Combining Coordination Mechanisms to Improve the Performance of Multi-Robot Teams

Ehsan Nasroullahi
Oregon State University
Corvallis, OR, 97331 USA
nasroumo@engr.orst.edu

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

Abstract

Coordination is essential to achieving good performance in cooperative multiagent systems. To date, most work has focused on either implicit or explicit coordination mechanisms, while relatively little work has focused on the benefits of combining these two approaches. In this work we demonstrate that combining explicit and implicit mechanisms can significantly improve coordination and system performance over either approach individually. First, we use *difference* evaluations (which aim to compute an agent's contribution to the team) and stigmergy to promote implicit coordination. Second, we introduce an explicit coordination mechanism dubbed Intended Destination Enhanced Artificial State (IDEAS), where an agent incorporates other agents' intended destinations directly into its state. The IDEAS approach does not require any formal negotiation between agents, and is based on passive information sharing. Finally, we combine these two approaches on a variant of a team-based multi-robot exploration domain, and show that agents using a both explicit and implicit coordination outperform other learning agents up to 25%.

1 Introduction

Coordinating a team of agents such that they collectively achieve a common goal is a complex problem within the field of multiagent systems [15]. Improving coordination in multiagent systems will benefit many application domain including Unmanned Aerial Vehicles (UAV) swarms, search and rescue mission, exploration, and sensor networks [2, 8, 1]. In general, coordination mechanisms can be broken down into two main categories: implicit and explicit coordination mechanisms. Implicit coordination relies solely upon an agent's observation of its environment to make decisions, while explicit coordination involves direct interaction and exchange of information between two or more agents. Implicit coordination mechanisms tend to be limited by observation restrictions and explicit methods are typically limited by communication restrictions. In many real-world domains agents have access to limited amounts of both observation and communication. In such cases, maximizing the benefit of both types of information by concurrently using implicit and explicit mechanisms is likely to be advantageous. We propose using a combination of explicit and implicit coordination mechanisms to improve coordination and performance over either method individually.

In this work we combine two well known implicit coordination mechanisms (coupled policy evaluations and stigmergy) with our a novel explicit coordination mechanism (communicating agents' *Intended Destination Enhanced Artificial State (IDEAS)*) to improve coordination in the Cooperatively Coupled Rover Domain (CCRD). The CCRD is an extension of the Continuous Rover Domain [2, 9], in which a set of rovers must coordinate their actions to collectively optimize coverage over a set of environmental points of interest (POIs) and individual rovers can observe any given POI. The CCRD increases the coordination complexity by requiring teams of agents to observe each POI. Here, agents must not only optimize coverage as a collective, but they must form teams and the teams must coordinate within themselves to optimize coverage of their POI as well [10]. This is difficult because there are two different coordination problems going on concurrently. First, at a high level all agents within the system must coordinate to provide optimal coverage of POIs. Second, agents must coordinate amongst themselves to form teams and the agents comprising the teams must coordinate their actions to optimally select and cover a given POI (teams either observe a POI together or not at all). This tight coupling between agents both at a system level and at a team level present a complex coordination problem.

The key contributions of this paper are as follows:

- Introduce a novel explicit coordination mechanism (*IDEAS*) in the Cooperatively Coupled Rover Domain.
- Combine implicit (coupled policy evaluation and stigmergy) and explicit (*IDEAS*) coordination mechanisms to improve performance of multi-rover teams.

The remainder of this paper is structured as follows. Section 2 provides background information on implicit coordination, explicit coordination, and multiagent coordination. Section 3 provides an introduction to the Continuous Rover Domain (CRD) used in this work. Section 4 provides an overview of the algorithms, evaluation functions, and methods used in this work. Section 5 contains the experimental results. Finally, Section 6 contains the discussion and conclusions of this work.

2 Background and Related Work

Implicit coordination relies solely upon an agent's observation of its environment to make decisions. One example used in this work is stigmergy, by which agents coordinate by communicating with each other through the environment [7, 13]. Another example used in this work is coupled policy evaluations (both *global* and *difference* policy evaluations), which allow agents to implicitly coordinate their actions based upon the policy evolutionary received [1, 2]. Here, agents have a policy evaluation that is reflective of how the agents collectively performed at achieving a particular objective. In this work, we focus primarily upon *difference evaluations* which are shaped policy evaluations designed to promote coordination in large multiagent systems [2, 19]. Difference evaluations evaluate each agent's impact on the system by comparing the systems performance both with and without the agent. If the agent made a positive contribution to the system, its policy evaluation is positive, if it harmed the system its policy evaluation is negative.

Explicit coordination involves direct interaction and exchange of information between two or more agents. Examples of explicit coordination are auctions, bidding, and other forms of negotiations [12, 14, 16, 18]. However, these methods are frequently complex and require a back-and-forth dialogue between agents before reaching agreement and taking any action, thus slow system response time[5]. Also, in many real world problems communication is limited in bandwidth and availability. In this work, we propose to utilize a novel explicit coordination mechanism called *IDEA* which only requires passive information sharing between agents. In particular, agents passively communicate the action that they currently intend to take based upon their perceived system state. This information provides agents with an effective "look ahead" at what other agents intend to do during the next time step, allowing them to adjust their actions to compensate accordingly. A similar approach was taken in [3], however they only took into account actions already taken by agents.

Coordination is essential in order to exploit the resources (high bandwidth communications and a diverse array of sensors and actuators) available on modern robots [6, 4]. Through sharing information and leveraging each others resources, a group of robots can truly be more than the sum of its parts [6]. Cooperative coordination is key to optimizing the performance of multirobot teams. Several algorithms and approaches have been developed including auction-based dynamic task allocation and ad-hoc team formation [6, 17, 20]. Although many existing algorithms have been shown to work well at improving coordination within multirobot teams, relatively few have explored the potential of combining different types of coordination mechanisms together in order to improve performance. The goal of this work is to explore such combinations.

3 Cooperatively Coupled Rover Domain

In this section we discuss the Cooperatively Cpupled Rover Domain (CCRD) that has been adapted and modified from the original Continuous Rover Domain [2, 9, 10]. The CCRD contains a set of rovers (agents) that are able to move around in a two dimensional plane to observe Points of Interest (POIs). Each of these POIs has a value (V_i) assigned to them. The goal of the agents is to implicitly form teams and for the teams to optimize their coverage of environmental POIs.

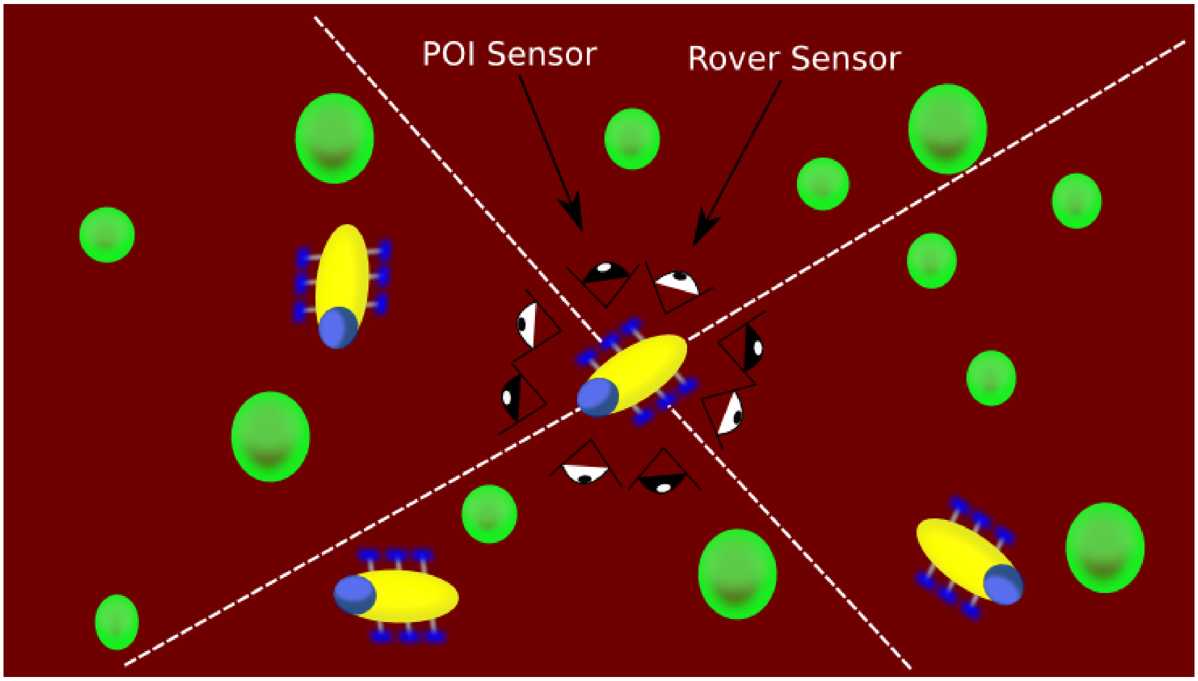

Figure 1: The figure shows how rovers observe their environment. Each rover has a set of 8 sensors (4 POI sensors and 4 rover sensors). The sensors provide coverage over 4 quadrants, where each quadrant contains one POI sensor and one rover sensor (Section 3).

In the CCRD, each agent has two types of sensors (POIs, Rover) and a total of 8 sensors (four of each type). Each rovers' point of view is divided into four quadrants, each quadrant contains both a POI and a rover sensor.[1] At every given time t, each of the sensors return a density of POIs or rovers (respectively) in their quadrant. The value of this density is the sum of the values of each of the POIs or rovers divided by their Euclidean distance from the sensor.[2] This yields the POI and rover density values for each quadrant (Equations 1 and 2, respectively):

$$s_{1,q,\eta,t} = \sum_{i \in POI} \frac{V_i}{\delta(L_i, L_{\eta,t})} \tag{1}$$

$$s_{2,q,\eta,t} = \sum_{\eta \in Rover} \frac{1}{\delta(L_{\eta'}, L_{\eta,t})} \tag{2}$$

where $s_{1,q,\eta,t}$ and $s_{2,q,\eta,t}$ are the POI and rover sensor readings respectively for quadrant q at time t for agent η, $\delta(x, y)$ is a Euclidean norm function, L_i is the location of POI i, and $L_{\eta,t}$ is the location of rover η at time t. The variables $s_{1,q,\eta,t}$ and $s_{2,q,\eta,t}$ represent the system state in this domain.

In order to earn credit for observing a given POI, a team of M rovers must observe it together (if fewer than M rovers observe the POI or the M rovers are not in the observation distance no credit is

[1] The orientation of these quadrants are based upon the rovers heading. The quadrant axes are oriented according to the rover's current heading.

[2] $\delta(x, y) = max\{||x - y||^2, d^2\}$, where d is the minimum distance to prevent undefined number.

given). The teams are implicitly formed and are defined as being the closest M rovers to a given POI. The goal of agents in this domain is for rovers to collectively position themselves such that teams of M rovers are optimally observing each POI. The resultant system objective then becomes:

$$G(z) = \sum_{t} \sum_{i \in POI} \sum_{m=1}^{M} \frac{V_i}{\delta_{i,t}^{m}} \tag{3}$$

where G is the system performance when M number of rovers are required to observe each POI, V_i is the value of each POI, $\delta_{i,t}^{m}$ is the distance between the i^{th} POI and the m^{th} closest rover to it at time step t. The value $\delta_{i,t}^{m}$ can be expounded as follows:

$$\Delta_{i,t} \equiv \{\delta(L_i, L_{\eta_1,t}), \delta(L_i, L_{\eta_2,t}) \dots \delta(L_i, L_{\eta_N,t})\} \tag{4}$$

where $\Delta_{i,t}$ contains the set of Euclidean norm distances between the i^{th} POI and the η^{th} rover at time step t.

$$\delta_{i,t}^{1} = min(\Delta_{i,t}) \tag{5}$$

$$\delta_{i,t}^{2} = min(\Delta_{i,t} \; - \; \{\delta_{i,t}^{1}\}) \tag{6}$$

$$\delta_{i,t}^{3} = min(\Delta_{i,t} \; - \; \{\delta_{i,t}^{1}\} \; - \; \{\delta_{i,t}^{2}\}) \tag{7}$$

where $\delta_{i,t}^{1}$, $\delta_{i,t}^{2}$, and $\delta_{i,t}^{3}$ are the distance between the i^{th} POI and the closest, second closest, and third closest rovers respectively to the POI at time step t.

4 Methodology

This section describes our algorithms, evaluation functions, and agents. In this work, each agent was a neuroevolutionary learner that evolved a pool of neural network policies. We used two different policy evaluations to rank the policies, each of which is described in section 4.4.

4.1 Neuroevolution

In domains with continuous action and state spaces like the CCRD, neuroevolutionary algorithms have been shown to be effective [2, 11]. In this work, each agent is a neuroevolutionary learner. Here, each agent evolves its policy using a pool of 10 neural network policies. Each of these neural networks had 10 hidden units, 8 inputs, 2 outputs (input and output are bounded [0,1]), and utilized sigmoid activation functions. Algorithm 1 shows the process that we used to select, mutate, evaluate, and rank our policies. The algorithm is a population based search with the IDEAS explicit coordination mechanism added into it (Algorithm 1). In our algorithm we initialize rank of all of the policies to zero, used ϵ-greedy to select the next policy, 5000 episode (T_{max}), 15 time step (t_{max}), 0.3 as mutation constant (ζ), 0.1 as learning rate (α).

Algorithm 1: This algorithm presents the process of evolving policies. In this algorithm ϵ is exploration rate, ζ is mutation rate, α is learning rate and R is the reward that each agent gets from its evaluation function.

```
Initialize $N$ networks at $T = 0$
for $T < T_{max}$ do
    1. Select a policy $N_i$ from population
            With probability $\epsilon$: $N_i \leftarrow N_{random}$
            With probability $1 - \epsilon$: $N_i \leftarrow N_{best}$
    2. Mutate the selected policy $N_i$:
            with probability $\zeta$: Mutate $N_i$
            with probability $1 - \zeta$: Do Not Mutate $N_i$
    3. Control robot with $N_i$ for next episode
    for $t < t_{max}$ do
        if IDEAS then
            Add IDEAS (Algorithm 2)
        else
            Update State
        Simulate
        Take an Action
        $Rank \leftarrow (1 - \alpha) * Rank + \alpha * R$
    4. Ranking the $N_i$ base on performance
            $N_i$ Rank $\leftarrow Rank$
    5. Replace $N_{worst}$ with $N_i$
```

4.2 IDEAS for Explicit Coordination

The actions are chosen by the agent's current policy, generated using the policy search algorithm described in Section 4.1. In the CCRD the movement of each rover was governed by these actions as follows:

$$dx = d(O_1 - 0.5) \tag{8}$$

$$dy = d(O_2 - 0.5) \tag{9}$$

where $d/2$ is the maximum distance a rover can move in a given direction at a single time step (10 units), O_1 and O_2 are the x and y outputs from the agents current policy (N_i), and dx and dy are the actions selected by policy.

We now introduce an agent's *IDEAS*. In this domain, an agent's IDEAS includes the POI that the agent is currently headed toward along with its Euclidian distance to that POI. This information is explicitly broadcast between agents within the system, allowing the agents to effectively "know" what other agents plan to do in the following time step so they can coordinate and react accordingly. When agents share information about what they intend to do, prior to actually taking any actions it can lead to better coordination and improved performance. This is because agents are able to actively account for each others' actions, without losing any time (agents are able to gain the insight from a future time step, without losing the time associated with taking that step). In order to determine an agents' IDEAS, the agent observes its environment and based upon what it observes it selects its current intended action (the agent observes the environmental state S and feeds it into its current policy to get out its intended action ($\vec{a} =< dx, dy >$). The agent then uses that "intended" action to predict its next move.In particular, the agent predicts the POI it will be closest to next time step and calculates its predicted distance to that POI. Each agent calculates this piece of information and all agents passively share this piece of information with each other. This piece of information is denoted as $s_{\eta,c}$:

$$s_{\eta,c} = \frac{V_{C_{POI}}}{\delta(L_{C_{POI}}, L_\eta)} \tag{10}$$

where $s_{\eta,c}$ is the portion of agent η's state vector that contains its predicted distance from $V_{C_{POI}}$, which is the value of the closest POI to agent η, $\delta(L_{C_{POI}}, L_\eta)$ is the Euclidean norm function, $L_{C_{POI}}$ is the location of the closest POI to rover η, and L_η is the predicted location of agent η at time $t+1$ based upon its current state and predicted action. This information represents the value of the POI that is predicted to be closest to agent η at time step $t+1$, divided by the predicted distance between the POI and agent η at time step $t+1$. This piece of information is what each agent explicitly shares via passive broadcast within each other in system. Each agent receives the set of all $s_{\eta,c}$ values $\{\mathbf{s}\}$. The next section discusses how this particular set of information is utilized by other agents.

4.3 IDEAS: Modified State

Each agent receives the set of $\{\mathbf{s}\}$ values (described in previous section) and uses them as a scaling factor I_i for their state information (I_i seen in Equation 11). We scale them such that if an agent is the closest to a POI, its value of that POI remains high (scaled by 1.0) and if there are many other rovers closer to the POI the POIs value is scaled by a number less than 1.0 represented by $s_{\eta,c}$. Here, an agent would interpolate each of the received set of $\{s\}$ values according to the following:

$$I_i = \begin{cases} 1 & \text{Rover is the closet to the POI} \\ s_{\eta,c} & \text{Rover is not the closest to the POI} \end{cases} \tag{11}$$

where the values I_i are scaling values between 0 and 1.0 that will be used to modify the agents current perceived state.

Each agent uses the information of I_i values to modify its perceived environmental state during time step t. Here, the state information would change as follows:

$$s_{1,q,\eta,t} = \sum_{i \in POI} \frac{I_i V_i}{\delta(L_i, L_{\eta,t})} \tag{12}$$

where $s_{1,q,\eta,t}$ is a modified version of agent η's system state information that incorporates the IDEAS from other agents (Equation 11), q is the quadrant, t is the time step, and i indexes the POI. Here, the state information is modified such that it reduces the observed value of POIs that are predicted to be heavily observed by agents and maintains high values for POIs that are not predicted to be observed heavily. The algorithm used to calculate each agents current IDEAS, communicate them to other agents, convert them into scalar values for scaling the state, and adjust the perceived system state can be found in Algorithm 2.

Now that we introduce agents' IDEAS and explain how they modify their state, we explain how we actually implement this in our simulation. In every simulation all agents receive state information from their sensors. Then they run their selected network (N_i) with their perceived states as inputs, to generate intended actions. Then, based on the intended actions agents update their state (S') to include the IDEAS which were explained in Section 4.2. Then their updated states are used to generate the action to take.

4.4 Policy Evaluations

One of the most important decisions that has to be made in neuroevolution is what policy evaluation each agent will use to evolve its set of policies. The first and most direct policy evaluation is to let each agent use the global system objective as the agent policy evaluation. However, in many domains, especially domains involving large numbers of agents, such a policy evaluation often leads to slow

Algorithm 2: The implementation of IDEAS.

```
Update the states ($S$) for all agents
for $t < t_{max}$ do
    1. Simulate using $N'$
    2. Observe Next State ($S'$)
    3. Extract IDEAS from $S'$
    4. $S \leftarrow$ Update $S$ (Equation 12)
```

evolution. This is because each agent has relatively little impact on its own evaluation. For instance if there were 100 agents and an agent takes an action that improves the system evaluation, it is likely that some of the 99 other agents will take poor actions at the same time, and the agent that took a good action will not be able to observe the benefit of its action. In this work, agents receiving the system performance G as its policy evaluation received values according to Equation 3.

In this work, we also utilize difference evaluations of the form [2, 19]:

$$D_\eta(z) \equiv G(z) - G(z - z_\eta) \tag{13}$$

where z_η is the action of agent η. All the components of z that are affected by agent η are removed from the system. Intuitively this causes the second term of the difference evaluation to evaluate the performance of the system without η and therefore D evaluates the agent's contribution to the system performance. Here, we derive D for the Cooperatively Coupled Rover Domain where the number of rovers required to form a team to observe a POI is $M = 3$. In this case, combining Equations 3 and 13 yields the equation for the difference policy evaluations used in this work:

$$D_\eta(z) = \sum_t \sum_{i \in POI} \sum_{m=1}^{M} \left\{ \frac{V_i}{\delta_{i,t}^m} - \frac{V_i}{\delta_{i,t}^{m \neq \eta}} \right\} \tag{14}$$

where D is the difference evaluation for agent η, M is the number of agents require to observe the i^{th} POI with the value of V_i, $\delta_{i,t}^m$ is the Euclidean distance from the m^{th} closest agent to the i^{th} POI, and $\delta_{i,t}^{m \neq \eta}$ is the Euclidean distance from the n^{th} closest agent to the i^{th} POI where agent m is not η. In this case D is non-zero in three key situations: 1) when the rover is the closest rover to the i^{th} POI, 2) when the rover is the second closet rover to the POI, and 3) when the rover is the third closest rover to POI. In order to compute D_η we need to remove the impact of agent η from the system and calculate G without it, by replacing agent η in the calculation with the fourth closest rover to the POI. Had agent η not been present, the fourth closest agent would have been part of the 3 agent team instead. Comparing the performance with and without the agent provides solid feedback in the agents contributions to the system.

5 Results

We now apply our approach in the CCRD (Section 3) with both static and dynamic POI's (Section 5.1 and 5.2), under various observation distances. These experiments require teams of 3 rovers to observe a given POI. Requiring team formation places significant coordination requirements on the agents within the system because agents must rely on each other in order to accomplish their goals. We test the effects of combining various agent evaluations, stigmergy, and IDEAS to determine how best to combine the benefits of these approaches. In these experiments, the minimum observation radius for POIs (the furthest away a rover from any team can be from a POI and still receive credit for observing it) changes, ranging from 10 to 100 units.

These experiments were run for 20 statistical runs, 5000 episodes (40 time steps per episode), in the world that contains 40 rovers and 66 POIs, which have a fixed location at all the time. Rovers are initially placed in between a high value POI (value of 10) and 65 low value POIs (value of 3) at 260 units from left and 50 units from the bottom.

5.1 Static Environment

The first set of experiments observes the performance of agents using implicit coordination via coupled evaluation functions as well as agents using combined coupled evaluations, stigmergy, and IDEAS information under varying observation restrictions (Figure 2). Intuitively, as the observation radius of agents increases, the overall system performance should increase (agents have access to more information). Initially, this performance increase is observed (Figure 2), where agents using only G and D respectively, improve performance as observability increases. However, as the observability continues to increase, agents using global evaluations begin to struggle due to too much information. Agents using global evaluations receiving too much information cannot distinguish the impact of their own actions on their evaluations versus the impact of other agents, making it difficult to learn. Agents using difference evaluations (D) on the other hand continue to perform well (outperforming other methods (R and G) by as much as 200%). This is because agents using D are able to effectively determine their own impact on the learning signal they received.

Although agents using implicit coordination via coupled policy evaluations were able to achieve good performance, we also tested the performance of agents using a combination of coordination mechanisms in the static CCRD (Figure 2). Here, agents used coupled policy evaluations (G and D, respectively), stigmergy, and the IDEAS coordination mechanism. In this case stigmergy introduces a change in the environment, the values of POIs decrease by 3% each time they are observed by a team of rovers. Experiments were also conducted to test the performance of coupled policy evaluations with stigmergy as well as coupled policy evaluations with IDEAS, however these results were excluded as they behaved statistically similar to agents using only coupled policy evaluations.

The results in Figure 2 show that combining implicit (coupled policy evaluations and stigmergy) and explicit (IDEAS) coordination mechanisms in the CCRD outperform other methods by as much as 25%. The reason that this combination works is because 1) agents' using difference policy evaluations receive a quality learning signal that promotes system-centric coordination 2) stigmergy provides an environmental cue to agents that emphasizes under observed POIs, and 3) IDEAS provide rovers with an effective "look ahead" of other rovers actions allowing them to act accordingly (agents may decide to pursue areas that are less heavily trafficked, or to attempt to form a team with another agents).

5.2 Dynamic Environment

Now that we have demonstrated the benefits of combining implicit and explicit coordination mechanisms, we want to extend this one step further by demonstrating the robustness of such an approach. In these experiments, agents in the CCRD (with 3 rover teams) have a dynamic environment, where the location and values of POIs change every episode. Dynamically changing the environment each learning episode increases learning difficulty and makes it harder for rovers to coordinate their policies. As seen in Figure 3, agents using global policy evaluations (both with and without additional coordination mechanisms such as stigmergy and IDEAS) perform approximately 50% better than they did in the static CCRD. This is because the agents are effectively learning policies that randomly wander and when the POIs are randomly placed throughout the environment every episode, so over a number of episodes they end up running across more POIs on average. However, they still perform poorly compared to agents using difference policy evaluations. This is because difference policy evaluations are able to account for the "randomness" of the environment and design agent policies that compensate for dynamic environmental factors. Agents using difference policy evaluations coupled with stigmergy and IDEAS perform agents using only difference policy evaluations by approximately 25% under different observation restrictions, and they outperform all other methods by as much as 60%.

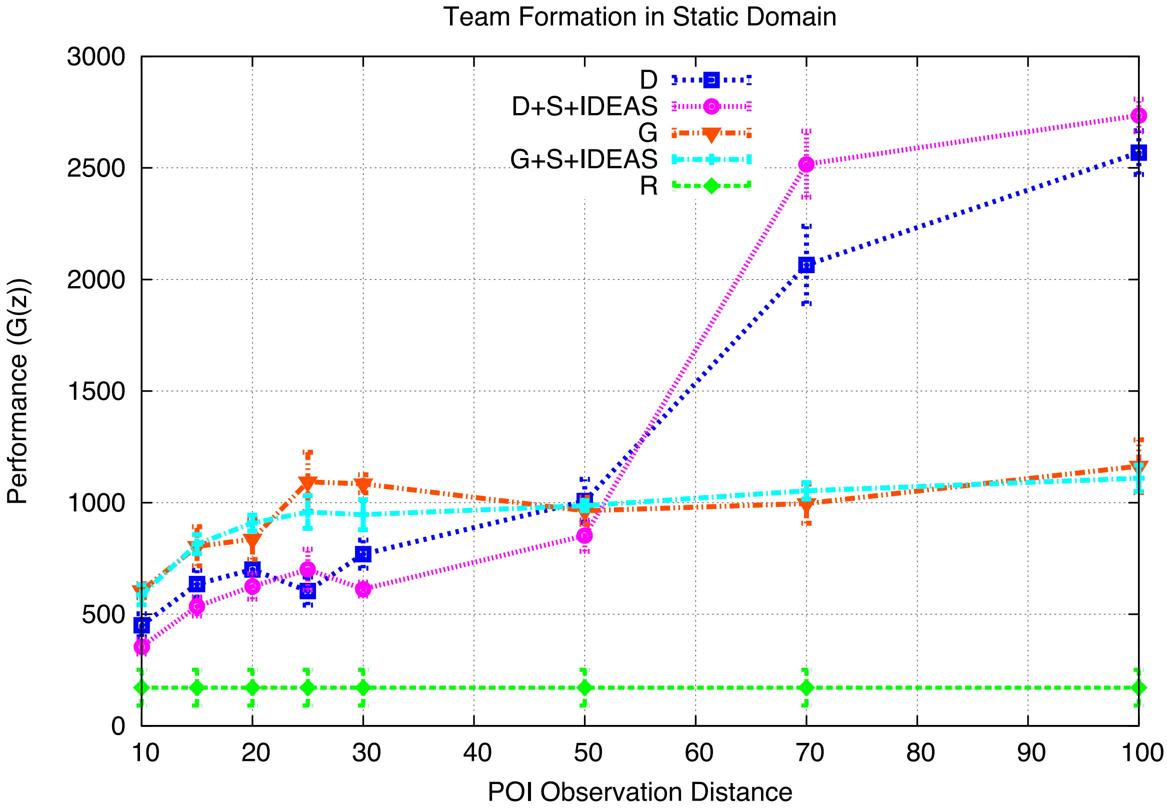

Figure 2: CCRD with 40 rovers in a static environment, where teams of 3 agents are required to observe each POI. Agents using $D + S + IDEAS$ outperform all other methods once the observation distance is above 60 units. Policy evaluations coupled with stigmergy (G+S,D+S) and IDEAS (G+IDEAS,D+IDEAS) were excluded from this figure because they were statistically similar to G and D (respectively).

6 Discussion

Coordinating the actions of disparate agents such that they collectively complete a complex task is a key problem that must be addressed in order to advance the field of multiagent systems. Although many implicit and explicit mechanisms for solving such coordination problems exist, they are frequently unable to fully address the coordination issues involved due to limited observation and communication restrictions. In order to improve performance of these methods, we selected implicit and explicit coordination mechanisms whose benefits were complementary under limited observation and communication restrictions. In particular, we utilized a combination of two implicit coordination mechanisms (coupled policy evaluations and stigmergy) and one novel explicit coordination mechanism (IDEAS) in the Cooperatively Coupled Rover Domain (CCRD) under limited observability. Overall, combining coupled evaluations, stigmergy, and IDEAS coordination mechanisms resulted in up to 25% improved performance over other approaches.

Combining the benefits of these coordination mechanisms enabled improved performance under varying observation restrictions because the mechanisms were complementary. Coupling policy evaluations enables agents to attempt to work together as a collective unit, what is good for an individual is good for the team. However, under limited observability, agents receive limited information and their policy evaluations become less reflective of the overall team performance and instead emphasize the performance in their local region. This is addressed by allowing agents to passively communicate their IDEAS (explicit coordination), which allows agents in different areas to implicitly “skip” information

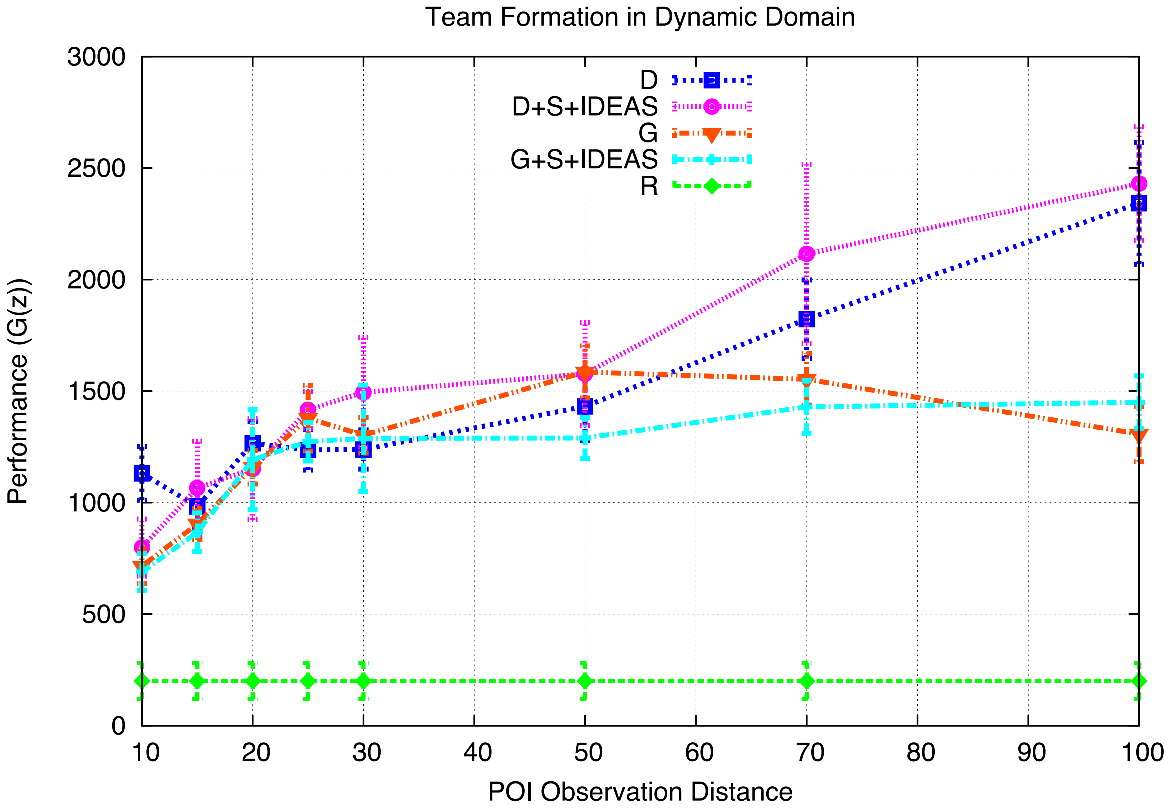

Figure 3: CCRD with 40 rovers in a dynamic environment, where teams of 3 agents are needed to observe each POI. As seen, agents using $D + S + IDEAS$ outperform all other methods once the observation distance is above 60 units.

across the system to each other, improving their ability to globally coordinate their actions (agents may effectively coordinate through interacting with other agents, though they may never interact directly). Finally, stigmergy provides an environmental que that impacts agents locally in a way that has global repercussions. As POI's were observed, their values decreased. This means that if a POI has been heavily observed in the past, although there are currently no rovers near it, as a new rover comes across it they will know to look elsewhere for a higher value POI, effectively encouraging the rovers to disperse and search other areas of the domain.

Although we used specific coordination mechanisms, there are undoubtably more combinations of implicit and explicit coordination mechanisms that will improve performance in many multiagent system domains. In general, implicit coordination mechanisms rely heavily upon agents' observation of the environment and tend to be limited by observation restrictions, while explicit coordination mechanisms rely heavily upon direct agent-to-agent information sharing and negotiation and are typically limited by communication restrictions. In most real-world multiagent system domains both observation and communication restrictions exist and when they do, a combination of implicit and explicit coordination mechanisms will likely be advantageous over either method individually.

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] A. K. Agogino and K. Tumer. Efficient evaluation functions for evolving coordination. *Evolutionary Computation*, 16(2):257–288, 2008.

[3] Hamid R. Arabnia, editor. *Proceedings of the 2006 International Conference on Artificial Intelligence, ICAI 2006, Las Vegas, Nevada, USA, June 26-29, 2006, Volume 1*. CSREA Press, 2006.

[4] D. Avrahami-Zilberbrand and G. Kaminka. Towards dynamic tracking of multi-agents teams: An initial report. *American Association for Artificial Intelligence (AAAI)*, 2007.

[5] P. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, pages 23–46, 2005.

[6] B. Gerkey and M. Mataric. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5), October 2002.

[7] Karuna Hadeli, Paul Valckenaers, Constantin B. Zamfirescu, Hendrik Van Brussel, Bart Saint Germain, Tom Holvoet, and Elke Steegmans. Self-organising in multi-agent coordination and control using stigmergy. In *Engineering Self-Organising Systems*, pages 105–123, 2003.

[8] B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. In *The Knowledge Engineering Review*. Cambridge University Press, 2005.

[9] M. Knudson and K. Tumer. Coevolution of heterogeneous multi-robot teams. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Portland, OR, July 2010.

[10] M. Knudson and K. Tumer. Robot coordination with ad-hoc team formation (extended abstract). In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1141–1142, Toronto, Canada, May 2010.

[11] M. Knudson and K. Tumer. Adaptive navigation for autonomous robots. *Robotics and Autonomous Systems*, 59:410–420, June 2011.

[12] P. Krysta, T. Michalak, T. Sandholm, and M. Wooldridge. Combinatorial auctions with externalities. *9th International Conference on Autonomous Agents and Multiagent Systems*, May 2010.

[13] N. Monekosso, P. Remagnino, and A. Szarowicz. An improved Q-learning algorithm using synthetic pheromones. *Lecture Notes in Computer Science*, 2296, 2001.

[14] R. Nair, M. Tambe, and S. Marsella. Role allocation and reallocation in multiagent teams: Towards a practical analysis. *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems*, 2003.

[15] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:2005, 2005.

[16] V. Robu, I. Vetsikas, E. Gerding, and N. Jennings. Flexibly priced options: A new mechanism for sequential auction markets with complementary goods (extended abstract). *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 1485–1486, May 2010.

[17] P. Stone, G. Kaminka, S. Kraus, and J. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. *American Association for Artificial Intelligence (AAAI)*, 2010.

[18] M. Tambe. Implementing agent teams in dynamic multi-agent environments. *Applied Artificial Intelligence*, 1997.

[19] K. Tumer and D. Wolpert, editors. *Collectives and the Design of Complex Systems*. Springer, New York, 2004.

[20] L. Vig and J. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4), August 2006.