

Coordinating Simple and Unreliable Agents

Kagan Tumer
NASA Ames Research Center
Mailstop 269-4
Moffett Field, CA 94035, USA
ktumer@mail.arc.nasa.gov

ABSTRACT

Coordinating a large number of simple and unreliable agents to achieve a global objective poses a unique design challenge: How to achieve robust system level behavior in spite of the computational/reliability limitations of the agents. We investigate agent coordination in the difficult optimization problem of selecting the subset of sensing devices with distortions that provides the smallest average distortion [1]. We approach problem by assigning an agent to each device and having that agent use a simple reinforcement learning algorithm to determine whether to be part of the aggregate device. Our results show that the right agent reward structure provides significant improvements over both traditional search methods and traditional multi-agent methods. Furthermore, even in extreme cases of agent failures (i.e., half the agents fail during the simulation) this approach still outperforms a failure-free and centralized search algorithm.

1. INTRODUCTION

Coordinating simple and unreliable agents to achieve a global objective is different than traditional coordination problem in that: (i) agents have limited sensing and decision making capabilities; (ii) agents are not be able to model the other agents in the system; and (iii) agents are likely to fail (up to 50% of them) during the simulation. We propose addressing this problem by having Reinforcement-Learning (RL) agent [3] aim to maximize their own utilities and let the system reward structure overcome the limited observation and robustness problems. The design problem we face then, is to determine how best to set the utility functions of the agents in a way that will lead to good values of the global utility, without involving difficult to scale external mechanisms. Note that though the agents have simple decisions to make, this is still fundamentally a multi-agent problem: Each agent autonomously makes a decision at each time step based on its estimate of the reward it will receive; and the system is fully distributed as each agent has full autonomy over its actions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.
Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

For the joint action of agents working in such a system to provide good values of the global utility, we must both ensure that the agents do not work at cross-purposes, and that each one has a learning problem that is solvable with limited resources. Typically these two requirements are in conflict with one another. For example, providing each agent with the global utility will ensure that they will not work at cross purposes. However, each agent then faces a difficult signal to noise problem as the reward of each agent depends on the actions of all the other agents. At the other extreme, providing each agent with a simple, local utility function will provide a clear signal, but may not necessarily lead the system to high values of global utility. The challenge is to find the best trade-off between these two requirements.

2. COMBINATION OF FAULTY DEVICES

The faulty device combination problem arises when many simple and noisy observational devices (e.g., nano or micro devices, low power sensing devices) attempt to accurately determine some value related to the phenomenon they're observing [1]. Each device provides a single number that is slightly off, similar to sampling a Gaussian centered on the value of the real number. The problem is to choose the subset of a fixed collection of the devices so that the average (over the members of the subset) distortion is as close to zero as possible. Formally, the problem is to minimize

$$\epsilon \equiv \frac{|\sum_{j=1}^N n_j a_j|}{\sum_{k=1}^N n_k}, \quad (1)$$

where $n_j \in \{0, 1\}$ is whether device j is or is not selected, and there are N devices in the collection, having associated distortions $\{a_j\}$. This is a hard optimization problem that is similar to known NP-complete problems such as subset sum or partitioning [1], but has two twists: the presence of the denominator and that $a_j \in R \forall j$. We denote the global utility function by $G = -\epsilon$ (so that to be consistent with the concept of “utility”, the goal is to “maximize” G). The system is composed of N agents, each responsible for setting one of the n_j . The problem we face is in determining the agent utilities so that, as the agents learn to maximize their own utilities, they also maximize G .

2.1 Expected Difference Utility

The first agent utility we discuss is the **Estimated Difference Utility**, EDU, given by [5, 6]:

$$EDU_i \equiv G(z) - E_{z_i}[G(z)|z_{-i}] \quad (2)$$

where z is the full system state, z_i is the components of z that depend on agent i 's actions, z_{-i} are the components of z that do not depend on the actions of agent i , and $E_{z_i}[G(z)|z_{-i}]$ is the expected value of G over the possible actions of agent i . Using Eq. 1, EDU leads to:

$$\begin{aligned} EDU_i(z) = & - \frac{|\sum_{j=1}^N n_j a_j|}{\sum_{k=1}^N n_k} \\ & + \left(p(n_i = 1) \frac{|\sum_{j \neq i}^N n_j a_j + a_i|}{\sum_{k \neq i}^N n_k + 1} \right. \\ & \left. + p(n_i = 0) \frac{|\sum_{j \neq i}^N n_j a_j|}{\sum_{k \neq i}^N n_k} \right) \end{aligned} \quad (3)$$

where $p(n_i = 1)$ and $p(n_i = 0)$ give the probabilities that agent i set its n_i to 1 or 0 respectively. In what follows, we will assume that those two actions are equally likely (i.e., for all agents i , $p(n_i = 1) = p(n_i = 0) = 0.5$).

Depending on which action agent i chose (0 or 1), EDU(z) reduces to:

$$0.5 \left(\frac{|\sum_{j=1}^N n_j a_j - a_i|}{\sum_{k=1}^N n_k - 1} - \frac{|\sum_{j=1}^N n_j a_j|}{\sum_{k=1}^N n_k} \right) \text{ if } n_i = 1, \quad (4)$$

or:

$$0.5 \left(\frac{|\sum_{j=1}^N n_j a_j + a_i|}{\sum_{k=1}^N n_k + 1} - \frac{|\sum_{j=1}^N n_j a_j|}{\sum_{k=1}^N n_k} \right) \text{ if } n_i = 0. \quad (5)$$

Note that in this formulation, EDU provides a very clear signal. Because EDU is aligned with G [5] (i.e., actions that improve EDU also improve G) if EDU is positive, the action taken by agent i was beneficial to G , and if EDU is negative, the action was detrimental to G . Thus an agent trying to maximize EDU will efficiently maximize G , without explicitly trying to do so. Furthermore, note that the computation of EDU requires very little information. Any system capable of broadcasting G can be minimally modified to accommodate EDU. For each agent to compute its EDU, the system needs to broadcast the two numbers needed to compute G : the number of devices that were turned on (i.e., the denominator in Equation 1) and the associated subset distortion as a real number (i.e., the numerator in Equation 1 before the absolute value operation is performed. Based on those two numbers, the agent can compute its EDU.

2.2 Wonderful Life Utility

The second utility we consider is the **Wonderful Life Utility** [5, 6], given by:

$$WLU_i \equiv G(z) - G(z_{-i}). \quad (6)$$

The major difference between EDU and WLU is in how they handle z_{-i} . EDU provides an estimate of agent i 's impact by sampling all possible actions of agent i whereas WLU simply removes agent i from the system. For this application, the WLU becomes:

$$WLU_i(z) = - \frac{|\sum_{j=1}^N n_j a_j|}{\sum_{k=1}^N n_k} + \frac{|\sum_{j \neq i}^N n_j a_j|}{\sum_{k \neq i}^N n_k} \quad (7)$$

Note however, that unlike with EDU, the action chosen by agent i has a large impact on the WLU. If agent i chooses action 0, the two terms in Equation 7 are identical, resulting

in a WLU of zero. Depending on which action agent i chose (0 or 1), WLU reduces to 0 if $n_i = 0$ or:

$$\frac{|\sum_{j=1}^N n_j a_j - a_i|}{\sum_{k=1}^N n_k - 1} - \frac{|\sum_{j=1}^N n_j a_j|}{\sum_{k=1}^N n_k} \text{ if } n_i = 1. \quad (8)$$

In this formulation, unlike EDU, WLU provides a clear signal only if agent i had chosen action 1. In that case, a positive WLU means that the action was beneficial to G , and a negative WLU means that the action was detrimental for G . However, if agent i had chosen action 0, it receives a reward of 0 regardless of whether that action was good or bad for G . This means that on average half the actions an agent takes will be random as far as G is concerned.

3. EXPERIMENTAL RESULTS

Each agent used a simple reinforcement learning algorithm, where each action is associated with the value of the utilities it has yielded. Agents choose actions to maximize those utility values. However, to spur exploration, rather than selecting the action with the highest expected utility, the agents used a Boltzman distribution over those utility values to select an action. In order to allow the agents to gather data, the agents took random actions for the first 20 time steps. The results are based on 20 different $\{a_i\}$ configurations, where each $\{a_i\}$ is selected from a Gaussian distribution with zero mean and unit variance. For each configuration, the experiments were run 50 times (i.e., each point on the figures is the average of $20 \times 50 = 1000$ runs).

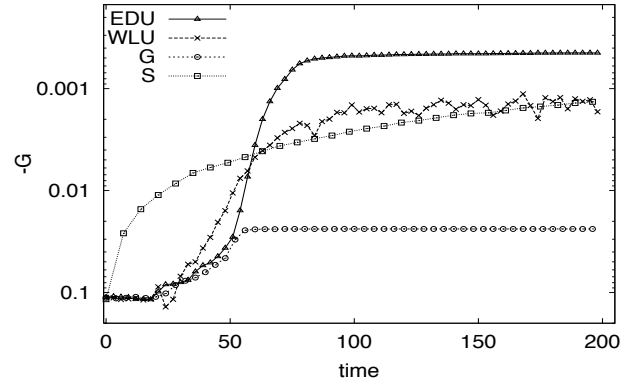


Figure 1: System Performance, N=100.

3.1 Agent Utility Performance

Figure 1 shows the simulation results. The graphs labeled G , EDU and WLU show the performance of agents using G , EDU and WLU as their reward, respectively. S shows the performance of local search where new n_i 's are generated at each step by perturbing the current state and selected if the solution is better than the current best solution (in the experiments reported here, 25% of the actions were randomly changed at each time step, though somewhat surprisingly, the results are not particularly sensitive to this parameter). This local search algorithm provides the performance of an algorithm with centralized control. Because the runs are only 200 time steps long, algorithms such as simulated annealing do not outperform local search: there is simply no time for an annealing schedule. Agents use G directly have

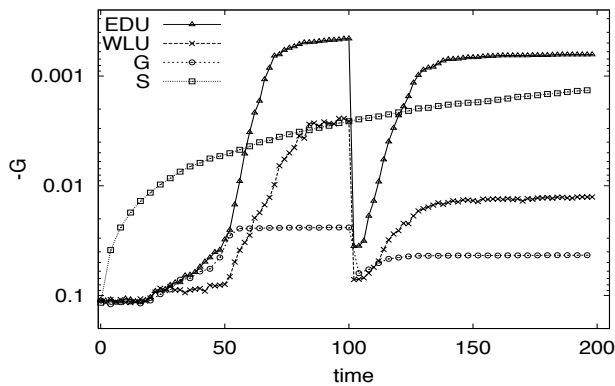


Figure 2: System performance for 100 agents, 20% of which fail at time $t=100$.

a difficult time learning. The noise in the system is simply too large for such agents to learn how to select their actions. WLU performs at the level of the centralized algorithm. Because agents only receive useful feedback when they take one of the two actions, the performance is between that of G and EDU . EDU outperforms both the other RL algorithms and the centralized search algorithm. This is due to EDU receiving a clear signal for each of the two actions.

3.2 Handling Agent Failures

In order to evaluate the robustness of the proposed utility functions for multiagent coordination, we tested the performance of the system when a subset of the agents failed during the simulation. At $t = 100$, a certain percentage of agents failed (e.g., were turned off) simulating hazardous condition with agents malfunctioning. The relevance of this experiment was in determining whether the proposed utility functions require all or a large portion of the agents to perform well to be effective, or whether they can handle sudden changes to their environment.

Figure 2 shows the performance of 100 agents where 20% of the agents fail at time step $t = 100$. The results of the centralized search algorithm with no failures (“S” from Section 3.1), is also included for comparison. In these experiments, the agent learning algorithms were not adjusted to account for the change in the environment. In agents that continued to function, the learning proceeded as though nothing had happened. As a consequence, not only did the agents need to overcome the sudden change in their task but they had to do so with parameters tuned to the previous environment. Despite these limitations, EDU recovers rapidly from having 20% of the agents fail. Note this is a powerful results: a distributed algorithm with only 80% functioning agents, each tuned to a different environment outperforms a 100% functioning centralized algorithm.

Figure 3 shows the performance of EDU when the percentage of agent failures increases from 10 to 50%. For comparison purposes, the search results (From Section 3.1) are also included. After the initial drop in performance when the agents stop responding, the system using EDU recovered rapidly and outperformed the fully functioning and centralized search algorithm. These results demonstrate both the adaptability of EDU and its robustness to failures of individual agents, even in extreme cases where half the agents fail during the run.

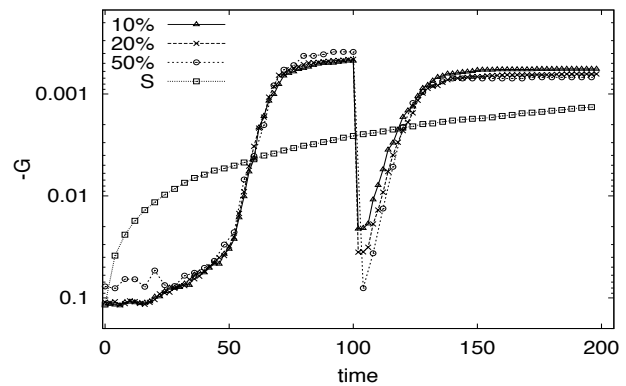


Figure 3: Effect of agent failures on EDU for 100 agents (S has no agent failures).

4. DISCUSSION

The combination of faulty devices is a simple abstraction of a problem that will loom large in the near future: How to coordinate a large number of agents – many of which may have limited access to information and perform unreliably – to achieve a prespecified global objective. This problem is fundamentally different from traditional multi-agent problems in at least three ways: (i) the agents have limited sensing and decision making capabilities; (ii) the agent do not model the actions of other agents; and (iii) the agents are unreliable and failure-prone.

The work summarized in this article is based on ensuring coordination among simple and unreliable agents by providing them with the proper reward structure. This approach is well-suited for addressing coordination in large scale cooperative multi-agent systems where the agents do not have pre-set and possibly conflicting objectives, or when the agents do not need to hide their objectives. The results shows the promise of this method by providing improvements over both a centralized algorithm and a multi-agent system using a team game approach. Furthermore, when as many as half the agents fail during simulations, the proposed method still outperforms a fully functioning centralized search algorithm.

Acknowledgments: The author would like to thank David Wolpert for invaluable discussions and for bringing the faulty devices problem to his attention, and Adrian Agogino for his many comments.

5. REFERENCES

- [1] D. Challet and N. F. Johnson. Optimal combinations of imperfect objects. *Phys. Rev. Let.*, 89:028701, 2002.
- [2] T. Hogg and B. A. Huberman. Controlling smart matter. *Smart Materials and Structures*, 7:R1–R14, 1998.
- [3] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [4] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.
- [5] K. Tumer and D. Wolpert. A survey of collectives. In *Collectives and the Design of Complex Systems*, pages 1,42. Springer, 2004.
- [6] D. H. Wolpert and K. Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(2/3):265–279, 2001.