Product distribution theory for control of multi-agent systems

Chiu Fan Lee
Physics Department, Oxford University
Parks Road, Oxford OX1 3PU, U.K.
c.lee1@physics.ox.ac.uk

David H. Wolpert
MS 269-1, NASA Ames Research Center
Moffett Field, CA 94035, USA
dhw@ptolemy.arc.nasa.gov

Abstract

Product Distribution (PD) theory is a new framework for controlling Multi-Agent Systems (MAS's). First we review one motivation of PD theory, as the information-theoretic extension of conventional full-rationality game theory to the case of bounded rational agents. In this extension the equilibrium of the game is the optimizer of a Lagrangian of the (probability distribution of) the joint state of the agents. Accordingly we can consider a team game in which the shared utility is a performance measure of the behavior of the MAS. For such a scenario the game is at equilibrium — the Lagrangian is optimized — when the joint distribution of the agents optimizes the system's expected performance. One common way to find that equilibrium is to have each agent run a reinforcement learning algorithm. Here we investigate the alternative of exploiting PD theory to run gradient descent on the Lagrangian. We present computer experiments validating some of the predictions of PD theory for how best to do that gradient descent. We also demonstrate how PD theory can improve performance even when we are not allowed to rerun the MAS from different initial conditions, a requirement implicit in some previous work.

1. Introduction

Product Distribution (PD) theory is a recently introduced broad framework for analyzing, controlling, and optimizing distributed systems [9, 10, 11]. Among its potential applications are adaptive, distributed control of a Multi-Agent System (MAS), (constrained) optimization, sampling of high-dimensional probability densities (i.e., improvements to Metropolis sampling), density estimation, numerical integration, reinforcement learning, information-theoretic bounded rational game theory, population biology, and management theory. Some of these are investigated in [2, 1, 8, 3].

Here we investigate PD theory's use for adaptive, distributed control of a MAS. Typically such control is done by having each agent run its own reinforcement learning algorithm [4, 13, 14, 12]. In this approach the utility function of each agent is based on the **world utility** $G(x)$ mapping the joint move of the agents, $x \in X$, to the performance of the overall system. However in practice the agents in a MAS are bounded rational. Moreover the equilibrium they reach will typically involve mixed strategies rather than pure strategies, i.e., they don't settle on a single point x optimizing $G(x)$. This suggests formulating an approach that explicitly accounts for the bounded rational, mixed strategy character of the agents.

Now in any game, bounded rational or otherwise, the agents are independent, with each agent i choosing its move x_i at any instant by sampling its probability distribution (mixed strategy) at that instant, $q_i(x_i)$. Accordingly, the distribution of the joint-moves is a product distribution, $P(x) = \prod_i q_i(x_i)$. In this representation of a MAS, all coupling between the agents occurs indirectly; it is the separate distributions of the agents $\{q_i\}$ that are statistically coupled, while the actual moves of the agents are independent.

PD theory adopts this perspective to show that the equilibrium of a MAS is the minimizer of a Lagrangian $\mathcal{L}(P)$, derived using information theory, that quantifies the expected value of G for the joint distribution $P(x)$. From this perspective, the update rules used by the agents in RL-based systems for controlling MAS's are just particular (inefficient) ways of finding that minimizing distribution.

Various techniques for modifying the agents' utilities from G to improve convergence have been used with such RL-based systems [13]. As illustrated below, by viewing the problem as one of Lagrangian-minimization, PD theory can be used to derive corrections to those techiques. In addition, PD theory suggests novel ways to find the equilbirum, e.g., applying any of the powerful search techniques for continuous variables, like gradient descent, to find the P optimizing $\mathcal{L}$. By casting the problem this way in terms of finding an optimal P rather than finding an optimal x, we can exploit the power of search techniques for continuous variables even when X is a discrete, finite space. Moreover, typically such search techniques can be run in a highly distributed fashion, since P is a product distribution. Finally, the techniques for modifying agents' utilities in RL-based systems often require rerunning the MAS from different initial conditions, or equivalently exploiting knowledge of the functional form of G to evaluate how particular changes to the initial conditions would affect G. PD theory provides us with techniques for improving convergence even when we have no such knowledge, and when we cannot rerun the system.

In the next section we review the game-theory motivation of PD theory. We then present details of our Lagrangian-minimization algorithm. We end with computer experiments involving both controlling a spin glass (a system of coupled binary variables) and controlling the agents in a variant of the bar problem [13]. These experiments both validate some of the predictions of PD theory of how best to modify the agents' utilities from G, and of how to improve convergence when one cannot rerun the system.

2. Bounded Rational Game Theory

In this section we motivate PD theory as the information-theoretic formulation of bounded rational game theory.

2.1. Review of noncooperative game theory

In noncooperative game theory one has a set of N **players**. Each player i has its own set of allowed **pure strategies**. A **mixed strategy** is a distribution $q_i(x_i)$ over player i's possible pure strategies. Each player i also has a **private utility** function g_i that maps the pure strategies adopted by all N of the players into the real numbers. So given mixed strategies of all the players, the expected utility of player i is $E(g_i) = \int dx \; \prod_j q_j(x_j) g_i(x)$ [1].

In a **Nash equilibrium** every player adopts the mixed strategy that maximizes its expected utility, given the mixed strategies of the other players. More formally, $\forall i, q_i = \text{argmax}_{q'_i} \int dx \; q'_i \prod_{j \neq i} q_j(x_j) \; g_i(x)$. Perhaps the major objection that has been raised to the Nash equilibrium concept is its assumption of **full rationality** [5, 6]. This is the assumption that every player i can both calculate what the strategies $q_{j \neq i}$ will be and then calculate its associated optimal distribution. In other words, it is the assumption that every player will calculate the entire joint distribution $q(x) = \prod_j q_j(x_j)$. If for no other reasons than computational limitations of real humans, this assumption is essentially untenable.

2.2. Review of the maximum entropy principle

Shannon was the first person to realize that based on any of several separate sets of very simple desiderata, there is a unique real-valued quantification of the amount of syntactic information in a distribution $P(y)$. He showed that this amount of information is (the negative of) the Shannon entropy of that distribution, $S(P) = -\int dy \; P(y) ln[\frac{P(y)}{\mu(y)}]$. So for example, the distribution with minimal information is the one that doesn't distinguish at all between the various y, i.e., the uniform distribution. Conversely, the most informative distribution is the one that specifies a single possible y. Note that for a product distribution, entropy is additive, i.e., $S(\prod_i q_i(y_i)) = \sum_i S(q_i)$.

Say we given some incomplete prior knowledge about a distribution $P(y)$. How should one estimate $P(y)$ based on that prior knowledge? Shannon's result tells us how to do that in the most conservative way: have your estimate of $P(y)$ contain the minimal amount of extra information beyond that already contained in the prior knowledge about $P(y)$. Intuitively, this can be viewed as a version of Occam's razor. This approach is called the maximum entropy (maxent) principle. It has proven useful in domains ranging from signal processing to supervised learning [7].

2.3. Maxent Lagrangians

Much of the work on equilibrium concepts in game theory adopts the perspective of an external observer of a game. We are told something concerning the game, e.g., its utility functions, information sets, etc., and from that wish to predict what joint strategy will be followed by real-world players of the game. Say that in addition to such information, we are told the expected utilities of the players. What is our best estimate of the distribution q that generated those expected utility values? By the maxent principle, it is the distribution with maximal entropy, subject to those expectation values.

To formalize this, for simplicity assume a finite number of players and of possible strategies for each player. To agree with the convention in other fields, from now on we implicitly flip the sign of each g_i so that the associated player i wants to minimize that function rather than maximize it. Intuitively, this flipped $g_i(x)$ is the "cost" to player i when the joint-strategy is x, though we will still use the term "utility".

Then for prior knowledge that the expected utilities of the players are given by the set of values $\{\epsilon_i\}$, the maxent estimate of the associated q is given by the minimizer of the Lagrangian

$$\mathcal{L}(q) \equiv \sum_i \beta_i [E_q(g_i) - \epsilon_i] - S(q) \quad (1)$$

[1] Throughout this paper, the integral sign is implicitly interpreted as appropriate, e.g., as Lebesgue integrals, point-sums, etc.

$$= \sum_i \beta_i[\int dx \ \prod_j q_j(x_j)g_i(x) - \epsilon_i] - S(q)(2)$$

where the subscript on the expectation value indicates that it evaluated under distribution q, and the $\{\beta_i\}$ are "inverse temperatures" implicitly set by the constraints on the expected utilities.

Solving, we find that the mixed strategies minimizing the Lagrangian are related to each other via

$$q_i(x_i) \propto e^{-E_{q_{(i)}}(G|x_i)} \qquad (3)$$

where the overall proportionality constant for each i is set by normalization, and $G \equiv \sum_i \beta_i g_i$ [2]. In Eq. 3 the probability of player i choosing pure strategy x_i depends on the effect of that choice on the utilities of the other players. This reflects the fact that our prior knowledge concerns all the players equally.

If we wish to focus only on the behavior of player i, it is appropriate to modify our prior knowledge. To see how to do this, first consider the case of maximal prior knowledge, in which we know the actual joint-strategy of the players, and therefore all of their expected costs. For this case, trivially, the maxent principle says we should "estimate" q as that joint-strategy (it being the q with maximal entropy that is consistent with our prior knowledge). The same conclusion holds if our prior knowledge also includes the expected cost of player i.

Modify this maximal set of prior knowledge by removing from it specification of player i's strategy. So our prior knowledge is the mixed strategies of all players other than i, together with player i's expected cost. We can incorporate prior knowledge of the other players' mixed strategies directly, without introducing Lagrange parameters. The resultant **maxent Lagrangian** is

$$\begin{aligned}\mathcal{L}_i(q_i) &\equiv \beta_i[\epsilon_i - E(g_i)] - S_i(q_i) \\ &= \beta_i[\epsilon_i - \int dx \ \prod_j q_j(x_j)g_i(x)] - S_i(q_i)\end{aligned}$$

solved by a set of coupled **Boltzmann distributions**:

$$q_i(x_i) \propto e^{-\beta_i E_{q_{(i)}}(g_i|x_i)}. \qquad (4)$$

Following Nash, we can use Brouwer's fixed point theorem to establish that for any non-negative values $\{\beta\}$, there must exist at least one product distribution given by the product of these Boltzmann distributions (one term in the product for each i).

The first term in $\mathcal{L}_i$ is minimized by a perfectly rational player. The second term is minimized by a perfectly *irrational* player, i.e., by a perfectly uniform mixed strategy q_i. So β_i in the maxent Lagrangian explicitly specifies the balance between the rational and irrational behavior of the player. In particular, for $\beta \to \infty$, by minimizing the Lagrangians we recover the Nash equilibria of the game. More formally, in that limit the set of q that simultaneously minimize the Lagrangians is the same as the set of delta functions about the Nash equilibria of the game. The same is true for Eq. 3.

Eq. 3 is just a special case of Eq. 4, where all player's share the same private utility, G. (Such games are known as **team games**.) This relationship reflects the fact that for this case, the difference between the maxent Lagrangian and the one in Eq. 2 is independent of q_i. Due to this relationship, our guarantee of the existence of a solution to the set of maxent Lagrangians implies the existence of a solution of the form Eq. 3. Typically players a will be closer to minimizing their expected cost than maximizing it. For prior knowledge consistent with such a case, the β_i are all non-negative.

For each player i define

$$f_i(x, q_i(x_i)) \equiv \beta_i g_i(x) + \ln[q_i(x_i)]. \qquad (5)$$

Then we can maxent Lagrangian for player i is

$$\mathcal{L}_i(q) = \int dx \ q(x) f_i(x, q_i(x_i)). \qquad (6)$$

Now in a bounded rational game every player sets its strategy to minimize its Lagrangian, given the strategies of the other players. In light of Eq. 6, this means that we interpret each player in a bounded rational game as being perfectly rational for a utility that incorporates its computational cost. To do so we simply need to expand the domain of "cost functions" to include probability values as well as joint moves.

Often our prior knowledge will not consist of exact specification of the expected costs of the players, even if that knowledge arises from watching the players make their moves. Such alternative kinds of prior knowledge are addressed in [10, 11]. Those references also demonstrate the extension of the formulation to allow multiple utility functions of the players, and even variable numbers of players. Also discussed there are **semi-coordinate** transformations, under which, intuitively, the moves of the agents are modified to set in binding contracts.

3. Optimizing the Lagrangian

First we introduce the shorthand

$$\begin{aligned}[G \mid x_i] &\equiv E(G \mid x_i) &(7)\\ &= \int dx' \delta(x'_i - x_i) G(x) \prod_{j \neq i} q_i(x'_i), &(8)\end{aligned}$$

[2] The subscript $q_{(i)}$ on the expectation value indicates that it is evaluated according the distribution $\prod_{j \neq i} q_j$.

where the delta function forces $x'_i = x_i$ in the usual way. Now given any initial q, one may use gradient descent to search for the q optimizing $\mathcal{L}(q)$. Taking the appropriate partial derivatives, the descent direction is given by

$$\triangle q_i(x_i) = \frac{\delta \mathcal{L}}{\delta q_i(x_i)} = [G|x_i] + \beta^{-1} \log q(x_i) + C \quad (9)$$

where C is a constant set to preserve the norm of the probability distribution after update, i.e., set to ensure that

$$\int dx_i q_i(x_i) = \int dx_i (q_i(x_i) + \triangle q_i(x_i)) = 1. \quad (10)$$

Evaluating, we find that

$$C = -\frac{1}{\int dx_i 1} \int dx_i \Big\{ [G|x_i] + \beta^{-1} \log q_i(x_i) \Big\}. \quad (11)$$

(Note that for finite X, those integrals are just sums.)

To follow this gradient, we need an efficient scheme for estimation of the conditional expected G for different x_i. Here we do this via Monte Carlo sampling, i.e., by repeatedly IID sampling q and recording the resultant private utility values. After using those samples to form an estimate of the gradient for each agent, we update the agents' distributions accordingly. We then start another block of IID sampling to generate estimates of the next gradients.

In large systems, the sampling within any given block can be slow to converge. One way to speed up the convergence is to replace each $[G \mid x_i]$ with $[g_i \mid x_i]$, where g_i is set to minimize bias plus variance [9, 11]:

$$g_i(x) := G(x) - \int dx_i \frac{L^{-1}_{x_i}}{\int dx'_i L^{-1}_{x'_i}} G(x_{-i}, x_i), \quad (12)$$

where L_{x_i} is the number of times the particular value x_i arose in the most recent block of L samples. This is called the **Aristocrat Utility** (AU). It is a correction to the utility of the same name investigated in [13] and references therein.

Note that evaluation of AU for agent i requires knowing G for all possible x_i, with $x_{(i)}$ held fixed. Accordingly, we consider an approximation, which is called the **Wonderful Life** utility (WLU), in which we replace the values $\frac{L^{-1}_{x_i}}{\int dx'_i L^{-1}_{x'_i}}$ defining agent i's AU with a delta function about about the least likely (according to q_i) of that agent's moves. (This is version of the utility of the same name investigated in [13] and references therein.)

Below we present computer experiments validating the theoretical predictions that AU converges faster than the team game, and that the WLU defined here converges faster than its reverse, in which the delta function is centered on the *most* likely of the agent's moves.

Both WLU and AU require recording not just $G(x)$ for the Monte Carlo sample x, but also G at a set of points related in a particular way to x. When the functional form of G is known, often there is cancellation of terms that obviates this need. Indeed, often what one does need to record in these cases is easier to evaluate than is G. However when the functional form of G is not known, using such private utilities would require rerunning the system, i.e., evaluating G for many points besides x.

PD theory provides us an alternative way to improve the convergence of the sampling. This alternative exploits the fact that the joint distribution of all the agents is a product distribution. Accordingly, we can have the agents all announce their separate distributions $\{q_i\}$ at the end of each block. By itself, this is of no help. However say that x as well as $G(x)$ is recorded for all the samples taken so far (not just those in the preceding block). We can use this information as a training set for a supervised learning algorithm that estimates G. Again, this piece of information is of no use by itself. But if we combine it with the announced $\{q_i\}$, we can form an estimate of each $[G \mid x_i]$.

This estimate is in addition to the estimate based on the Monte Carlo samples — here the Monte Carlo samples from all blocks are used, to approximate $G(x)$, rather than to directly estimate the various $[G \mid x_i]$. Accordingly we can combine these two estimates. Below we present computer experiments validating this technique.

4. Experiments

4.1. Known world utilities

We first consider the case where the functional form of the world utility is known. Technically, the specific problem that we consider is the equilibration of a spin glass in an external field, where each spin has a total angular momentum $3/2$. The problem consists of 50 spins in a circular formation, where every spin is interacting with three spins on its right, three spins on its left as well as with itself. There are also external fields which interact with each individual spin differently. The world utility is thus of the following form:

$$G(x) = \sum_i h_i x_i + \sum_{<i,j>} J_{ij} x_i x_j, \quad (13)$$

where $\sum_{<i,j>}$ means summing over all the interacting pairs once. In our problem, the set elements in the set $\{h_i\}$ and $\{J_{ij}\}$ are generated uniformly at random form -0.5 to 0.5. The algorithm for the Lagrangian estimation goes as follows:

1. Each spin is treated as an agent which possesses a probability distribution on his set of actions: $\{q_i(x_i) \mid x_i \in \sigma_i \equiv \{-1, 1\}\}$, which is initially set to be uniform.

2. Each agent picks its choice of state according to the probability distribution for L times sequentially, where L is the Monte Carlo block size. We denote the number of state x_i picked by agent i by L_{x_i}. We require L_{x_i} to be non-empty for all $x_i \in \omega_i$, i.e., if some $L_{x_i} = 0$, we randomly pick a sample x' and set $x_i = x'_i$ so that $L_{x_i} = 1$. This process is to ensure that we can get conditional expected values $[G|x_i]$ for all $x_i \in \sigma_i$. It should be noted though that it violates the assumptions of IID sampling underpinning the derivation of the private utilities minimizing bias plus variance.

3. The gradients for each individual component is calculated based on the L samples taken from the previous step (c.f. eq. 9), and gradient descents are performed for all i simultaneously. Since all probabilities must be positive, for each component i, the magnitude of descent is halved if $q_i(x_i)$ is no longer positive for some x_i.

4. Repeat steps 2 and 3.

In figure 1, we have shown a comparison of three different ways of doing the descent direction estimation in step 3 above. Team game means that we use $[G|x_i]$ to get the descent directions, weighted Aristocratic Utility corresponds to using the formula in eq. 12 to get the descent directions, and uniform Aristocratic Utility corresponds to simplifying the functions $\{g_i\}$ to

$$\hat{g}_i(x) := G(x) - \frac{1}{|\sigma_i|} \sum_{x_i \in \sigma_i} G(x_{-i}, x_i). \qquad (14)$$

In figure 1, we see that weighted AU outperforms uniform AU except at $\beta^{-1} = 0.2$. This unexpected result at $\beta^{-1} = 0.2$ may be due to the limitation on the size of L. (Recall that we have required that $L_{x_i} \neq 0$, and if it ever does, we randomly pick a sample x' and set $x_i = x'_i$ so that $L_{x_i} = 1$.) Hence, as shown in figure 3, the number of $L_{x_i} = 1$ is greater when $\beta^{-1} = 0.2$ than that when $\beta^{-1} = 0.6$. This demonstrates that at $\beta^{-1} = 0.2$, quite a few redistributions of the samples are happening and hence the size of L has to be enlarged to get decent statistics. The speculation is further strengthened by comparing correct WLU (where $\frac{L^{-1}_{x_i}}{\int dx'_i L^{-1}_{x'_i}}$ defining agent i's AU are replaced with a delta function about about the least likely (according to q_i) of that agent's moves) and incorrect WLU (where the same quantities are replaced with a delta function about about the most likely (according to q_i) of that agent's moves) with different sample size L. As shown in figures 4 and 5, the increase in sample size does amend the problem caused by resampling.

5. Unknown world utilities

We now consider the case where the explicit formula for the world utility is not known and hence the calculations for WLU, uniform AU and weighted AU are not possible. Recall that for this case we require that each player not only submits her choices of actions during each Monte Carlo block, but her probability distribution as well. Although this brings a constant overhead to the transmission, this becomes negligible when L is large.

The problem we consider here is a 100-agent 4-night bar problem [14]. In this problem, each agent's strategy set consists of four elements: $\{1, 2, 3, 4\}$. The world utility is of the form:

$$G(x) = -50 \times \sum_{k=1}^{4} e^{-f_k(x)/6} \qquad (15)$$

where $f_k(x) = \sum_i \delta(x_i - k)$, i.e., $f_k(x)$ is the number of agents attending the bar at night k. The precise algorithm is as follows:

1. Each agent possesses a probability distribution on her set of actions: $\{q_i(x_i) \mid x_i \in \omega_i\}$, which is initially set to be uniform.

2. Each agent picks its state according to the probability distribution for L times sequentially, where L is the Monte Carlo block size, as well as her probability distribution $\{q_i(x_i) \mid x_i \in \omega_i\}$. Again, we require L_{x_i} to be non-empty for all $x_i \in \omega_i$, i.e., if some $L_{x_i} = 0$, we randomly pick a sample x' and set $x_i = x'_i$ so that $L_{x_i} = 1$.

3. Denote the set of samples in the L Monte Carlo step by S, each agent generates a set of artificial data points according to agents' probability distributions, and denote those by A_i. Then we define the following quantity:

$$\bar{G}_{x_i} \;:=\; \frac{1-\alpha}{|S|} \sum_{x' \in S} \delta(x_i - x'_i) G(x) \qquad (16)$$
$$+\frac{\alpha}{|A_i|} \sum_{x' \in A_i} \delta(x_i - x'_i) \hat{G}_S(x) \qquad (17)$$

where α is a weighting parameter between 0 and 1 and $\hat{G}$ is defined by:

$$\hat{G}_S(x) := \frac{\sum_{x' \in S} d(x, x') G(x')}{\sum_{x' \in S} d(x, x')} \qquad (18)$$

where $d(\ .\ ,\ .\ )$ is some appropriate metric. In the present 100-agent 4-night bar problem, $d(x, x') := e^{-2 \times \sum_{k=1}^{4} |f_k(x) - f_k(x')|}$ where the functions $\{f_k(.)\}$ are as defined in eq. 15.

4. Each agent updates her probability distribution according to the gradients calculated as in eq. 9 but with $[G|x_i]$ replaced by $\bar{G}_{x_i}$. Again, for each agent i, the

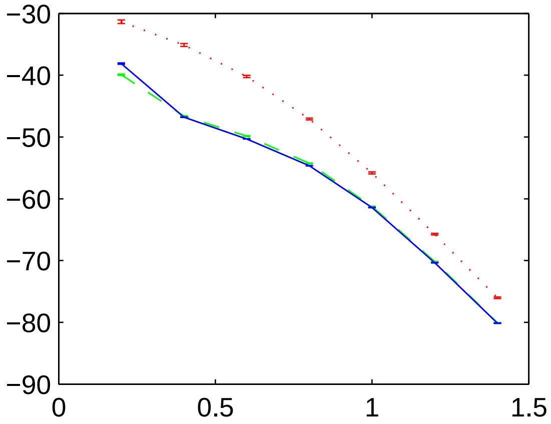

Figure 1. Plots of β^{-1} vs. the Lagrangian for different utilities. The curves are generated by plotting the Lagrangian at the 20th timestep, i.e., after 20 descents. The initial step sizes are set to be 0.2 times the gradients. Also, $L = 100$ and a total of 40 simulations are performed. (Red dotted line: Team game, green dashed line: uniform AU, blue solid line: weighted AU.)

magnify of descent is halved if $q_i(x_i)$ is no longer positive for some x_i'.

5. Repeat steps 2 to 4.

Given that the utility function is reasonably smooth, it is natural to expect that the estimates aided by artificial data points will provide an improvement. And this is indeed shown in figure 6 with varying α. The same experiments are also performed for the 50-spin model introduced in section 4.1 but with a new metric $d(x, x') = \sum_{i=1}^{50} \delta(x_i - x_i')$. The results are shown in figure 7.

An immediately improvement to the scheme above is to realize that there is no need to restrict ourselves to data available in that particular time step in calculating $\bar{G}_{x_i}$ in eq. 16. Namely, unlike step 3 above, we can accumulate "true" data from previous steps in calculating $\bar{G}_{x_i}$ which will certainly improve the accuracy of the estimation. To illustrate this idea, at time step t, we define S' to be the set true samples drawn at time step t and with the set of true samples drawn at time step $t-1$, and we calculate $\bar{G}_{x_i}$ as:

$$\bar{G}_{x_i} := \frac{1-\alpha}{|S|} \sum_{x' \in S} \delta(x_i - x_i') G(x) \tag{19}$$

$$+ \frac{\alpha}{|A_i|} \sum_{x' \in A_i} \delta(x_i - x_i') \hat{G}_{S'}(x) \,. \tag{20}$$

The results for the bar problem are shown in figure 8.

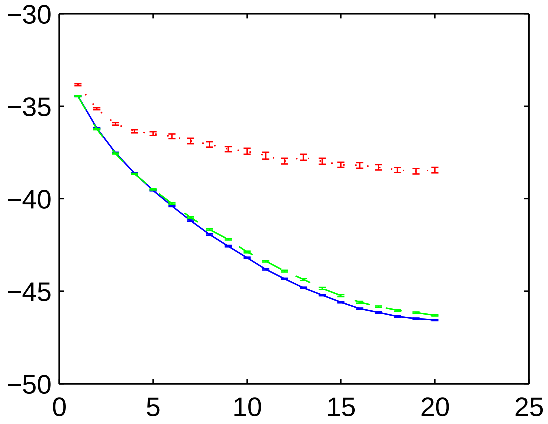

Figure 2. The time series of Lagrangian along the 20 time steps (curves generated at $\beta^{-1} = 0.6$). (Red dotted line: Team game, green dashed line: uniform AU, blue solid line: weighted AU.)

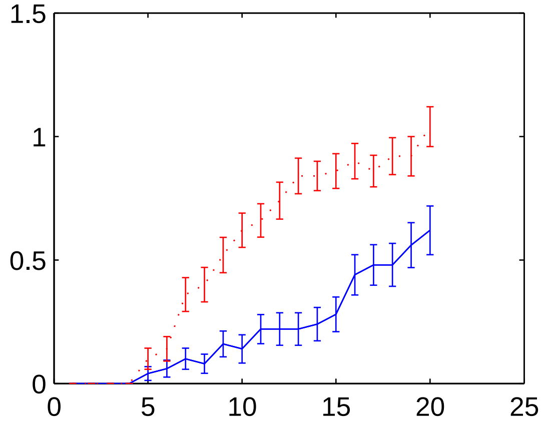

Figure 3. Plots of time step versus $\frac{1}{50} \sum_i \sum_{x_i \in \sigma_i} \delta(L_{x_i} - 1)$ at different temperatures. (Red dotted line: $\beta^{-1} = 0.2$, blue solid line: $\beta^{-1} = 0.6$.)

6. Conclusion

Product Distribution (PD) theory is a recently introduced broad framework for analyzing, controlling, and optimizing distributed systems [9, 10, 11]. Here we investigate PD theory's use for adaptive, distributed control of a MAS. Typically such control is done by having each agent run its own reinforcement learning algorithm [4, 13, 14, 12].

In this approach the utility function of each agent is based on the world utility $G(x)$ mapping the joint move of

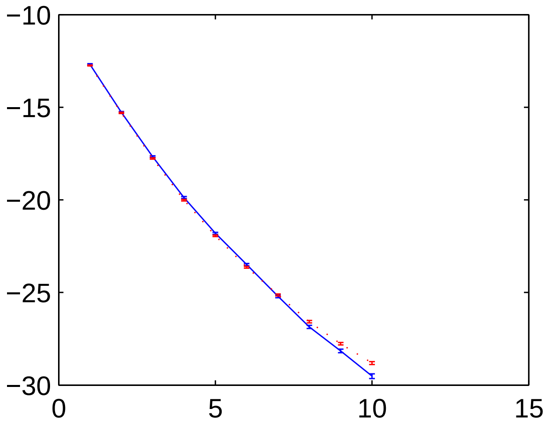

Figure 4. The time series of Lagrangian along the 20 time steps with sample size $L = 100$ (curves generated at $\beta^{-1} = 0.2$). (Red dotted line: correct WLU, blue solid line: incorrect WLU.)

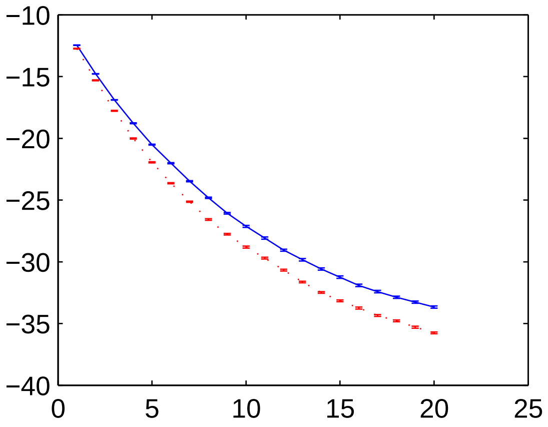

Figure 5. The time series of Lagrangian along the 20 time steps with sample size $L = 200$ (curves generated at $\beta^{-1} = 0.2$). (Red dotted line: correct WLU, blue solid line: incorrect WLU.)

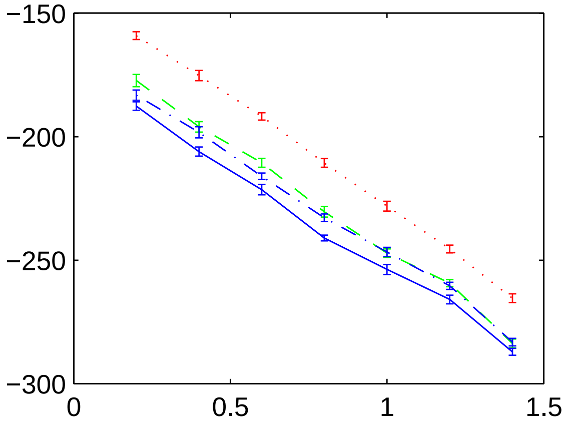

Figure 6. Plots of β^{-1} vs. the Lagrangian with various weighing parameters α for the bar problem. The curves are generated by plotting the Lagrangian at the 20th timestep. (Red dotted line: no data augmentation, green dashed line: data aug. with $\alpha = 0.3$, blue solid line: data aug. with $\alpha = 0.5$, black dashed dotted line: data aug. with $\alpha = 0.7$.)

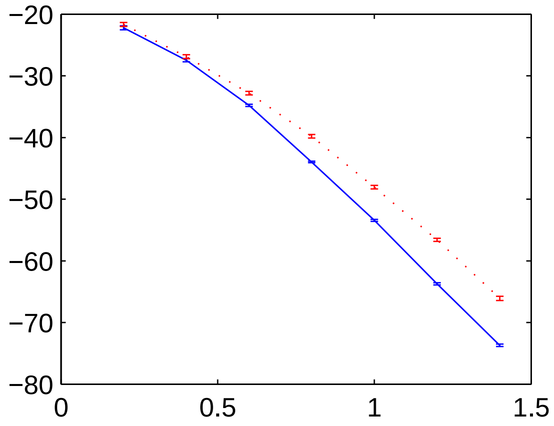

Figure 7. Plots of β^{-1} vs. the Lagrangian for the 50-spin model. The curves are generated by plotting the Lagrangian at the 20th timestep. (Red dotted line: no data augmentation, blue solid line: data aug. with $\alpha = 0.5$.)

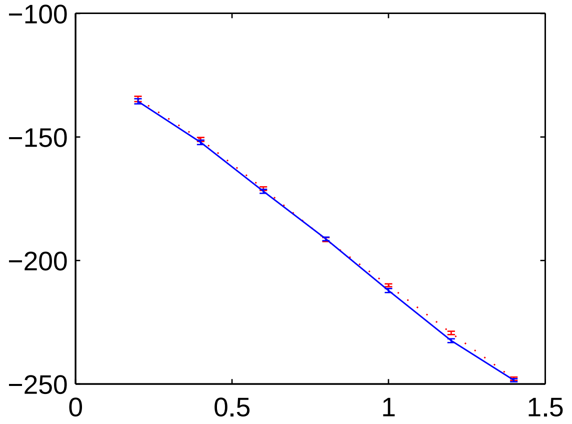

Figure 8. Plots of β^{-1} vs. the Lagrangian with data accumulation (blue line) and without data accumulation (red dotted line) for the bar problem. The curves are generated by plotting the free energies at the 10th timestep. The initial step sizes are set to be 0.2 times the gradients. Also, the number of true data $|S|$ is 20, the number of data stored is 20 and the number of artificial data $|A_i|$ is 40. A total of 80 simulations are performed.

the agents, $x \in X$, to the performance of the overall system. However in practice the agents in a MAS are bounded rational. Moreover the equilibrium they reach will typically involve mixed strategies rather than pure strategies, i.e., they don't settle on a single point x optimizing $G(x)$. This suggests formulating an approach that explicitly accounts for the bounded rational, mixed strategy character of the agents.

PD theory directly addresses these issues by casting the control problem as one of minimizing a Lagrangian of the joint probablity distribution of the agents. This allows the equilibrium to be found using gradient descent techniques. In PD theory, such gradient descent can be done in a distributed manner.

We present experiments validating PD theory's predictions for how to speed the convergence of that gradient descent. We then present other experiments validating the use of PD theory to improve convergence even if one is not allowed to rerun the system (an approach common in RL-based schemes). These results demonstrate the power of PD theory for providing a principled way to control a MAS in a distributed manner.

Acknowledgements CFL thanks NASA and NSERC (Canada) for financial support.

References

[1] S. Airiau and D. H. Wolpert. Product distribution theory and semi-coordinate transformations. 2004. Submitted to AAMAS 04.

[2] N. Antoine, S. Bieniawski, I. Kroo, and D. H. Wolpert. Fleet assignment using collective intelligence. In *Proceedings of 42nd Aerospace Sciences Meeting*, 2004. AIAA-2004-0622.

[3] S. Bieniawski and D. H. Wolpert. Adaptive, distributed control of constrained multi-agent systems. 2004. Submitted to AAMAS 04.

[4] R. H. Crites and A. G. Barto. Improving elevator performance using reinforcement learning. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems - 8*, pages 1017–1023. MIT Press, 1996.

[5] D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. MIT Press, Cambridge, MA, 1998.

[6] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

[7] D. Mackay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[8] S. Bieniawski W. Macready and D.H. Wolpert. Adaptive multi-agent systems for constrained optimization. 2004. Submitted to AAAI 04.

[9] D. H. Wolpert. Factoring a canonical ensemble. 2003. cond-mat/0307630.

[10] D. H. Wolpert. Bounded rational games, information theory, and statistical physics. In D. Braha and Y. Bar-Yam, editors, *Complex Engineering Systems*, 2004.

[11] D. H. Wolpert. Generalizing mean field theory for distributed optimization and control. 2004. Submitted.

[12] D. H. Wolpert and K. Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(2/3):265–279, 2001.

[13] D. H. Wolpert and K. Tumer. Collective intelligence, data routing and braess' paradox. *Journal of Artificial Intelligence Research*, 2002.

[14] D. H. Wolpert, K. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters*, 49(6), March 2000.