Coalition Formation meets Information Theory

Victor Palmer[*] and **Thomas Ioerger**
Department of Computer Science
Texas A&M University
301 Harvey R. Bright Bldg
College Station, TX. 77843-3112
vpalmer | ioerger@cs.tamu.edu

Abstract

The process of coalition formation, where distinct autonomous agents come together to act as a coherent group is an important form of interaction in multi-agent systems. Previous work has focused on developing coalition formation algorithms which seek to maximize some coalition structure valuation function $V(\mathcal{CS})$. However, for many real-world systems, evaluation of $V(\mathcal{CS})$ must be done empirically, which can be time-consuming, and when evaluation of $V(\mathcal{CS})$ becomes too expensive, value-based coalition formation algorithms can become unattractive. In this work we present a algorithm for forming high value coalition structures when direct evaluation of $V(\mathcal{CS})$ is not feasible. We present the IBCF (Information-Based Coalition Formation) algorithm, which does not try to directly maximize $V(\mathcal{CS})$, but instead seeks to form coalitions which possess maximum amounts of information about how environmental states and agent actions relate to external reward. Such information maximization strategies have been found to work well in other areas of artificial intelligence, and we evaluate the performance of the IBCF algorithm on two multi-agent control domains (multi-agent pole balancing and the SysAdmin network management problem) and compare the performance of IBCF against relevant state-of-the-art algorithms in each domain.

Introduction

Background

The process of coalition formation, where distinct autonomous agents come together to act as a coherent group is an important form of interaction in multi-agent systems. Partitioning a collection of agents into judiciously chosen coalitions can result in significant performance benefits to the entire agent system such as reduced agent coordination costs and training times, and increased access to pooled resources. The use of coalitions has been advocated in a wide variety of domains including electronic business (Tsvetovat & Sycara 2000) (purchasing agents pooling their requirements to obtain larger discounts), grid-computing (I. Foster & Tuecke 2001), and e-business (T. Norman 2004) (where agents can come together to form organizations to fill market niches). The coalition formation process can be viewed as being composed of three main activities (T. Sandholm & Tohme 1999):

1. *Coalition Structure (CS) Generation*: The creation of coalitions such that agents in a given coalition coordinate their actions while agents in different coalitions do not. This involves partitioning the agents into disjoint sets, and such a partition is called a *coalition structure*. For example, in a multi-agent system composed of three agents n_1, n_2, n_3, there are seven possible coalitions: $\{n_1\}, \{n_2\}, \{n_3\}, \{n_1, n_2\}, \{n_2, n_3\}, \{n_3, n_1\}, \{n_1, n_2, n_3\}$. A valid coalition structure would be $\mathcal{CS} = \{\{n_1\}, \{n_2, n_3\}\}$

2. *Coalition Value Maximization*: In this step, the resources of the individual coalition agents are pooled to maximize the payoff to each coalition. In a market-based environment, this might mean literally combining financial resources to gain more bargaining sway, or in the context of a system of agents trained with reinforcement learning for example, this would involve training a reward-maximizing policy over a coalition's agents.

3. *Payoff Distribution*: Since coalition formation generally occurs in the context of autonomous, self-interested agents, at some point reward distributed to a coalition must be parceled to its member agents, and there are various common methods for doing this - equally among members, proportional to the agent's contribution, etc.

Recent work on coalition formation has concentrated on the first of these steps (CS generation), and we will continue this tradition here. Generally, CS generation takes place in the context of some valuation function $V(\mathcal{CS})$ which measures the fitness of a given coalition structure. If the task structure of the environment is known ahead of time, this function may be directly evaluatable as in (Shehory & Kraus 1998), while in other contexts, one may use the sum of individual agent reward payoffs (which would require performing all three steps of the coalition formation process).

In general, the goal of CS-generation is to construct a coalition structure which maximizes this valuation func-

[*]This work was supported by a Fannie and John Hertz Foundation fellowship.

tion. Many authors have proposed algorithms for coalition formation in different situations, such as when optimality bounds are required (T. Sandholm & Tohme 1999; Dang & Jennings 2004), or when coalition size can be bounded from above (Shehory & Kraus 1998). While much of this work has centered on deterministic algorithms, the field has seen some application of stochastic methods as well (particularly genetic algorithms (Sen & Dutta 2000)).

Generally - certainly in the work cited above - CS-generation requires access to the valuation function $V(\mathcal{CS})$ (for example, in the Sen & Dutta work, $V(\mathcal{CS})$ is needed to perform fitness evaluation in the genetic algorithm). And in fact, many accesses to $V(\mathcal{CS})$ may be required before a CS generation algorithm terminates. If the nature of the agent system, its environment, the task structure, etc. are known *a priori*, evaluating this function may be able to be done directly (as in (Shehory & Kraus 1998)). However, in other cases, such as when dealing with coalition formation among physical robotic systems, valuing a coalition structure may need to be done empirically by retraining the system to respect the CS and then letting the system interact with the environment, which might be an expensive undertaking.

Our Contribution

In this present work we look at the problem of coalition structure generation when repeated evaluation of $V(\mathcal{CS})$ is not feasible. While this may seem a little backward - attempting to maximize the value of a CS without having access to the CS valuation function, similar problems in AI-related fields have been successfully addressed by the application of *information-theory based methods* (e.g. (Battiti 1994)). For example, in the classifier training community, one faces similar difficulties with the problem of feature selection. In classifier feature selection, one attempts to reduce the dimensionality of a training corpus by discarding all but a few features. This training corpus is used to train a classifier, and one would like to keep features which make the error rate of the final trained classifier is as low as possible. The 'obvious' solution of iteratively trying different sets of features until minimum classifier error is achieved is untenable because of the possibly significant time required to retrain the classifier. The solution is to use information-theory-based methods to construct a feature set which conveys the most information possible about class labels. While this approach does not guarantee a high-performance classifier, it tries to provide the classifier training algorithm with most informative data set possible, which hopefully will result in a high-performance classifier post-training.

In our present coalition structure generation problem, we will take a similar approach. Instead of directly attempting to maximize the coalition valuation function $V(\mathcal{CS})$, we will attempt to form coalitions such that, post-coalition formation, as much information as possible is retained about how environmental states and agent actions relate to external rewards. Because we will need a training-corpus to calculate coalition information content, we will here concentrate exclusively on the use of coalition formation in a corpus-based reinforcement learning context (such as Least-Squares Policy Iteration (Lagoudakis & Parr 2003), where experience data is collected in one stage and training occurs afterward). We will then present the IBCF (Information-Based Coalition Formation) algorithm for generating these information-maximizing coalitions, and finally demonstrate the application of the IBCF algorithm on a multi-agent pole balancing problem and a multi-agent network control problem.

Preliminaries

Reinforcement Learning Basics

We work in the typical (multi-agent) reinforcement learning paradigm (Sutton & Barto 1998) in which a set of learning agents interact with a Markov decision process (MDP). Specifically, let us have a set of N agents $\{n_1, n_2, \ldots, n_N\}$, each of which possess a set of sensors $S(n_i) = \{s_j, \ldots, s_k\}$ and a set of actions $A(n_i) = \{a_l, \ldots, a_m\}$. Let us take the total number of sensors for all agents in the system to be N_S and the total number of actions for all agents to be N_A. We will take each sensor as generating a scalar output in the real numbers at any given point in time ($s_i(t) \in \mathbb{R}$).

The state, agent actions and reward at each time-step $t \in \{0, 1, 2, \ldots\}$ are denoted $\mathbf{s_t} \in \mathcal{S} \equiv S(n_1) \otimes S(n_2) \otimes \ldots \otimes S(n_N) \in \mathbb{R}^{N_S}$, $\mathbf{a_t} \in \mathcal{A} \equiv A(n_1) \otimes A(n_2) \otimes \ldots \otimes A(n_N) \in \mathbb{Z}^{N_A}$, and $r_t \in \mathcal{R}$ respectively. The environment's dynamics are characterized by state transition probabilities, $\mathcal{P}(\mathbf{s}, \mathbf{a}, \mathbf{s}') = Pr\{\mathbf{s_{t+1}} = \mathbf{s}' | \mathbf{s_t} = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\}$, and expected rewards $\mathcal{R}(\mathbf{s}, \mathbf{a}) = E\{r_{t+1} | \mathbf{s_t} = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\}$.

System agents decide what action to take at each time-step by following a policy $\pi(\mathbf{a_t}; \mathbf{s_t})$ such that the probability that the system agents will (collectively) perform action $\mathbf{a_t}$ when the environment is in state $\mathbf{s_t}$ is $\pi(\mathbf{a_t}; \mathbf{s_t})$. There are several measures of policy quality, though the one we will be most interested in here is long-term reward $\rho(\pi)$:

$$\rho(\pi) = E\{\sum_{t=1}^{\infty} \gamma^{t-1} r_t | \mathbf{s_0}, \pi\} \quad (1)$$

for some scalar *discounting factor* $0 < \gamma < 1$.

Coalition Formation Basics

A coalition of agents $C = \{n_i, \ldots, n_k\}$ groups a set of agents together such that they are able to act in a coordinated fashion. Here, we will take a *coalition structure* $\mathcal{CS} = \{C_1, \ldots, C_{|\mathcal{CS}|}\}$ to be a set of coalitions such that the system agents $\{n_i\}$ are disjointly split up into $|\mathcal{CS}|$ coalitions, and such that every agent is assigned to some coalition in the coalition structure. When needed, we will refer to the i^{th} coalition in a coalition structure as $\mathcal{CS}(i)$.

Reduced Environments and Policies

Imagine that we are interested only in the set of agents in a single coalition $C = \{n_j, \ldots, n_k\}$. If we only consider this set of agents, we only have access to the sensors possessed by the agents $n_i \in C$. We will let $\mathbf{s_t}^C$ refer to the reduced system state obtained by taking the state vector $\mathbf{s_t}$ and deleting any components generated from sensors possessed by agents not in C. Likewise, we will let $\mathbf{a_t}^C$ refer to the actions taken by only those agents in C. Finally, we will say that a policy $\mathbf{a_t}^C = \pi^C(\mathbf{s_t}^C)$ is a policy which controls

agents in coalition C by responding to the sensor readings of the agents in coalition C.

Coalition Structure Valuation

The value of a coalition structure $\mathcal{CS} = \{C_i\}$ will be determined by training and evaluating $|\mathcal{CS}|$ optimal policies π^{C_i}, one policy to govern the agents in each coalition $C_i \in \mathcal{CS}$. Specifically, we will here say that the value of a coalition structure $V(\mathcal{CS})$ is:

$$V(\mathcal{CS}) = \rho(\{\pi^{C_i}\}) = E\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r_t | \mathbf{s_0}, \{\pi^{C_i}\}\Big] \quad (2)$$

Training Corpus

Although many variations of reinforcement learning occur on-line, with the agent(s) under training interacting with the environment in real-time, we assume here that we have collected a set of agent experiences beforehand, and then perform reinforcement learning over this *training corpus*. That is, imagine that we have collected a number of *experience tuples* (collectively a training corpus $\mathcal{T}$) by allowing our agent system to interact with the environment:

$$\mathcal{T} = \{\langle \mathbf{s_t}, \mathbf{a_t}, \mathbf{s'_t}, r_t \rangle\} \quad (3)$$

If we train a policy over the system as a whole, without the consideration of coalitions, we would be able to use $\mathcal{T}$ directly. However, with the introduction of coalitions, policies would be trained over each coalition separately since inter-coalitional agents do not coordinate their actions. This would have the effect that, for coalition C_i, the training corpus would become:

$$\mathcal{T}^{C_i} = \{\langle \mathbf{s_t}^{C_i}, \mathbf{a_t}^{C_i}, \mathbf{s_t}'^{C_i}, r_t \rangle\} \quad (4)$$

such that the corpus $\mathcal{T}^{C_i}$ used to train the agents in C_i may not consider any of the sensor outputs or actions of agents $n_j \notin C_i$.

Our Approach

As promised in the introduction, we will view the coalition formation process from an information theoretic perspective. Specifically, we will view the act of training our agents via reinforcement learning as the following: a training corpus $\mathcal{T}$ which contains information relating the actions of system agents and the state of the environment to the reward received by the agent system is transformed into a reward-maximizing policy by the reinforcement learning algorithm.

Now, because agents only coordinate intra-coalitionally, agents only have access to the sensors and knowledge of the actions of agents in their own coalition. In this context, imagine the effect of coalition formation from the perspective of some agent n_i. If n_i were not in a coalition, n_i would have access to the entire training corpus $\mathcal{T}$, and all the information contained therein about how to go about obtaining reward. However, if we place n_i in a coalition C_i along with several other agents, suddenly n_i only has access to the sensor readings of the other agents in its own coalition. Since n_i could not use the sensor output of extra-coalitional agents for policy execution anyway, the contribution to the training corpus $\mathcal{T}$ from extra-coalitional agents cannot affect the final policy of agent n_i. Thus, as far as the reinforcement learning process is concerned, placing n_i in a coalition is equivalent to altering the training corpus $\mathcal{T}$ to remove the contributions from extra-coalitional agents (effectively transforming $\mathcal{T} \Rrightarrow \mathcal{T}^{C_i}$). Since $\mathcal{T}$ is being altered, coalition formation will necessarily have an impact on the information contained in $\mathcal{T}$ relating environmental states and agent actions to reward. As did the work on feature selection and classifier training, we will proceed with the assumption that a CS which maximizes the amount of information coalitions' possess about the relation between state/actions and reward, will, if not guarantee us an optimally performing CS, will at least result in *better-performing* coalition structures than those with coalitions which provide less information about reward. As such, we will be interested in methods to attempt to maximize this state/action/reward information.

Information Theory

We can write down the mutual information in the training corpus relating states and actions to reward as the following:

$$I(r; \mathbf{s}, \mathbf{a}, \mathbf{s}') = \sum_{\mathbf{a}} \int_r \int_{\mathbf{s}} \int_{\mathbf{s}'} P(\mathbf{s}, \mathbf{a}, \mathbf{s}', r) \log \frac{P(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)}{P(\mathbf{s}, \mathbf{a}, \mathbf{s}')P(r)} \quad (5)$$

We can also speak of the information available to agents in a coalition C_i:

$$I(C_i) \equiv I(r; \mathbf{s}^{C_i}, \mathbf{a}^{C_i}, \mathbf{s}'^{C_i}) \quad (6)$$

$$= \sum_{\mathbf{a}^{C_i}} \int_r \int_{\mathbf{s}^{C_i}} \int_{\mathbf{s}'^{C_i}} P(\mathbf{s}^{C_i}, \mathbf{a}^{C_i}, \mathbf{s}'^{C_i}, r) \log \frac{P(\mathbf{s}^{C_i}, \mathbf{a}^{C_i}, \mathbf{s}'^{C_i}, r)}{P(\mathbf{s}^{C_i}, \mathbf{a}^{C_i}, \mathbf{s}'^{C_i})P(r)} \quad (7)$$

For a given coalition structure $\mathcal{CS}$, each of the $|\mathcal{CS}|$ coalitions will have some $I(C_i)$. We can also define the *average* coalitional mutual information across the coalition structure:

$$I(\mathcal{CS}) = \frac{1}{|\mathcal{CS}|} \sum_{C_i \in \mathcal{CS}} I(C_i) \quad (8)$$

Since our goal is to maximize the amount of information possessed by each coalition, we will attempt to maximize this quantity. As pointed out in (Battiti 1994), it is generally not possible to optimize such an expression for mutual information content with computational effort which scales reasonably with number of information variables. However, as we pointed out last section, even if we could find such an information maximizing coalition structure, it would not guarantee us that optimally-*performing* coalition structures would result. Thus, as (Battiti 1994) points out, it is reasonable to use approximate algorithms for maximizing information, and as such here we will use a simple greedy approach.

IBCF Coalition Formation Algorithm

We now present an algorithm to form a coalition structure by greedy maximization of mean coalition information:

IBCF Algorithm

```
Set CS = {{n1},{n2},...,{nN}}
Set DONE = FALSE, MAX_MEAN = -1

While DONE = FALSE

    DONE = TRUE

    For i,j = 1 to |CS|
        Set CS' = CS
        Merge CS'(i) and CS'(j)

        TEST_MEAN = I(CS')
        If TEST_MEAN > MAX_MEAN
            CS'' = CS'
            MAX_MEAN = TEST_MEAN
            DONE = FALSE
        End
    End

    If DONE = FALSE
        CS = CS''
    End
End
```

Computational Requirements

Since in each iteration of the main while loop two coalitions are merged, the outer loop can execute at most N (number of system agents) times. The double inner For loops operate over pairs of coalitions, such that the total number of evaluations of evaluations of I before termination is at worst:

$$\sum_{x=1}^{N} x^2 = \frac{1}{6}N(N+1)(2N+1) \propto O(N^3) \qquad (9)$$

There are several methods for practically evaluating I: one can simply construct a histogram of the various probability distributions and calculate information directly, or one can use more advanced level-of-detail approaches such as Fraser's algorithm (Fraser & Swinney 1986). Regardless, in the worst case, one can always use the brute force histogram construction method. In this case, calculating mutual information over n variables will require $O(C^n)$ bins, where C is the number of segments along any given dimension of the histogram. Since our expression for mutual information involves the reward variable, and 3 state/action variables (s, a, s') for each agent, we would require $O(C^{3N+1})$ bins for information evaluation over N agents. This may seem a steep requirement (though, the reader should note that this requirement could be much less steep under a more sophisticated estimation scheme such as Fraser's algorithm), but one must remember that we work in a reinforcement learning context. In general, since the action space of a set of agents scales exponentially in the number of agents, training an N-agent system even with modern reinforcement learning algorithms (e.g. Least-Squares Policy Iteration, (Lagoudakis & Parr 2003)) involves $O(C^N)$ computation *anyway*.

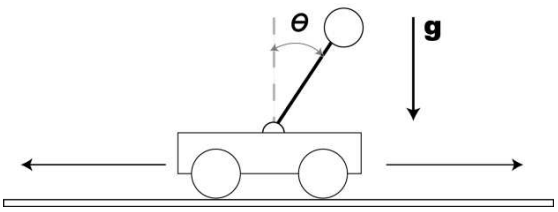

Figure 1: The cart-pole balancing problem.

Experiment 1: Multi-Agent Cart-Pole Balancing Problem

One of the classic problems in reinforcement learning is the cart-pole balancing problem. In this problem we have a cart of mass M capable of frictionless, one-dimensional motion. Attached to this cart via a hinge is a pole with a mass m on top. Gravity acts to pull the top mass down, but the pole and mass can be kept in the upright position by judicious movements of the cart.

Agents in this system are capable of applying some impulse either to the left or to the right of the cart, and reward is given out to the agents if the pole is within some angle tolerance of the upright position. System parameters are specified in two variables, θ, the angle of the pole away from the upright position, and $\omega = \dot{\theta}$, the angular velocity of the pole.

We performed a multi-agent version of this simulation to demonstrate our coalition formation algorithm. Specifically, in our system, each agent was given a single sensor which measured some linear combination of θ and ω (to enable direct combination of these two variables, both θ and ω were normalized to have zero mean and unit variance before use in all cases). Specifically, each agent was given a single sensor η_i:

$$\eta_i = cos(\phi_i)\theta + sin(\phi_i)\omega \qquad (10)$$

And as usual, each agent was capable of imparting a force from the left or right to the cart. This problem is interesting, because individually, each agent has insufficient information to solve the pole-balancing problem. That is, neither θ or ω alone is enough to solve the pole balancing problem, and neither is any linear combination of these two variables. Agents must be in coalitions such that other agents supply missing pieces of information to allow both agents to come to a joint decision as to what force to apply to the cart.

Pre-training, we ran our IBCF algorithm as described above on the cart-pole agents. Intuitively, one would expect that the maximum amount of reward-related information would occur in two-agent coalitions $\{n_1, n_2\}$ such that $\{\phi_1 = x, \phi_2 = x + \pi/2\}$, with higher-agent coalitions offering no informational advantage. That is, such a two-agent coalition has all the information contained in the original θ and ω variables - and in fact, the two variables θ and ω can be recovered by a rotation of $\{\eta_1, \eta_2\}$ by an angle $-x$.

We collected a training corpus by allowing $N = 8$ agents to interact with a pole-cart system. All physical constants and simulation methods were taken directly from the single-agent pole-balancing example presented in (Lagoudakis &

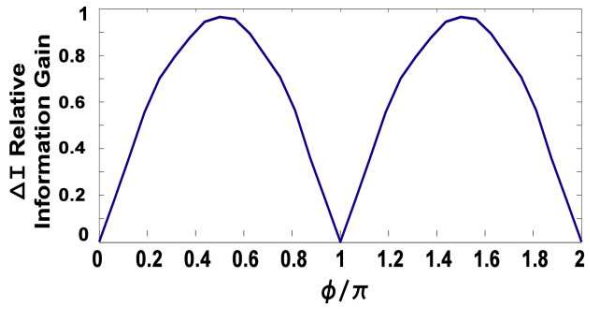

Figure 2: Relative information gain between the single-agent coalitions $C_1 = \{n_1\}, C_2 = \{n_2\}$ with $\phi_1 = 0, \phi_2 = \phi_1 + \phi$, and the two-agent coalition $C_3 = \{n_1, n_2\}$. Specifically, $\Delta I = \frac{I(C_3) - (I(C_1)+I(C_2))/2}{(I(C_1)+I(C_2))/2}$. Notice that when $\phi = 0$ the sensors of the two agents are identical and no information gain is achieved. The maximum is achieved at $\phi = \pi/2$ when the two sensor outputs are 'orthogonal'.

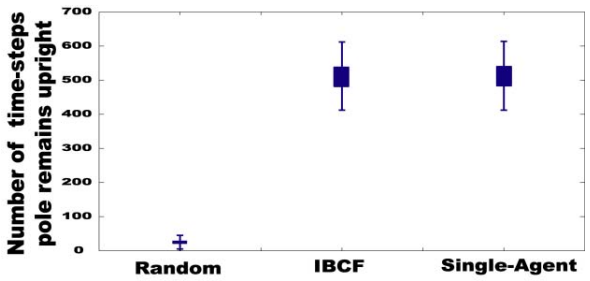

Figure 3: Results of the cart-pole balancing experiment.

Parr 2003). During experience collection, agents followed a random policy, applying force from the left or right with equal probability. We initialized the sensor angles to be the following $\phi_1 = 0, \phi_2 = \pi/2, \phi_3 = \pi/8, \phi_4 = \pi/8 + \pi/2, \phi_5 = \pi/4, \phi_6 = \pi/4 + \pi/2, \phi_7 = 3\pi/4, \phi_8 = 3\pi/4 + \pi/2$. After the training corpus was collected, we ran IBCF over the corpus, which produced the following coalition structure:

$$\mathcal{CS}_{\text{IBCF}} = \{\{n_1, n_2\}, \{n_3, n_4\}, \{n_5, n_6\}, \{n_7, n_8\}\} \tag{11}$$

As expected, the algorithm matched agents with complementary sensors such that intra-coalition mutual information between states/actions and reward was maximized. After coalition structure generation, we trained each coalition in isolation to balance the pole using the reinforcement learning scheme described in (Lagoudakis & Parr 2003) (Least-Squares Policy Iteration). After training was complete, we measured the average time the system was able to keep the pole balanced upright (with all agents in all coalitions able to apply forces simultaneously, of course). For comparison, we also measured the average 'upright time' when randomly generated coalition structures (made up of strictly two-agent coalitions) were used, and the average upright time for a single agent balancing the pole with access with both θ and ω directly (essentially reproducing the single-angle case presented in (Lagoudakis & Parr 2003) ... in this domain, the single-agent policy will perform optimally). For each case, we allowed the system to try to balance the pole 500 times, and the average upright times are plotted in Figure 3. The IBCF-coalition based results are nearly identical to the single-agent results, as expected, since each coalition has as much information about the system as does the single agent. Note that we did not compare against a direct search for an optimal coalition structure because the rather long training/evaluation times for this domain made this untenable.

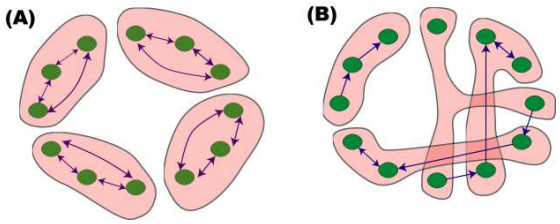

Figure 4: Two network topologies with the coalitions produced by IBCF drawn on the figure.

Experiment 2: Multi-Agent Network Control Problem (SysAdmin)

The SysAdmin problem, as presented in (D. Guestrin & Parr 2001) simulates the management of network by a system of agents. The problem roughly goes as follows: there is a network of N machines (connected to each other through some network topology), each of which is managed by an agent. These machines are designed to run processes, and when a machine completes a process, its agent is given some reward. Specifically, machines are capable of being in three 'stability' states STATE = {HEALTHY, UNSTABLE, DEAD} and three 'process' states LOAD = {IDLE, PROCSES RUNNING, PROCSS COMPLETE}. Healthy machines turn into unstable machines with some probability, and unstable machines turn into dead machines with some probability. Unstable machines take longer to complete processes (and hence generate less reward on average) and dead machines cannot run processes at all. To make matters more complicated, unstable and dead machines can send bad packets to their neighbors, causing them to go unstable and eventually die as well. At each time-step, each agent must decide whether or not to reboot its machine. Rebooting returns the machine to the healthy state, but at the cost of losing all running process (which effectively incurs some reward penalty since all work done on processes up to this point will be lost).

For our experiment here, we generated $N = 12$ machine networks with randomly generated network topologies. Specifically, each potential connection in the network (144 possible connections) had a probability (we used 0.05) of being turned on. For each network, we collected a training corpus by having each agent follow a random policy and

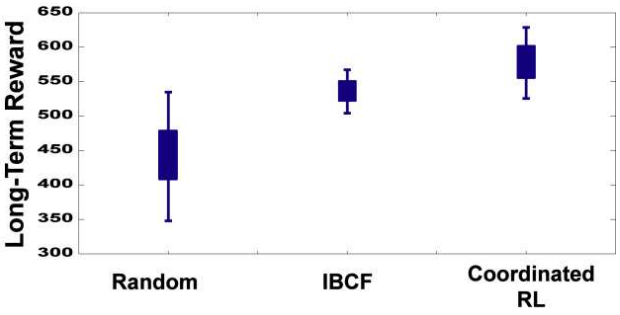

Figure 5: The results of the SysAdmin simulation.

reboot its machine with a 50% probability. All other simulation parameters were as described in the (D. Guestrin & Parr 2001) work. After the corpus was collected, we ran IBCF on the data. To speed up training (and because our networks generally ended up relatively sparse - usually two connections per node), we capped our coalition sizes at three agents (which we did by adjusting our algorithm such that when a coalition reached three agents in size, it was no longer considered for mergers). After coalition generation, we trained each coalition in isolation, using Least-Squares Policy Iteration, as outlined in (Lagoudakis & Parr 2003). For comparison, we also trained our agents under randomly generated coalition structures (with the same three-agent maximum coalition size), and also trained our network agents using Coordinated Reinforcement Learning (CRL) as described in (D. Guestrin & Parr 2001). Coordinated Reinforcement Learning is useful on domains where much is known about interdependencies between agents pre-training, and has been shown to produce policies which are near optimal on this particular SysAdmin problem domain (D. Guestrin & Parr 2001). While direct comparison between CRL and our coalition generation algorithm is really not reasonable since CRL requires significant additional information about the domain (besides the training corpus), the results provide a useful idea of relative level of performance of policies on this domain. The results are plotted in Figure 5 for 500 generated networks in each series. Note that once again we did not compare against a direct search for an optimal CS because of the computational costs.

It is interesting to observe the coalitions that our algorithm generated for various network topologies. Notice in Figure 4a that for a system where nodes are strictly connected to two neighbors (and an appropriate three-agent coalition structure is obvious), our algorithm generates coalitions reflecting these dependencies. Other more complex topologies resulted in different coalition structures, for example in Figure 4b, though the interdependencies of the network are still somewhat represented.

Conclusions

We have examined the issue of coalition structure generation from an information theoretical viewpoint. Specifically, we concentrated on reinforcement-learning-based domains where one has pre-learning training-corpus of collected experiences. We showed how one could extract statistical information from this corpus to determine coalition structures which maximized the information that each coalition possesses about the nature of how environmental states and agent actions relate to external reward, the assumption being that larger amounts of such information should result in the generation of better, higher-valuation policies.

We applied our algorithm to two multi-agent reinforcement domains: the multi-agent pole-balancing problem (as in (Lagoudakis & Parr 2003)) and the SysAdmin, network management problem (as in (D. Guestrin & Parr 2001)), and the system performance resulting from training over coalition structures generated with our IBCF algorithm compared favorably to other, specialized peer training algorithms in each domain.

References

Battiti, R. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks* 5(4).

D. Guestrin, D. K., and Parr, R. 2001. Multiagent planning with factored mdps. In *NIPS-14*.

Dang, V., and Jennings, N. 2004. Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems*, 563–571.

Fraser, A., and Swinney, H. 1986. Independent coordinates for strange attractors from mutual information. *Physical Review A* 33(2):1134–1140.

I. Foster, C. K., and Tuecke, S. 2001. The anatomy of the grid. *The International Journal of High Performance Computing Applications* 15(3):200–222.

Lagoudakis, M., and Parr, R. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* 4:1107–1149.

Sen, S., and Dutta, P. 2000. Searching for optimal coalition structures. In *Fourth International Conference on Multi-Agent Systesm (ICMAS'00)*, 287.

Shehory, O., and Kraus, S. 1998. Methods for task allocation via agent coalition formation. *Artificial Intelligence Journal* 101(1-2):165–200.

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

T. Norman, A. Preece, e. a. 2004. Agent-based formation of virtual organizations. *International Journal of Knowledge Based Systems* 17(2-4):103–111.

T. Sandholm, K. Larson, M. A. O. S., and Tohme, F. 1999. Coalition structure generation with worst case guarantees. *Artificial Intelligence* 111(1-2):209–238.

Tsvetovat, M., and Sycara, K. 2000. Customer coalitions in the electronic marketplace. In *Proceedings of the Fourth International Conference on Autonomous Agents*, 263–264.