Distortion of Agent States For Improved Coordination

By
Connor Yates

A THESIS

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Associate)

Presented May 23^{rd}, 2017
Commencement June 2017

AN ABSTRACT OF THE THESIS OF

Connor Yates for the degree of Honors Baccalaureate of Science in Computer Science presented on May 23^{rd}, 2017. Title: Distortion of Agent States For Improved Coordination

Abstract approved:

Dr. Kagan Tumer

Many real world problems have partial solutions or intermediary steps which can lead toward solving the problem. When assembling robotic teams to solve these problems, we have intuition about which intermediary steps are more useful than others. We examine methods to identify and apply our designer intuition onto tightly coupled multiagent problems. In a method analogous to potential based reward shaping, we shape the perceived value of points of interest (POI) in a ground-rover observation problem based on the potential for further team coordination if the observing agent goes to observe that POI. These state distortion methods utilize information from the current world, and as such are constructed independent of the learning method used. Methods for direct state shaping from Nasroullahi [1], which are based on future prediction about other agents' actions, are extended into the future. From this initial work, we were inspired to create new methods that which readily scale to POI problems of arbitrary coupling dimension. These new methods show a no performance degradation in less coupled domains, and sustained operational capacity in more difficult, tightly coupled problems where traditional methods break down. This field of direct state distortion for increased cooperation and performance is relatively unexplored, and we finally lay out future directions of this area of work.

Key Words: multiagent learning, state-shaping methods, tightly coupled problems

Corresponding e-mail address: yatesco@oregonstate.edu

Distortion of Agent States For Improved Coordination

By
Connor Yates

A THESIS

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Associate)

Presented May 23^{rd}, 2017
Commencement June 2017

Honors Baccalaureate of Science in Computer Science project of Connor Yates presented on May 23^{rd}, 2017

APPROVED:

Dr. Kagan Tumer, Mentor, representing Robotics

Dr. Geoff Hollinger, Committee Member, representing Robotics

Dr. Jen Jen Chung, Committee Member, representing Robotics

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University Honors College. My signature below authorizes release of my project to any reader upon request.

Connor Yates, Author

Contents

1 INTRODUCTION

Many problems in the real world benefit from multiple robots working as a team. General exploration, search and rescue, and rearranging furniture in rooms are all tasks which benefit from multiple robots working together to accomplish the final goal. These tasks are different, however, in that they do not require the same level of coordination between the agents. Mapping a room can be done by a single agent, but can be done more quickly if multiple robots are working together. These types of task exhibit *loose coupling*, where more agents are beneficial but not necessary. Alternatively, moving a large piece of furniture cannot be done by a single small robot. Multiple robots are required to work together to try to lift and move a large couch at once. This exhibits *tight coupling* where the robot's success is tightly bound to the actions of other robots, as well as their own action. Additionally, tightly coupled problems can vary in difficulty. A chair may need two robots to move, whereas a large couch could require three or four.

When agents learn to solve tightly coupled problems, current methods of learning via reward shaping can be ineffective on problems with a high degree of coupling. The global reward, which measures the success of an entire team, will reward agents who can work together to accomplish a shared goal. A team of robots whose task is to clean a large floor would receive a large reward if they clean a large section of the floor. In multiagent reinforcement learning, the agents would receive the same global reward. If one robot cleaned the entire floor while the rest spun in place, the entire team would be rewarded equally. Obviously, only one of the policies in this team was useful, but the global reward does not capture this information. To account for this, the difference reward [2], which measures my contribution to the team's global reward, will reward agents who are large contributors to solving a problem. By removing oneself from the world, an agent can see how much impact it had upon the team. In the above example, when the robots which spin around needlessly are removed, they see no effective change in team performance. These agents would receive no reward from the difference reward. The agent which did the work would receive all the reward, and would thus be able to learn much more quickly that what it is doing is correct.

The global reward (G), and the difference reward (D) can be used in tightly coupled problems, but their effectiveness is lessened as the degree of coupling increases. These tasks depend upon all the agents taking the correct joint action at the same time. The floor cleaning robots, with no interdependence on their team's success, can learn what good actions to take without worrying about cooperation from others. Most importantly, while agents using the difference reward can identify when their actions help the team, they do not help identify the proper action when the reward is dependent upon the joint action of the team.

The entire joint-action space for a team of agents is exponentially large, and quickly becomes infeasible to directly search across. To address this deficiency, two

general approaches have been used. Both methods attempt to produce a smoother signal for the agent to learn from. The first is to change the reward signal sent to the agents, to not rely on the piecewise reward signal presented by D [3]. The second method is to create a richer area for the agent to take actions on. To elaborate, consider any general objection localisation problem, such as search and rescue or scientific point-of-interest (POI) observation. Our basic agent perceives a POI as a binary "thing here" or "no thing here." This provides a minimal working view of the world. We can modify observations to have more information about the usefulness of a POI given an observing agent. In searching and observing a POI, "thing here which is close by to me" and "thing here which is far away from me" is more useful, because the agent can make a decision to pursue the close by POI, since it will most likely reach it more quickly, expending less energy.

Continuing with that idea, we can phrase a POI in terms of the group of agents as a whole. This idea now takes into account how other agents in the system are needed to work together simultaneously on a POI. This is useful, and necessary, in tightly coupled tasks where success is dependent on others. Now, a POI described as "thing here, close to me, with others close by" is much more attractive than "thing here, close to me, but everyone else is far away". In the latter scenario, we now have the knowledge to be able to start looking for other POI's to pursue, since the agent might not be likely to get additional help from other agents. Additionally, the POI described as "thing here, a little ways away from me, and everyone else is very close" should be a more attractive option; an agent which can complete a team to observe a POI will benefit from viewing these POI in a more attractive light.

By viewing the value of a POI as a function of other's potential ability to observe it, we can introduce the notion of intent into the problem. Agents intent can be incorporated into the state by changing the value of a POI based on the perceived ability for other agents to observe the POI. This work was first approached by Nasroullahi [1], where the value of a POI in the state was directly modified based on a simple formula, making a POI where the agent is likely to have an impact seem larger (and more attractive) than other POI's. This approach was domain specific, and relied on knowing the degree of coupling for the present problem in order to calculate a modification of the POI value.

This thesis uses this initial idea of modifying the state from Nasroullahi [1] as a starting point for the questions: What are alternative methods that would be more generalizable to the degree of coupling, and less domain specific, and how do they perform as the size of agent teams increase?

The rest of this paper is structured as follows. Background information on the learning methods and tightly coupled domains are presented in Section 2. A discussion on potential based reward shaping follows, along with comparisons to the first examples of direct state shaping for improvement in tightly coupled domains. The background information in Section 2 then concludes with an overview of intent in formal logic and AI problems. Section 3 covers the formulae introduced in this work

and their intentions. Sections 4 and 5 cover the experimental domains and direct results. Section 6 discusses these results in the context of the previous work and the idea of intent in multiagent systems. Section 7 concludes the paper with a discussion on the future avenue of research.

2 Background

There are four main sections of background theory this work uses and builds upon. Neural networks and evolutionary learning are utilized as the decision making and learning methods. More importantly, the idea of passively estimating intent of other agents uses theory from potential based reward shaping and human-intent research. This is examined in the context of previous state shaping methods employed by Nasroullahi [1].

2.1 Reward Structures

This work utilizes two main types of rewards in the evolutionary search: global rewards ($G(z)$) and difference rewards ($D(z)$). Global rewards measure the total system performance, whereas difference rewards measure the impact of the individual agent on the system [4]. These are functions that operate on the joint state of the world, z, as seen at a given time step. When talking about the difference reward D, the notation z_i will be used to denote the state of the world from the perspective of agent i, which allows the agent to analyze their own impact upon the world. Recent work into reward structures has developed new methods for tightly coupled domains [3], showing improvement over both D and G. The problem domain examined in this work is tightly coupled, but at a low degree of coupling, allowing adequate analysis to take place with D.

The global reward (G) aims to capture the overall performance of a team of agents. The domain specific implementation of G used in this work is shown in Equation 1.

$$G(z) = (\sum_{p \in POI} value(p) \times observed(p)) - \sum_{a \in agents} stepcost(a) \tag{1}$$

The difference rewards are generally calculated with $D(z_i) = G(z) - G(z \setminus a_i)$ [4]. Since we are looking at the impacts of the agent over the full time step series, any POI's the agent helped to return are removed in the $G(z \setminus a_i)$ term. The functional

form of D in this domain becomes

$$\begin{aligned} D(z_i) &= G(z) - G(z \setminus a_i) \\ &= G(z) - (G(z) - \sum_{p \in POI} value(p) \times observedby(p, a_i)) \\ &= \sum_{p \in POI} value(p) \times observedby(p, a_i) \end{aligned}$$

where *observedby* is a binary function returning 1 if agent a_i observed POI p, and 0 if not. Of important note, the novel state shaping methods presented in Section 3 do not depend upon a specific reward structure. Since the method impacts the state and not the rewards structures, this method can be used with any multiagent reward structure.

2.2 Neural Networks and Evolutionary Learning

In this work we use neural networks as control modules that take in the state of the world the agent sees, and outputs a single action. These controlling functions undergo evolutionary learning. Control policies are run in a world, and then rated with a reward signal. Typically, the global reward (G) or the difference reward (D) are used. We use D as the baseline for comparison, as it is the superior reward signal. After the trial is run and the neural networks are associated with a reward, a simple version of cooperative co-evolutionary algorithms (CCEA) is employed to select the best networks. The networks are sorted, and the lowest performing are removed from the pool. Then, the remaining are cloned and randomly mutated to bring the population back up to level. This method selects for the best performing agents. Thus, if the reward signal is adequate, this searches for the optimal policy to solve a given problem domain.

2.3 Potential-based reward shaping

Potential based reward shaping is a practice in reinforcement learning which incorporates heuristic-based knowledge of the world into the reward signal sent to the agent [5]. In this manner, the additional information can be incorporated via designer knowledge [6] or learned through interactions with the system [7]. Potential based reward shaping provides multiagent systems with benefits to time-to-convergence [8] but will alter the exploration, creating new joint action policies [5]. Potential based reward shaping is effective because it helps exploit information present in the system to help the agent learn. Through this, it creates a better signal for the agent to learn from. We use this idea of modifying information to help find solutions later in the derivation of state deformation methods.

2.4 Intent of other agents

If I, as a human researcher, am observing rocks and see two fellow researchers head to a large, interesting rock it would make sense for me to follow because I extrapolate their goal (observing that rock) from their perceived motions and positions. Conversely, if I see a large table but no one else near, I'm not going to try to lift it on my own, especially if I knowing that others are not going to come and help. If I know help will come, then waiting briefly before the help arrives is beneficial. I don't have to take any additional actions, but can still receive the reward for completing the task by waiting for others. We want to leverage the intent of other agents in the system, mainly because intuitively it makes sense to. Thus, we want to see how we can use this idea of intent in robotic situations. In order to do this, we need to know the extent human-gauged intent, and the plausibility of applying it to purely robotic scenarios.

In human-human and human-robot scenarios, intent has been leveraged for performing complex actions. But first, we must assume that the agents around us are acting intelligently, with an intent driving them. This can be an explicit internally held intent, as in intentional agent decision making [9] or simply the implicit general intent to solve the problem of the domain. Framed this way, every intelligent agent in a scenario trying to solve a specific task will have an associated intent to solve that task. Therefore, as long as we avoid domains where the agent has no goal or metric for success, a general intent can always be inferred.

Intent of others is useful in imitation learning [10, 9]. Alternatively, others' intent can be internally modeled as a Markov decision process (MDP) and applied to competitive games, such has humans vs robots billiards [11]. However, with this formulation in large multiagent problems, agents may need to maintain an MDP for each agent, which would eventually require an intractable amount of storage and computation time. We want to find a more general, less computationally intensive method to try to capture the other agents' intent.

2.5 Existing state shaping methods

In the past, state shaping mostly refered to methods for reducing the dimensionality of the states given to an agent [12, 13, 14]. One of the main qualities focused upon by Andre and Cheng is the idea of reducing the dimensionality of the state in order to create a simpler problem domain to learn [12, 14]. Kheradmandian focuses on creating abstractions for states through machine learning techniques [13], in an attempt to simplify the state space being learned upon.

While these methods are direct methods for state shaping, they focus on changing the dimensionality, not the values, inside the state. This is the major difference between the previous works and explicit coordination methods leveraged by Nasroullahi [1]. In this, the values in the state are directly modified by a heuristic function, reminiscent of the potential-based reward shaping methods. In the paper, a tightly-coupled POI observation domain is examined. The state used is a quadrant based

observation of the nearby POI and agents. The world is divided into quadrants relative to the observing agent. This results in an 8-dimensional state, where each dimension is calculated as the value of the POI or agent divided by its distance from the observing agent, summed for each POI or agent in that quadrant (Equation 2).

$$State\,Quadrant\,Value = \sum_{p \in quadrant} \frac{value(p)}{dist(a,p)} \tag{2}$$

To improve the coordination in the joint action space, the perceived values of the POI are directly modified by a scalar multiple, changing Equation 2 to

$$State\,Quadrant\,Value = \sum_{p \in quadrant} \frac{value(p)}{dist(a,p)} \times s(p) \tag{3}$$

where the scaling value $s(p)$ is defined as

$$s_{POI} = \frac{value(POI)}{dist(a_c, POI)} \tag{4}$$

where $value_{POI}$ is the original value of the POI, and a_c is the closest agent to the current POI [1]. The idea behind this method is to implicitly communicate the utility of a POI to other agents in the system. When another agent is close to a POI, and needs a helping agent to completely mark the POI, nearby agents will see the perceived value of the POI stay high or increase, while other POI lose value.

3 New State Shaping Methods

One immediate drawback to the state shaping method used in [1] is the inability to differentiate POIs in domains with a degree of coupling greater than 2. For example, in a domain with coupling of 5, the s_{POI} value produced by Equation 4 is the same for POI with 4 nearby agents, and 1 nearby agent. Since the formulation is directly dependent upon the problems degree of coupling, it is obvious that this method would not generalize to higher degree coupling problems.

3.1 Min-Max State Distortion

To address this, we developed a new formula for distorting the state, which directly incorporates the degree of coupling into the formulation in a manner that does not increase the computational complexity beyond what is required for Equation 4.

For an agent to use this model of intent in a tightly coupled system, they must be able to reason not only about what other agents might do, but also which selection of agents is best situated to complete the task.

This idea is what inspires the novel POI distortion equation, which evaluates an agent's potential contribution to a POI in comparison to what an agent thinks others will do. We call this the impact,

$$Impact(a,p) = \frac{max\left(min\left(\frac{d_{jc}}{d_a}, 1\right), \frac{1}{2}\right)}{max\left(min\left(\frac{d_c}{d_a}, 1\right), \frac{1}{2}\right)} \tag{5}$$

Where d_a is the distance between the observing agent and the poi,$dist(a,p)$. In a similar manner the closest agent to the POI is $d_c = dist(a_{closest}, p)$. The distance between the j^{th} closest agent and the POI is $d_{jc} = dist(a_{jth\ closest}, p)$. We set j to the degree of coupling minus one, so the j^{th} closest agent is a part of the observing team, but is not the last member of the team.

The effect of this equation is visualized in three different scenarios in Figure 1. In it, the POI, as the black circle, has a coupling weight of three. Two agents, as the white squares, are the two closest agents. When a third agent looks at the POI, it will be distorted depending on its distance away. The magnitude of the distortion is represented by the color gradient.

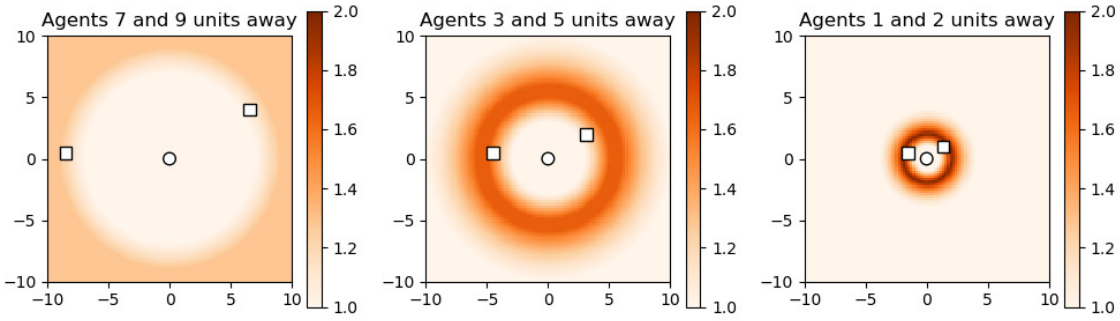

Figure 1: The magnitude of deformation that an agent would see around a POI using the min-max distortion. The black circle represents the POI in question, and the two black-bordered white squares represent other agents in the system.

3.2 Gaussian State Distortion

One alternative we looked to explore was the magnitude by which distortion would occur. The min/max method provides a scaling value between 1 and 2, inclusive. In order to create a distortion value which would create a scale between $\frac{1}{n}$ and n, where n is some integer, we used a Gaussian function, based off the observing agent's distance to the POI. By itself, this would create a function with a range between 0 and 1. To

create a larger scaling magnitude, and to incorporate the relative distances of other agents, we introduce a term in front of the main Gaussian function, $\frac{n}{d_c+d_{jc}}$, which both provides a larger maximum scaling (n), and devalues POI with agent teams far away ($\frac{1}{d_c+d_{jc}}$). This is written in full in Equation 6.

$$Impact(a,p) = \left(\frac{n}{d_c + d_{jc}}\right) e^{-\frac{(d_a-d_c)^2+(d_a-d_{jc})^2}{n}} \tag{6}$$

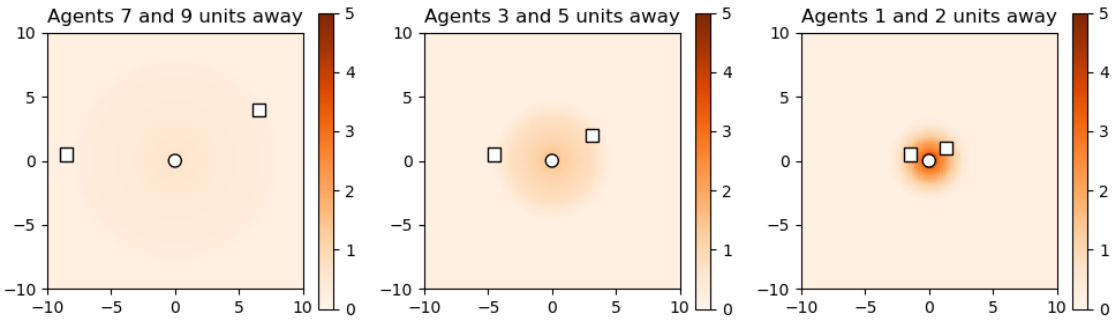

Figure 2: The magnitude of deformation that an agent would see around a POI using the Gaussian distortion. The black circle represents the POI in question, and the two black-bordered white squares represent other agents in the system.

4 Methodology

4.1 Cooperative Rover Observation Domain

In our simplified rover domain, teams of rovers are placed in a rectangular world populated with a set number of Points of Interest (POI).The task is to find and observe the POI in the world. The POI have a coupling degree, and when the number of agents adjacent to the POI are equal to or greater than the coupling degree, the POI is marked as "observed" and removed from the world.

Formally, the evolutionary algorithm is looking for a neural network policy which maximizes the reward, G (Equation 1. In order to optimize Equation 1, the first sum should be maximized, while the second minimized. Thus, given a team of agents $a_1, a_2, \ldots, a_n$ in a walled world with POI $p_1, p_2, \ldots, p_m$, what policy given to all agents will minimize the steps s needed to observe all POI? This search is also constricted by the tightly coupled nature of the task. An policy optimal policy must take actions to maximize G in the presence of the rest of the team. This impacts how the sum of observed is generated, as this sum

$$\sum_{p \in POI} value(p) \times observed(p)$$

can be rewritten in terms of specific observing agents

$$\sum_{p \in POI} value(p) \times max(observedby(p, a_1) \times observedby(p, a_2) \times observedby(p, a_3) \ldots \times observedby(p, a_n))$$

where $a_1, a_2, \ldots, a_n$ are a distinct subset of agents. Since the $observedby(p, a_i)$ function returns 1 or 0, the maximum of the product of observing agents will either be 1 if enough agents are present, or 0 otherwise. An optimal policy will position agents such that successful subsets $a_1, a_2, \ldots, a_n$ can be constructed for all POI during the trial.

Agents receive a reward equal to the value of a POI (in this case, 5) for each POI observed, and received a -0.1 reward for each movement.

A single trial involves a specified number of epochs, in which a world was generated and agents placed in the world. Agents were given a strict time limit to operate in the world, to keep the simulations quick and avoid agents wandering about needlessly incurring negative reward. Teams were assembled at random.

4.2 Cooperative Coevolutionary Algorithms - CCEA

Policies were trained via a basic cooperative co-evolutionary search. Coevolutionary methods have been shown to work well in homogeneous multiagent domains [15]. The specific algorithm used is presented in Figure 3. Policy fitness is based on the reward they received after each series of time-steps in the world. In our experiments, the fitness $F(z) = G(z)$ or $F(z) = D(z)$. While evolutionary policy learning was used, the work presented does not explicitly rely on evolutionary methods.

Our implementation of CCEA uses a simple 50% survival rate at each epoch. The lowest 50% performing agents were killed off, and the top 50% were cloned and mutated by 10%. The 10% mutation involves randomly sampling 10% of the weights in the neural network, and adjusting the weights by a relative $\pm 10\%$. Policies are collected into one large population before being divided into teams. This causes the policies to search for an optimal policy that can be used on any agent, in any team.

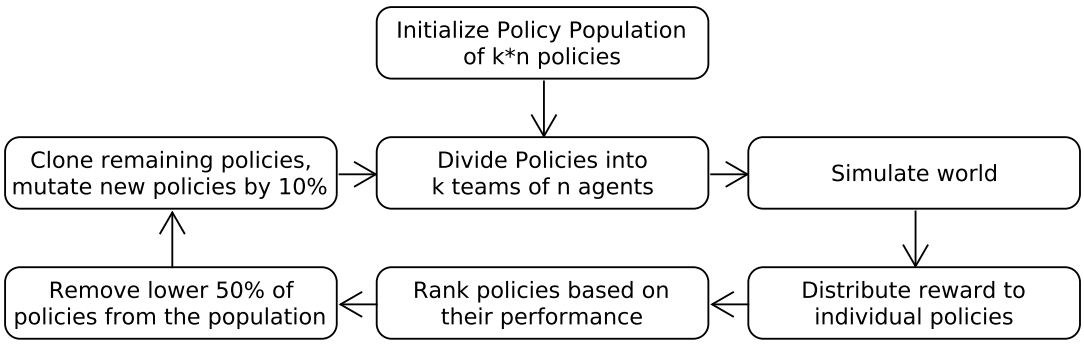

Figure 3: The overview of CCEA as implemented in this work. The process has no definite end, but is allowed to loop as long as learning occurs.

4.3 Rover Domain Simulator

A C++ simulator was written to control the state interactions in the rover domain. The Fast Artificial Neural Network library FANN [16] was used to implement the neural networks, and our own code was written for CCEA and NN mutation. The simulator controls a 4-connected world, and would instantiate the agents and POI in the world. The simulator would randomly position the agents and POI in each world if requested. A new world was created for each generation of learning. In cases where agent or POI positions were specified outright, these initial positions are retained across generations. Reward signals were calculated at the end of each generation, which were used for the generations' evolutionary learning stage.

4.4 Experiments

To test the methods, worlds were constructed with variations on three main parameters: the ratio of POI to agents, the size of the agent teams, and the type of distortion used on the state. 10 statistical trials were run for each experiment.

In these tests, agents start evenly spaced in a straight line one edge of the world. This straight line configuration does not necessarily test the POI picking ability, since a general trend in one direction could find POI *if the agents are already grouped*. Until they are in groups, they cannot solve the problems. However, once they do group up, this world should be easy to get a large reward on. Similarity, the grouped initial configuration looks to see how agents can move as a single unit toward the next best POI. As a team, they have the ability to observe a POI, but they need to coordinate their actions in order to do so.

Each trial was run for 5000 epochs, and compared against a baseline test in the same world configuration without the state deformation. In the smaller (30 unit x 30 unit) world, agents could move for 40 time steps. In the smaller (50 unit x 50 unit)

world, agents could move for 60 time steps.

5 Results

For the trials with a coupling of 2, there is no statistical difference between teams using state distortion and those not using distortion. For tests with a coupling of 3, the teams using distortion methods would see performance increases as more POI were introduced than agents. The graphs of team performance for these scenario are seen in Figures 4, 5, 6, 8. Figure 7 shows the point where the trend breaks down, and a ratio of POI to agents being 1 provided the most benefit for using state distortion.

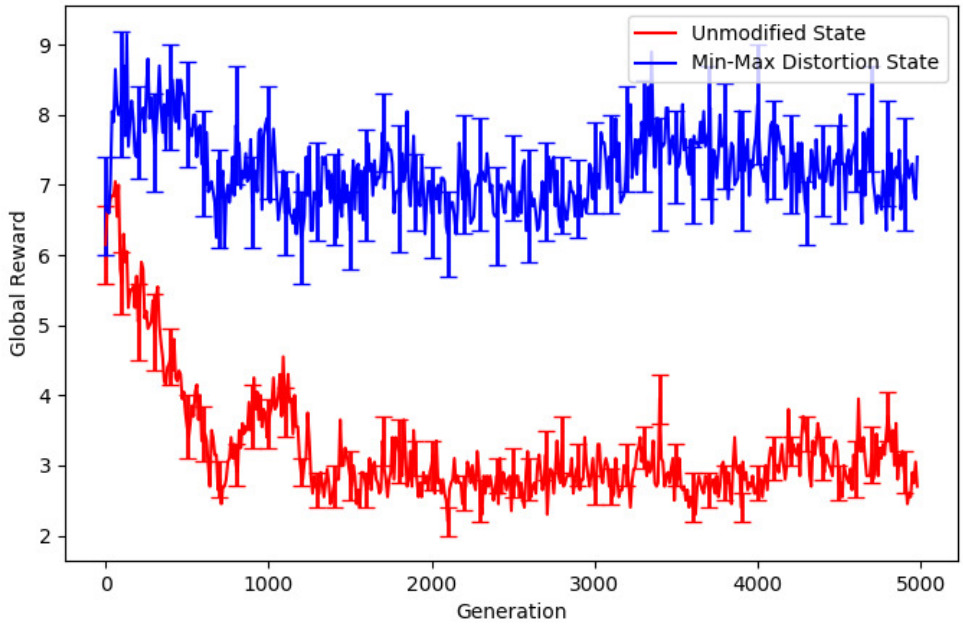

Figure 4: Performance with 10 agents, 15 POI, and coupling of 3. The min/max scaling method preserves performance in the more difficult domain where learning with D degrades quickly, and fails to find a successful policy.

6 Discussion

This data shows a better performance than methods without the state-shaping used. But why is this occurring? By looking at the paths the rovers take, we can infer how their behavior is changing based on these methods, and see what effects they are instilling in the population.

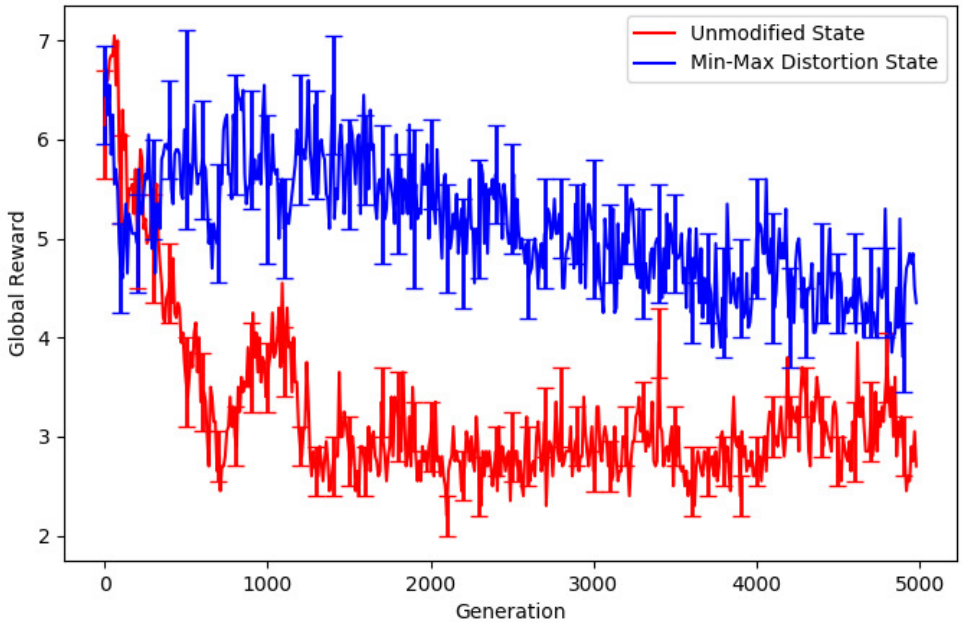

Figure 5: Performance with 10 agents, 15 POI, and coupling of 3. Gaussian distortion was used with a maximum magnitude of 2. This preserves some performance relative to learning with D, but does not converge toward a higher value.

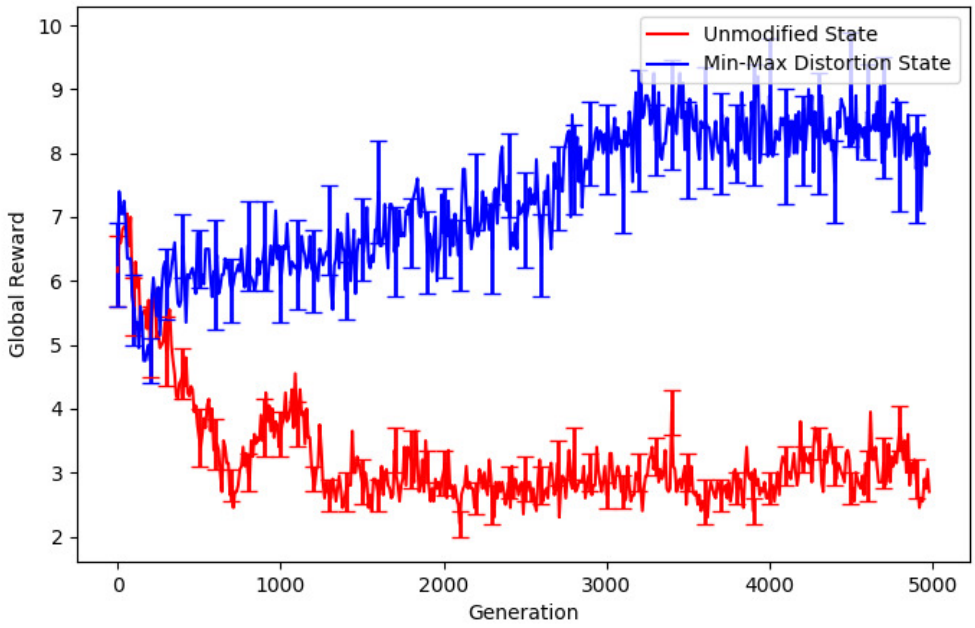

Figure 6: Performance with 10 agents, 15 POI, and coupling of 3. Gaussian distortion was used with a maximum magnitude of 4. This increases performance relative to baseline learning with D, and performs better than Gaussian distortion with magnitude 2 (Figure 5).

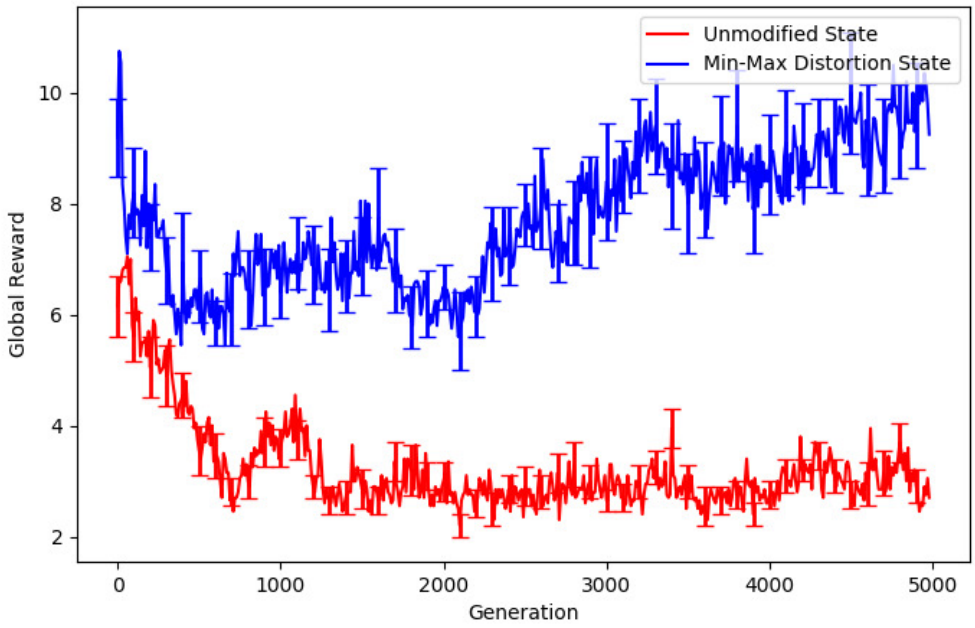

Figure 7: Performance with 25 agents, 25 POI, and coupling of 3. Gaussian distortion was used with a maximum magnitude of 4. As the size of the agent teams are increased, the distorted state agents outperform agents learning only with D. However, the gains are less pronounced than with the smaller problem (Figure 6).

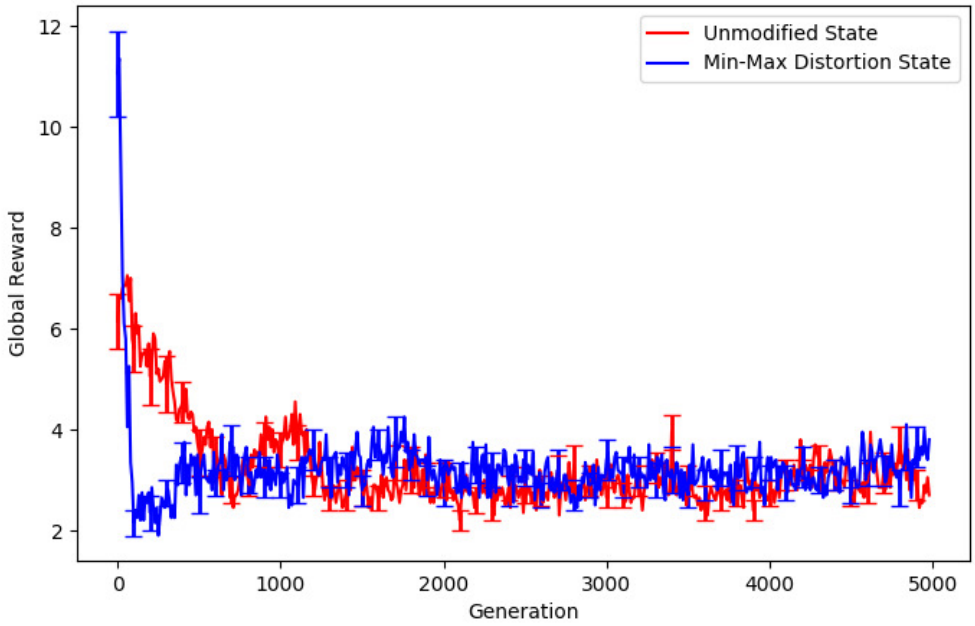

Figure 8: Performance with 25 agents, 30 POI, and coupling of 3. Gaussian distortion was used with a maximum magnitude of 4. The trend seen in smaller worlds breaks down, and both the Gaussian distortion and baseline with D fail to find a satisfactory policy in this larger world.

Firstly, we look at the min/max formula. The increase in performance can be attributed to the state shaping helping agents decide which option is the best to pursue. When an agent approaches two POIs with different numbers of agents around them, the POI closer to being completed will be highlight more strongly, letting the agent know that it is closer to completion, as seen in Figure 1 and 2.

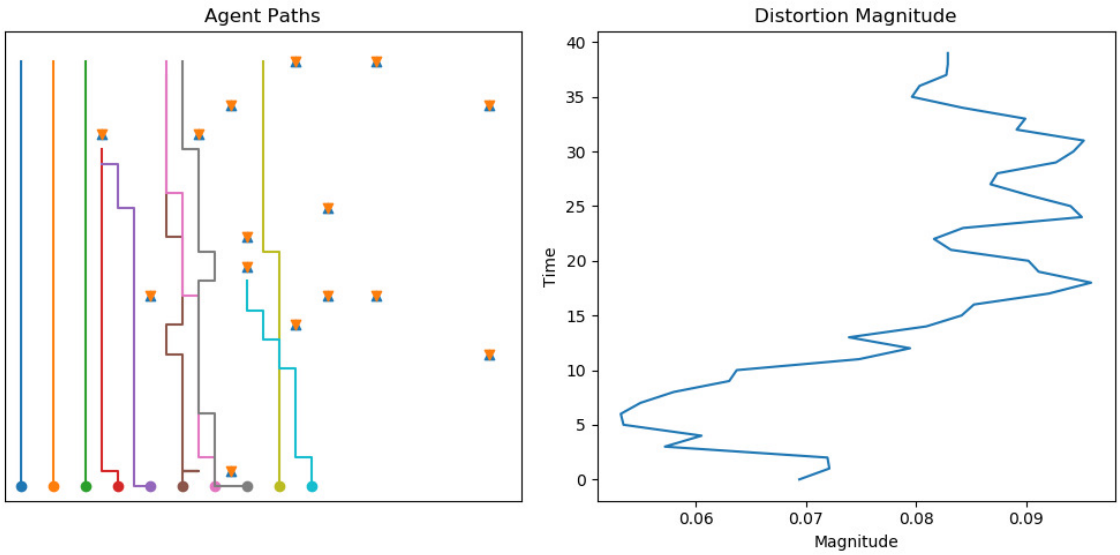

Figure 9: The paths of agents during generation 1000 are shown alongside the average distortion value perceived by agents these agents during the generation. In the left figure, agents start at the dots.

Similar phenomena are observed in with the Gaussian state deformation. The agents learn to move as a coordinated team when working with a coupling of 3, particularly under situations where POIs outnumber agents. When the larger scaling magnitude of 4 is used, the successes become more pronounced. Most interestingly, the successful trend in situations where POIs outnumber agents breaks down on the largest case tested, with 25 agents and 30 POIs. With 25 agents, the world with 25 POIs performed best under the Gaussian state distortion with magnitude 4 (Figure 7), whereas the situation with 25 agents and 30 POI fails just like the standard learning without state distortions (Figure 8).

This effect is shown by the trajectories of a team of agents starting in a line (Figures 9, 10, 11). Figures 9 and 10 shows one trial's starting and ending generations from the successful tests shows of 10 agents and 15 POI from Figure 6. Agents form teams as they move toward POI in the later generations, leading to the increased performance. This corresponds with the average POI distortion value seen by agents during this generation, shown on the right. We see the increase in distortion corresponds to an increase in performance through the generations of learning. This also corresponds with the paths and distortions in Figure 11, which is the last generation

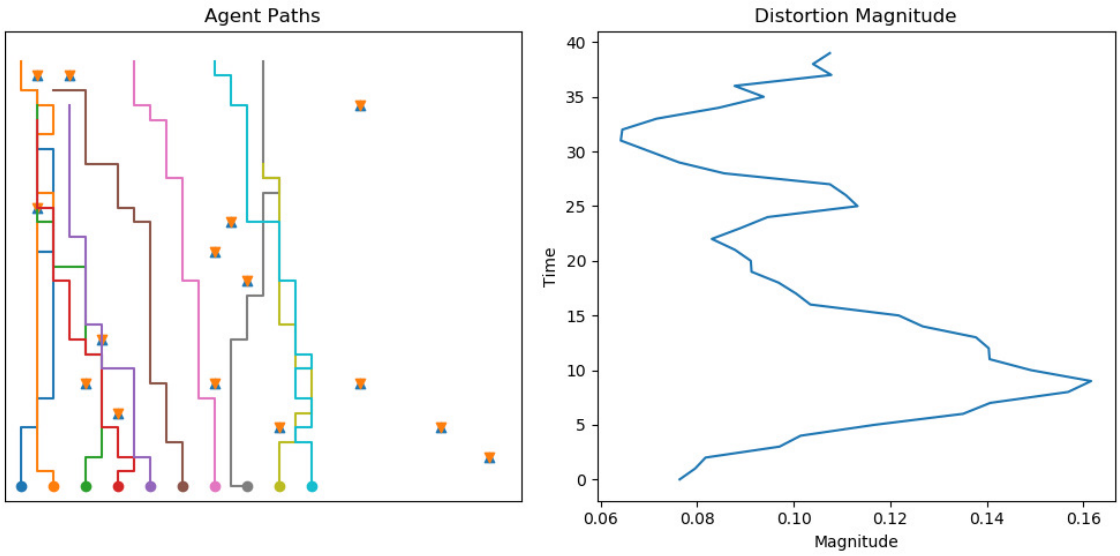

Figure 10: The paths of agents during generation 5000 are shown alongside the average distortion value perceived by agents these agents during the generation. In the left figure, agents start at the dots.

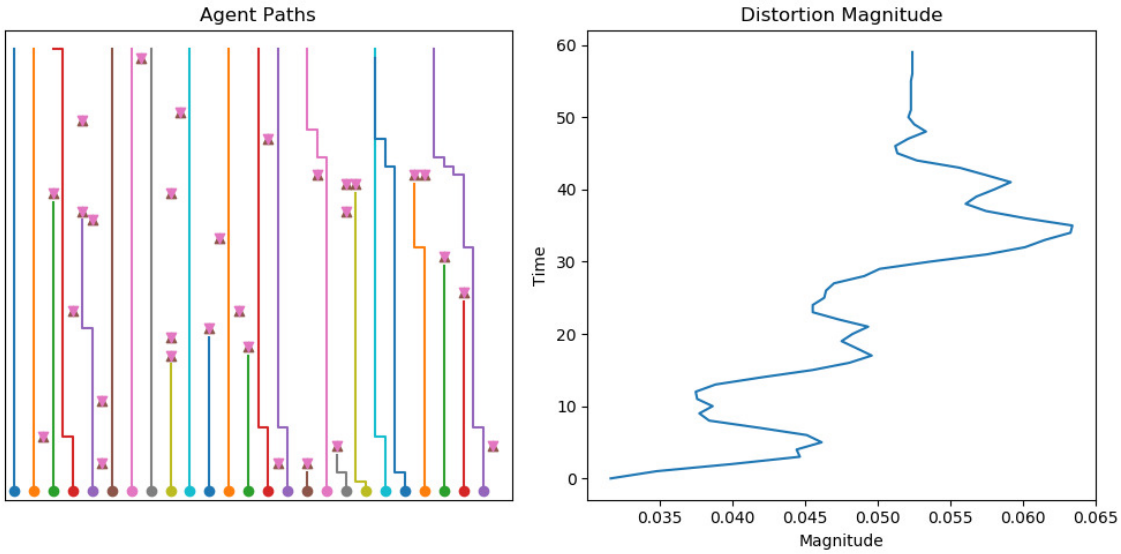

Figure 11: The paths of agents during generation 5000 are shown alongside the average distortion value perceived by agents these agents during the generation. In the left figure, agents start at the dots.

from the failing tests of 25 agents, 30 POI visualized in figure 8. Once again, the poor performance of the team corresponds to the small average distortion during the epoch.

7 Concluding Remarks and Future Work

In this paper, we analyze current methods to gauge intent in multiagent systems. We also present computationally simple methods to help agents gauge their potential impact on the system by directly distorting sensor values before they are assembled into the state. The new methods for potential based state distortion show improvement in tightly coupled situations where preserving teams throughout operation is beneficial. The idea of distorting values in the state, much like distorting reward signals in potential based reward shaping, is relatively unexplored, and offers a new expansion of that research. For example, how would shaping the state by focusing on the agents, not the POI impact the coordination effects. Instituting shaping on the agents would create a general solution that would not be restricted to POI observation domains, and would work on all multiagent problems. The impact of potential based state shaping on multiagent domains which are not POI based is unexplored, and could yield interesting results.

References

[1] E. Nasroullahi and K. Tumer, "Combining coordination mechanisms to improve performance in multi-robot teams," *Artificial Intelligence Research*, vol. 1, no. 2, pp. 1–10, 2012. [Online]. Available: http://www.sciedu.ca/journal/index.php/air/article/view/1141

[2] M. Colby and K. Tumer, "Shaping fitness functions for coevolving cooperative multiagent systems," in *Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems*, Valencia, Spain, June 2012, pp. 425–432.

[3] A. Rahmattalabi, "D++: Structural Credit Assignment in Tightly Coupled Multiagent Domains," Ph.D. dissertation, Oregon State University, 2016.

[4] A. K. Agogino and K. Tumer, "Analyzing and visualizing multiagent rewards in dynamic and stochastic domains," *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 2, pp. 320–338, 2008.

[5] S. Devlin and D. Kudenko, "Theoretical considerations of potential-based reward shaping for multi-agent systems," *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pp. 225–232, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2030503

[6] J. Randløv and P. Alstrøm, "Learning to Drive a Bicycle using Reinforcement Learning and Shaping," *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 463–471, 1998.

[7] M. Grzes and D. Kudenko, "Plan-based reward shaping for Multi-Agent reinforcement learning," *Intelligent Systems, 2008.*, vol. 00, pp. 1–20, 2014. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs%7B_%7Dall.jsp?arnumber=4670492

[8] S. Devlin and D. Kudenko, "Dynamic Potential-Based Reward Shaping," *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pp. 433–440, 2012.

[9] A. N. Meltzoff, "The 'like me' framework for recognizing and becoming an intentional agent," *Acta Psychologica*, vol. 124, no. 1, pp. 26–43, 2007.

[10] P. L. Jackson, A. N. Meltzoff, and J. Decety, "Neural circuits involved in imitation and perspective-taking," *NeuroImage*, vol. 31, no. 1, pp. 429–439, 2006.

[11] T. Nierhoff, K. Leibrandt, T. Lorenz, and S. Hirche, "Robotic Billiards: Understanding Humans in Order to Counter Them," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1889–1899, 2016.

[12] D. Andre and S. J. Russell, "State Abstraction for Programmable Reinforcement Learning Agents," in *Proceedings of the 18th National Conference on Artificial Intelligence*, 2002, pp. 119–125. [Online]. Available: http://www.aaai.org/Papers/AAAI/2002/AAAI02-019.pdf

[13] G. Kheradmandian and M. Rahmati, "Automatic abstraction in reinforcement learning using data mining techniques," *Robotics and Autonomous Systems*, vol. 57, no. 11, pp. 1119–1128, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2009.07.002

[14] Z. Cheng and L. E. Ray, "State abstraction in reinforcement learning by eliminating useless dimensions," *Proceedings - 2014 13th International Conference on Machine Learning and Applications, ICMLA 2014*, pp. 105–110, 2015.

[15] S. G. Ficici, O. Melnik, and J. B. Pollack, "A Game-Theoretic and Dynamical-Systems Analysis of Selection Method in Coevolution," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 580–602, 2005.

[16] S. Nissen, "Implementation of a fast artificial neural network library (fann)," *Report, Department of Computer Science University of Copenhagen (DIKU)*, vol. 31, p. 29, 2003.