Multiagent Reinforcement Learning in a Distributed Sensor Network with Indirect Feedback

Mitchell Colby
Oregon State University
442 Rogers Hall
Corvallis, OR 97331
colbym@engr.orst.edu

Kagan Tumer
Oregon State University
204 Rogers Hall
Corvallis, OR 97331
kagan.tumer@oregonstate.edu

ABSTRACT

Highly accurate sensor measurements are crucial in order for power plants to effectively operate, as well as to predict and subsequently prevent any potentially catastrophic failures. As the cost of sensors decreases while their power increases, distributed sensor networks become a more attractive option for implementation in power plants. In this work, we investigate the use of a distributed sensor network to achieve highly accurate measurements. We apply shaped rewards to local components and use a simple learning algorithm at each sensor in order to maximize those rewards. Our results show that the measurements from a sensor network trained using shaped rewards are up to two orders of magnitude more accurate than a sensor network trained with a traditional global reward. Further, the algorithm proposed scales well to large networks, and is robust to measurement noise and sensor failures.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: Distributed Systems

General Terms

Algorithms, Experimentation

Keywords

Multiagent learning, Coordination

1. INTRODUCTION

One of the most significant challenges to developing efficient energy sources is addressing how to control and optimize power plants. As power plants become more complex, a distributed control strategy will become necessary [13]. Rather than a central controller making decisions for an entire plant, subsystems within the plant must be adaptively controlled independently, while maintaining effectiveness on an overall system level [8, 11]. In order for any control process to be effective, accurate feedback about the environment is required, which is typically provided by sensors.

Sensors are becoming smaller, less expensive, more computationally powerful, and more capable of operating in harsh environments [4], allowing for the implementation of large sensor networks in industrial power plants. Given the complexity of power plants and the power of new sensors, a distributed sensor network is a natural system to implement in a power plant [5].

Increasing the number of sensors in a system provides many benefits. First, with a large network of sensors, the network can compensate for sensor failures without losing significant performance [10]. Second, with the ability of sensors to preprocess data, distributed sensor networks can give much more useful data relating to system wide performance than a standard set of sensors [14]. Third, interacting sensors can give system level information that is not available from simply aggregating sensor information [9]. Finally, a distributed sensor network is often capable of self-organization, which is extremely helpful in area surveillance [7].

The contribution of this work are as follows:

- Extend the Defect Combination Problem to power plant applications by providing indirect feedback using a system model.
- Use shaped rewards on the modified Defect Combination Problem in order to minimize measurement error in a distributed sensor network operating in a model power plant.
- Show that the average difference reward meets real-world performance requirements, including robustness to noise and sensor failures, as well as accurate tracking performance.

We develop a methodology to provide indirect feedback based on information that is easily obtained in order to train a network of sensors to have high-accuracy measurements. In systems such as power plants, increasing the accuracy of sensor readings is crucial, because small changes in the system state must be detected not only to maintain effective plant performance but to also predict and subsequently prevent potentially catostrophic failures [3]. Through the use of indirect feedback and reward shaping, we achieve sensor measurements which are two orders of magnitude more accurate than measurements using global rewards. This work is in conjunction with the National Energy Technology Laboratory (NETL), which is interested in new methodologies for controlling and sensing in complex next-generation power plants.

The rest of this paper is organized as follows. Section 2 gives background information on the Rankine cycle domain and reward structures used, as well as past approaches to

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.
Copyright © 2012, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

such sensor network problems. Section 3 defines the modified Defect Combination Problem for use in the Rankine cycle. Section 4 defines the agent learning algorithm, as well as derives the reward structures for the Rankine cycle Defect Combination Problem. Section 5 defines the experiments conducted, and presents the experimental results. Finally, Section 6 discusses the results, draws conclusions, and identifies area of future work.

2. BACKGROUND

The following sections describe the Difference Reward and the Expected Difference Reward, the Rankine cycle power plant, the Defect Combination Problem, and past related work.

2.1 Difference Reward

The Difference Reward $D_i(z)$ is defined as [12]:

$$D_i(z) = G(z) - G(z_{-i}) \tag{1}$$

where $G(z)$ is the global reward, and $G(z_{-i})$ is the global reward without the influence of agent i. Intuitively, the Difference Reward gives agent i's specific impact on the system performance. Note that:

$$\frac{\partial D_i(z)}{\partial a(i)} = \frac{\partial G(z)}{\partial a(i)} \tag{2}$$

where $a(i)$ is agent i. Thus, an agent acting to increase the Difference Reward will also act to increase the global reward. This property is termed *factoredness* [1]. Further, because the Difference Reward only depends on the actions of agent i, noise from other agents is reduced in the feedback given by D_i. This property is termed *learnability* [1].

An extension of the Difference Reward, the Expected Difference Reward $ED_i(z)$, is defined as in [12]:

$$ED_i(z) = G(z) - E_i(a)[G_z] \tag{3}$$

where $E_i(a)[G(z)]$ is the expected value of the global reward across all actions that agent i may take. In the case of a discrete action space, Equation 3 becomes:

$$ED_i(z) = G(z) - \sum_{a \in A} P_i(a) G_i(z_a) \tag{4}$$

where $P_i(a)$ is the probability that agent i takes action a, and $G_i(z_a)$ is the global reward when agent i takes action a. While the Difference Reward gives the impact of agent i on the global reward, the Expected Difference Reward gives the expected impact of agent i on the global reward. The difference reward has been shown to promote good learned policies in many domains, including rover coordination [1] and distributed sensor network control [12].

2.2 Rankine Cycle

We develop a sensor network in a well known power generation system: a vapor power Rankine cycle [6]. The Rankine cycle is one of the simplest thermodynamic power-producing cycles, and serves as a testbed to demonstrate the effectiveness of our approach. It is important to note that although the Rankine cycle is not complex enough to demonstrate effectiveness of control algorithms, it is adequate to demonstrate the effectiveness of our sensor network training algorithm. In our approach, the model is treated as a black box, and only the output of the model is used to train sensors operating within that model. Thus, the complexity of the model is irrelevant in our approach. In a Rankine cycle, the working fluid passes through a boiler and becomes saturated vapor. Next, the fluid goes through the turbine, which results in an energy output which is used to produce electricity. The fluid then passes through a condenser and becomes a saturated liquid. Finally, the fluid passes through a pump and returns to the boiler, completing the cycle. The Rankine cycle is shown in Figure 1. For the purposes of this analysis, we make the following assumptions:

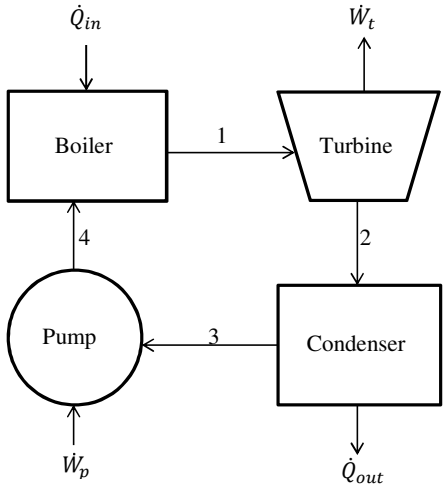

Figure 1: A vapor power Rankine cycle. The working fluid travels through a boiler, turbine, condenser, and pump in succession. The work output of the turbine is used to generate electricity.

A1. Each component of the cycle is considered to be a control volume.
A2. All processes of the working fluid are internally reversible.
A3. The turbine and pump operate adiabatically.
A4. Kinetic and potential energy effects are negligible.
A5. Saturated vapor enters the turbine. Condensate exits the condenser as a saturated liquid.
A6. The working fluid is water.

As seen in Figure 1, there are four distinct states in the Rankine cycle, each of which lies between two of the components. At each state, the working fluid has an enthalpy h, which is a thermodynamic value indicating the energy stored in the working fluid. The enthalpy of a fluid is a function of temperature and pressure. The system performance is related to the enthalpy h_s at each plant state s by the following relations:

$$\frac{\dot{W}_t}{\dot{m}} = h_1 - h_2 \tag{5}$$

$$\frac{\dot{Q}_{out}}{\dot{m}} = h_2 - h_3 \tag{6}$$

$$\frac{\dot{W}_p}{\dot{m}} = h_4 - h_3 \tag{7}$$

$$\frac{\dot{Q}_{in}}{\dot{m}} = h_1 - h_4 \tag{8}$$

where $\dot{m}$ is the mass flow rate of the working fluid, $\dot{W}_t$ is the work output of the turbine, $\dot{Q}_{out}$ is the heat output of

the condenser, $\dot{W}_p$ is the work input to the pump, and $\dot{Q}_{in}$ is the heat input to the boiler. In order to evaluate these relations, the enthalpy of the working fluid at each state must be determined by measuring the temperature and pressure at each state; this requires the development of a sensing policy. Such a policy was studied in the Defect Combination Problem.

2.3 Defect Combination Problem

The Defect Combination Problem (DCP) assumes that there exists a set of imperfect sensors X which have constant attenuations due to manufacturing defects or imperfections [2, 12]. Each sensor $x_i \in X$ has an attenuation a_i in its measurement. Thus, if sensor x_i is measuring some value A, its measurement is $A + a_i$. The DCP involves choosing a subset of the sensors such that the aggregate attenuation of the combined readings is minimized, which is equivalent to maximizing:

$$G(z) = -\frac{\left|\sum_{i=1}^{N} n_i a_i\right|}{\sum_{i=1}^{N} n_i} \tag{9}$$

where N is the number of sensors in the system, $n_i \in \{0, 1\}$ is an indicator function based on whether the sensor is "on" or "off," and G is the aggregated attenuation of the combined sensor readings.

There are two key drawbacks to the DCP in the context of real-world sensor network applications. First, the objective function (Equation 9) requires knowledge of the attenuation of each sensor. However, it is extremely unlikely to know the attenuation of each individual sensor in a real-world application. Secondly, the DCP assumes constant attenuations a_i for each sensor x_i. In reality, a sensor will have noise in its reading, such that the attenuation will be $N(\sigma, a_i)$, where $N(\sigma, \mu)$ is a normally distributed random variable with mean μ and standard deviation σ. Thus, the DCP provides an inadequate framework to train real-world sensor networks. For a real-world application, the objective function must include readily available information about the system, and the sensors should have noise in their measurements.

2.4 Related Work

Early work on the DCP defined each sensor as an agent, which then chose to be on or off [12]. The agents learned with a standard Q-learning algorithm, and agents received either the global reward, the difference reward, or the expected difference reward. Both the difference reward and expected difference reward are factored with respect to the global reward, such that agents acting to improve their private reward functions also acted to increase the global reward. Further, these reward structures have high learnability. Ultimately, the difference reward and expected difference reward yielded significantly better sensing policies than the global reward in the DCP. Although sensor attenuation was minimized in this work, the reward structure does not allow for implementation in real-world sensor networks, because it requires knowledge of the attenuation of each sensor. We aim to extend this work by implementing a model of the system into the reward structure, in order to allow for implementation in real-world applications.

3. DCP FOR POWER PLANTS

We apply a modified version of the DCP to a Rankine cycle power plant. There is a set of motes X_s at each of the four plant states, where $s \in \{1, 2, 3, 4\}$ is the state of the power plant (see Figure 1). Each mote $x_{s,i} \in X_s$ has sensors capable of measuring temperature and pressure, the two parameters needed to determine the enthalpy of the working fluid. The sensors in mote $x_{s,i}$ have a mean temperature attenuation $t_{s,i}$, and a mean pressure attenuation $p_{s,i}$. Further, each sensor has an associated measurement noise defined by the Gaussian distribution, where σ_t and σ_p are the standard deviations for temperature and pressure attenuations respectively. Thus, the temperature and pressure attenuations of the sensors on each mote are given by the following normal distributions:

$$e_{T,s,i} = N(\sigma_t, t_{s,i}) \tag{10}$$

$$e_{P,s,i} = N(\sigma_p, p_{s,i}) \tag{11}$$

Each mote is considered to be an agent. First, an agent decides whether to be "on" or "off." If an agent decides to be "on," then it must decide if it will measure temperature, pressure, or both temperature and pressure. The goal of the agents is to collectively take actions which will minimize the aggregate error in temperature and pressure readings. The aggregate attenuation for temperature at a state s is defined as:

$$g_{T,s} = \frac{\sum_{i=1}^{N_s} n_{s,i} e_{T,s,i}}{\sum_{i=1}^{N_s} n_{s,i}} \tag{12}$$

where N_s is the number of motes in state s, and $n_{s,i} \in \{0, 1\}$ denotes whether mote $x_{s,i}$ is measuring temperature or not. Similarly, the aggregate attenuation for pressure at state s is defined as:

$$g_{P,s} = \frac{\sum_{i=1}^{N_s} n_{s,i} e_{P,s,i}}{\sum_{i=1}^{N_s} n_{s,i}} \tag{13}$$

where $n_{s,i} \in \{0, 1\}$ denotes whether mote $x_{s,i}$ is measuring pressure or not. From Equations 12 and 13, the measured values of temperature and pressure at state s are:

$$T_{s,sensed} = T_s + g_{T,s} \tag{14}$$

$$P_{s,sensed} = P_s + g_{P,s} \tag{15}$$

where T_s and P_s are the true temperature and pressure at state s, respectively. Equations 12 and 13 can not be used to provide feedback to learning agents, because they can not be calculated directly in real-world applications, because the attenuation of each sensor is not known. However, using the system model and knowledge of the control inputs, the enthalpy at each state may be analytically determined. Thus, the enthalpy found from the sensor readings may be compared with the true enthalpy (found with system model) to determine the accuracy of the sensor network.

The enthalpy of the working fluid is a thermodynamic property which quantifies the level of energy in the fluid. Enthalpy change in a fluid corresponds to the fluid either absorbing or expelling energy, and is used to determine power levels in a power cycle. In the Rankine cycle power plant, the control inputs are $\dot{Q}_{out}$, $\dot{W}_p$, $\dot{Q}_{in}$, and $\dot{m}$, and are known values. Thus, using Equations 5 through 8 in addition to the assumptions made about the Rankine cycle, the enthalpy values h_1 through h_4 may be directly determined. The enthalpy at each state is also estimated by the sensor network,

where the estimation of enthalpy is defined by:

$$h_{s,sensed} = f(T_{s,sensed}, P_{s,sensed}) \quad (16)$$

where $f(T, S)$ is the enthalpy equation based on thermodynamic empirical data, and $T_{s,sensed}$ and $P_{s,sensed}$ are the sensed temperature and pressure values, as defined in Equations 14 and 15. The error in the enthalpy reading at a given state is thus:

$$h_{s,error} = |h_s - h_{s,sensed}| \quad (17)$$

where h_s is the true enthalpy of state s, found using the Rankine cycle model. The error in enthalpy gives an indication of the effectiveness of the sensors measuring temperature and pressure. The objective of the entire sensor network is to minimize the total attenuation of enthalpy measurements at each state, which is equivalent to maximizing:

$$G(z) = -\sum_{s=1}^{4} h_{s,error} \quad (18)$$

The key difference between this approach and the DCP is the fact that the objective function given by Equation 18 uses data which is readily available in order to judge sensor efficacy. Recall that the DCP objective function (Equation 9) includes individual sensor attenuations, which are extremely impractical to obtain, especially as the size of the sensor network grows. Thus, the modification we have made to the DCP allows for implementation in real-world applications.

4. AGENT LEARNING

Agent learning in the Rankine cycle DCP is achieved through standard multiagent Q-learning, shown in Algorithm 1. Each agent maintains a private Q-table, and updates the Q-table at each time step based on the reward that the agent receives.

Algorithm 1 Reinforcement Learning Algorithm

Each agent i generates a randomly seeded Q-table Q_i;
$episode = 1$;
while $episode < maxEpisodes$ **do**
1. Each agent i selects an action from Q-table using ϵ-greedy;
2. Calculate measured temperature and pressure at each state (Eqns. 14 and 15);
3. Calculate measured enthalpy at each state (Eqn. 16);
4. Calculate true enthalpy at each state (Eqns. 5 through 8);
5. Calculate system objective (Eqn. 18);
6. Calculate rewards R_i for each agent (D, ED, AD, or G);
7. Q-update: $Q_i(a) \leftarrow Q(a)(1-\alpha) + \alpha R_i$;
8. $episode = episode + 1$;
end while

The following sections derive each reward (D, ED, AD, and G) used for learning.

4.1 Global Reward

The simplest approach to solving the learning problem is to simply use the Global Reward to provide feedback to each agent. Although this method is simple, it is an ineffective approach. As the number of sensors in the network grows, the Global Reward provides poor agent-specific impact. For example, in a sensor network with 1000 motes, the Global Reward provides poor feedback on the action selection of one particular agent, because of the small impact on G that one agent has. For this reason, it is not expected that using the Global Reward for feedback will provide satisfactory results, but will provide a good baseline for comparison of our shaped rewards D and ED.

4.2 Difference Reward

We will now derive the Difference Reward for the Rankine cycle DCP. Recall from Equation 1, the Difference Reward is defined as:

$$D_i(z) = G(z) - G(z_{-i}) \quad (19)$$

There are four actions $a \in A$ that each agent may take in the Rankine cycle DCP, where $A = \{a_1, a_2, a_3, a_4\}$, and:

- a_1: sense nothing
- a_2: sense pressure
- a_3: sense temperature
- a_4: sense both temperature and pressure

Removing an agent from the system is equivalent to having it sense nothing, or to take action a_1. The aggregate attenuation for temperature at a state s without the impact of agent i is:

$$g_{T,s,-i} = \frac{\sum_{j=1,j\neq i}^{N_s} n_{s,j} e_{T,s,i}}{\sum_{j=1,j\neq i}^{N_s} n_{s,j}} \quad (20)$$

The aggregate attenuation for pressure at a state s without the impact of agent i is:

$$g_{P,s,-i} = \frac{\sum_{j=1,j\neq i}^{N_s} n_{s,j} e_{P,s,j}}{\sum_{j=1,j\neq i}^{N_s} n_{s,j}} \quad (21)$$

Thus, the temperature and pressure measurements at any state s without the impact of agent i are given by:

$$T_{s,sensed,-i} = T_s + g_{T,s,-i} \quad (22)$$

$$P_{s,sensed,-i} = P_s + g_{P,s,-i} \quad (23)$$

The estimated enthalpy at state s without the effects of agent i is given by:

$$h_{s,sensed,-i} = f(T_{s,sensed,-i}, P_{s,sensed,-i}) \quad (24)$$

The error in the enthalpy measurement without the effects of agent i is:

$$h_{s,error,-i} = |h_s - h_{s,sensed,-i}| \quad (25)$$

The total system reward without the effects of agent i is:

$$G(z_{-i}) = -\sum_{i=1}^{4} h_{s,error,-i} \quad (26)$$

The Difference Reward for the Rankine cycle DCP is thus:

$$D_i(z) = G(z) - G(z_{-i}) \quad (27)$$

where $G(z)$ is defined in Equation 18 and $G(z_{-i})$ is defined in Equation 26.

4.3 Expected Difference Reward and Average Difference Reward

We will now derive the Expected Difference Reward for the Rankine cycle DCP. Recall that the Expected Difference Reward is defined by:

$$ED_i(z) = G(z) - \sum_{a \in A} P_i(a) G_i(z_a) \tag{28}$$

In order to calculate the Expected Difference Reward for some agent i, we must determine the global reward when agent i selects any action $a \in A$. We begin by defining temperature and pressure indicator functions based on the action selected:

$$n_t(a) = \begin{cases} 0, & a = a_1 \\ 0, & a = a_2 \\ 1, & a = a_3 \\ 1, & a = a_4 \end{cases} \tag{29}$$

$$n_p(a) = \begin{cases} 0, & a = a_1 \\ 1, & a = a_2 \\ 0, & a = a_3 \\ 1, & a = a_4 \end{cases} \tag{30}$$

The aggregate attenuation for temperature at state s when agent i takes action a is:

$$g_{T,s,i}(a) = \frac{\left[\sum_{j=1,j\neq i}^{N_s} n_{s,j} e_{T,s,j}\right] + n_t(a) e_{T,s,i}}{n_t(a) + \sum_{j=1,j\neq i}^{N_s} n_{s,j}} \tag{31}$$

The aggregate attenuation for pressure at state s when agent i takes action a is:

$$g_{P,s,i}(a) = \frac{\left[\sum_{j=1,j\neq i}^{N_s} n_{s,j} e_{P,s,j}\right] + n_p(a) e_{P,s,i}}{n_p(a) + \sum_{j=1,j\neq i}^{N_s} n_{s,j}} \tag{32}$$

The temperature and pressure measurements at state s when agent i takes action a are thus:

$$T_{s,sensed,i}(a) = T_s + g_{T,s,i}(a) \tag{33}$$

$$P_{s,sensed,i}(a) = P_s + g_{P,s,i}(a) \tag{34}$$

The estimated enthalpy at state s when agent i takes action a is:

$$h_{s,sensed,i}(a) = f(T_{s,sensed,i}(a), P_{s,sensed,i}(a)) \tag{35}$$

The error in the enthalpy measurement at state s when agent i takes action a is:

$$h_{s,error,i}(a) = |h_s - h_{s,sensed,i}(a)| \tag{36}$$

The total system reward when agent i takes action a is thus:

$$G_i(z_a) = -\sum_{s=1}^{4} h_{s,error,i}(a) \tag{37}$$

The probabilities in the Expected Difference Reward are based on previous actions taken by each agent. Initially, each probability $P(a)$ is set to $1/|A|$, where $|A|$ is the cardinality of the set A. During learning, each agent keeps a record of the number of times it has taken every action. These records allow for direct calculation of the probabilities. Suppose that agent i has taken action a_k a total of c_k times, then the probability of agent i taking a particular action a_j is given by:

$$P_i(a_j) = \frac{c_j}{\sum_{k=1}^{4} c_k} \tag{38}$$

With the probabilities defined, we can now calculate the Expected Difference Reward as:

$$ED_i(z) = G(z) - \sum_{j=1}^{4} P_i(a_j) G_i(z_{a_j}) \tag{39}$$

We can define a similar reward, the Average Difference Reward, which is equivalent to the expected difference reward when the probabilities of taking each action are assumed to be constant:

$$AD_i(z) = G(z) - \frac{1}{4} \sum_{j=1}^{4} G_i(z_{a_j}) \tag{40}$$

Table 1: Experiment Parameters

Parameter	Value
α	0.3
ϵ	0.0002
Episodes	1000
Statistical Runs	2000

Table 2: Controller Parameters

Parameter	Value
$\dot{Q}_{out}$	169.75 MW
$\dot{W}_p$	844.1 kW
$\dot{Q}_{in}$	269.77 MW
$\dot{m}$	$3.77 \cdot 10^5$ kg/h

5. EXPERIMENTS AND RESULTS

The following sections describe the experiments conducted with the Rankine cycle DCP, as well as the results for each experiment. For each experiment, the number of agents in each state was varied from 50 to 1000. The experimental parameters were all set as in Table 1, unless otherwise noted. For each plot, the error in the mean $\sigma/\sqrt{N}$ is reported, where $N = 2000$ is the number of statistical runs.

5.1 Enthalpy Measurement

The first experiment involves measuring the enthalpy at each state in the Rankine cycle during steady-state operation without the presence of sensor noise. Constant control inputs were maintained, consistent with typical steady-state operation [6], given in Table 2. In each power plant state, the agents were trained in order to minimize enthalpy attenuation at that state. The rewards tested are the global reward, the difference reward, the expected difference reward, and the average difference reward. The policies learned using these reward structures are compared with a random policy, where each agent randomly selects an action at each episode with uniform probability. The enthalpy measurement experiment give a baseline result to compare against when analyzing sensor failures and measurement noise. The learning plot for 1000 agents in the Rankine cycle DCP is shown in Figure 2, and the scaling results when varying between 50 and 1000 agents are shown in Figure 3. As seen

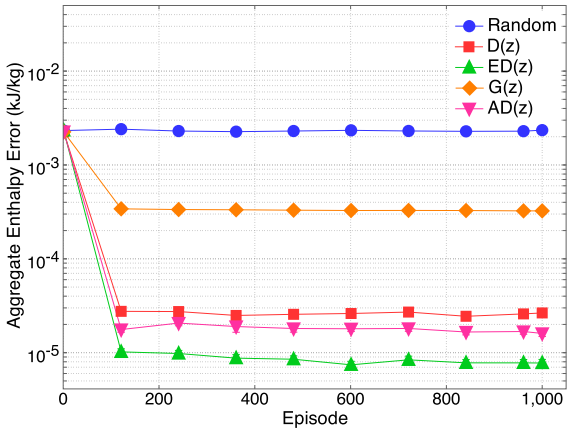

Figure 2: Rankine Cycle DCP with 1000 agents and no noise or failures.

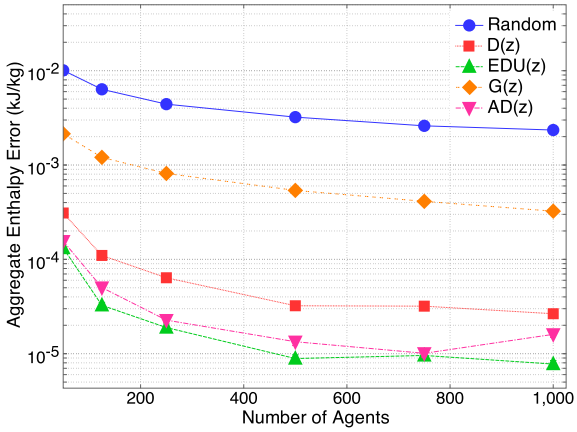

Figure 3: Rankine Cycle DCP with no noise or failures - scaling.

in Figure 2, the random reward policy performed the worst, as expected. $G(z)$ yielded slightly better results than random, but still yielded poor performance due to learning noise associated with $G(z)$. $AD(z)$ and $D(z)$ yielded similar performance, with $AD(z)$ performing slightly better. Finally, $ED(z)$ yielded the best results.

As seen in Figure 3, $ED(z)$ and $AD(z)$ yielded the best results, regardless of the number of agents in the system. The reason these reward structures perform better than $D(z)$ can be attributed to the fact that they provide non-zero feedback when an agent chooses to sense nothing. In contrast, $D(z)$ gives a feedback of zero when an agent chooses to sense nothing, whether that action is benificial or detrimental to the system. $ED(z)$ and $AD(z)$ both give meaningful feedback regardless of the action selected by the agent, which yields better learned performance.

5.2 Enthalpy Measurement with Sensor Failures

The second experiment involves determining the effects of sensor failures on the system. For this experiment, the network is allowed to train for 1000 episodes, as in the enthalpy measurement experiment (Section 5.1). After 1000 episodes, a percentage of the sensors fail, and the network retrains for another 1000 episodes to compensate for the sensor failures. The level of sensor failures is set at 15%. This experiment gives an indication of how robust the sensor network is to agent failures, which is a crucial property for a sensor network operating in a real-world domain such as a power plant. The results for 500 agents with 15% sensor failure are shown in Figure 4, and the scaling results when varying the number of agents between 50 and 1000 are shown in Figure 5. As seen in Figure 4, after 15% of the sensors fail, $D(z)$ and

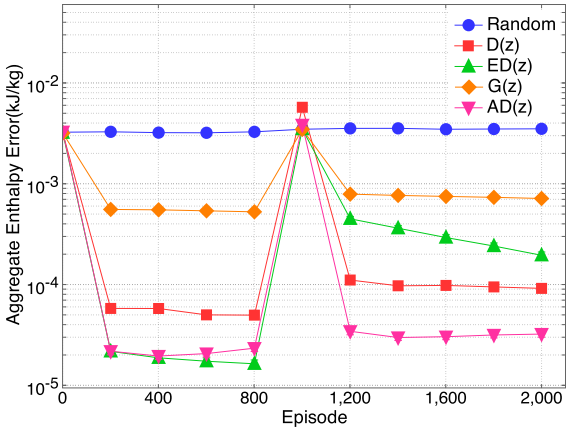

Figure 4: Rankine Cycle DCP with 500 agents and 15% failures.

$AD(z)$ are able to recover over 95% of lost performance. However, $ED(z)$ is unable to recover, and actually performs worse than $D(z)$. This is due to the fact that $ED(z)$ is calculated by tracking the probabilities of each agent taking an action. After the agents fail, these probabilities are no longer accurate, because they depended on the system prior to agent failure. Once the agents fail, these incorrect probabilities actually corrupt the learning signal, resulting in poor performance. As seen in Figure 5, $ED(z)$ does not recover

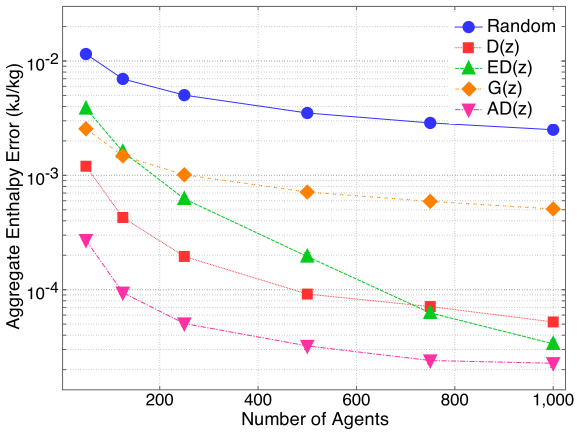

Figure 5: Rankine Cycle DCP with 15% failure - scaling.

from failures and perform better than $D(z)$ until there are at least 750 agents in the system. Our results show that as the system size increases, uncertainty about action selection for

each agent also increases, resulting in a more uniform probability distribution when calculating $ED(z)$. Thus, as the system size increases, agent failures impact $ED(z)$ less and less. When agent failures are present, $AD(z)$ consistently performs better than all other reward structures tested.

5.3 Enthalpy Measurement with Sensor Noise

The third experiment involves determining the effects of measurement noise on the sensor network. For this experiment, sensor noise is set at 15%. Noise is defined by altering the value of the standard deviation of the normal distribution from which sensor readings are drawn. For a particular noise value ϕ, the standard deviation is defined as:

$$\sigma(\phi) = \frac{\phi}{3} \tag{41}$$

Thus, for a particular noise value chosen, there is a probability of 0.997 that the noise will be less than or equal to the chosen value. This experiment gives an indication of how robust the sensor network is to measurement noise, which is always present in real-world sensor network applications, and thus must be accounted for by sensor network control. The results for 1000 agents with 15% sensor noise are shown in Figure 6. As seen in Figure 6, when noise is present in the

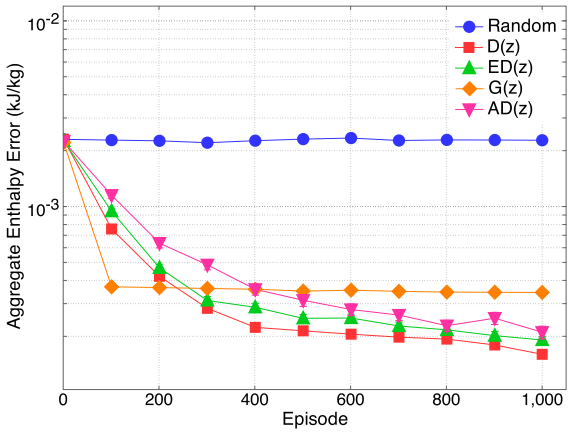

Figure 6: Rankine Cycle DCP with 1000 agents and 15% noise.

system, $D(z)$, $ED(z)$, and $AD(z)$ all provide almost identical performance. As measurement noise is introduced to the system, the probabilities while calculating $ED(z)$ become less certain (i.e. the probability distribution becomes more uniform), and $ED(z)$ becomes almost identical to $AD(z)$. $D(z)$ is able to effectively filter much of the sensor noise, because it only depends on the action of a single agent.

5.4 Enthalpy Measurement with Sensor Failures and Sensor Noise

The fourth experiment involves determining the effects of both sensor failures and measurement noise on the sensor network. This experiment is identical to the sensor failure experiment (Section 5.2), with the addition of measurement noise as in the noise experiment (Section 5.3). This experiment gives insight to how well the sensor network would perform in real-world applications, where both sensor noise and sensor failures must be adequately addressed in order to maintain acceptable network performance. The results for 1000 sensors with 15% measurement noise and 15% agent failure are shown in Figure 7. As seen in Figure 7, $D(z)$,

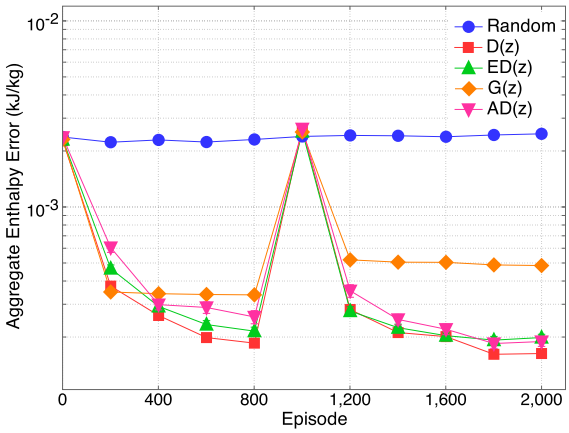

Figure 7: Rankine Cycle DCP with 1000 agents and 15% noise and failures.

$ED(z)$, and $AD(z)$ all perform almost identically. Again, $ED(z)$ and $AD(z)$ are essentially equivalent, because noise in the system results in the probability distribution for calculating $ED(z)$ near-uniform, making it almost identical to the true uniform distribution used to calculate $AD(z)$. An interesting result is the fact that $ED(z)$ is able to recover from failure when noise is present, even though it was unable to recover from failure when there was no noise. This can be attributed to the effect the noise has on the probability distribution for calculating $ED(z)$. Based on these results, we conclude that $AD(z)$ is the best reward choice for training agents in the distributed sensor network. $AD(z)$ is robust to measurement noise, sensor failures, and a combination of measurement noise and sensor failures. Other reward structures struggle in at least one of these situations, while $AD(z)$ provides consistently good feedback for learning agents.

5.5 Temperature Tracking

The final experiment involves training a distributed sensor network with $AD(z)$, and then using this sensor network to track the temperature at the turbine outlet during heatup. The temperature at the turbine outlet was raised from 200^oC to 315^oC. We add artificial oscillations to the temperature profile in order to add complexity to the problem. This experiment gives insight to how the sensor network can track changing parameters, which is a crucial element for a sensor network giving feedback to a plant controller. There are 1000 sensors, and the sensors have 15% measurment noise. The results for the temperature tracking experiment are shown in Figure 8. As seen in Figure 8, there is no observable difference between the true temperature profile and the measured temperature profile. This shows that the trained sensor network is capable of tracking system parameters, which is essential for the plant control which makes use of these parameters. Thus, in addition to being robust to agent noise and failures, $AD(z)$ can provide accurate measurements to the system controller, making it an ideal reward structure for use in real-world power plant control applications.

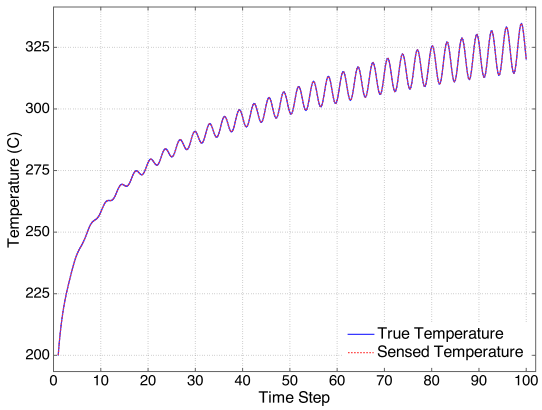

Figure 8: Temperature tracking at turbine outlet. There is no discernible difference between the true and measured temperatures.

6. DISCUSSION

This research investigated how to train a distributed sensor network operating in a power plant. We extended the Defect Combination Problem to utilize readily available information from the system model in order to train a sensor network. Then, we tested $G(z)$, $D(z)$, $ED(z)$, and $AD(z)$ as reward structures to train the sensor network. Our results show that variants of the Difference Reward provide sensor network performance that is almost two orders of magnitude more accurate than when using the global reward. Further, we show that $AD(z)$ is robust to sensor failures, measurement noise, and both sensor failures and measurement noise occuring simultaneously. Finally, we show that the sensor network trained with $AD(z)$ can accurately track dynamic system parameters, which is a crucial feature of a sensor network giving feedback to the plant controller. Thus, $AD(z)$ is an ideal reward structure for use in real-world sensor network applications.

Future work involves testing this sensor training algorithm on a more realistic simulator and to combine the sensor network with a controller. Although the Rankine cycle is not an accurate model of a real-world power plant, the algorithm we utilized is independent of the model; the model is treated as a black box, and only the model output affected the sensor network feedback. The complexity of the model does not affect how the sensor network learns, and is not necessary to test the sensor network itself. However, the complexity of the model is important when considering a controller, because the complexity of the control algorithm typically increases with system complexity. By incorporating a more realistic plant model, we can test the sensor network while it is working in conjunction with a plant controller. Ultimately, the sensor network must provide accurate measurement data in order for correct control decisions to be made, so the sensor network and controller should be tested together to give more meaningful insight on the quality of the distributed sensor network.

Acknowledgements This work was partially supported by DOE NETL grant DE-FE000085.

7. REFERENCES

[1] A. K. Agogino and K. Tumer. Analyzing and visualizing multiagent rewards in dynamic and stochastic environments. *Journal of Autonomous Agents and Multi-Agent Systems*, 17(2):320–338, 2008.

[2] D. Challet and N. F. Johnson. Optimal combinations of imperfect objects. *Physical Review Letters*, 89(2):028701, 2002.

[3] H. Chung, Z. Bien, J. Park, and P. Seong. Incipient multiple fault diagnosis in real time with application to large-scale systems. *Nuclear Science, IEEE Transactions on*, 41(4):1692 –1703, Aug 1994.

[4] A. Hussey, A. Nasipuri, R. Cox, and J. Sorge. Feasability of using a wireless mesh sensor network in a coal-fired power plant. *Proceedings of the IEEE SoutheastCon 2010*, pages 384–389, 2010.

[5] R. Lin, Z. Wang, and Y. Sun. Wireless sensor networks solutions for real time monitoring of nuclear power plant. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 4, pages 3663 – 3667 Vol.4, June 2004.

[6] M. J. Moran and H. N. Shapiro. *Fundamentals of Engineering Thermodynamics*. John Wiley and Sons, Inc., fifth edition, 2004.

[7] A. Rogers, A.and Farinelli and N. Jennings. Self-organising sensors for wide area surveillance using the max-sum algorithm. *Sensors Peterborough NH*, pages 84–100, 2010.

[8] A. Schroder, A. Schnettler, B. Schowe-von der brelie, I. Laresgoiti, and J. Lopez. Intelligent self-describing power grids. In *Electricity Distribution - Part 2, 2009. CIRED 2009. The 20th International Conference and Exhibition on*, page 1, June 2009.

[9] S. Sitharama Iyengar, Q. Wu, and N. Rao. Networking paradigm for distributed sensor networks. In *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings of the Second IEEE International Workshop on*, pages 284 –290, Sept. 2003.

[10] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE*, 7(5):16 –27, Oct 2000.

[11] R. Swartz, J. Lynch, and C.-H. Loh. Near real-time system identification in a wireless sensor network for adaptive feedback control. In *American Control Conference, 2009. ACC '09.*, pages 3914 –3919, June 2009.

[12] 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*, pages 173–188. Springer, 2005.

[13] A. Venkat, I. Hiskens, J. Rawlings, and S. Wright. Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, pages 1192–1206, 2008.

[14] Q. Wu, N. Rao, J. Barhen, S. Iyenger, V. Vaishnavi, H. Qi, and K. Chakrabarty. On computing mobile agent routes for data fusion in distributed sensor networks. *Knowledge and Data Engineering, IEEE Transactions on*, 16(6):740 – 753, June 2004.