 0.50$ and $P_{n_j=1} = 0.50$, EDR_j becomes the following for the standard DCP problem

(Section 3.1):

$$EDR_j = \begin{cases} 0.50 \dfrac{\sum_{i \neq j}^{N} n_i a_i - a_j}{\sum_{i \neq j}^{N} n_i - 1} - 0.50 \dfrac{\sum_{i=1}^{N} n_i a_i}{\sum_{i=1}^{N} n_i}, & \text{if } n_j = 1 \\ 0.50 \dfrac{\sum_{i \neq j}^{N} n_i a_i + a_j}{\sum_{i \neq j}^{N} n_i + 1} - 0.50 \dfrac{\sum_{i=1}^{N} n_i a_i}{\sum_{i=1}^{N} n_i}, & \text{if } n_j = 0 \end{cases}$$

EDR_j provides a clear learning signal: if it is positive, the action taken by agent j was beneficial to system performance, and if EDR_j is negative, the action was harmful to system performance. Agents trying to maximize EDR_j will implicitly maximize system performance simultaneously (Section 2.3). Both D_j and EDR_j require very little information to compute. Any system capable of broadcasting G can be minimally modified to accommodate D_j or EDR_j.

Though the simplest way to organize a multiagent system is to have no teams and no hierarchical organization, as agent scaling increases, coordination can become too complex for a non-hierarchical system to be effective. We address this shortcoming by incorporating teams and hierarchical organization into the system.

4.1.2 Hierarchically coordinated teams

In this section, we address the coordination issues introduced by large multiagent systems by first creating individual teams of agents within the system, and then superimposing a hierarchical control layer on top of each team. In this setting, individual teams are treated as though they were a single sensor and each 'team sensor' is controlled by a single control agent. These top-layer control agents are responsible for coordinating the actions of the teams. This results in a two-layer hierarchical network structure, which reduces agent-to-agent coordination complexity and information overhead within the system (Figure 1, right). As seen in the right side of Figure 1, the bottom layer consists of teams of C sensing agents (each sensing agent is randomly assigned to a single team). Each team acts independently to optimize its own internal objective, which is simply to minimize its own attenuation. Here, the teams do not directly communicate, instead they rely upon the top-layer control agents to choose when the team will and will not participate in system-level sensing. Thus, the control agent placed over each team effectively becomes a 'high-level sensor' whose attenuation is equal to the aggregate attenuation of the team it controls. These control agents form their own group and coordinate in order to optimize the system-level attenuation by choosing when individual teams participate in system-level sensing (Section 4.1.2).

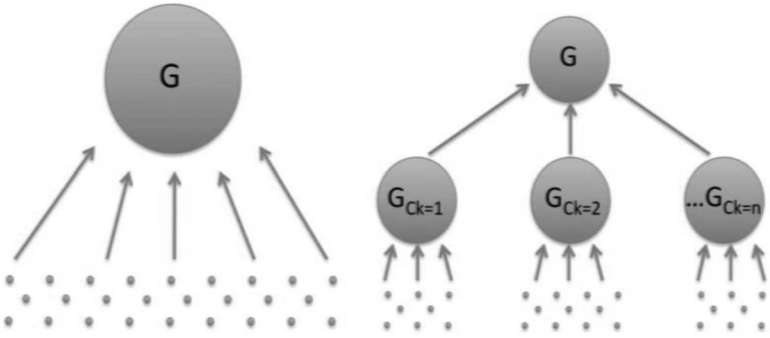

Figure 1 When no teams or hierarchical organization is present (left), agents are required to coordinate directly with all other agents to optimize the global objective G. We reduce this coordination requirement by adding in a two-layer hierarchical structure (right). Here, sensing agents are partitioned into separate teams and coordinate to optimize their team objective G_{c_k} according to Equation (4). Then, a control agent is assigned over each team, and the control agents coordinate to optimize the system-level objective G (Section 4.1.2)

Algorithm 1: Learning in hierarchically coordinated teams: In the DCP, the sensing agents and control agents were both ε-greedy reinforcement learners. Due to the sensitivity of this domain, the learning for the sensing agents and the control agents were separated. First, the sensing agents learned how to coordinate the actions of their individual teams while the control agents behaved randomly. Then, learning was turned off for the sensing agents and they followed their fixed learned policies while the control agents began to learn. The primary reason for training these two types of agents separately is that due to the combinatorial nature of the DCP, it is difficult if not impossible for the control agents to effectively coordinate the actions of the teams until the teams are following fixed policies.

Given a set of N sensing agents and M control agents
Randomly partition agents into M teams and create a control agent for each team
for $Run = 1 \rightarrow Run_{\text{Max}}$ **do**
  for $Episode = 1 \rightarrow \frac{Episode_{\max}}{2}$ **do**
    Sensing Agents Select Action (ε-greedy)//Sensing agents are learning
    Control Agents Select Random Action//Control agents behave randomly
    Calculate System Performance:

$$G_H = \frac{\left|\sum_{k=1}^{K} A_{c_k} n_k\right|}{\sum_{k=1}^{K} N_{ck} n_k} = \frac{\left|\sum_{i=1}^{N} n_i a_i\right|}{\sum_{i=1}^{N} n_i}$$

    Calculate Sensing Agent Rewards (G_{c_k}, D_{c_k}, or EDR_{c_k})
    Perform Value Update for Sensing Agents//Only sensing agents are learning
  end for
  for $Episode = \frac{Episode_{\max}}{2} \rightarrow Episode_{\max}$ **do**
    Sensing Agents Select Actions Greedily//Sensing agents use their fixed learned policies
    Control Agents Select Action (ε-greedy)//Control agents are learning
    Calculate System Performance:

$$G_H = \frac{\left|\sum_{k=1}^{K} A_{c_k} n_k\right|}{\sum_{k=1}^{K} N_{c_k} n_k} = \frac{\left|\sum_{i=1}^{N} n_i a_i\right|}{\sum_{i=1}^{N} n_i}$$

    Calculate Control Agent Rewards (G_H, D_H, or EDR_H)
    Perform Value Update for Control Agents//Only control agents are learning
  end for
end for

In the hierarchical setting, agents in the bottom layer attempted to optimize the attenuation of their individual teams for a single reading (according to Equation (4)), while the control agents dictated both if and when each team would participate in the aggregated system sensor reading. Thus, instead of turning 'on' or 'off' like the sensing agents, the control agents each turned an entire team on or off (Algorithm 1). In the DCP, the top-level control agents coordinated in order to optimize the standard DCP system objective (Equation (4)):

$$G_H = \frac{\left|\sum_{k=1}^{K} A_{c_k} n_k\right|}{\sum_{k=1}^{K} N_{c_k} n_k} = \frac{\left|\sum_{i=1}^{N} n_i a_i\right|}{\sum_{i=1}^{N} n_i} \tag{5}$$

where G_H is the objective of the control agents in the hierarchical system for the standard DCP (equivalent to the system objective in the DCP—Equation (4)), A_{c_k} the aggregated attenuation of team c_k, N_{c_k} the total number of active devices in team c_k, K the total number of teams (N/C), $n_k \epsilon\{0,1\}$ depending on whether the agent i governing team k chose to turn team c_k *on* or *off*. The goal of the control agents G_H is to combine the team attenuations A_{c_k} and participations N_{c_k} in such a way that they optimize the system-level attenuation G.

In the DCP, the hierarchical control agents are attempting to optimize the system performance, G, directly by coordinating the actions of individual teams in the system. In this setting, the difference and EDRs can be derived by combining Equations (2) and (5), and Equations (3) and (5), respectively. We first

derived the rewards for hierarchical control agents using difference rewards, by combining Equations (2) and (5), yielding the following:

$$D_{j,H} = \begin{cases} \frac{\left|\sum_{k=1}^{K} n_k A_{c_k}\right|}{\sum_{k=1}^{K} n_k} - \frac{\left|\sum_{k\neq j}^{K} n_k A_{c_k}\right|}{\sum_{k\neq j}^{K} n_k}, & \text{if } n_j = 1 \\ 0, & \text{if } n_j = 0 \end{cases}$$

where $D_{j,H}$ is the difference reward of control agent j, which is in control of team c_j (i.e. control agent j chooses whether or not team c_j is turned 'on' or 'off' with respect to the system objective G), k an individual control agent, K the total number of control agents in the system, A_{c_k} the aggregated attenuation of team c_k, and n_k the total number of sensors participating in sensing for team k. Again, the difference reward is designed to provide each control agent with feedback that tells it how its actions impacted the overall system performance, but if the control agent chooses to turn its team 'off' frequently, it will receive rewards that are effectively random with regards to the system performance. Next, we derived the EDR for control agents in the standard DCP by combining Equations (3) and (5), which yielded the following:

$$EDR_{j,H} = \begin{cases} 0.50\frac{\sum_{k\neq j}^{K} n_k A_{c_k} - A_{c_j}}{\sum_{k\neq j}^{K} n_k - n_j} - 0.50\frac{\sum_{k=1}^{K} n_k A_{c_k}}{\sum_{k=1}^{K} n_k}, & \text{if } n_j = 1 \\ 0.50\frac{\sum_{k\neq j}^{K} n_k A_{c_k} + A_{c_j}}{\sum_{k\neq j}^{K} n_k + n_j} - 0.50\frac{\sum_{k=1}^{K} n_k A_{c_k}}{\sum_{k=1}^{K} n_k}, & \text{if } n_j = 0 \end{cases}$$

where $EDR_{j,H}$ is the EDR of hierarchical control agent j, which is in control of team c_j (i.e. control agent j chooses whether or not team c_j is turned 'on' or 'off' with respect to the system objective G), k an individual control agent, K the total number of control agents in the system, A_{c_k} the aggregated attenuation of team c_k, and n_k the total number of sensors participating in sensing for team k.

5 Experiments and results

We conducted the following set of experiments:

(1.) The DCP with no teams (Section 3.1).
(2.) The DCP with hierarchically coordinated teams (Section 3.1).
(3.) The DCP with failures using hierarchically coordinated teams (Section 3.1).

All experiments were simulated and the results shown here are the corresponding results from these empirical experiments. There were five different types of agents used. The first type of agents are controlled by a single centralized algorithm, which simply turns on the single best sensing device for each time step the best sensor (TBS). Although selecting the best sensor is conceptually simple, it is a centralized algorithm and requires global coordination. Selecting the best *single sensor* is fundamentally different than choosing the best subset of sensing devices such that their collective readings result in a better attenuation than any single device can achieve independently. The second type of agents behave completely randomly (R). The next three types of agents are learning agents attempting to optimize global (G), D, or EDR structures. These rewards were derived separately for agents in the no teams and hierarchically coordinated teams experiments (Section 4).

At the beginning of each experimental run, the attenuations a_i for each agent were drawn from a Gaussian distribution of 0 mean and unit variance. All experiments had 10000 episodes, were averaged over $r = 100$ statistical runs, and were plotted with the error of the mean $\sigma/\sqrt{r}$ (the error in the mean is plotted in Figures 2–7, but it is so small that it is frequently not visible). The results are statistically significant as we performed a t-test with $p = 0.05$ for all experiments. The learning rate was set to $\alpha = 0.05$ (performance was not overly sensitive to α). In the non-hierarchical team-free experiments, all agents had an exploration rate of $\varepsilon = (1)/(N)$, where N was the number of sensing agents in the system. In hierarchical team-based experiments, agents in the bottom

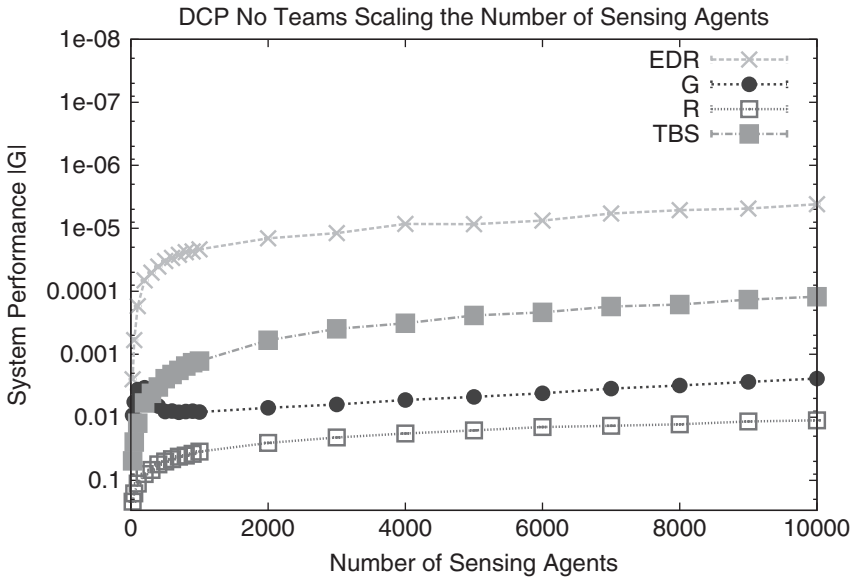

Figure 2 Scaling the number of sensing agents in the Defect Combination Problem (DCP) with no teams. As seen, with up to 1000 agents present in the system, agents using *D* and expected difference reward (*EDR*) rewards outperform all other methods by up to nearly three orders of magnitude. When the system is scaled further, agents using *D* have diminishing performance due to a lack of reward feedback (Section 4.1.1), while agents using *EDR* rewards continue to outperform all other methods with up to 10000 sensing agents

layer had an exploration rate of (1)/(*C*), where *C* is the number of agents per team, and the top layer had an exploration rate of (*C*)/(*N*) (exploration was inversely proportional to the number of agents coordinating together in a particular group). All value tables and Q-tables were initialized to 0. For all agents, for the first 20 time steps, learning was turned off and agents chose random action selections. After the first 20 steps, learning was turned on for 60 agents at a time until all of the agents were learning, in the mean time agents who had not been switched on continued performing randomly.[1]

5.1 No teams in the Defect Combination Problem

The first set of experiments shows the performance of agents solving the DCP problem using learning without teams or hierarchical organization. Here, each agent must coordinate directly with all other agents in the system. In these experiments (Figures 2–4) agents using random action selections, *R*, utilizes approximately half of the sensors each time step, but performs poorly since the selection of which sensors is completely random. Similarly, agents using a global reward *G* turn on approximately half of the sensing devices and make better decisions selecting which sensors to turn on. This results in *G* outperforming *R* by approximately an order of magnitude in most settings (Figures 2–4). However, agents using *G* still have difficulty differentiating the impact of their own actions on their reward signal from the actions of other agents (in this setting, each agent's reward signal is directly impacted by the actions of all other agents). This is because with *G*, all agents receive the system performance as their reward signal, regardless of how their own actions impacted the system performance. This makes it difficult for these agents to coordinate their actions, inhibiting system performance.

D and *EDR* address this shortcoming by effectively filtering out the impact of other agents on an agents' reward signal and accounting for each agent's individual contribution to the system performance. When relatively few agents are present in the system, agents using *D* outperform many other methods (Figure 2). However, as seen in Figures 2 and 3, when scaling is increased agents using *D* perform poorly in this experiment. This is because *D* only provides constructive feedback when the agent elects to be active. Agents using *D* receive a reward of 0 when they choose to remain off so they do not receive enough

[1] Allowing all agents to begin learning simultaneously created a "spike" into the system which significantly slowed down learning. The gradual introduction of the learning agents is softens this discontinuity in learning (Tumer, 2005).

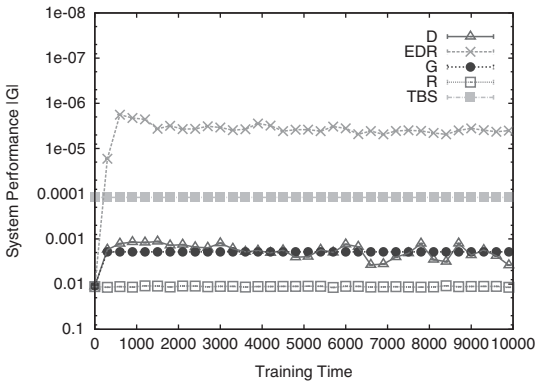

Figure 3 Performance for 10000 sensing agents in the standard Defect Combination Problem (no hierarchies). Agents must choose whether to be 'on' or 'off'. As seen, agents using expected difference reward (*EDR*) obtain significantly better aggregated attenuation than the best single sensor TBS. Agents using *EDR* rewards perform well because these rewards promote agent-to-agent coordination and decision making

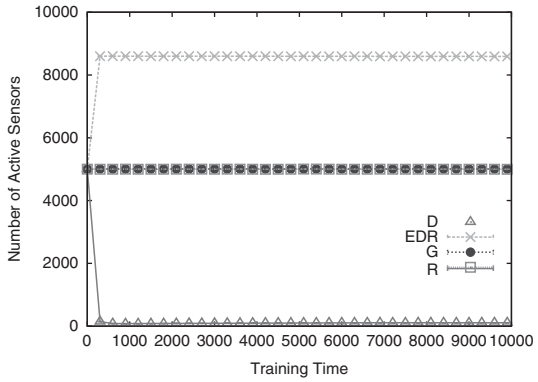

Figure 4 Performance for 10000 sensing agents in the standard Defect Combination Problem (no hierarchies). *G* and *R* both use nearly the optimal number of sensors 50% (Challet & Johnson, 2002), but achieve poor performance. Although *D* and expected difference reward (*EDR*) are both difference rewards, they lead to very different policies. *D* uses few sensors, while *EDR* uses nearly 80% of the sensing devices

feedback to successfully coordinate their actions (Section 4.1.1). This is because if an agent using *D* initially harms the system and gets a negative reward, it chooses to remain 'off' a majority of the time (except for a small amount of exploration). If the agent continues to get negative rewards for being turned on while it explores, it will be hard to overcome. This is because while the value of being turned off remains at 0, the negative rewards for being turned on effectively 'stack' and discourage the agent from turning on. It would take a significant amount of 'positive' reward to make up for these initial negative rewards. One solution to this problem would be to decrease the value of the learning rate α, which would effectively truncate the effective build up of 'negative' rewards on the value. On the other hand, the EDR structure, *EDR*, does not give a 0 reward to an agent for being on or off, instead it gives an estimated value of the agents cumulative impact on the system over time based upon its historic action selections (Section 4.1.1), resulting in better performance than *D* in this case (Figures 2 and 3). As seen, in this setting *EDR* significantly outperforms all other rewards (Figures 2 and 3).

It is clear from this experiment that the way an agent handles the information it receives drastically impacts the performance. Agents using *G*, *D*, and *EDR* received the exact same information, yet agents using difference rewards were able to routinely outperform agents using a traditional global rewards by approximately two or three orders of magnitude in most settings (agents using *D* experienced difficulty learning with more than 1000 agents in the system due to the lack of reward feedback described previously). Here, both *D* and *EDR* reduce the overall coordination complexity for individual agents by filtering much of the noise of other agents' actions from each agent's reward signal (Section 2.3). It is interesting to note that although *D* and *EDR* achieve similar

performance in most settings, they both arrive at very different policies. Agents using D typically learn to use <10% of the overall sensors, while agents using *EDR* use nearly 80% (Figure 4). Both of these solutions used far from the theoretical optimal number of sensors, which was determined to be 50% in Challet and Johnson (2002). The key difference between these two reward structures is how they account for an individual agents' contribution to the system. D emphasizes the impact of an agents' action during a single time step, removing the agent from the system for the current time step and determining how it impacted the system performance. The problem with emphasizing a single time step in a hard combinatorial optimization problem such as this one is that the environment changes too rapidly for a single episode to provide significant enough feedback that agents can coordinate their actions properly. This is why D performs better with a lower learning rate α. A lower learning rate allows the agent to effectively observe multiple episodes and have them count with the same weight in the decision-making process. Instead of requiring a low α, *EDR* is designed to provide an agent with a view of how it impacted the system over multiple episodes. This reward explicitly accounts for the historical behavior of an agent and leverages that information into obtaining how the agent's behavior generally impacts the system performance.

These results tell us that shaped rewards alone may not be enough to maximize system performance for this problem. The inability of agents to achieve an optimal solution stemmed from the fact that each agent received information involving every other agent in the system. Even though difference rewards filter this information and improve performance, it is clear from these results that difference rewards alone are not enough to handle the coordination complexities present in such large multiagent systems (Figure 2). Now that we have demonstrated the shortcoming of using only reward shaping with difference rewards to scale to large multiagent systems of up to 10000 devices, we will demonstrate how hierarchical organization can be combined with these reward shaping techniques to improve coordination and performance in these systems.

5.2 Hierarchically coordinated teams in the Defect Combination Problem

Next, we implement a two-layer hierarchy into the DCP with 10000 sensing agents (Section 4.1.2). We conduct four experiments, where agents were randomly grouped into teams of $C = 25$, 50, 100, and 200, respectively. A single control agent was then placed over each individual team and the control agents coordinate the actions of the teams in order to optimize the system objective (Section 4.1.2). Adding a two-layer team-based hierarchical structure to this system reduces the communication overhead for each agent by ~99% (each individual agent in the system only has to coordinate directly with a fraction of the other agents in the system). This reduces the amount of noise an agent has to deal with in regards to its own reward signal, resulting in a cleaner signal and better action selections.

This hierarchical approach assigns a control agent for each team which determines how the team participates in sensing. This approach addresses the two key issues that inhibited the performance in when no teams were present. First, it reduces the agent-to-agent coordination complexity by adding structure and organization to the system. Agents now only need to coordinate with other agents in their team (agents in the top level of the hierarchy form their own team). Second, the information sharing and processing requirements are reduced by ~99% for all agents within the system. Here, individual sensing agents continue to optimize their local team objective G_{c_k}, while the hierarchical agents directly optimize their own reward G_H (which is the DCP system objective G in these experiments). Here, the team-based sensing agents continued optimizing their individual team objective, G_{c_k}, while the top-layer control agents focused directly on coordinating the actions of the teams to directly optimize the system objective (Equation (5)).

Here, the top- and bottom-level teams were trained separately (Algorithm 1). First, the team-based agents learned for 5000 time steps while the control agents took random actions and did not learn. Then, the bottom-level agents' learning was turned off and they followed their learned policies while the top-layer agents' learned for the next 5000 time steps. This was done for two key reasons: (1) the actions of the agents in the bottom layer were independent of agents in the top layer, and (2) due to the combinatorial optimization nature of the DCP, the control agents could not make optimal decisions until the actions of the teams were set. The separate training of the top and bottom levels of the hierarchy is responsible for the learning spike at 5000 time steps in Figures 5–7.

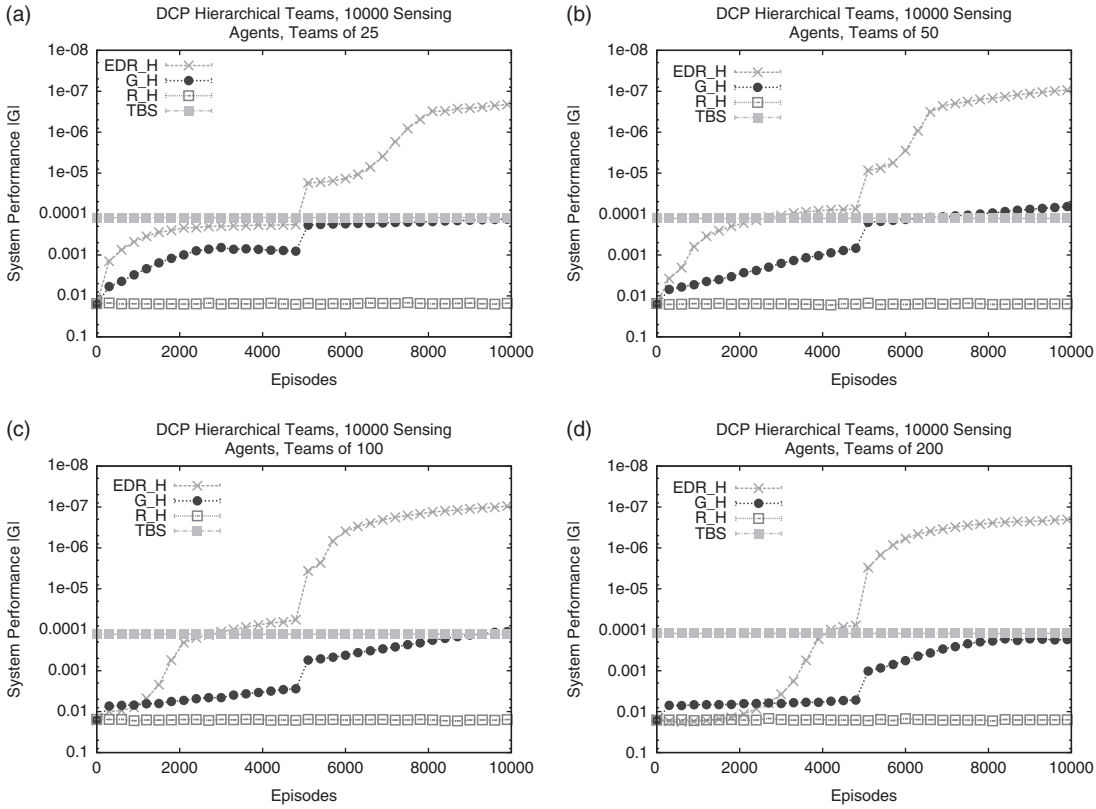

Figure 5 Performance for 10000 sensing agents in the standard Defect Combination Problem (DCP) with hierarchical organization and teams of $C = 25$, 50, 100, and 200 (Section 4.1.2). Agents are randomly partitioned into separate teams and control agents are placed over the top of each team to coordinate how teams participate in the system. The transition between team-based agents learning between episodes 0 and 5000 and the control agents learning from after episode 5000 to episode 10000 is the reason for the discontinuity in learning performance around 5000 episodes in each graph involving hierarchical organization (Algorithm 1). Here, for the first 5000 episodes, performance improves as individual teams improve their attenuations. Then, for the next 5000 episodes, performance increases as control agents learn to coordinate the behavior of the teams in the system. As seen, shaped difference rewards coupled with hierarchical organization (D_H and EDR_H) outperform all other approaches (Figures 2–3). The hierarchy dictates the control flow and reduces the information overheads, while difference rewards improve agent decision making by making efficient use of locally available information. EDR = expected difference reward

As seen in Figure 5, hierarchical organization benefits agents using global, difference, and EDR structures (G_H, D_H, and EDR_H). In fact, these results show that coupling hierarchical organization and difference rewards can outperform either approach individually by orders of magnitude. Agents using the global, difference, and EDR structures with hierarchical organization all significantly improve their performance over the team-free setting (Figure 2). Observing the performance of random agents in a hierarchical setting (R_H) shows that although hierarchical organization can reduce information overheads, reduce processing requirements, and dictate the control flow of the system, without a good decision-making algorithm, the system performs poorly. Similarly, observing the performance of agents using traditional global reward based learning for decision making also achieve relatively low performance (agents using global rewards and hierarchical organization G_H are barely able to achieve the same performance as the best single sensing device in the system, TBS). This shows that simply adding hierarchical organization to the system may not be enough to maximize system performance. Adding a hierarchy reduces coordination complexity and information overheads, but it does not attempt to optimize agent decision making given the information each agent receives.

Agents utilizing global rewards achieve nearly an order of magnitude better performance when a hierarchical structure was added to the system compared with global rewards with no teams. This is because, in addition to reducing the information overhead, the hierarchical structure allows teams to coordinate their actions together to

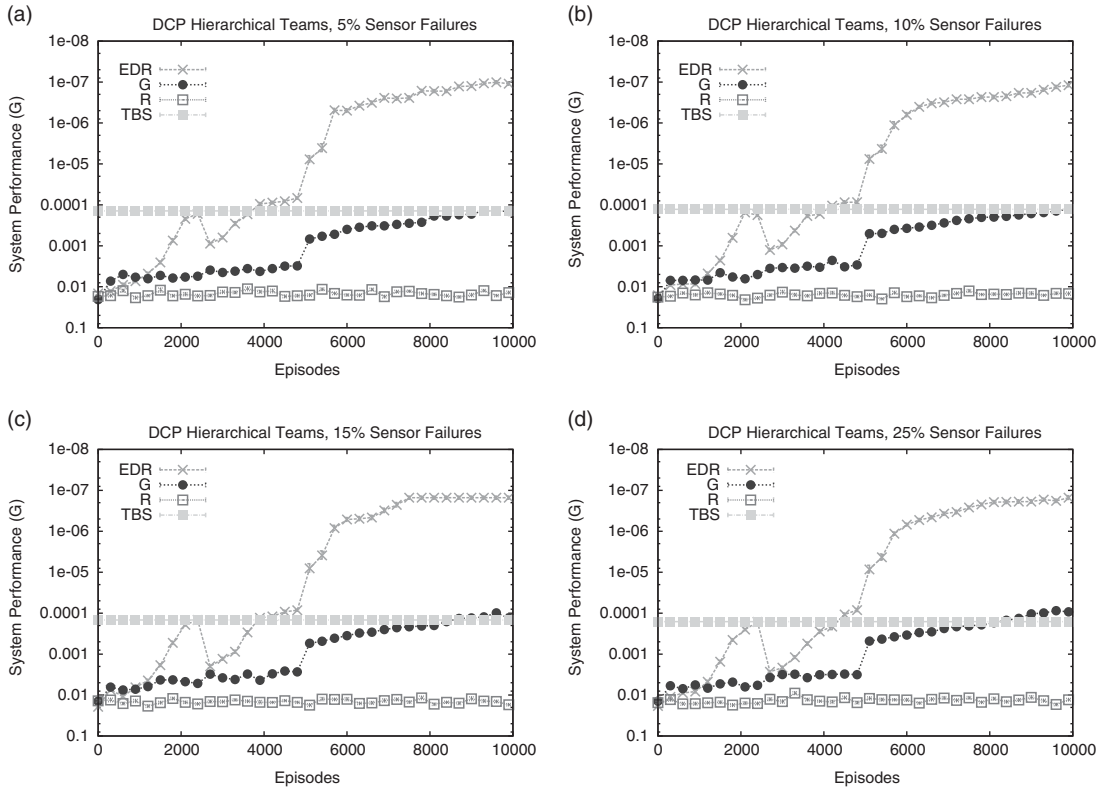

Figure 6 Performance for 10000 sensing agents in 10000 sensing agents solved the standard Defect Combination Problem (DCP) with hierarchical organization and teams of $C = 25$, 50, 100, and 200 (Section 4.1.2); 10000 sensors solving the DCP with hierarchical teams (Section 4.1.2); 10% of the bottom layer agents fail after 2500 time steps. The discontinuity at 5000 time steps is due to hierarchical learning (Section 5.2). As seen, when 10% of the sensing devices fail, the remaining 90% are able to coordinate to adapt their behavior and recover most of the lost system performance. EDR = expected difference reward

improve system performance. Through reducing the information overhead, agents are able to better determine their own individual impact on their rewards. This allows them to make better decisions when attempting to optimize their individual reward both in a team setting as well as a control agent setting. Despite the benefits from the addition of a hierarchical structure, agents using traditional global reward structures and hierarchical organization (G_H) were still unable to achieve the same performance as agents using shaped rewards without a hierarchical structure (Figures 2–3). This is why agents using *D* and *EDR* rewards in a non-hierarchical setting where the information overhead and coordination complexity remain high still outperform a traditional global reward G_H in a hierarchical structure. However, agents using a combination of a hierarchical structure and shaped rewards outperform non-hierarchical approaches by orders of magnitude (Figure 5), which supports the fact that hierarchical structures and shaped rewards offer complimentary benefits in large-scale multiagent systems. Hierarchical organization dictates the control flow and reduces the information overheads, while shaped rewards improve agent decision making given the information each received.

5.3 Hierarchically coordinated teams with failures in the Defect Combination Problem

Now that we have established that a combination of shaped rewards and hierarchical organization can dramatically improve the performance of large multiagent systems, we want to demonstrate the robustness of such an approach to component failures. In the context of this experiment an agent (controller or sensor) getting stuck *on* will constitute a failure. This type of failure would occur when sensors within the system fail to coordinate their actions with other agents, forcing the other agents to adapt their policies to account for these rouge agents. As failures in the top and bottom layers of the hierarchy may impact the system

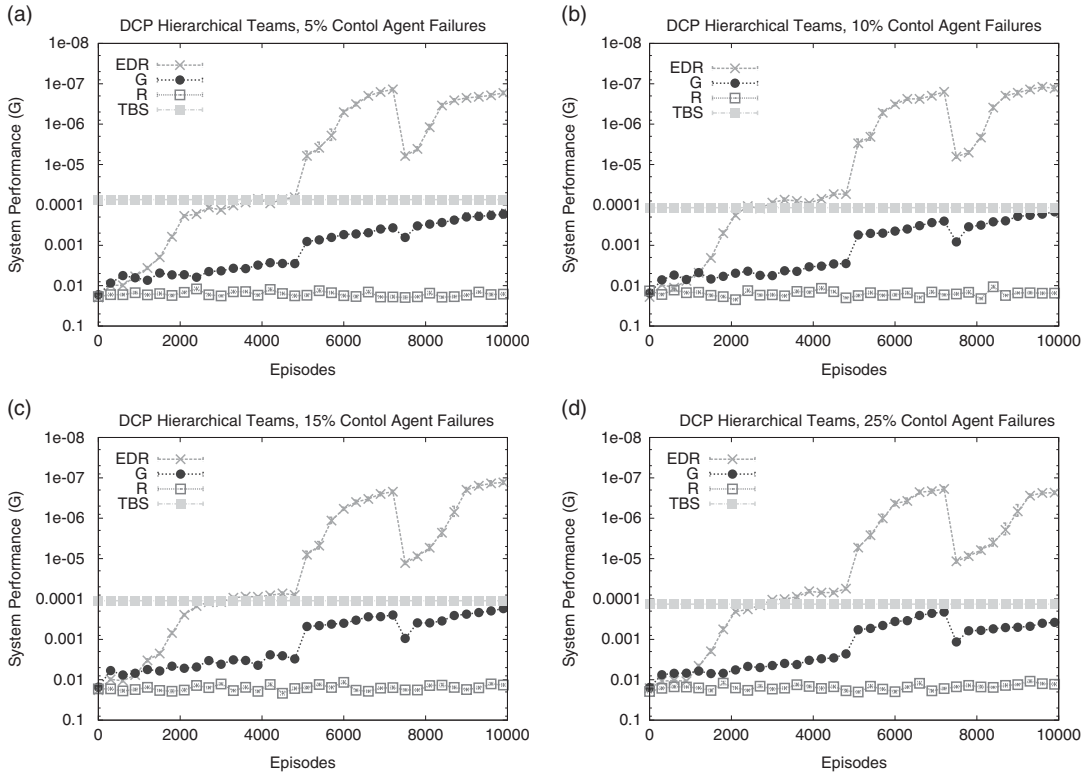

Figure 7 Performance for 10000 sensing agents in 10000 sensing agents solved the standard Defect Combination Problem (DCP) with hierarchical organization and teams of $C = 25$, 50, 100, and 200 (Section 4.1.2); 10000 sensors DCP with hierarchical teams (Section 4.1.2); 10% of top-layer control agents fail after 7500 time steps. The discontinuity at 5000 time steps is due to hierarchical learning (Section 5.2) and the discontinuity at 7500 episodes is due to the occurrence of failures. As seen, a combination of shaped rewards and hierarchies proves robust to top-layer failures. Agents using D_H and EDR_H far outperform agents using a standard global reward G_H. EDR = expected difference reward

differently, we perform a separate experiment for each case. In the first set of experiments 10 – 50% of the bottom-level sensors fail after 2500 time steps (Figure 6), while the other sensors continue learning. In the next set of experiments 10 – 50% of the top-level control agents fail at time step 7500, while the others continue learning (Figure 7). In both cases, the team-based agents learn for the first 5000 episodes and the hierarchical control agents learn for the second 5000 episodes (Algorithm 1).

As seen in Figure 6, a combination of difference rewards and hierarchies is robust to failures within each individual team. Here, a portion of the agents in each individual team fail after 2500 time steps and the remaining sensing devices need to coordinate their actions with these defective devices in order to recover system performance. Due to the reduced coordination requirements imposed by the hierarchical organization, team-based sensing agents only need to coordinate their actions with 100 other agents. These reduced coordination requirements coupled with agents using difference rewards enable them to coordinate in order to regain the performance lost due to failures. In the next experiment (Figure 7), a portion of the control agents failed, each one impacting an entire team of sensing agents. However, as the individual teams maintained relatively low attenuations, when control agents failed and remained on, the remaining control agents were still able to coordinate their actions in order to achieve good performance even in the presence of failures.

6 Discussion

In very large multiagent systems complete information sharing between agents to promote coordination is often impractical. Even when complete information is available, there is frequently too much information for each agent to process. In such systems, agents frequently encounter two key problems: (1) increased coordination requirements, and (2) increased information sharing and processing requirements (agents

frequently receive more information than they can effectively process). We address both of these issues by combining two well-known coordination mechanisms, hierarchical organization, and shaped difference rewards. Hierarchies dictate the control flow and information handling, lowering the 'per agent' coordination complexity in the system. Here, hierarchical organization governed the control flow of the system and reduced the information sharing and processing requirements of individual agents by ~99%. On the other hand, difference rewards act to optimize information processing, serving as an information filter (extracting only the specific information relative to a particular agent) and promote agent coordination. Difference rewards filtered the information each agent received, extracting only the specific information relative to that particular agent. Our results show that a combination of shaped difference rewards and hierarchical organization can improve coordination, scalability, and performance in large multiagent systems. Combining these approaches led to approximately three orders of magnitude improvement over either method individually in the DCP, and shows promise for other large multiagent domains including sensor networks, aerial vehicle coordination, and network traffic management.

This work showed the potential advantages of combining coordination algorithms in ways that leverage their benefits. Although many coordination algorithms exist throughout the literature, they have primarily been used independently and relatively little work has focused on the performance increases attainable by combing them. Future work would include finding new combinations of coordination algorithms that can be used to improve both agent-to-agent coordination as well as overall scalability. In particular, selecting coordination mechanisms that are synergistic and not only work well together but actually magnify each others benefits. This could be done by defining a set of metrics and characteristics of coordination mechanisms, which could then be used to determine when coordination mechanisms may be merged together to improve performance.

Acknowledgments

This work was partially supported by the National Science Foundation under grant 0931591 and the National Energy Technology Laboratory under grant DE-FE0000857.

References

Agogino, A., HolmesParker, C. & Tumer, K. 2012. Evolving large scale UAV communication system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Philadelphia, PA, July.

Agogino, A. & Tumer, K. 2008. Analyzing and visualizing multi-agent rewards in dynamic and stochastic domains. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)* **17**(2), 320–338.

Barrett, S., Stone, P. & Kraus, S. 2011. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *Proceedings of 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, May.

Bharathidasan, A. & Ponduru, V. 2003. Sensor networks – an overview. *IEEE Potentials*.

Challet, D. & Johnson, N. 2002. Optimal combination of imperfect objects. *Physics Review Letters* **89**, 028071.

Devlin, S. & Kudenko, D. 2011. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Farinelli, A., Rogers, A. & Jennings, N. 2008. Maximising sensor network efficiency through agent-based coordination of sense/sleep schedules. In *Workshop on Energy in Wireless Sensor Networks*.

Grzes, M. & Kudenko, D. 2010. Online learning of shaping rewards in reinforcement learning. *Neural Networks* **23**, 541–550.

Hayden, S., Carrick, C. & Yang, Q. 1999. A catalog of agent coordination patterns. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*.

HolmesParker, C., Agogino, A. & Tumer, K. 2012. Evolving distributed resource sharing for cubesat constellations. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Philadelphia, PA, July 2012.

HolmesParker, C., Agogino, A. & Tumer, K. 2013. Exploiting structure and utilizing agent-centric rewards to promote coordination in large multiagent systems (extended-abstract). In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Horling, B. & Lesser, V. 2005. A survey of multiagent organizational paradigms. *Knowledge Engineering Review* **19**(4), 281–316.

Horling, B., Mailler, R. & Lesser, V. 2004. A case study of organizational effects in a distributed sensor network. In *Proceedings of the International Conference on Intelligent Agent Technology*.

Howley, E. & Duggan, J. 2011. Investing in the commons: a study of openness and the emergence of cooperation. *Advances in Complex Systems* **14**.

Knudson, M. & Tumer, K. 2010. Coevolution of heterogeneous multi-robot teams. In *Genetic and Evolutionary Computation Conference (GECCO)*.

Kok, J. & Vlassis, N. 2006. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research (JMLR)* **7**, 1789–1828.

Mehta, N., Ray, S., Tadepalli, P. & Dietterich, T. 2008. Automatic discovery and transfer of maxq hierarchies. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

Ng, A., Harada, D. & Russell, S. 1999. Policy invariance under reward transformations: theory and application to reward shaping. In *Proceedings of International Conference on Machine Learning*.

Panait, L. & Luke, S. 2005. Cooperative multi-agent learning – the state of the art. *Journal of Autonomous Agents and MultiAgent Systems (JAAMAS)* **11**(3), 387–434.

Rogers, A., Farinelli, A. & Jennings, N. 2010. Self-organising sensors for wide area surveillance using the max-sum algorithm. In *Self-Organizing Architectures*, **6090**, 84–100. Lecture Notes in Computer Science, Springer.

Sutton, R. & Barto, A. 1998. *Reinforcement Learning An Introduction*. MIT Press.

Tambe, M., Bowring, E., Jung, H., Kaminka, G., Maheswaran, R., Marecki, J., Modi, P., Nair, R., Okamoto, S., Pearce, J., Paruchuri, P., Pynadath, D., Scerri, P., Schurr, N. & Varakantham, P. 2005. Conflicts in teamwork – hybrids to the rescue. In *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Tham, C. & Renaud, J. 2005. Multi-agent systems on sensor networks a distributed reinforcement learning approach. In *Intelligent Sensors, Sensor Networks and Information Processing Conference (ISSNIP)*.

Tumer, K. 2005. Designing agent utilities for coordinated, scalable, and robust multiagent systems. In *Challenges in the Coordination of Large Scale Multiagent Systems*, P. Scerri, R. Mailler & R. Vincent (eds). Springer, 173–188.

Vinyals, M., Rodriguez-Aguilar, J. & Cerquides, J. 2010. A survey on sensor networks from a multiagent perspective. *The Computer Journal*.

Vrancx, P., Verbeeck, K. & Nowe, A. 2008. Decentralized learning in Markov games. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* **38**(4), 976–981.

Williamson, S., Gerding, E. & Jennings, N. 2009. Reward shaping for valuing communications during multi-agent coordination. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Wolpert, D. H. & Tumer, K. 2001. Optimal payoff functions for members of collectives. *Advances in Complex Systems* **4**(2/3), 265–279.

Xu, Y., Scerri, P., Yu, B., Okamoto, S., Lewis, M. & Sycara, K. 2005. An integrated token-based algorithm for scalable coordination. In *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Zhang, C., Abdallah, S. & Lesser, V. 2009. Integrating organizational control into multi-agent learning. In *Procceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.