Dirichlet-Multinomial Counterfactual Rewards for Heterogeneous Multiagent Systems*

Gaurav Dixit[1], Nicholas Zerbel[1], and Kagan Tumer[1]

Abstract— **Multi-robot teams have been shown to be effective in accomplishing complex tasks which require tight coordination among team members. In homogeneous systems, recent work has demonstrated that "stepping stone" rewards are an effective way to provide agents with feedback on potentially valuable actions even when the agent-to-agent coupling requirements of an objective are not satisfied. In this work, we propose a new mechanism for inferring hypothetical partners in tightly-coupled, heterogeneous systems called Dirichlet-Multinomial Counterfactual Selection (DMCS). Using DMCS, we show that agents can learn to infer appropriate counterfactual partners to receive more informative stepping stone rewards by testing in a modified multi-rover exploration problem. We also show that DMCS outperforms a random partner selection baseline by over 40%, and we demonstrate how domain knowledge can be used to induce a prior to guide the agent learning process. Finally, we show that DMCS maintains superior performance for up to 15 distinct rover types compared to the performance of the baseline which degrades rapidly.**

I. INTRODUCTION

Multi-robot (multiagent) coordination is a difficult control problem, particularly in tightly-coupled multiagent environments where agents must take complimentary actions at similar times in order to achieve an objective [1]. Despite the inherent difficulties with multi-robot coordination, it is possible to accomplish a wide variety of complex tasks using multi-robot teams. For example, multi-robot teams could be used effectively for tasks such as exploration [2], [3], air traffic control [4]–[7], and satellite orbital configuration [8].

Recent work introduced the idea of D_{++} structural credit assignment which provides agents with "stepping stone" rewards in tightly-coupled, homogeneous systems by allowing agents to infer hypothetical partners when needed [9], [10]. These stepping stone rewards guide agents to potentially valuable actions even if the agent-to-agent coupling for a given action has not been satisfied yet. However, with the default counterfactual partner selection method, agents are unable to select appropriate partners when operating in heterogeneous teams. This represents a serious coordination problem since many real-world multi-robot problems require team members to have different functions and capabilities.

In this work, we introduce Dirichlet-Multinomial Counterfactual Selection (DMCS) which uses Bayesian inference to effectively turn the problem of selecting appropriate counterfactual partners into a multi-armed bandit problem [11]. To generate effective stepping stone rewards, agents must learn to pick agents of types that can satisfy the coordination requirements of the environment. Thus, effective counterfactual selection depends on being able to model the environment. Assuming the underlying structure of the reward dynamics of the environment is distributed using some generating distribution, the prevalence of agent types can be modeled as a multinomial distribution. The posterior of choosing a counterfactual agent type needs to incorporate uncertainty to encourage exploration in terms of counterfactual selection. We can express this formulation with Bayesian inference that uses a Dirichlet conjugate prior with a multinomial distribution. Using DMCS, we show that D_{++} can be extended to heterogeneous, tightly-coupled multiagent systems.

To validate this approach, we apply D_{++} with DMCS to a heterogeneous version of the tightly-coupled multi-rover exploration problem. We show that, with DMCS, agents are able to learn which counterfactual partners should be sampled to provide them with the most effective stepping stone rewards. We show that agents using D_{++} with DMCS are able to learn to coordinate in a tightly-coupled, heterogeneous system while agents using D_{++} with standard selection fail to coordinate. We also show that DMCS outperforms random selection of partners by over 40%, and achieves superior performance for up to 15 distinct rover types compared to the performance of the baseline. Finally, we show that domain knowledge can be used in DMCS to induce prior beliefs to improve agent learning.

The contributions of this work are to:

- Introduce a Bayesian inference based selection mechanism for choosing hypothetical partners which improves learned coordination behaviors in heterogeneous, tightly-coupled systems;
- Reduce the difficulty in learning effective coordination strategies by incorporating prior domain knowledge into the selection mechanism.

The remainder of this work is organized as follows: Section II covers the necessary background including related work. Section III discusses the tightly-coupled, heterogeneous rover domain. Section IV formally introduces DMCS. In Section V we describe the setup of our experiments, and, in Section VI, we discuss the results of this work. Finally, in Section VII, we present our conclusions and identify possible areas for future work.

*This work was partially supported by the National Science Foundation Grant IIS-1815886, the Air Force Office of Scientific Research grant FA9550-19-1-0195, and the National Aeronautics and Space Administration Grant 80NSSC18K0941.

[1] The authors are with the Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, 1500 SW Jefferson Way, Corvallis, OR, USA [dixitg, zerbeln, kagan.tumer]@orgeonstate.edu

II. BACKGROUND AND RELATED WORK

In this section we cover the necessary background information which supports this work. We conclude this section with a discussion on related works.

A. Multiagent Reinforcement Learning

Multiagent reinforcement learning is challenging due to the inherent non-stationarity of the environment created by multiple agents learning simultaneously [12], [13]. Independent Q-learning agents often do not perform well and cannot learn tasks that require coordination. Some of the recent methods to address coordination and non-stationarity have focused on using an actor-critic architecture with a centralized critic [14]. Variants of these methods often use joint states and actions of all the agents in the system to learn coordination policies. These methods fail to scale to environments with a large number of agents and actions. The learning architectures are tightly coupled to the number of agents in the system making it difficult to learn and operate when agents might fail or new agents are introduced. The learning is also slow when the rewards are sparse and require tight coordination.

B. Reward Shaping

In cooperative multiagent systems, where agents need to learn coordinated behaviors, the design of agent reward functions has a significant impact on agent-to-agent interactions and the performance of the overall system [12], [13], [15]. The key to constructing good agent reward functions is to balance the feedback an agent receives (based on its individual actions) with the system-level coordination with other agents [13], [16]. Ideally, any increase in an individual agent's reward will simultaneously increase the overall performance of the system. This process of determining agent credit assignment is referred to as reward shaping [17], [18], and it is a powerful tool utilized in multiagent learning [19].

C. Difference Evaluations

Multiagent learning can be difficult due to the reward-signal noise generated from other agents learning simultaneously; however, learning distributed policies has been shown to improve team performance in loosely-coupled multiagent systems [7], [12], [20]–[23]. Learning coordination implicitly in this manner is made possible by shaping agent reward signals so that any increase in an individual agent's reward simultaneously increases the overall system score [2], [16], [24], [25]. Difference evaluations (Eqn. 1) are a shaped reward which provides agents with information relating its individual actions to the impact it made on the system [18], [24]. Specifically, difference evaluations remove much of the noise generated from multiple agents acting in a system by leveraging a counterfactual argument (often considered as a "null" action) and comparing what the current system score is with a hypothetical score. This "null" action is one example of a counterfactual utilized in difference evaluations; however, other actions may be used so long as the action is independent of the individual agent's selected action. With this setup, any improvement in an agent's difference reward simultaneously improves the overall performance of the system.

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

In Eqn. 1, z represents the current joint-state of the agents in the system, $D_i(z)$ is the difference reward received by agent i, $G(z)$ is the actual system score (or global reward), and $G(z_{-i} \cup c_i)$ is the global reward with a counterfactual action, c_i, taken in place of agent i's actual action.

Previous works have demonstrated that difference evaluations perform extremely well in loosely-coupled systems with homogeneous agents [6], [16], [17], [24]. However, difference evaluations struggle to achieve good system performance in tightly-coupled systems unless reward functions are specifically altered to encourage teaming [26]. One issue with using difference evaluation functions in tightly-coupled multiagent systems is that the rewards received by an agent are extremely sparse [9], [10]. For example, an agent may discover a potentially beneficial action to take; however, without other agents there to reinforce that action, the agent would not receive a reward. Even in loosely coupled multiagent systems, it can be difficult for an agent to "stumble" upon a good action. In a tightly-coupled system, multiple agents would need to "stumble" upon a good action at the same time which is statistically unlikely to happen within a viable timeframe [9], [10].

D. D_{++} Evaluation

Although difference evaluations allow for cooperative learning implicitly, it is generally ineffective in tightly-coupled systems due to reward sparsity. To contend with the reward sparsity, D_{++} structural credit assignment (introduced by Rahmattalabi *et al* [9], [10]) introduces the concept of "stepping stone" rewards which help guide agents to potentially beneficial actions even when a team member is not yet present to take a complimentary action. Instead of inferring a counterfactual action taken by agent i, D_{++} evaluations introduce counterfactual partners which are copies of the current agent inferred as partners. We further define D_{++} evaluations below

$$D_{++}(i,n) = \frac{G(z_{+(\cup_{i=1,\ldots,n})}) - G(z)}{n} \tag{2}$$

where n represents the number of counterfactual partners added, $D_{++}(i,n)$ represents the reward received by agent i with n counterfactual partners, and $G(z_{+(\cup_{i=1,\ldots,n})})$ represents what the global score would be with if there were n agents taking supporting the action. Dividing by the n term in Eqn. 2 normalizes the reward with respect to the number of added counterfactual partners.

E. Related Work

To address the challenge of multi-robot coordination in multiagent systems, some researchers utilize direct communication between team members to develop coorindation behaviors [3], [27], [28]. Sometimes, that communication is centralized as seen in *Pusher-Watcher* (by Gerkey and Matarić [1]) which utilizes a centralized, auction-based system to allocate agents to specific tasks. In other examples, the communication is distributed and is managed using communications protocols as is the case in many approaches to RoboCup robotic soccer teams [27]. In both of these examples, communication is explicit and relies on a communications channel strong enough to transmit data. There are also several works which deal with communications in tightly-coupled multiagent systems and heterogeneous multiagent systems [3], [27], [28]. Although this research shows that communication, when handled properly, is extremely effective in improving coordination among team members, communication is often expensive to implement and extremely limited by the environment [26]. For example, in an exploration domain it is likely that rovers will travel to remote regions in order to explore where high-fidelity communication cannot be established with the rest of the team [3]. This may lead to sub-optimal behaviors as robots explore areas already explored by another robot.

Other works focus on implicitly learning distributed agent policies [12], [20]–[22] to address the multi-robot coordination challenge. For example, the work by Pagello *et al* [28] demonstrates an approach to robotic soccer using implicit, stigmergic communications in which robots infer actions based on observing the actions of their teammates. In this work, we introduce a new method for selecting counterfactual partners which allows agents to access more informative stepping stone rewards.

III. ROVER COORDINATION PROBLEM

In this work, we investigate the performance of DMCS in a tightly-coupled, multi-robot exploration domain known as the rover domain. In the classic example of the rover domain, a team of rovers on Mars are tasked with exploring points of interest (POI) spread out across a two-dimensional, continuous space. Each POI has a different observational value associated with it, and rovers only receive rewards for observations which are made within a certain radius of the POI, R_{obs}. Only the closest observation of a POI is counted towards the overall system; therefore, even if multiple rovers are observing the same POI, only the closest of those observations is counted. For that reason, an optimal strategy in the classic rover domain is for rovers to disperse and explore different regions of the map. Additionally, each rover is equipped with two different sensors: one sensor detects POIs on the map, and the second sensor detects other rovers on the map.

A. Tightly-Coupled Rover Domain

The tightly-coupled version of the rover domain modifies the domain mechanics to make it a requirement that POIs be observed by multiple rovers at the same time in order for rewards to be given. Similar to the classic rover-domain, only the closest observations of a POI count towards the system goal. Based on the required number of observations, m, the observations of the m closest rovers are counted towards the system score. We can express the system reward function as given in Eqn. 3.

$$G(z) = \sum_i \sum_j \sum_k \frac{V_i N^1_{i,j} N^2_{i,k}}{0.5(\delta_{i,j} + \delta_{i,k})} \tag{3}$$

In Eqn. 3, V_i represents the value associated with POI i, $\delta_{i,j}$ is the distance between POI i and the closest rover j, $\delta_{i,k}$ is the distance between POI i and the second closest rover k, and both N values are constants which denote whether or not the coupling requirement is satisfied. In other words $N^1_{i,j}$ and $N^2_{i,k}$ are determined by equations 4 and 5, respectively.

$$N^1_{i,j} = \begin{cases} 1, & \text{if } \delta_{i,j} \leq R_{obs} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$N^2_{i,k} = \begin{cases} 1, & \text{if } \delta_{i,k} \leq R_{obs} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

B. Hetereogeneous Multi-Rover Problem

In this work, we modify the tightly-coupled rover problem further by introducing heterogeneous rover teams. These rovers follow the same set of rules in terms of movements and observation distances. Additionally, the rovers' sensors are density based, and they detect the density of POIs and other rovers per quadrant relative to the sensing rover's position. The POI and rover density sensors are described further by equations 6 and 7, respectively.

$$S^D_{POI} = \frac{1}{N_{i,q}} \sum_{i \in q} \frac{V_i}{\delta_{i,j}} \tag{6}$$

$$S^D_{Rover} = \frac{1}{N_{k,q}} \sum_{k \in q} \frac{1}{\delta_{j,k}} \tag{7}$$

In Eqn. 6, q represents the quadrant being scanned, i designates the POI located in quadrant q, V_i is the value of POI i, and $\delta_{i,j}$ is the distance between the current rover, j, and POI i. In Eqn. 7, k represents other rovers in q, and $\delta_{j,k}$ represents the distance between the rover, j, and a different rover, k. Furthermore, $N_{i,q}$ and $N_{k,q}$ represent the number of POIs and rovers detected in quadrant q, respectively.

In addition to multiple rover types, the POI also have different criteria which must be satisfied in order for them to be considered observed. This point is illustrated further by Fig. 1 where there are three rover types present: circle, star, and diamond. Each POI, represented by a triangle, must be observed by a rover of each type as indicated. This figure demonstrates the importance of selecting the appropriate counterfactual partners so that an agent receives an accurate stepping stone reward from D_{++}.

IV. DIRICHLET-MULTINOMIAL COUNTERFACTUALS

In this section we formalize DMCS and discuss the multiagent reinforcement learning algorithm used in the multi-rover exploration domain.

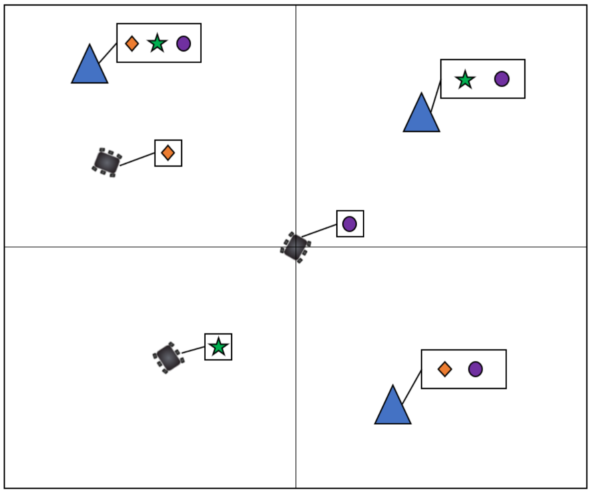

Fig. 1. Heterogeneous rover domain with three exploratory rovers and three POIs (represented as triangles). Each rover is of a different type (diamond, circle, or star), and each POI must be observed by rovers of certain types as indicated.

A. *Dirichlet-Multinomial Counterfactual Selection*

To extend D_{++} to heterogeneous multiagent systems, we introduce a new method for choosing counterfactual partners called Dirichlet-Multinomial Counterfactual Selection (DMCS). Agents using D_{++} with DMCS, choose counterfactual partners from a learned posterior distribution over the prevalence of agent types. The underlying model is a multinomial distribution with parameters p_k over k agent types. The parameter vector for this multinomial is drawn from a Dirichlet distribution, which forms the conjugate prior distribution for multinomial likelihood. The choice of Dirichlet distribution follows from the rover domain assumptions, but inference using other suitable distributions would also lead to similar results. The concentration hyperparameter vector α induces a prior in the Dirichlet distribution. This prior can come from domain knowledge and represents the prior belief in gain from selecting a particular type of agent as a counterfactual. A single point estimate (for instance, the mean of the distribution), helps agents pick a type as a counterfactual partner. This can be expressed analytically as:

$$E[P_i|\tau,\alpha] = \frac{c_i + \alpha_i}{n + \sum_k \alpha_k} \tag{8}$$

In Eqn. 8, c_i is the number of counterfactual agents of each type that generated a stepping stone reward, n is the total number of counterfactual agents sampled so far and α represents the prior belief. This approach does not account for the uncertainty associated due to the limited amount of exploration and observed evidence. A Bayesian approach however, will help model this uncertainly by generating a posterior distribution and simultaneously provide point estimates for greedy selection. The goal is to estimate the posterior distribution for the probability of selecting each type, p, conditioned on the observations τ and prior belief vector α, given by $(P|\tau,\alpha)$.

We start by building the model using prior belief α and observations τ, and then use it to sample from the posterior to approximate the posterior with Markov Chain Monte Carlo (MCMC) methods. We use MCMC methods because exact inference is tractable only for conjugate distributions. In particular, we use the No-U-Turn Sampler (NUTS) [29] because it can reach a good estimate of the posterior with only a few initial observations. A trace is generated by drawing 1000 samples from the posterior in three chains (with discarding). As the number of samples increases, the estimated posterior converges to the true posterior. After sufficient observations, the posterior's credible interval will be small enough to correctly model the underlying structure of the POI constraints in the environment.

This approach effectively turns counterfactual sampling into a multi-armed bandit problem [11] where the agent will be more likely to sample certain counterfactual partners based on expected stepping stone rewards. Although this approach is not guaranteed to always choose the best counterfactual partners for a given POI like a point estimate based approach would, its associated uncertainly is necessary for exploration and dealing with mixed POI constraints.

B. *D_{++} with DMCS*

The overall structure of D_{++} (presented by Rahmattalabi *et al* [9], [10]) makes it possible to add in new methods of counterfactual selection without changing the overall functionality of the algorithm. We demonstrate this point further in Algorithm 1 where $D_i(z)$ is the difference reward for agent i, $D_{++}(n)$ is used to express D_{++} being calculated with n number of counterfactuals, and N_A refers to the total number of agents required by coupling. With this setup, agents are still able to leverage difference evaluations when the difference reward is informative; however, counterfactual partner selection is now handled using DMCS (Step 9) when agents estimate stepping stone rewards using D_{++}.

Algorithm 1: D_{++} with DMCS

```
 1: Calculate $D_i(z)$ using Eqn. 1
 2: Calculate $D_{++}(N_A - 1)$ using Eqn. 2
 3: if $D_{++}(N_A - 1) \leq D_i(z)$ then
 4:    return $D_i(z)$
 5: else
 6:    $n = 0$
 7:    while $n < N_A - 1$ do
 8:       $n = n + 1$
 9:       Sample $n$ counterfactuals from the current
          estimated posterior
10:       Calculate $D_{++}(n)$ using Eqn. 2
11:       if $D_{++}(n) > D_{++}(n - 1)$ then
12:          Return $D_{++}(n)$
13:       end if
14:    end while
15: end if
16: Return $D_i(z)$
```

V. Experimental Setup

We carry out experiments in the heterogeneous, tightly-coupled rover domain to investigate the performance of DMCS and the effect of prior beliefs. Each agent is a fully connected feed-forward neural network that maps the state vector $(4+4n)$ for n agent types to two action values, corresponding to (x, y) that lie in $[0, 1]$. The weights are updated using policy gradient methods. The environment is 30 x 30 units, with 18 POIs and 10 rovers. In the first state, s_0, of every episode, the POIs and the agents are randomly placed within the environment. For every agent, a type is picked from a uniform distribution $\mathcal{U}(0, t)$. The observation radius is bound between $[2, 5]$. In the first experiment, the POIs have a coupling requirement of three. Agent types that can successfully observe a POI to get a reward are sampled from a multinomial.

Random sampling from a uniform distribution serves as a type aware baseline to compare against DMCS. Agents sample counterfactual partners randomly from among the types of agents present in the system. It has an advantage over D_{++} because it allows agents to sample partners from types other than its own. However, random selection does not scale well as more agent types are introduced into the system. Finally, in a system with just one type, this essentially collapses to homogeneous D_{++} using standard counterfactual selection.

VI. Results and Discussion

In this section, we compare the performance of D_{++} using DMCS against D_{++} using uniform, random partner selection (baseline), D_{++} using standard partner selection, and learning with global reward feedback in the tightly-coupled multi-rover exploration problem. First, we examine the behavior of DMCS in a homogeneous multi-rover system to verify that altering the selection mechanism for counterfactual partners does not change the functionality of D_{++}. We then demonstrate the performance of DMCS in the heterogeneous multi-rover system as well as the influence of prior belief on the learning process. Finally, we test the scalability of DMCS for increasing numbers of rover types.

A. Homogeneous Multi-Rover System

To calibrate DMCS, we compare D_{++} with standard counterfactual selection against D_{++} using DMCS, D_{++} using uniform, random sampling (baseline), as well as global reward as feedback for agent learning in a homogeneous system. This system is identical to the tightly-coupled rover problem used in the work by Rahmattalabi *et al* [9]. Figure 2 shows overall system score achieved by the rover teams per episode of learning. The results show that, in a homogeneous system, DMCS and uniform sampling achieve the same level of performance as D_{++} with standard counterfactual selection. This is consistent with expectations since all rovers are of the same type in a homogeneous system, and the type of partner selected is irrelevant to receiving an accurate stepping stone reward.

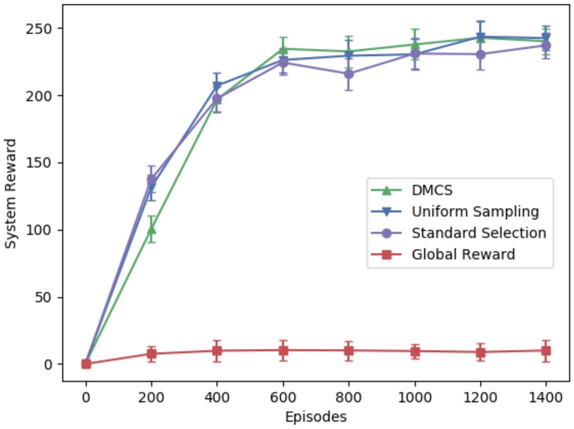

Fig. 2. Homogeneous agents: System reward earned by rover teams per episode. There are 18 rovers and 10 POIs on a 30x30 world. Each POI requires observations from 3 rovers. As expected, all selection mechanisms perform equally well.

B. Heterogeneous Multi-Rover System

To demonstrate the performance of DMCS in heterogeneous systems, we compare DMCS with uniform sampling, D_{++} with standard counterfactual selection, and using the global reward for agent learning in a 30x30 world with 18 rovers, 10 POIs, and three rover types. We assume no domain knowledge and start with a uniform prior for this experiment. These results, which are representative of the results in different sizes of rover systems, are illustrated by Fig. 3. As indicated by the results, uniform sampling can produce decent coordination policies after some exploration. However, DMCS learns faster than the baseline because a few observations of stepping stone reward generating counterfactuals are enough to start capturing the underlying POI constraint distribution. This results in more insightful rewards than uniform sampling where agents are likely to make poor choices for counterfactual partners. Agents using DMCS also develop superior coordination strategies as indicated by the improvement in system rewards earned. Agents using both the global reward $G(z)$ and D_{++} with standard counterfactual selection fail to learn coordinated behaviors due to the sparsity in rewards received.

To demonstrate the scalability of DMCS, we performed additional tests with differing numbers of rover types. The results of these tests are presented in Fig. 4. With a more diverse population, DMCS performs significantly better than both the baseline and standard counterfactual selection. As the number of rover types increases, the performance of uniform random sampling degrades as it becomes more likely that agents will select partners which do not provide them with informative stepping stone rewards. The results also indicate that the performance of standard counterfactual selection degrades, almost to 0.0, after nine rover types.

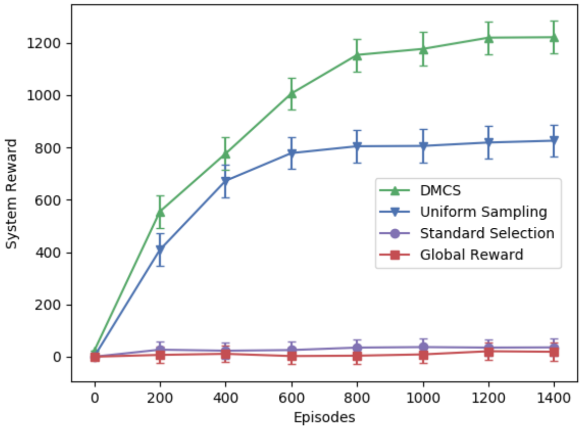

Fig. 3. Heterogeneous agents: System reward earned by rover teams per episode. There are 18 rovers and 10 POIs on a 30x30 world. Observation constraints for each POI is generated by randomly sampling from a distribution over agent types.

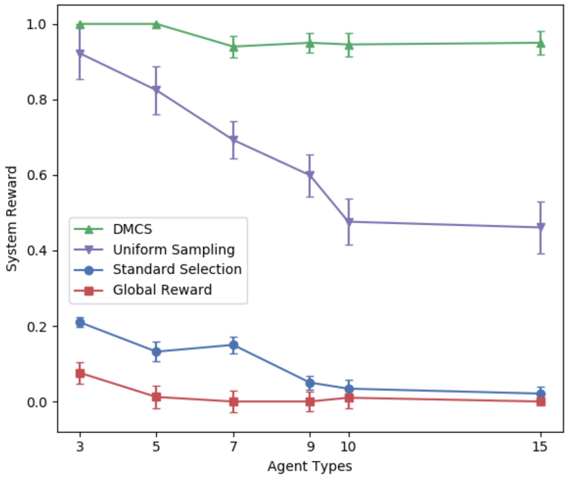

Fig. 4. Scalability of DMCS: Normalized system reward earned by rover teams of increasing heterogeneity. There are 27 rovers and 16 POIs in a 30x30 world. Observation constraint for each POI is generated by randomly sampling from a distribution over agent types.

C. Influence of Prior Belief on Learning

Figure 5 captures the effect of prior belief on the learned posterior. The choice of the prior hyperparameter α depends on the confidence in an agent's belief. For large values of α, the prior will weigh in more than the observed evidence and agents will need to observe the generated stepping stone reward for many counterfactuals to overcome this prior. For smaller values of α, the evidence has a stronger influence on the distribution and will quickly remove the bias introduced by the prior. The values for α can themselves come from a distribution. For example, if the rovers have explored similar terrain before, the learned posterior can be used as the prior for the next terrain exploration task.

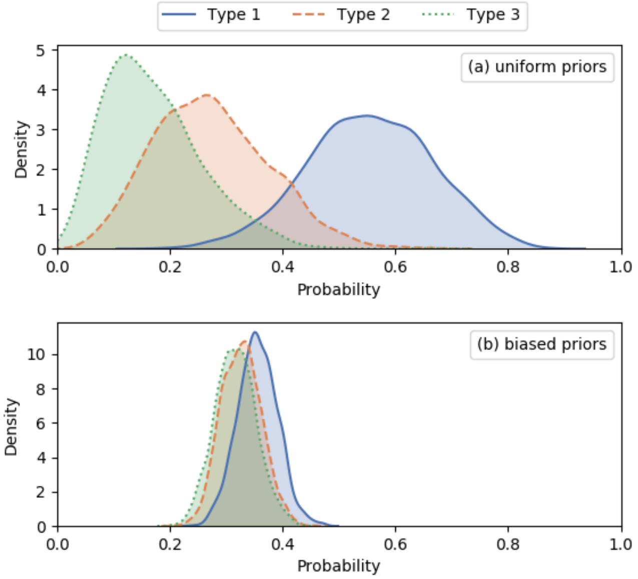

Fig. 5. Influence of prior belief on the learned posteriors for a system with 3 agent types. Sub-figure (a) shows learned posteriors after starting with a uniform prior where all types are equally likely to be selected as partners. The Agent learns the true posteriors after sufficient observations of partner selection. Sub-figure (b) shows learned posteriors after starting with a strong, incorrect prior. The posteriors do not converge to the true value and the probability of choosing an agent of any type is close to 0.33 which implies agents did not learn which partner should be selected.

VII. Conclusions

In this work, we introduced a new method of selecting counterfactual partners, DMCS, which allows agents in tightly-coupled, heterogeneous systems to utilize D_{++} to learn which counterfactual partners should be selected to produce more informative stepping stone rewards. Using a learned posterior distribution over agent types, we see that agents using D_{++} with DMCS were able to efficiently choose counterfactual partners; whereas, agents using D_{++} with standard selection were unable to learn in the heterogeneous system. We also see that DMCS results in over 40% improvement in system performance over the random partner selection baseline, and we see how domain knowledge can be used to induce a prior to guide agents in the learning process. Finally, the results demonstrate the impressive scalability of DMCS which maintained superior performance as the number of rover types in the system was increased (up to 15 rover types) while the performance of uniform sampling degraded rapidly.

DMCS assumes that agents have prior knowledge about other agents which are available to team up with. In future work, we will explore how DMCS may be used if this assumption is not true. Additionally, we will explore how DMCS performs in problems where diverse agents with different action spaces must coordinate such as search and rescue problems where robots partner with non-robot partners or multi-robot exploration problems involving coordination between land, water, and aerial vehicles.

References

[1] B. P. Gerkey and M. J. Mataric, "Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 464–469.

[2] A. Agogino and K. Tumer, "Efficient evaluation functions for multi-rover systems," in *Genetic and Evolutionary Computation Conference*, Springer. Springer, 2004, pp. 1–11.

[3] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[4] "Implicit adaptive multi-robot coordination in dynamic environments," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 5168–5173, 2015.

[5] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on automatic control*, vol. 43, no. 4, pp. 509–521, 1998.

[6] K. Tumer and A. Agogino, "Distributed agent-based air traffic flow management," *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07*, p. 1, 2007.

[7] K. Tumer and A. K. Agogino, "Improving air traffic management with a learning multiagent system," *Intelligent Systems*, vol. 24, no. 1, Jan/Feb 2009.

[8] S. Damiani, G. Verfaillie, and M.-C. Charmeau, "An earth watching satellite constellation: How to manage a team of watching agents with limited communications," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. ACM, 2005, pp. 455–462.

[9] A. Rahmattalabi, J. J. Chung, M. Colby, and K. Tumer, "D++: Structural credit assignment in tightly coupled multiagent domains," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, vol. 2016-Novem. IEEE, 2016, pp. 4424–4429.

[10] A. Rahmattalabi, "D++: Structural Credit Assignment in Tightly Coupled Multiagent Domains," Master's Thesis, Oregon State University, 2017.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2012.

[12] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[13] K. Tuyls and K. Tumer, "Multiagent Learning," in *Multiagent Systems*, 2nd ed., G. Weiss, Ed. London, England: The MIT Press, 2016, ch. 10, pp. 423–484.

[14] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *CoRR*, vol. abs/1706.02275, 2017. [Online]. Available: http://arxiv.org/abs/1706.02275

[15] P. J. Hoen, K. Tuyls, L. Panait, S. Luke, and J. La Poutré, "An Overview of Cooperative and Competitive Multiagent Learning," *Learning and Adaption in Multi-Agent Systems*, pp. 1–50, 2005.

[16] A. K. Agogino and K. Tumer, "Analyzing and visualizing multiagent rewards in dynamic and stochastic domains," vol. 17, no. 2, pp. 320–338, 2008.

[17] M. Colby and K. Tumer, "Shaping Fitness Functions for Coevolving Cooperative Multiagent Systems," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, no. June. Valencia, Spain: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 4–8.

[18] L. Yliniemi and K. Tumer, "Multi-objective multiagent credit assignment through difference rewards in reinforcement learning," in *Asia-Pacific Conference on Simulated Evolution and Learning*, Springer. Springer, 2014, pp. 407–418.

[19] S. Devlin, M. Grześ, and D. Kudenko, "An Empirical Study of Potential-Based Reward Shaping and Advice in Complex, Multi-Agent Systems," *Advances in Complex Systems*, vol. 14, no. 02, pp. 251–278, 2011.

[20] M. Bowling and M. Veloso, "Simultaneous adversarial multi-robot learning," in *IJCAI*, vol. 3, 2003, pp. 699–704.

[21] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous agents and multi-agent systems*, vol. 11, no. 3, pp. 387–434, 2005.

[22] K. Tuyls, P. J. Hoen, and B. Vanschoenwinkel, "An evolutionary dynamical analysis of multi-agent learning in iterated games," *Autonomous Agents and Multi-Agent Systems*, vol. 12, no. 1, pp. 115–153, 2006.

[23] A. Iscen, A. Agogino, V. SunSpiral, and K. Tumer, "Controlling tensegrity robots through evolution," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 1293–1300.

[24] A. Agogino and K. Tumer, "Efficient evaluation functions for evolving coordination," *Evolutionary Computation*, vol. 16, no. 2, pp. 257–288, 2008.

[25] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, vol. 99, 1999, pp. 278–287.

[26] M. Knudson and K. Tumer, "Coevolution of heterogeneous multi-robot teams," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 127–134.

[27] L. Iocchi, D. Nardi, M. Piaggio, and A. Sgorbissa, "Distributed coordination in heterogeneous multi-robot systems," *Autonomous Robots*, vol. 15, no. 2, pp. 155–168, 2003.

[28] E. Pagello, A. DAngelo, F. Montesello, F. Garelli, and C. Ferrari, "Cooperative behaviors in multi-robot systems through implicit communication," *Robotics and Autonomous Systems*, vol. 29, no. 1, pp. 65–77, 1999.

[29] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo," 2011.