The Knowledge Engineering Review, Vol. 00:0, 1–24. © 2004, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

Combining Reward Shaping and Hierarchies for Scaling to Large Multiagent Systems

Author 1

Address 1
E-mail: Email 1

Author 2

Address 2
E-mail: Email 2

Author 3

Address 3
E-mail: Email 3

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. Here, we combine shaped difference rewards and hierarchical organization in two domains with up to 10,000 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 within two variations of the Defect Combination Problem. 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 and 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 and Lesser, 2005; Kok and Vlassis, 2006; Tambe et al., 2005; Vrancx et al., 2008; Xu et al., 2005). Although significant progress has been made towards improving coordination and scalability with each of these methods individually, relatively little work has focused on leveraging the complementary benefits of these approaches. 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. In particular, we combine hierarchical organization to reduce each agent's communication overhead with reward shaping techniques that attempt to make optimal use of information locally available to agents in order to improve agent-to-agent coordination, increase scalability, and improve performance in large multiagent systems.

Reward shaping has been shown to drastically improve coordination, scalability, and performance in multiagent systems (Agogino and Tumer, 2008; Grzes and Kudenko, 2010; Williamson et al., 2009). 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, and distributed sensor networks (Agogino and Tumer, 2008; Agogino et al., 2012; HolmesParker et al., 2012; Knudson and Tumer, 2010; Tumer, 2005). 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 and 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 and 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 hierarchichal 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 and 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 two variations of the Defect Combination Problem (DCP) described in Section 3.1 (Challet and 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 Defect Combination Problem (DCP). Section 3.1 describes the two variations of the Defect Combination Problem. 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 Defect Combination Problem (DCP) utilized statistical physics to determine the theoretical optimal performance based upon the number of sensors (Challet and 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 nonhierarchical 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 nonhierarchical 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 and Lesser, 2005). All hierarchies have some form of these characteristics, although they may be implicitly present and not formally developed (Horling and 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 and Lesser, 2005; Horling et al., 2004; 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 and Lesser, 2005; Horling et al., 2004).[1] 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 2-layer hierarchy to enable decision making and decrease coordination requirements for agents.

[1] A comprehensive list of organizational structures and methods can be found in (Horling and Lesser, 2005).

2.2 Reward Shaping

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

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

where $Q(s,a)$ is the Q-value associated with the agent taking action a in state s, r is the standard reward, a' is an alternate action, s' is an alternate state, α is the learning rate, γ is the discount factor, and $F(s,s')$ is 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 (Devlin and Kudenko, 2011; Ng et al., 1999). 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 and Tumer, 2008; Devlin and Kudenko, 2011; Grzes and Kudenko, 2010; Williamson et al., 2009). Prior to introducing the shaped rewards used in this work, we will introduce two key characteristics of shaped rewards and two metrics for shaped rewards.

2.3 Factoredness and Learnability

Ideally, a reward should provide an agent with two key pieces of information: 1) How its action impacted the overall system performance, and 2) How its action impacted the reward it received. Feedback on how its own actions impacted the system performance allows agents to make decisions that are in-line with the system objective. Providing agents with feedback on how its individual actions impacted the reward it received allows the agent to adapt its actions in order to benefit both itself and the system.

This first property has been formalized for an agent j, by defining the **degree of factoredness** (also presented in (Tumer and Wolpert, 2004; Wolpert and Tumer, 2001; Wolpert et al., 1999)) between the agent-reward g_j and system reward G at state z, as:

$$F_{g_j} = \frac{\sum_z \sum_{z'} u\left[(g_j(z) - g_j(z'))(G(z) - G(z'))\right]}{\sum_z \sum_{z'} 1} \tag{2}$$

where the states z and z' only differ in the state of agent j, and $u[x]$ is the unit step function, equal to 1 if $x > 0$. The numerator keeps track of the number of state pairs (z, z') for which the agent-reward, $g_j(z) - g_j(z')$, and system reward, $G(z) - G(z')$, are aligned (have the same sign). A high degree of factoredness means that agents improving their own local reward are concurrently improving the system performance, while agents harming their local reward are also harming system performance.

The second property has been defined as **learnability**, which is the degree to which an agents reward, g_j, was impacted by its own actions as opposed to the actions of other agents. The learnability of a reward, g_j, for agent j, evaluated at z can be quantified as follows:

$$L_{g_j} = \frac{||g_j(z) - g_j(z - z_j + z'_j)||}{||g_j(z) - g_j(z' - z'_j + z_j)||} \tag{3}$$

where in the numerator z' differs from z only in the state of agent j, and in the denominator the state of all other agents is changed from z to z'. Intuitively, the learnability provides a ratio

between the portion of the agents reward signal that depended upon its own actions (signal), and the portion of its reward signal that depended upon the actions of all other agents (noise). The higher the learnability, the easier it is for an agent to learn an accurate mapping between its actions and its rewards.

2.4 Difference Rewards

Difference rewards were designed using the theory of collectives developed at NASA Ames Research Center by Wolpert et al.(HolmesParker et al., 2012; Agogino and Tumer, 2006; HolmesParker and Agogino, 2011; Tumer, 2005; Tumer and Wolpert, 2004; Wolpert et al., 1999). A complete definition and description of the difference reward can be found in (Agogino and Tumer, 2008; Tumer and Wolpert, 2004; Wolpert and Tumer, 2001; Wolpert et al., 1999). Difference rewards have been shown to work well in a number of domains and conditions (Agogino et al., 2012; HolmesParker et al., 2012; HolmesParker and Agogino, 2011; HolmesParker and Tumer, 2012; Agogino and Tumer, 2008; Tumer, 2005). In this work, we focus on a particular variation of difference rewards known as expected difference rewards, which have the following form (Agogino and Tumer, 2008):

$$EDR_j \equiv G(z) - E_{z_j}[G(z)|z_{-j}] \tag{4}$$

where G is the system objective, z is the complete system state vector, z_{-j} contains all the variables not affected by agent j, and $E_{z_j}[G(z)|z_{-j}]$ gives the expected value of G over the possible actions of agent j. Such rewards are factored because the second term does not depend on j's actions (Tumer, 2005). Furthermore, they usually have far better learnability than does a team reward, because the second term of EDR_j, which removes a lot of the effect of other agents (i.e., noise) from j's reward. 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. There are two key advantages to using EDR_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 (Agogino and Tumer, 2008; Tumer, 2005). Second, because the second term does not depend on the actions of agent j, any action by agent j that improves EDR, also improves G (the derivatives of EDR and G with respect to j are the same) (Agogino and Tumer, 2008; Tumer, 2005).

Any system capable of broadcasting the system performance G or passing state-vector information can be minimally modified to allow agents to independently calculate their own expected difference reward (Tumer, 2005). This information is commonly shared within distributed sensor networks, as there are intermittent data sinks that analyze, package, and re-broadcast data (Williamson et al., 2009; Farinelli et al., 2008). Additionally, systems such as the Aegis Ballistic Missile Defense System package system-level data and re-distribute it to individual nodes within the system for decenralized decision making (Lamber and Sinno, 2011).

3 Domains

3.1 The Defect Combination Problem (DCP)

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 (Farinelli et al., 2008; Rogers et al., 2010; Williamson et al., 2009). In this work, a set of up to 10,000 sensing devices must coordinate their sensing schedules in order to optimize their aggregated attenuation within a sensor network. This work focuses on the Defect Combination Problem (DCP) domain introduced in (Challet and Johnson, 2002). 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:

$$G = \frac{\left| \sum_{i=1}^{N} n_i a_i \right|}{\sum_{i=1}^{N} n_i} \tag{5}$$

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

This is an NP-complete optimization problem (Challet and 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 ($1 \leq K \leq N$).[2] To illustrate this, consider the case where there are 6 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 6 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 to from $|0.01|$ to $|0.18|$. Finding good solutions requires a great deal of coordination between sensors, as any one sensor can heavily impact the system performance.

3.2 The Time-Extended Defect Combination Problem (TEDCP)

Frequently, distributed sensor networks are comprised of battery-powered sensors which are deployed within an envirnoment to make observations. In such settings, the individual sensing nodes must not only coordinate in order to observe the environment, but they must also coordinate their actions such that they conserve resources such as power (battery life). Here, the majority of battery power is consumed on two key tasks: 1) transmitting information, and 2) taking measurements. The transmission of information can be reduced by applying an organizational structure (as is done in this work). Additionally, both the transmition of information as well as the frequency of taking measurements can be reduced for each sensor by having the sensors coordinate their sense/sleep schedules. In general, the goal would be to optimize the tradeoffs between observing the environment and minimizing the number of measurements performed by each individual sensing device.

In the original DCP problem (Equation 5) sensors must choose whether or not to participate in sensing, which involves a decision to be "on" or "off" in order to optimize their aggregated attenuations. We extend this one step further so that sensors must choose not only "if" they want to participate but "when" to participate, adding an additional degree of coordination complexity. Here, each agent may only participate in one of the M time steps per episode. This restriction couples the time slots, making this a significantly harder problem than solving multiple parallel versions of the DCP. In this setting agents try to optimize the average aggregated attenuation over M time slots, where the attenuation of a single time slot m is given by g_m:

[2]Statistical physics was used in (Challet and Johnson, 2002) to determine that the optimal percentage of active devices in the DCP should be 50% to optimize attenuation. However, no non-exhaustive method of selecting the optimal subset of devices was proposed.

$$g_m = \frac{\left| \sum_{i=1}^{N} n_{i,m} a_i \right|}{\sum_{i=1}^{N} n_{i,m}} \tag{6}$$

in this equation $n_{i,m} \in \{0, 1\}$ corresponding to whether sensor i chose to be "on" or "off" during time slot m (if agent i chose to participate during time-slot m, $n_{i,m}$ is 1, otherwise it is 0). Here, g_m represents the resultant attenuation for time slot m (analogous to Equation 5). Averaging the aggregated attenuation over M distinct time slots, the time-extended system objective becomes:

$$G_{TE} = \frac{1}{M} \sum_{m=1}^{M} g_m \tag{7}$$

where G_{TE} is the time-extended system objective, g_m is the aggregated attenuation of the sensors on time-slot m, and M is the total number of time-slots. Here the objective is to optimize the aggregated attenuation over M time steps.[3] It should be noted that there is no sequence of actions being taken in this case, instead, during each episode of learning agents are choosing which time-slot of the M time-slots they want to participate in as their action.

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 (Sutton and Barto, 1998) (though alternatives such as evolving neuro-controllers are also effective (Agogino and Tumer, 2004)). For complex delayed-reward problems, relatively sophisticated reinforcement learning systems such as temporal difference may have to be used. However, due to our agent selection and agent action set, the domains modeled in this paper only need to utilize immediate rewards. As a consequence, table-based immediate reward reinforcement learning is used. Our reinforcement learner is equivalent to an ϵ-greedy Q-learner with a discount rate of 0 (Sutton and Barto, 1998). At every episode an agent takes an action and then receives a reward evaluating that action. After taking action a and receiving reward r an agent updates its Q table (which contains its estimate of the value for taking that action (Sutton and Barto, 1998)) as follows:

$$Q(a) \leftarrow Q(a) + \alpha(r - Q(a)) \tag{8}$$

where a is the agents' action selection, r is the reward received for taking action a, α is the learning rate, and Q is the value associated with taking action a. At every time step the agent chooses the action with the highest table value with probability $1 - \epsilon$ and chooses a random action with probability ϵ.

4.1 Teams and Hierarchical Organization

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

4.1.1 No Teams

When there are no teams or organization present within a large multiagent system, all agents must coordinate directly together. Here, the ability of agents to learn to coordinate their actions is heavily impacted by the reward signal they receive. Throughout this work, agents receive learning signals via two different reward structures: global and expected difference rewards. When no teams

[3] Equation 7 reduces to Equation 5 when there is a single time slot ($M = 1$).

are present, global rewards provide agents with a learning signal that is equivalent to the system performance. Such 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 since 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). In the no teams setting, expected difference rewards help 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 2.4).

Here, we derive the expected 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 expected difference reward EDR_j for agent j is derived by combining Equations 4 and 5. 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) :

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

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 2.4). EDR_j rewards require very little information to compute and any system capable of broadcasting G can be minimally modified to accommodate 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 nonhierarchical system to be effective. We address this shortcoming by incorporating teams and hierarchical organization into the system.

4.1.2 Uncoordinated Teams

In contrast to the standard DCP problem approach in which all N agents observed each other and acted as a single group, we introduced a team-based approach. Here, we randomly partitioned the N agents into k teams, containing C_k agents, where each agent could only be a member of a single team. Random teams were assigned due to the NP-complete nature of the problem. The computational expense of intelligently assigning teams would be too high as it would require exhaustive search. Additionally, since there are no obvious ways of decomposing the system objective with respect to the teams, each team is treated as a separate DCP. The goal of each team is to optimize the aggregated attenuation of its C_k sensing devices (Equation 9). Agents within each team attempted to optimize their aggregated team attenuation according to the following:[4]

[4]In the team-based experiments (Sections 5.2-5.4), sensing agents' global, difference, and expected difference rewards were based upon the team objective (Equation 9).

$$G_{c_k} = \frac{|A_{c_k}|}{N_{c_k}} = \frac{\left|\sum_{i=1}^{C_k} n_i a_i\right|}{\sum_{i=1}^{C_k} n_i} \tag{9}$$

where G_{c_k} is the objective of team c_k, A_{c_k} is the aggregated attenuation of team c_k, N_{c_k} is the total number of active devices in team c_k, C_k is the number of sensing agents in team c_k, $n_i \in \{0, 1\}$ depending on whether sensor i chose to participate in sensing, and a_i is the attenuation of sensor i. A team approach is advantageous because it can reduce the coordination complexity of individual agents within the system by reducing the number of devices with which each agent has to communicate. Although this team formation approach reduces the coordination complexity and information overhead of the system, it may not lead to good system performance. This is because each team acts to optimize its own independent team objective G_{c_k}, without taking into account how its actions impact the overall system performance. We test this method because each team of C_k agents will have relatively low aggregated attenuations and by statistically averaging many teams with low aggregated attenuations an even lower attenuation may result.

Algorithm 1 – Team Formation: Given a set of N sensing agents, partition the sensing agents into equal teams. In the DCP, agents are randomly partitioned into teams due to the NP-complete nature of the problem. Any method of intelligently assigning teams would require extensive search and computational expense.

- Given: N sensing agents
- Initialize Agents

1. Randomly partition N sensing agents into M equal teams of size C
2. Assign a "local" team objective to each individual team (Equation 9):

$$G_{c_k} = \frac{\sum_{i=1}^{C} n_i a_i}{\sum_{i=1}^{C} n_i}$$

3. Assign agents individual rewards based upon their reward structures: G_{c_k} or EDR_{j,c_k}

We now derive the expected difference rewards for the team-based sensing agents in the standard DCP. In this setting, as discussed above, the sensing agents are randomly partitioned into teams and assigned team objectives G_{c_k}. Once assigned to teams, the sensing agents are not attempting to directly optimize the overall system objective G (Equation 5), but instead are actively attempting to optimize their team objective. This means that the expected difference rewards of agents in the team setting are based upon G_{c_k} instead of G. Thus, the expected difference rewards of team-based agents in the DCP can be derived by combining Equations 4 and 9. We derived the expected difference rewards for team-based sensing agents in the DCP by combining Equations 4 and 9, yielding the following:

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

where here we considered the case where the probability components of the expected difference reward structure for each agent were equivalent for each action "on" and "off", $P_{n_j=0} = 0.50$ and $P_{n_j=1} = 0.50$. Here, EDR_{j,c_k} is the expected difference reward for agent j which is a member of team c_k, n_i is an indicator which has a value of $n_i = 0$ if sensing agent i chose to be off, and $n_i = 1$ if sensing agent i chose to be turned on, and a_i is the attenuation of sensor i. This reward provides agent j with specific feedback on how it impacted the aggregated attenuation of team c_k. This will provide a positive learning signal if agent j was beneficial to the team's performance and a negative reward if agent j was detrimental to the team's performance. EDR_{j,c_k} provides a clear learning signal: if it is positive, the action taken by agent j was beneficial to team c_k's performance, and if EDR_{j,c_k} is negative, the action was harmful to team c_k's performance. Agents trying to maximize EDR_{j,c_k} will implicitly maximize team c_k's performance simultaneously (Section 2.4). Additionally EDR_{j,c_k} rewards require very little information to compute. Any system set up such that individual teams c_k are capable of broadcasting their team objective, G_{c_k}, to its members can be minimally modified to accommodate EDR_{j,c_k}.

4.1.3 Hierarchically Coordinated Teams

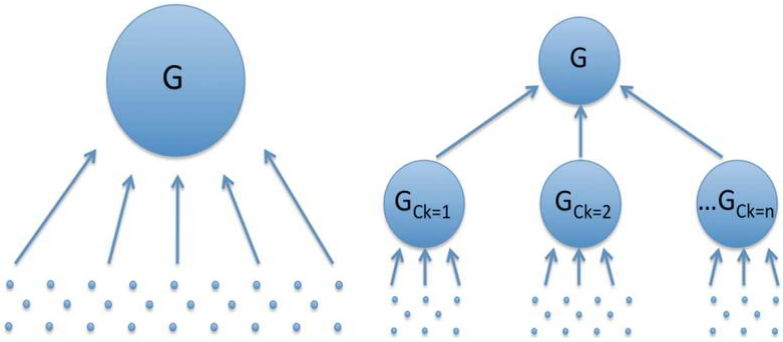

Figure 1: When no teams or hierarchical organization is present (left), agents are required to coordinate directly with all other agents to optimize the global objective G. We reduce this coordination requirement by adding in a two-layer hierarchical structure (right). Here, sensing agents are partitioned into separate teams and coordinate to optimize their team objective G_{c_k} (Equation 9). Then, a control agent is assigned over each team, and the control agents coordinate to optimize the system level objective G (Section 4.1.3).

As seen in the previous section, creating teams can decrease agent-to-agent coordination complexity and reduce information overhead. However, creating individual teams can be harmful to system performance if these teams fail to coordinate their actions well. We address this problem by superimposing a hierarchical control layer on top of each team (Algorithm 2). 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 2-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 agents, as described in Section 4.1.2. 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 4.1.3).

Algorithm 2 – Establish 2-Layer Hierarchical Organization: First, a set of N sensing agents are randomly partitioned into teams of equal size. Each team is assigned its own objective function, which its members (sensing agents) attempt to optimize. Hierarchical control agents are placed over each team and the control agents coordinate the actions of the teams in order to optimize the overall system performance G.

- Given: N sensing agents, M control agents
- Initialize Agents

1. Randomly partition N sensing agents into M equal teams of size C
2. Assign a "local" team objective to each individual team (Equation 9):

$$G_{c_k} = \frac{\sum_{i=1}^{C} n_i a_i}{\sum_{i=1}^{C} n_i}$$

3. Assign one control agent over each team
4. Assign a team objective to control agents (Equation 10):

$$G_H = \frac{\sum_{k=1}^{K} A_{ck} n_k}{\sum_{i=1}^{N} n_i}$$

In the hierarchical setting, agents in the bottom layer attempted to optimize the attenuation of their individual teams for a single reading (Equation 9), 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 3). In the DCP, the top level control agents coordinated in order to optimize the standard DCP system objective (Equation 5):

$$G_H = \frac{\left|\sum_{k=1}^{K} A_{c_k} n_k\right|}{\sum_{k=1}^{K} N_{c_k} n_k} = \frac{\left|\sum_{i=1}^{N} n_i a_i\right|}{\sum_{i=1}^{N} n_i} \tag{10}$$

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 5), A_{c_k} is the aggregated attenuation of team c_k, N_{c_k} is the total number of active devices in team c_k, K is the total number of teams (N/C), $n_k \in \{0, 1\}$ 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. The objective for the hierarchical control agents can be similarly derived for the TEDCP (it is a straight-forward extensiton that directly optimizes the system objective of the TEDCP - Equation 7), however it is excluded here for brevity.

In the standard 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, expected difference rewards for control agents can be derived by combining Equations 4 and 10, as follows:

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

where $EDR_{j,H}$ is the expected difference reward 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 is an individual control agent, K is the total number of control agents in the system, A_{c_k} is the aggregated attenuation of team c_k, and n_k is the total number of sensors participating in sensing for team k.[5]

5 Experiments and Results

We conducted the following set of experiments:

1. The DCP with no teams (Section 3.1).
2. The DCP with uncoordinated teams (Section 3.1).
3. The DCP with hierarchically coordinated teams. (Section 3.1).
4. The DCP with failures using hierarchically coordinated teams (Section 3.1).
5. The Time-Extended DCP with no teams (Section 3.2).
6. The Time-Extended DCP with hierarchical teams. (Section 3.2).

There were four different types of agents used. The first type of agents are controlled by a single centrallized algorithm, which simply selects turns on the single-best sensing device for each time step (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 (Section 3). In the TEDCP, the best sensor becomes the average of the best M sensors, where each of the best M sensors participates in exactly one time slot of sensing. Second, we consider the case where the behavior of the agents is completely random (R). The next two types of agents are learning agents attempting to optimize global (G) or expected difference reward (EDR) structures. These rewards were derived separately for agents in the no teams, uncoordinated 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 zero mean and unit variance. All experiments had 10,000 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 2-8, 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 nonhierarchical team-free experiments, all agents had an exploration rate of $\epsilon = \frac{1}{N}$, where N was the number of sensing agents in the system. In hierarchical and team based experiments, agents in the bottom layer had an exploration rate of $\frac{1}{C}$, where C is the number of agents per team, and the top layer had an exploration rate of $\frac{C}{N}$ (exploration was inversely proportional to the number of agents coordinating together in a particular group). All value tables and Q-tables were initialized to zero. 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

[5]Expected difference rewards can be derived similarly for agents in the Time-Extended Defect Combination Problem, but have been excluded here for brevity.

Algorithm 3 – 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
Instantiate Hierarchical Organization (Algorithm 2)
for $Run = 1 \rightarrow Run_{Max}$ **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:

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

    Calculate Sensing Agent Rewards: G_{c_k} or EDR_{c_k}
    Value Update for Sensing Agents (Equation 8) // 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 or EDR_H
    Value Update for Control Agents (Equation 8)// Only control agents are learning
  end for
end for

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.[6]

5.1 No Teams in the DCP

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

[6] 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).

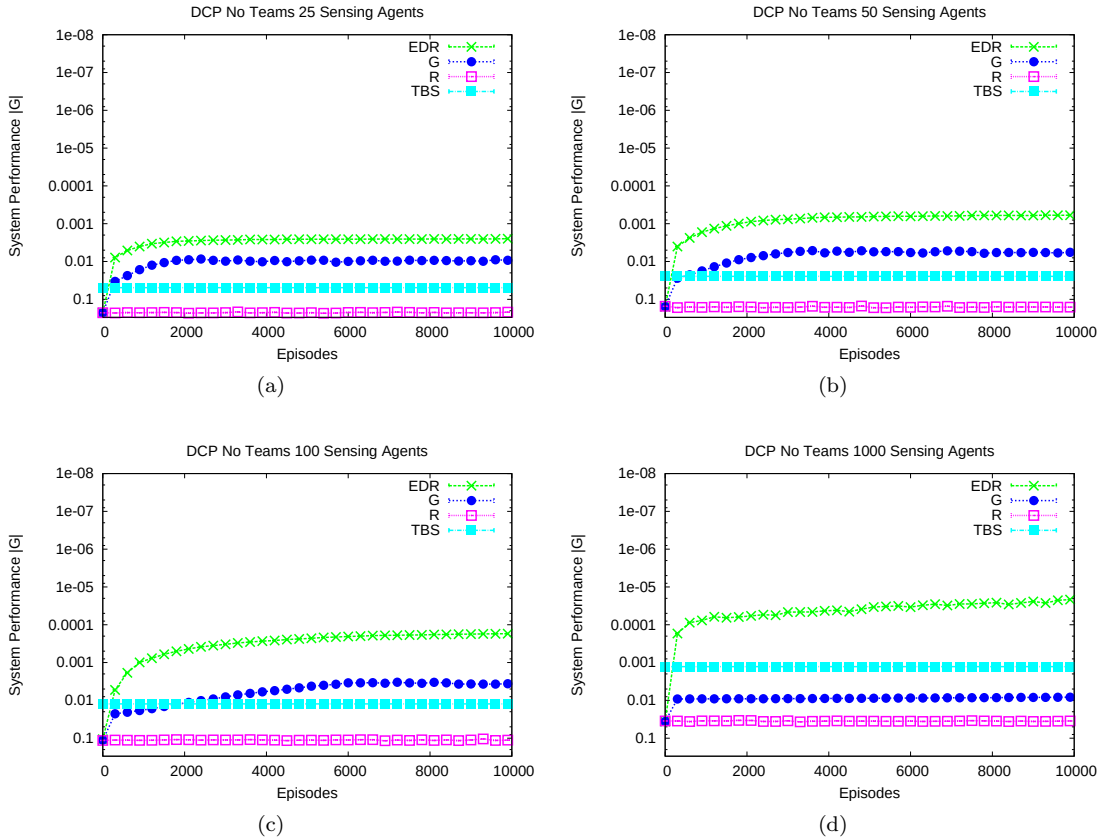

Figure 2: 25, 50, 100, and 1000 agents in the DCP with no teams (Section 4.1.1). As seen in these experiments, with up to 1000 sensing agents present in the system, agents using EDR rewards significantly outperform other methods, including a centralized search algorithm which turns on the most accurate sensing device in the system (TBS).

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.

Expected Difference Rewards, 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. Expected Difference Rewards, EDR, 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 G in this case (Figures 3 and 4). As seen, in this setting EDR significantly outperforms all other methods (Figures 3 and 4).

It is clear from this experiment that the way an agent handles the information it receives drastically impacts the performance. Agents using G and EDR received the exact same information, yet agents using expected difference rewards were able to routinely outperform agents using a traditional global rewards by approximately two or three orders of magnitude. Here, EDR rewards 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 2.4). EDR rewards are designed to provide an agent with a view of how it impacted the system over multiple

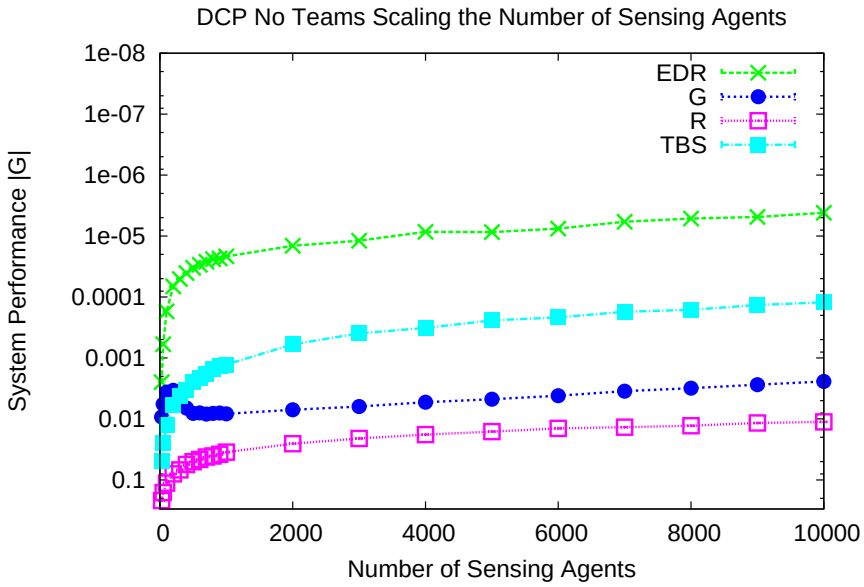

Figure 3: Scaling the number of sensing agents in the DCP with no teams. As seen, with up to 1000 agents presen in the system, agents using EDR rewards outperform all other methods by up to nearly 3 orders of magnitude. When the system is scaled further, agents using EDR rewards continue to outperform all other methods with up to $10,000$ sensing agents.

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. It is interesting to note that although agents using EDR achieve good performance, they typically use nearly 80% of the sensors in the system, which is far from the theoretical optimal number of sensors, which was determined to be 50% in (Challet and Johnson, 2002).

These results tell us that shaped rewards alone may not be enough to optimize 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 these difference rewards filter this information and improve performance, it is clear from these results that these rewards alone are not enough to handle the coordination complexities present in such large multiagent systems (Figure 4). Now that we have demonstrated the shortcoming of using only reward shaping with difference rewards to scale to large multiagent systems of up to 10,000 devices, we will implement two variations of system organization and discuss the benefits and drawbacks of each. In particular, we will implement a non-hierarchical team-based approach as well as a team-based approach involving 2-layer hierarchical organization.

5.2 Uncoordinated Teams in the DCP

In this set of experiments, we incorporated uncoordinated teams into a 10,000 sensing agent version of the DCP (Section 4.1.2). Here, we conducted a set of four experiments, where we randomly partitioned the 10,000 sensing agents into teams of $C = 25, 50, 100$, and 200 sensing agents, respectively (Figure 5). In this setting, each sensing agent can only be a member of a single team, and the goal of each team, G_{c_k}, is to optimize the aggregated attenuation of its own C sensing agents (Section 4.1.2, Equation 9). In this setting, each team acts independently to optimize its own attenuation and there is no coordination between the teams (there are no control agents present to coordinate the actions of the teams together in order to optimize the overall system performance G). By creating teams of agents, we effectively reduce the information

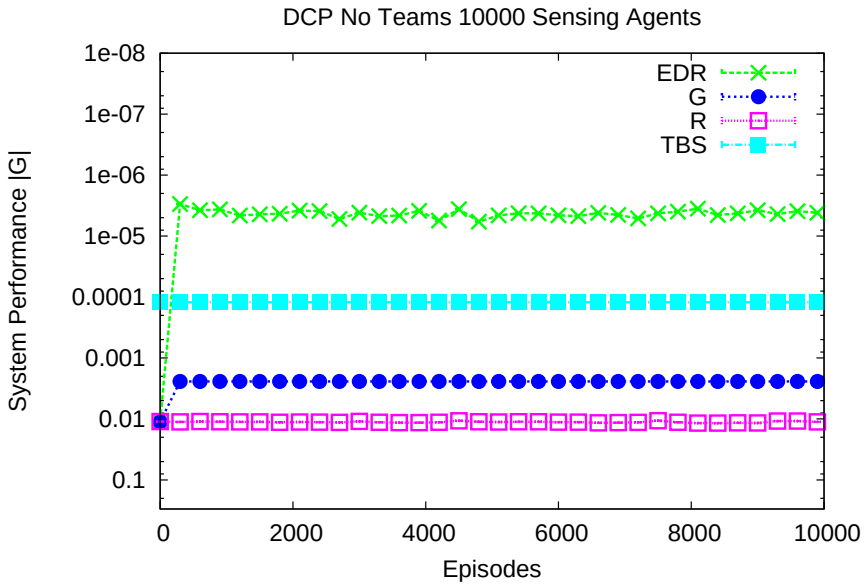

Figure 4: 10,000 sensors Defect Combination Problem (no hierarchies). Agents must choose whether to be "on" or "off". As seen, agents using EDR obtain significantly better aggregated attenuation than the best single sensor TBS (Section 5). Agents using EDR rewards perform well because these rewards promote agent-to-agent coordination and decision making.

sharing, processing requirements, and coordination complexity for agents within the system. Agents now only need to coordinate with the other sensing devices in their team. Unfortunately, although team formation promotes scalability, failure to coordinate the actions of individual teams can impede system performance. Here, small teams of agents attempt to optimize their own local reward, G_{c_k}, but are not aware of the actual system reward, G, that needs to be optimized. This is because individual teams are acting greedily with respect to their team objective, G_{c_k}, without any feedback on how their actions are impacting the system performance G.

As seen in Figure 5, team based agents using G_{c_k} continue to perform worse than team based agents using expected difference rewards EDR_{c_k}, especially as the size of individual teams is increased (this is because the larger the teams are, the more noisy the learning signal is for agents using global rewards). Even the amount of information each agent receives is reduced by approximately 99%, agents using G_{c_k} still have difficulty learning from their reward signals. Additionally, the overall system performance suffers because although the information overhead is reduced by approximately one hundred fold and it is easier for agents to deduce their individual impact on their rewards, the teams are not working together in an organized way, and are frequently interfering with each other. Agents using expected difference rewards EDR_{c_k} outperform G_{c_k} because these rewards attempt to leverage the information agents receive and use it to make optimal action selections with respect to their teams' performance. EDR_{c_k} do not perform as well as EDR did in the team-free non-hierarchical setting of Experiment 1 (Figure 4) because the teams are not coordinated; each team is individually trying to optimize its own 100 agent system objective G_{c_k} without accounting for how its actions impact the overall system performance G.

Agents using EDR_{c_k} do not perform as well as agents using expected difference rewards in the team-free setting (Figure 4), because the actions of the teams are not coordinated; each team is individually trying to optimize its own C agent team objective, G_{c_k}, without accounting for how its actions impact the overall system performance G. In this setting, the agents are optimizing their own C sensor team (effectively a C sensing agent instantiation of the DCP), but

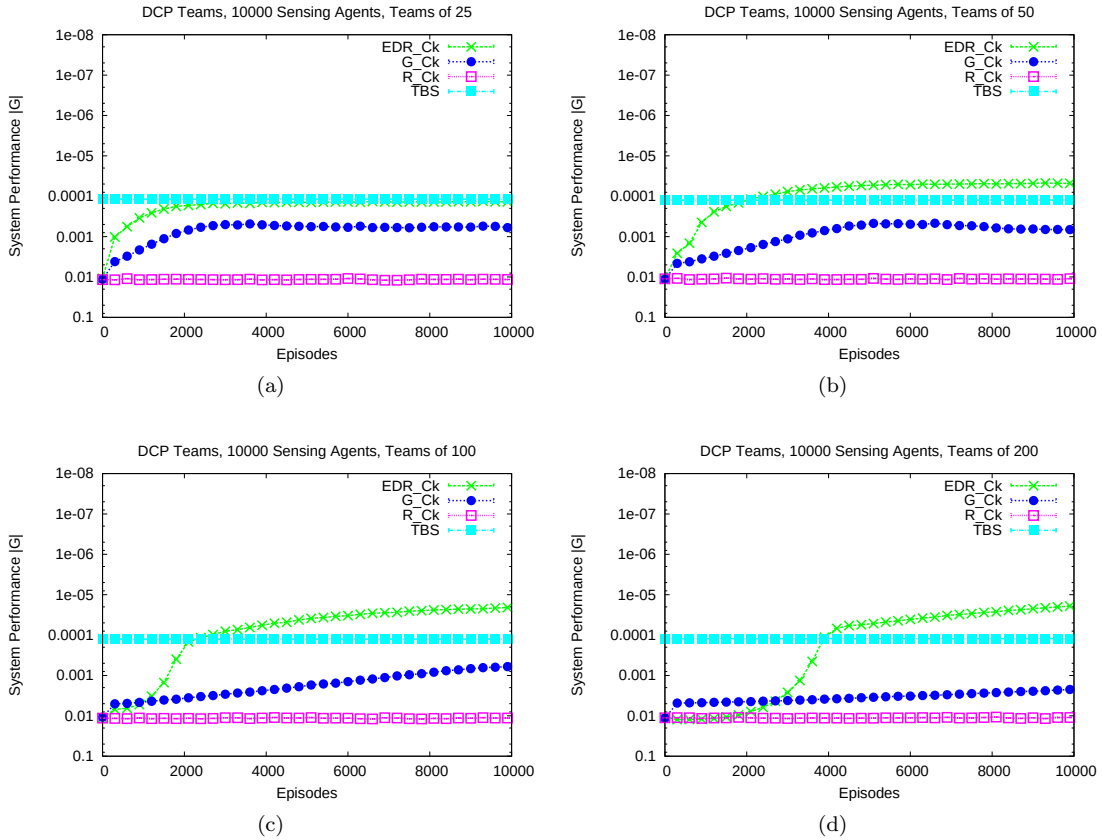

Figure 5: 10,000 sensors DCP with teams of $C = 25, 50, 100$, and 200 sensing agents (Section 4.1.2). Establishing teams reduces the amount of information each agent must receive and process by approximately 99% for teams of $C = 25$ to 200, however agents perform even worse than they did when no teams were present. This is because agents are directly attempting to optimize their own team's objective, without regard for how their actions impact the system-level performance (Section 4.1.2). Even still, teams of agents using EDR_{c_k} are able to outperform those using a standard global reward G_{c_k}, and generally perform as well or better than the best single sensing device (TBS).

the attenuations of the teams are then summed together in a suboptimal manner, resulting in suboptimal performance. The performance could vastly be improved if the teams were allowed to coordinate their actions to mutually benefit system performance. In the next experiment we address this by superimposing hierarchical control agents onto each team.

5.3 Hierarchically Coordinated Teams in the DCP

Next, we implement a 2-layer hierarchy into the DCP with 10,000 sensing agents (Section 4.1.3). 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 4.1.3). Adding a 2-layer team-based hierarchical structure to this system reduces the communication overhead for each agent by approximately 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

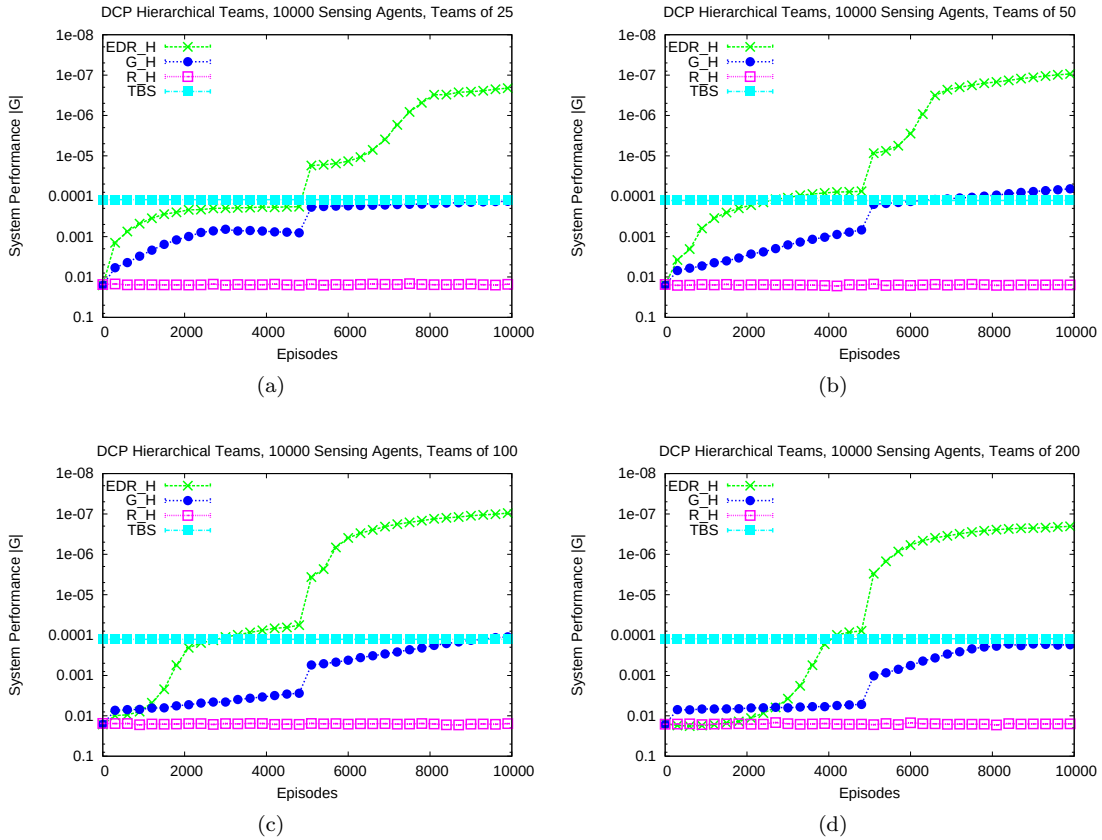

Figure 6: 10,000 sensors DCP with hierarchical organization and teams of $C = 25, 50, 100$, and 200 (Section 4.1.3). Agents are randomly partitioned into separate teams and control agents are placed over the top of each team to coordinate how teams participate in the system. The transition between team-based agents learning between episodes 0 to 5000 and the control agents learning from after episode 5000 to episode 10,000 is the reason for the discontinuity in learning performance around 5000 episodes in each graph involving hierarchical organization (Algorithm 3). 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 (EDR_H) outperform all other approaches (Figures 2-6). The hierarchy dictates the control flow and reduces the information overheads, while expected difference rewards improve agent decision making by making efficient use of locally available information.

in a cleaner signal and better action selections. Additionally, the presence of a control layer on top of the teams solves the team-based coordination problem from Section 5.2.

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 the two previous nonhierarchical experiments. 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). Secondly, the information sharing and processing requirements are reduced by approximately 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 10).

Here, the top and bottom level teams were trained separately (Algorithm 3). 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 6-8.

As seen in Figure 6, hierarchical organization benefits agents using global and expected difference reward structures (G_H and EDR_H). In fact, these results show that coupling hierarchical organization and expected difference rewards can outperform either approach individually by orders of magnitude. Agents using the global and expected difference reward structures with hierarchical organization all significantly improve their performance over both the team-free and the uncoordinated teams settings (Figures 4-5). 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 sytem 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 to global rewards with no teams and uncoordinated teams, respectively. This is because, in addition to reducing the information overhead, the hierarchical structure allows teams to coordinate their actions together to 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 2-6). This is why agents using EDR rewards in a nonhierarchical 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 nonhierarchical approaches by orders of magnitude (Figure 6), 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.

5.4 Hierarchically Coordinated Teams with Failures in the DCP

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. Since failures in the top

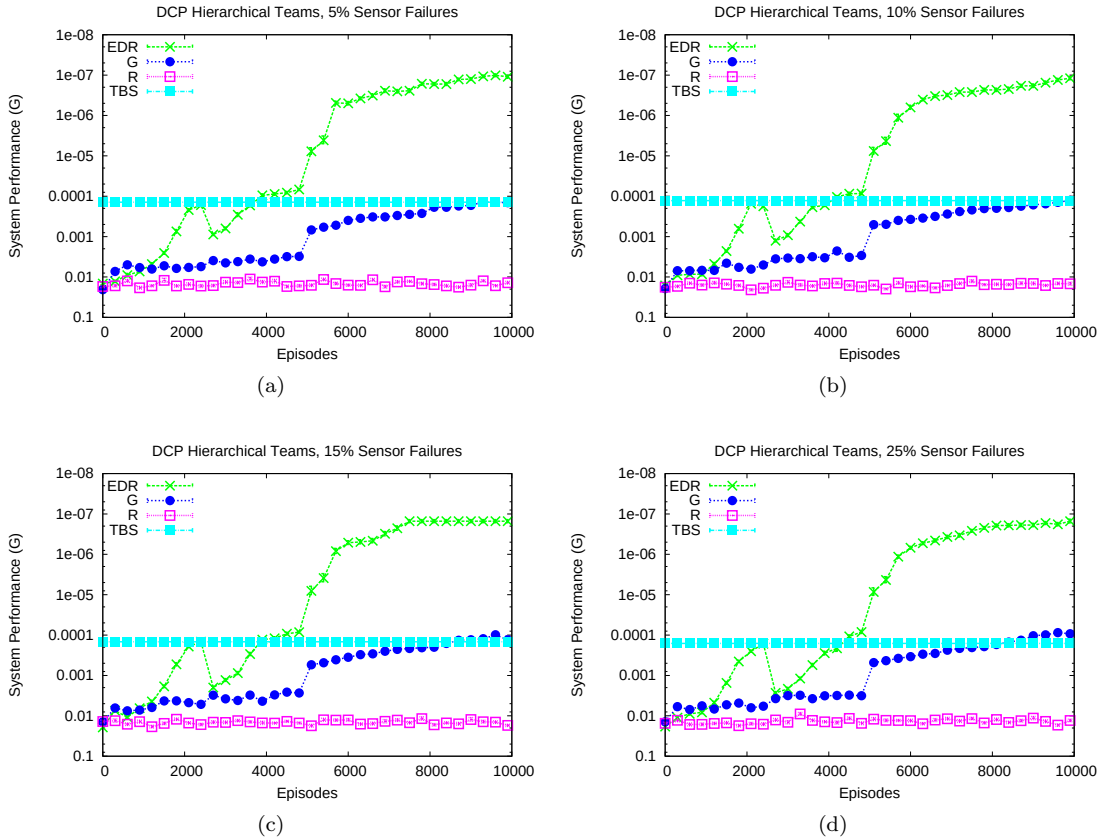

Figure 7: 10,000 sensors DCP with hierarchical organization and teams of $C = 25, 50, 100$, and 200 (Section 4.1.3). 10,000 sensors solving the DCP with hierarchical teams (Section 4.1.3). 10% of the bottom layer agents fail after 2500 time steps. The discontinuity at 5000 time steps is due to hierarchical learning (Section 5.3). 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 when learning with EDR rewards.

and bottom layers of the hierarchy may impact the system differently, we perform a separate experiment for each case. In the first set of experiments $10\% - 25\%$ of the bottom level sensors fail after 2500 time steps (Figure 7), while the other sensors continue learning. In the next set of experiments $10\% - 25\%$ of the top level control agents fail at time step 7500, while the others continue learning (Figure 8). 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 3).

As seen in Figure 7, 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 8), a portion of the control agents failed, each one impacting an entire team of sensing agents. However, since 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.

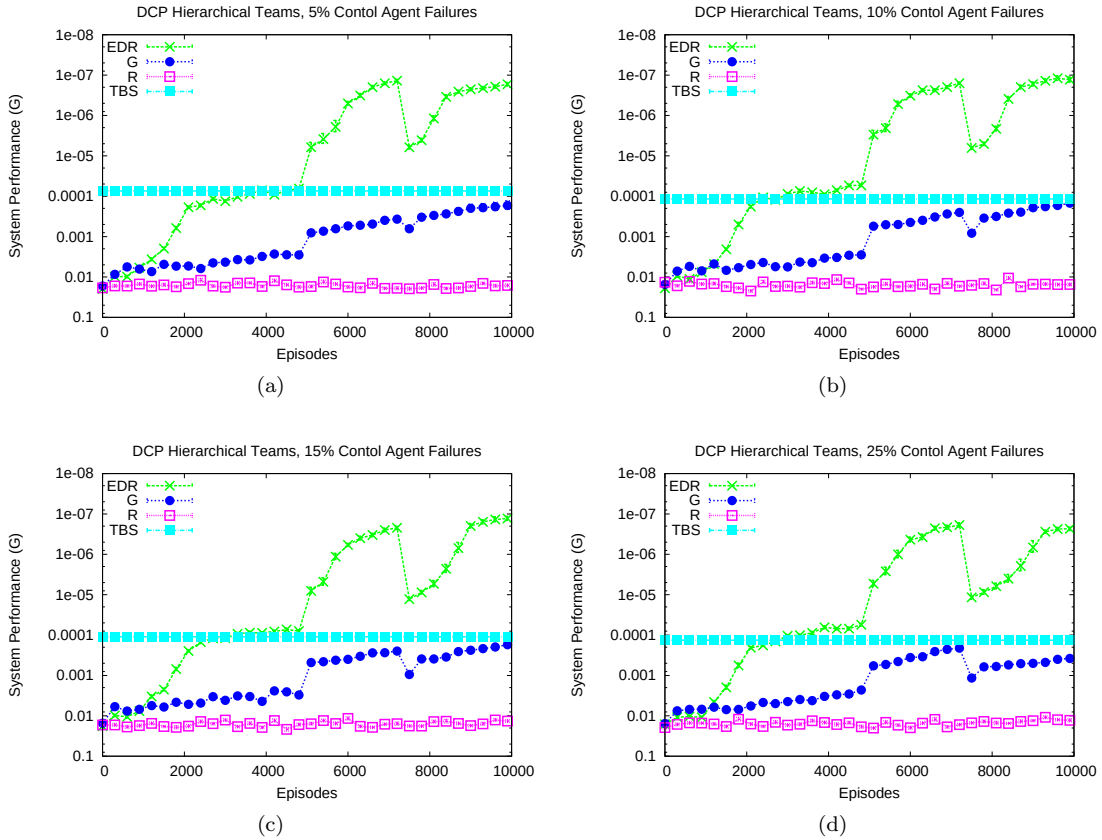

Figure 8: 10,000 sensors DCP with hierarchical organization and teams of $C = 25, 50, 100$, and 200 (Section 4.1.3). 10,000 sensors DCP with hierarchical teams (Section 4.1.3). 10% of top layer control agents fail after 7500 time steps. The discontinuity at 5000 time steps is due to hierarchical learning (Section 5.3) and the discontinuity at 7500 episodes is due to the occurence of failures. As seen, a combination of shaped rewards and hierarchies proves robust to top layer failures. Agents using EDR_H far outperform agents using a standard global reward G_H.

5.5 No Teams in The Time Extended Defect Combination Problem (TEDCP)

Next, we consider the Time-Extended version of the Defect Combination Problem. Here, the agents must collectively choose both "if" and "when" to participate in sensing. This problem presents a more complex coordination problem than the original DCP as it maintains the combinatorial optimization nature of the DCP problem but now, while at the same time providing agents with a significantly larger joint-action space. In these experiments, the TBS value is obtained by simply choosing the best sensing device available for each of the 10 time-slots. As seen here, agents using global rewards G have significant difficulty with this time-extended problem, due to the added coordination complexity. Agents using EDR outperform all other techniques. Agents using EDR continue to perform well as they are able to coordinate their actions and make decisions based upon the rewards they can expect to receive for different actions based upon previous experience. It is interesting to note however, that in the 10,000 sensor case agents perform approximately as well as 1000 agents in the standard DCP, which is what we would expect since there are effectively 1000 agents per time slot in this 10-slot version of the TEDCP.

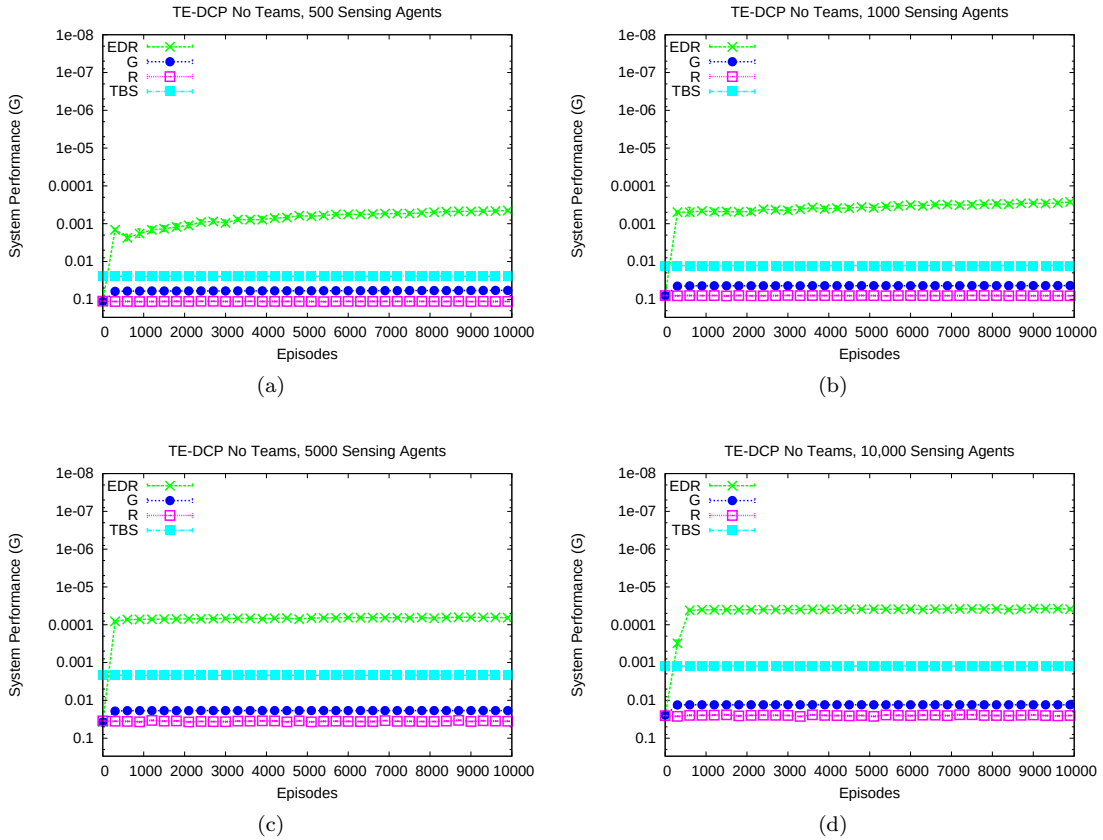

Figure 9: Agents solving a 10 slot version of the Time-Extended DCP when no teams are present. We include experiments with $N = 500, 1000, 5000$, and 10000 sensing agents, respectively. In this setting, the line TBS represents the centralized algorithm which activates the top 10 sensing devices (one per time slot in the TEDCP). As seen, EDR_H rewards drastically outperform agents using G_H, as well as the TBS algorithm. This is because these rewards promote agent coordination and good decision making.

5.6 Hierarchical Teams in The Time Extended Defect Combination Problem (TEDCP)

Continuing with the hierarchical approach, we now consider the Time-Extended DCP problem, where agents are trying to optimize G_{TE} (Section 3.2). Here, the 10,000 sensing agents are randomly grouped into teams of $C = 25, 50, 100$, and 200 devices, and a single control agent is placed on top of each team (Figure 1). The agents in the bottom layer optimize the attenuation of their team for the standard DCP (agents in the individual teams choose only whether to turn on and off). Here, the control agents in the top layer choose both "if" and "when" their team participates in sensing for the system (the control agents are responsible for choosing which time slot each team will participate in sensing).

As seen in Figure 10, EDR_H continue to outperform agents using G_H. Agents using G_H perform worse than they did in the original DCP problem because the amount of information they receive is the same and the coordination complexity of the problem has increased due to the additional time slots. Agents using G_H are still receiving too much information to be able to remove the noise from other agents off of their reward signal and the problem complexity is increased, resulting in poor performance. EDR_H rewards perform well in this problem, demonstrating that such rewards can be robust to increased problem and coordination complexities. Difference rewards allow agents to filter the information they receive to make

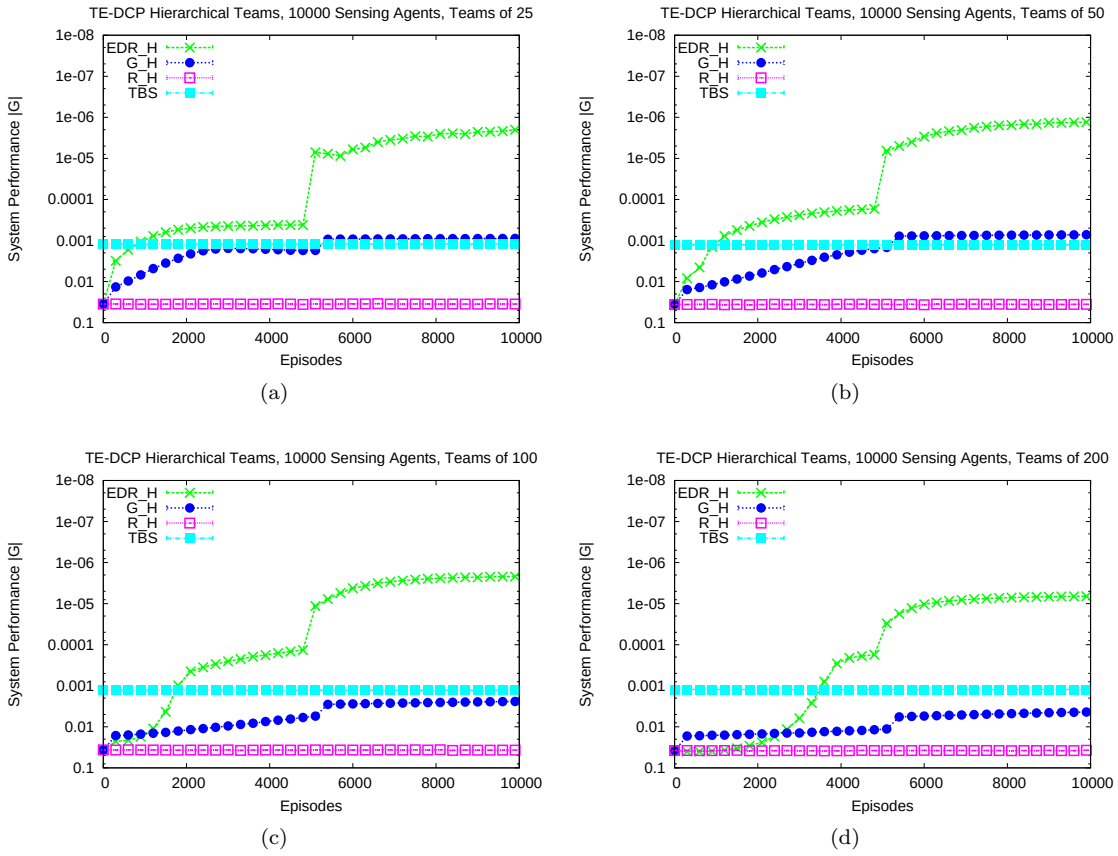

Figure 10: 10,000 sensing agents solving a 10 slot version of the Time-Extended DCP with hierarchical organization and teams of size $C = 25, 50, 100$, and 200, respecitvely (Section 4.1.3). EDR_H rewards drastically outperform agents using G_H and TBS. In addition to hierarchical organization which structures control flow and establishes coordination requirements, EDR_H rewards benefit agent decision making to improve system performance.

good decisions. Meanwhile, hierarchical organization reduces the coordination and information complexities. The combination of these two coordination methods results in significantly higher performance than either method independently.

This time-extended version of the Defect Combination Problem demonstrates that combining hierarchical organization with shaped difference rewards is not only resilient to scaling and component failures as shown previously, but that it is additionally robust to increased problem complexity. Here, the agents are responsible for coordinating not only "if" they should participate in sensing, but also "when" they should participate in sensing. Again, the combination of shaped difference rewards and hierarchical organization reduces the overall communication and information overheads for the agents in the system by up to 99% (agents only need to coordinate directly with a small fraction of their peers), while at the same time increasing system performance. These results also suggest that the approach is fairly robust to variations in the hierarchical organization of the system, as we show that similar performance is achieved for varying team sizes. These properties suggest that this approach would be widely applicable to a broad number of domains and applications.

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 approximately 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.

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. Our results show that a combination of shaped difference rewards and hierarchical organization can improve coordination, scalability, and performance in large multiagent systems. We demonstrated the robustness of our approach in two domains under varying conditions and in the presence of agent failures. Combining difference rewards and hierarchical organization led to approximately three orders of magnitude improvement over either method individually in the Defect Combination Problem and a Time-Extended version of the Defect Combination Problem. This work showed the potential advantages of combining coordination algorithms in ways that leverage their benefits, which can be utilized in other domains including sensor networks, aerial vehicle coordination, and network traffic management.

This work showed the benefits of combining reward shaping and hierarchical organization. However, it did not introduce a means of optimizing the organizational structure of the system. Future work includes extending our algorithm into domains where the problem structure can be directly exploited in order to establish optimal organization. Additionally, 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. We are currently pursuing both extensions of this work.

References

A. Agogino and K. Tumer. Efficient evaluation functions for multi-rover systems. In *The Genetic and Evolutionary Computation Conference (GECCO)*, pages 1–12, Seatle, WA, June 2004.

A. Agogino and K. Tumer. Quicr-learning for multi-agent coordination. *American Association for Artificial Intelligence (AAAI)*, 2006.

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

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

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

S. Devlin and D. Kudenko. 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)*, 2011.

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

M. Grzes and D. Kudenko. Online learning of shaping rewards in reinforcement learning. *Neural Networks*, 23, 2010.

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

C. HolmesParker and A. Agogino. Agent-based resource allocation in dynamic cubesat constellations. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2011.

C. HolmesParker and K. Tumer. Combining difference rewards and hierarchies for scaling to large multiagent systems. In *Proceedings of the Adaptive and Learning Agents Workshop (ALA)*, 2012.

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

B. Horling and V. Lesser. A survey of multiagent organizational paradigms. *Knowledge Engineering Review*, 2005.

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

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

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

J. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research (JMLR)*, 2006.

Lamber and Sinno. Bioinspired resource management for multi-sensor target tracking systems. In *MIT Lincoln Laboratory Project Report MD-26*, 2011.

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

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

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

A. Rogers, A. Farinelli, and N. Jennings. Self-organising sensors for wide area surveillance using the max-sum algorithm. *Lecture Notes in Computer Science. Self-Organizing Architectures*, 2010.

R. Sutton and A. Barto. *Reinforcement Learning An Introduction*. MIT Press, Cambridge, MA, 1998.

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

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

K. Tumer and D. Wolpert. The theory of collectives. *Collectives and the Design of Complex Systems*, 2004.

P. Vrancx, K. Verbeeck, and A. Nowe. Decentralized learning in markov games. *IEEE Transations on Systems, Man, and Cybernetics - Part B: Cybernetics*, 38(4), August 2008.

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

D. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. *Advances in Neural Information Processing Systems (NIPS)*, 1999.

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

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

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