Designing Agent Collectives For Systems With Markovian Dynamics

Paper ID:546

David H. Wolpert
NASA Ames Research Center
MS 269-2
Moffett Field, CA 94035
dhw@ptolemy.arc.nasa.gov

John W. Lawson
NASA Ames Research Center
MS 269-2
Moffett Field, CA 94035
lawson@ptolemy.arc.nasa.gov

ABSTRACT

The "Collective Intelligence" (COIN) framework concerns the design of collectives of agents so that as those agents strive to maximize their individual utility functions, their interaction causes a provided "world" utility function concerning the entire collective to be also maximized. Here we show how to extend that framework to scenarios having Markovian dynamics when no re-evolution of the system from counter-factual initial conditions (an often expensive calculation) is permitted. Our approach transforms the (time-extended) argument of each agent's utility function before evaluating that function. This transformation has benefits in scenarios not involving Markovian dynamics, in particular scenarios where not all of the arguments of an agent's utility function are observable. We investigate this transformation in simulations involving both linear and quadratic (nonlinear) dynamics. In addition, we find that a certain subset of these transformations, which result in utilities that have low "opacity (analogous to having high signal to noise) but are not "factored" (analogous to not being incentive compatible), reliably improve performance over that arising with factored utilities. We also present a Taylor Series method for the fully general nonlinear case.

1. INTRODUCTION

1.1 Background

In this paper we are concerned with large distributed collectives of interacting goal-driven computational processes, where there is a provided 'world utility' function that rates the possible behaviors of that collective [29, 27]. We are particularly concerned with such collectives where the individual computational processes use machine learning techniques (e.g., Reinforcement Learning (RL) [14, 20, 19, 23]) to try to achieve their individual goals. We represent those goals of the individual processes as maximizing an associated 'payoff' utility function, one that in general can differ from the world utility.

In such a system, we are confronted with the following inverse problem: *How should one initialize/update the payoff utility functions of the individual processes so that the ensuing behavior of the entire collective achieves large values of the provided world utility?* In particular, since in truly large systems detailed modeling of the system is usually impossible, how can we avoid such modeling? Can we instead leverage the simple assumption that our learnering algorithms are individually fairly good at what they do to achieve a large world utility value?

This problem is related to work in many other fields, including multi-agent systems (MAS's), computational economics, mechanism design, reinforcement learning, statistical mechanics, computational ecologies, (partially observable) Markov decision processes and game theory. However none of these fields is both applicable in large problems, and directly addresses the *general* inverse problem, rather than a special instance of it. (See [27] for a detailed discussion of the relationship between these fields, involving hundreds of references.) For example, the field of mechanism design is not generally applicable, being largely tailored to collectives of human beings, and in particular to the idiosyncracy of such collectives that their members have hidden variables whose values they "do not want to reveal". There is other previous work that does consider the general inverse problem, and even has each individual computational process (or "agent") use reinforcement learning [2, 7, 10, 15, 16]. However, in that work in general each process has the world utility function as its payoff utility function (i.e., implements a "team game" or an "exact potential game" [8]). Unfortunately, as expounded below and in previous work, this approach scales extremely poorly to large problems. (Intuitively, the difficulty is that each agent can have a hard time discerning the echo of its behavior on the world utility when the system is large; each agent has a horrible "signal-to-noise" problem.)

Intuitively, we are concerned with payoff utility functions that are "aligned" with the world utility, in that modifications a player might make that would improve its payoff utility also must improve world utility.[1] Fortunately the

---

[1]Such alignment can be viewed as an extension of the concept of incentive compatibility in mechanism design [9] to

equivalence class of such payoff utilities extends well beyond team-game utilities. In particular, in previous work we used the COllective INtelligence (COIN) framework to derive the 'Wonderful Life Utility' (WLU) payoff function [27] as an alternative to a team-game payoff utility. The WLU is aligned with world utility, as desired. In addition though, WLU overcomes much of the signal-to-noise problem of team game utilities [22, 29, 27, 31].

As an example, in some of our previous work we used the WLU for distributed control of network packet routing [29]. Conventional approaches to packet routing have each router run a shortest path algorithm (SPA). Unlike with a WLU-based collective, in SPA-based routing there is no concern for deleterious side-effects of routing decisions on the global goal (e.g., no concern for bottlenecks). We ran simulations demonstrating that a WLU-based collective has substantially better throughputs than the best possible SPA-based system [29], even though that SPA-based system had information denied the COIN system.

As another example, in [30] we considered the pared-down problem domain of a congestion game, in particular a more challenging variant of Arthur's El Farol bar attendance problem [1], sometimes also known as the "minority game" [6]. In this problem the individual processes making up the collective are explicitly viewed as 'players' involved in a non-cooperative game. Each player has to determine which night in the week to attend a bar. The problem is set up so that if either too few people attend (boring evening) or too many people attend (crowded evening), the total enjoyment of the attending players drops. Our goal is to design the payoff functions of the players so that the total enjoyment across all nights is maximized. In this previous work we showed that use of the WLU can result in performance *orders of magnitude* superior to that of team game utilities.

1.2 The Contribution of This Paper

In this paper we extend this previous work with an approach based on Transforming Arguments Utility functions (TAU) before the evaluation of those functions. The TAU process was originally designed to be applied to the individual utility functions of the agents in systems in which the world utility depends on the final state in an episode of variables outside the collective that undergo Markovian dynamics, with the update rule of those variables reflecting the state of the agents at the beginning of the episode. This is a very common scenario, obtaining whenever the agents in the collective act as control signals perturbing the evolution of a Markovian system.

In the previous version of the COIN framework, to achieve good signal-to-noise for such scenarios might require re-evolving the system from counter-factual initial states of the agents to evaluate each agent's reward for a particular episode. This can be computationally expensive. With TAU utility functions no such re-evolving is needed; the observed history of the system in the episode is transformed in a relatively cheap calculation, and then the utility function is evaluated with that transformed history rather than the actual one.

The TAU process has other advantages that apply even in scenarios not involving Markovian dynamics. In particular it allows us to employ the COIN framework even when not all arguments of the original utility function are observable, due for example to communication limitations. In addition, non-human agents, off-equilibrium behavior, etc.

certain types of TAU transformations result in utility functions that are not exactly aligned with the world utility, but have so much better signal-to-noise that the collective performs better when agents use those transformed utility functions than it does with exactly aligned utility functions.

In this paper computational experiments based on the Markovian and quadratic (nonlinear) update rules for the foregoing. In particular, in these experiments, we consider systems of 50 agents using a variety of world utilities and Markovian update rules. We compare the performance of using TAU utilities for the agents for linear and quadratic dynamics versus the performance using the corresponding team game utilities. We can also investigate systems having limited observability. In these cases, the performance with TAU utilities even robustly outperforms that of team game utilities in which there is full observability. We also find that the non-aligned, high signal-to-noise utilities consistently outperform their factored counterparts. We end with results using a Taylor Series method to address the more general nonlinear case than the quadratic one investigated here.

2. THE MATHEMATICS OF COLLECTIVE INTELLIGENCE

We view the individual agents in the collective as players involved in a repeated game.[2] Let Z with elements ζ be the space of possible joint moves of all players in the collective in some stage. We wish to search for the ζ that maximizes in some stage. We wish to search for the ζ that maximizes a provided **world utility** $G(\zeta)$. In addition to G we are concerned with utility functions $\{g_\eta\}$, one such function for each variable/player η. We use the notation $\hat{\eta}$ to refer to all players other than η.

2.1 Intelligence and the central equation

We wish to "standardize" utility functions so that the numeric value they assign to a ζ only reflects their ranking of ζ relative to certain other elements of Z. We call such a standardization of an arbitrary utility U for player η the "intelligence for η at ζ with respect to U". Here we will use intelligences that are equivalent to percentiles:

$$\epsilon_U(\zeta : \eta) \equiv \int d\mu_{\zeta_{\hat{\eta}}}(\zeta')\Theta[U(\zeta) - U(\zeta')] , \qquad (1)$$

where the Heaviside function Θ is defined to equal 1 when its argument is greater than or equal to 0, and to equal 0 otherwise, and where the subscript on the (normalized) measure $d\mu$ indicates it is restricted to ζ' sharing the same non-η components as ζ. In general, the measure must reflect the type of system at hand, e.g., whether Z is countable or not, and if not, what coordinate system is being used. Other than that, any convenient choice of measure may be used and the theorems will still hold. Intelligence value are always between 0 and 1.

Our uncertainty concerning the behavior of the system is reflected in a probability distribution over Z. Our ability to control the system consists of setting the value of some characteristic of the collective, e.g., setting the functions of the players. Indicating that value by s, our analysis revolves

---

[2]The full mathematics of the COIN framework, however, extends significantly beyond what is needed to address such games. See [28].

around the following central equation for $P(G \mid s)$, which follows from Bayes' theorem:

$$P(G \mid s) = \int d\vec{\epsilon}_G P(G \mid \vec{\epsilon}_G, s) \int d\vec{\epsilon}_g P(\vec{\epsilon}_G \mid \vec{\epsilon}_g, s) P(\vec{\epsilon}_g \mid s) , \quad (2)$$

where $\vec{\epsilon}_g \equiv (\epsilon_{g_{\eta_1}}(\zeta : \eta_1), \epsilon_{g_{\eta_2}}(\zeta : \eta_2), \cdots)$ is the vector of the intelligences of the players with respect to their associated functions, and $\vec{\epsilon}_G \equiv (\epsilon_G(\zeta : \eta_1), \epsilon_G(\zeta : \eta_2), \cdots)$ is the vector of the intelligences of the players with respect to G.

Note that $\epsilon_{g_\eta}(\zeta : \eta) = 1$ means that player η is fully rational at ζ, in that its move maximizes its utility, given the moves of the players. In other words, a point ζ where $\epsilon_{g_\eta}(\zeta : \eta) = 1$ for all players η is one that meets the definition of a game-theory Nash equilibrium [9]. Note that consideration of points ζ at which not all intelligences equal 1 provides the basis for a model-independent formalization of bounded rationality game theory, a formalization that contains variants of many of the theorems of conventional full-rationality game theory [25]. On the other hand, a ζ at which all components of $\vec{\epsilon}_G = 1$ is a local maximum of G (or more precisely, a critical point of the $G(\zeta)$ surface).

If we can choose s so that the third conditional probability in the integrand is peaked around vectors $\vec{\epsilon}_g$ all of whose componentsare close to 1, then we have likely induced large intelligences. If in addition the second term is peaked about $\vec{\epsilon}_G$ equal to $\vec{\epsilon}_g$, then $\vec{\epsilon}_G$ will also be large. Finally, if the first term is peaked about high G when $\vec{\epsilon}_G$ is large, then our choice of s will likely result in high G, as desired.

Intuitively, the requirement that the utility functions have high "signal-to-noise" (an issue not considered in conventional work in mechanism design) arises in the third term. It is in the second term that the requirement that the utility functions be "aligned with G" arises. In this work we concentrate on these two terms, and show how to simultaneously set them to have the desired form.

Details of the stochastic environment in which the collective operates, together with details of the learning algorithms of the players, are reflected in the distribution $P(\zeta)$ which underlies the distributions appearing in Equation 2. Note though that *independent of these considerations*, our desired form for the second term in Equation 2 is assured if we have chosen utility utilities such that $\vec{\epsilon}_g$ equals $\vec{\epsilon}_G$ exactly *for all* ζ. We call such a system *factored*. In game-theory language, the Nash equilibria of a factored collective are local maxima of G. In addition to this desirable equilibrium behavior, factored collectives automatically provide appropriate off-equilibrium incentives to the players (an issue rarely considered in game theory / mechanism design).

2.2 Opacity

We now focus on algorithms based on utility functions $\{g_\eta\}$ that optimize the signal/noise ratio reflected in the third term, subject to the requirement that the system be factored. To understand how these algorithms work, given a measure $d\mu(\zeta_\eta)$, define the **opacity** at ζ of utility U as:

$$\Omega_U(\zeta : \eta, s) \equiv \int d\zeta' J(\zeta' \mid \zeta) \frac{|U(\zeta) - U(\zeta'_\eta, \zeta_\eta)|}{|U(\zeta) - U(\zeta_\eta, \zeta'_\eta)|} , \quad (3)$$

where J is defined in terms of the underlying probability distributions,[3] and $(\zeta'_\eta, \zeta_\eta)$ is defined as the worldline whose η components are the same as those of ζ' while its η components are the same as those of ζ ([28]).

The denominator absolute value in the integrand in Equation 3 reflects how sensitive $U(\zeta)$ is to changing ζ_η. In contrast, the numerator absolute value reflects how sensitive $U(\zeta)$ is to changing $\zeta_{\hat\eta}$. So the smaller the opacity of a utility function g_η, the more $g_\eta(\zeta)$ depends only on the move of player η, i.e., the better the associated signal-to-noise ratio for η. Intuitively then, lower opacity should mean it is easier for η to achieve a large value of its intelligence.

To formally establish this, we use the same measure $d\mu$ to define opacity as the one that defined intelligence. Under this choice expected opacity bounds how close to 1 expected intelligence can be [28]:

$$E(\epsilon_U(\zeta : \eta) \mid s) \leq 1 - K, \text{where}$$
$$K \leq E(\Omega_U(\zeta : \eta, s) \mid s). \quad (5)$$

So low expected opacity of utility g_η ensure that a necessary condition is met for the third term in Equation 2 to have the desired form for player η. While low opacity is not, formally speaking, also sufficient for $E(\epsilon_U(\zeta : \eta) \mid s)$ to be close to 1, in practice the bounds in Equation 5 are usually tight.

2.3 Difference Utilities

It is possible to solve for the set of all utilities that are factored with respect to a particular world utility. Unfortunately, in general it is not possible for a collective both to be factored and to have zero opacity for all of its players. However consider **difference** utilities, which are of the form

$$U(\zeta) = G(\zeta) - \Gamma(f(\zeta)) \quad (6)$$

where $\Gamma(f)$ is independent of ζ_η. Any difference utility is factored [26], and under benign approximations, $E(\Omega_u \mid s)$ is minimized over the set of such utilities by choosing

$$\Gamma(f(\zeta)) = E(G \mid \zeta_{\hat\eta}, s) , \quad (7)$$

up to an overall additive constant. We call the resultant difference utility the **Aristocrat** utility (AU), loosely reflecting the fact that it measures the difference between a player's actual action and the average action.

If possible, we would like each player η to use the associated AU as its utility function to ensure good form for both terms 2 and 3 in Equation 2. This is not always feasible however. The problem is that to evaluate the expectation value defining its AU each player needs to evaluate the current probabilities of each of its potential moves. However if the player then changes its utility function to be the associated AU it will in general substantially change its ensuing behavior. (The player now wants to choose moves that maximize a different function from the one it was maximizing before.) In other words, it will change the probabilities of its moves, which means that its new utility function is in fact not the AU for its actual (new) probabilities.

There are ways around this self-consistency problem, but in practice it is often easier to bypass the entire issue, by

---

[3] Writing it out in full, $J(\zeta' \mid \zeta) \equiv J(\zeta_\eta, \zeta' \mid \zeta_{\hat\eta}, s)/P(\zeta \mid$ $\zeta_{\hat\eta}, s)$, with:

$$J(\zeta_\eta, \zeta' \mid \zeta_{\hat\eta}, s) \equiv \frac{P(\zeta_\eta \mid \zeta_{\hat\eta}, s) P(\zeta'_\eta \mid \zeta_{\hat\eta}, s) \mu(\zeta'_\eta)}{2} + \quad (4)$$
$$\frac{P(\zeta'_\eta \mid \zeta_{\hat\eta}, s) P(\zeta_\eta \mid \zeta'_{\hat\eta}, s) \mu(\zeta_\eta)}{2} .$$

giving each η a utility function that does not depend on the probabilities of η's own moves. One such utility function is the **Wonderful Life** Utility (WLU). The WLU for player η is parameterized by a pre-fixed **clamping parameter** CL_η chosen from among η's possible moves:

$$WLU_\eta \equiv G(\zeta) - G(\zeta_\eta, CL_\eta) . \tag{8}$$

WLU is factored regardless of the choice of clamping parameter. Furthermore, while not matching AU's low opacity, WLU usually has far better opacity than does a team game.

3. THE COIN FRAMEWORK FOR SYSTEMS WITH MARKOVIAN EVOLUTION

We consider games which consist of multi-step "episodes". Within each episode the entire system evolves in a Markovian manner from the initial moves of the players. We are interested in such games where some of the players η are not agents whose intial state is under control of a learning algorithm that we control, but rather constitute an "environment" for those controllable agents (i.e., where some of the players correspond to the state of nature).

Let A be the Markovian single step evolution operator of the entire system through an episode,

$$\vec{\zeta}_t = A\vec{\zeta}_{t-1} \tag{9}$$

Each component ζ_t^η, for example, could be a one-dimensional real number. The row vector A^η would then be η's update rule. Alternatively, each agent could be represented by one of N symbolic values. In that case, $\vec{\zeta}_t$ would be given in a unary representation as a vector in $\mathcal{R}^{N^{|\eta|}}$ (i.e. a Haar basis). Considering such large spaces are necessary to describe arbitrary, nonlinear dynamics as Markovian evolution. Here we will concentrate on the former case, where the moves of the players are all real numbers.

The full multiple time step evolution of an episode is given by single step operator in the usual way: Let

$$C = \begin{bmatrix} A \\ A^2 \\ A^3 \\ . \\ . \\ A^T \end{bmatrix}$$

where T is the number of time steps per episode. This operator applied to our initial state $\vec{\zeta}_0$ yields the entire "worldline" $\vec{\zeta}$, or time history, of the system.

$$\vec{\zeta} = C\vec{\zeta}_0. \tag{10}$$

We consider difference utility functions of the form

$$g_\eta(\vec{\zeta}) = G(C\vec{\zeta}_0) - \Gamma_\eta(F_\eta C\vec{\zeta}_0) \tag{11}$$

where G is the world utility function to be optimized. We will choose F_η so that the product $F_\eta C\vec{\zeta}_0$ is independent of agent η's actions. This is a necessary and sufficient condition for the associated difference utility $g_\eta(\vec{\zeta})$ to be factored with respect to the world utility G for any and all choices of Γ_η. In general, Γ_η can be chosen in such a way to optimize learnability. Here though, for simplicity, we choose $\Gamma_\eta = G$. Accordingly, application of the F_η operator is an instance of transforming the argument of the (second term of the) utility functions of the agents, i.e., it is a TAU process.

It is important to note that the particular form of C given above is not necessary for the results and methods of this paper to apply. In fact, there is no reason even to view the $COIN$-based choice of the g_η as optimizing G for a multi-step game involving a "dynamics" process. It can be viewed as simply optimizing some $G(C\vec{\zeta}_0)$ for some "abstract" function C. As we will see, a major advantage of our approach to optimizing functions of this form is that C only needs to be run once to set the values of g_η. Moreover, this single running of C is automatically done "by nature" as the system runs. There is no extra burden on the individual agents to perform calculations involving C, for example, to evaluate outcomes of counter-factual moves.

4. LINEAR EVOLUTION

4.1 Avoiding re-evolution of the system

We now consider the operator F_η for the case of linear evolution (i.e. C is a linear operator). For simplicity episodes are composed of one time step (i.e. $C = A$), and agents initially exist in one of two states (i.e., the players under our control can make one of two moves). As mentioned above, a sufficient condition for η's difference utility g_η to be factored is that the combination $F_\eta C\vec{\zeta}_0$ is independent of η's initial action. One way to accomplish this starts by clamping η's initial action, producing $\hat{C}L_\eta\vec{\zeta}_0$, where $\hat{C}L_\eta$ is a clamping operator represented by a decimated identity matrix with zero-valued diagonal element at position η. This clamped state must then be re-evolved to produce the desired combination, $C(\hat{C}L_\eta\vec{\zeta}_0)$.

Unfortunately doing this means re-evolving the entire system, which may be computationally prohibitive, especially if it must be done for each agent. We define, therefore, a post-evolution clamping operator F_η such that $F_\eta C = C(\hat{C}L_\eta)$, and therefore no re-evolving is needed once $C\vec{\zeta}_0$ has been evaluated (by nature). It follow that

$$F_\eta = C(\hat{C}L_\eta)C^{-1} . \tag{12}$$

The spectral structure of the operator F_η is readily determined. The eigenvalues are $\lambda_k^\eta = 1 - \delta_{k,\eta}$ where $\delta_{i,j}$ is the Kronecker delta. Corresponding eigenvectors are $\vec{e}_k^\eta = \vec{c}_k$ where $\{\vec{c}_k\}$ are the columns of the linear evolution operator C. Since they span the space the post-evolved state can be expanded in terms of these eigenvectors of F_η:

$$\vec{\zeta}_t = \sum_k a_k \vec{e}_k^\eta. \tag{13}$$

Application of F_η to the post-evolved state in this basis is straightforward. The result is $F_\eta\vec{\zeta}_t = \vec{\zeta}_t - a_\eta\vec{c}_\eta$ where a_η is the projection of $\vec{\zeta}_t$ in the direction of $\vec{c}_\eta$. Furthermore, since eigenvectors of F_η correspond to columns of C, the matrix C^{-1} acts as a projector onto this basis. Using this fact and recalling that $\vec{\zeta}_t = C\vec{\zeta}_t$, it can be shown that $a_\eta = \zeta_0^\eta$ i.e. it equals agent η's action at $t = 0$. Thus, F_η can be completely expressed in terms of observed post-evolution quantities:

$$F_\eta\vec{\zeta}_t = \vec{\zeta}_t - \zeta_0^\eta\vec{c}_\eta. \tag{14}$$

In this way we can calculate the result of clamping the initial state and re-evolving without performing that re-evolution.
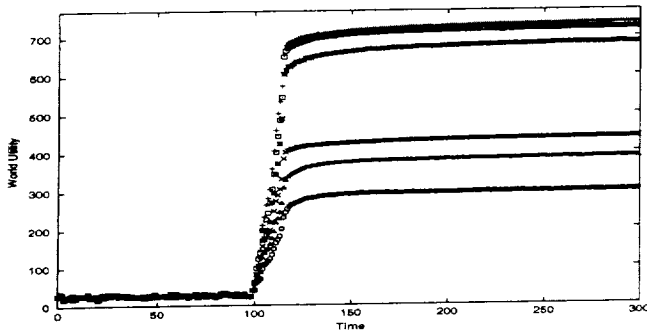
Figure 1: Performance for 50 agents with linear dynamics when the environment is set to zero at the beginning of each episode. Results for TAU g are represented by +, results for 75% observability TAU, $g^{75\%}$, are □, then applying L to the first as well as second terms gives the utility $g_{nf}^{75\%}$, with results depicted as *. $g^{25\%}$ is ×, $g_{nf}^{25\%}$ is △, and finally, G, the team game, is ⊙. Errorbars are too small to see.
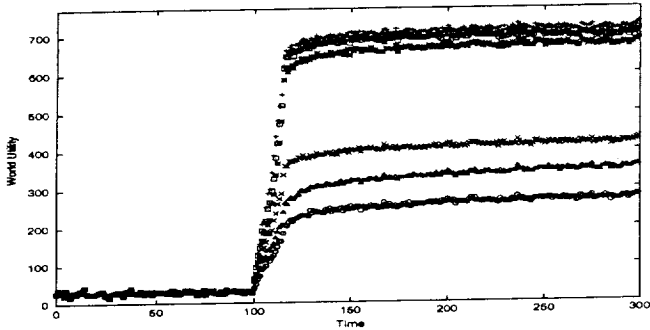

Figure 2: System performance for 50 agents with linear dynamics in a random environment. Key is same as Figure 1. The small degradation in performance due to randomness from the environment.

4.2 Observability restrictions

In practice, the full worldline of the system may not be fully observable to each agent. Such limited observability of a particular component may be determined by the problem. In other cases, due to communication constraints each agent is only allowed to observe a certain number of components, and must select which such components to observe, for example to optimize some auxiliary quantity like opacity. Similarly, the dynamics may not be known exactly to the agent; some rows of C may be uncertain to an agent, or simply cannot be determined. In these kinds of situations the g_η described above cannot be evaluated at the end of an episode by agent η, even if the value $G(\vec{\zeta_t})$ is globally broadcast to all agents.

The TAU approach outlined above is well-suited to address such situations. Formally, a decimated identity operator L can be defined whose diagonal elements are $\{0,1\}$ depending on whether or not they are observable. The corresponding factored utility for agent η is

$$g_\eta(\vec{\zeta_t}) = G(\vec{\zeta_t}) - G(LF_\eta\vec{\zeta_t}), \qquad (15)$$

where in general L may vary with η. Given global broadcast
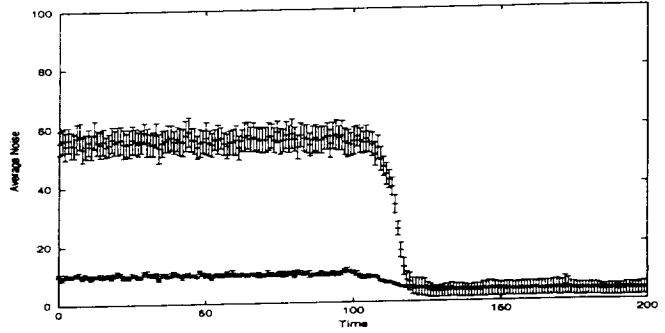

Figure 3: Comparison of average noise for factored $g^{75\%}$ (upper graph) and nonfactored $g_{nf}^{75\%}$ (lower graph) utility functions with 75 % observability. The first 100 time steps are the training period.

to all agents of the value of $G(\vec{\zeta_t})$, for each agent to evaluate this type of g_η only requires that those components of $F_\eta\vec{\zeta_t}$ that are non-zero (and therefore can vary) after application of the L operator be observed.

This difference utility has two main sources of noise, one from potentially poor choice of the clamping operator, and the other from the use of L in the second (subtracted) term but not in the first. To address that latter source of noise we can impose limited observability on the first term in addition to the second one, getting

$$g_\eta(\vec{\zeta_t}) = G(L\vec{\zeta_t}) - G(LF_\eta\vec{\zeta_t}). \qquad (16)$$

The new utility is not factored with respect to G. According to the central equation however, it may still result in better performance than when we don't have L in the first term, if the improvement in opacity more than offsets the loss of exact factoredness. In addition to the potential for such far superior opacity, this utility has the added advantage that now we don't even need to rely on global broadcast of $G(L\vec{\zeta_t})$ to evaluate g_η.

4.3 Experiments

Numerical simulations were performed with 50 agents. After an initial 100-episode training period, agents selected initial actions in each subsequent episode with the same reinforcement learning algorithm used in our previous work. All players underwent linear dynamics within each episode. The world utility function was a spin glass,

$$G_T = \sum_{i<j} J_{ij}\zeta_T^i\zeta_T^j. \qquad (17)$$

We collected statistics by averaging runs over many randomly set matrices A and coupling constants J_{ij}. These runs were for systems whose first 25% and 75% components at the end of the episode are observable, given some canonical ordering of agents. We considered both the case where the environment was initialized to zero (Figure 1) and where it was initialized randomly (Figure 2). We examined world utility value vs. episode number for six utility functions:
  1) TAU g for a fully observable system;
  2) TAU g for 75 % observability, $g^{75\%}$;
  3) The modification $g_{nf}^{75\%}$ giving a non-factored system, again with 75 % observability;
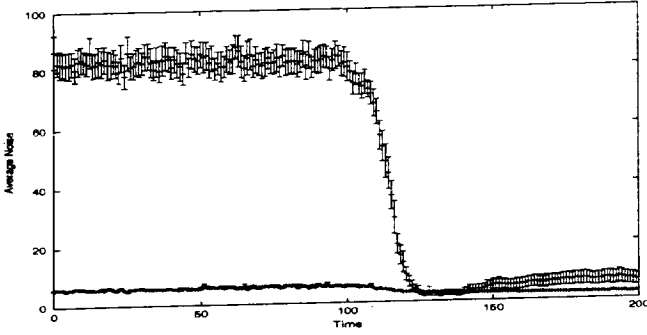 4) $g^{25\%}$ for a factored system with 25 % observability;

Figure 4: Comparison of average noise for factored $g^{25\%}$ (upper graph) and nonfactored $g_{nf}^{25\%}$ (lower graph) utilities with 25 % observability.
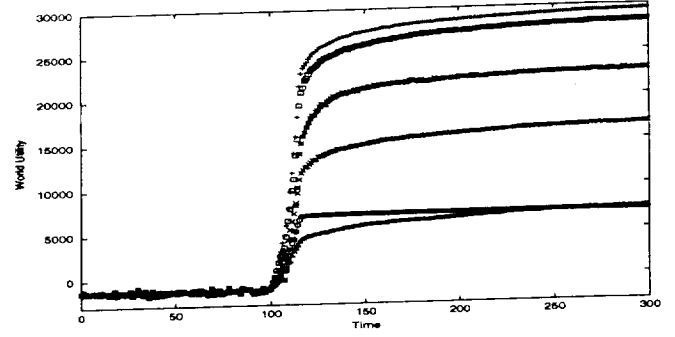

Figure 5: System performance for 50 agents executing quadratic dynamics. The environment has been initialized to zero. Key is same as in Figure 1.

5) $g_{nf}^{25\%}$ for a non-factored system with 25 % observability;

6) The team game, where every $g_\eta = G$.

Even the results for limited observability clearly outperform the corresponding team game in which there is full observability. Furthermore, for 75% observability, the nonfactored utilities (L in both terms) consistently outperform their factored counterpart. In these runs factoredness fell to approximately 90%, while noise levels in the utility functions were as shown in figures 3 and 4. The improvement in performance due to better signal-to-noise more than outweighs the degradation due to loss in factoredness.

5. NONLINEAR EVOLUTION

Generalizing these results to arbitrary nonlinear dynamics requires high dimensional representations. In particular, in the case where all agents' states are binary, the number of joint states grows as 2^N where N is the number of agents. The successive bits in such a representation can be indicated as $\{x_i\} \in \mathcal{B} \equiv \{-\infty, \infty\}$ where we have N bits altogether. Alternatively, we can expand the joint state in the basis of Walsh functions $(1, \{x_i\}, \{x_i x_{j \neq i}\}, ...)$ which spans the set of all functions taking elements of the space $\mathcal{B}$ to $\mathcal{B}$.

Doing this reduces the original nonlinear dynamics to linear dynamics, at the price of expanding the size of the space. As an example, in the case of a quadratic update rule, we can represent ζ_0 in terms of second order Walsh functions $\{x_i x_{j \neq i}\}$. Evolution of the system is accomplished by application of the associated evolution matrix C or A, yielding $\zeta_t = C\vec{\zeta}_0$. To obtain factored utility functions, analogous post-evolution operators F_η can be constructed. To ensure that the second term in the difference utility is independent of η, all terms involving x_η will have to be subtracted. In the quadratic case, N such terms will have to be subtracted whereas in the linear case there was only one term. We find

$$F_\eta \vec{\zeta}_t = \vec{\zeta}_t - \zeta_0^\eta \sum_i^N \zeta_0^i \vec{c}_{i,\eta} \qquad (18)$$

where $\vec{c}_{i,\eta}$ is the column of C corresponding to the Walsh function $x_i x_\eta$, $i = 1, .., N$. Results of experiments for this case with 50 agents are presented in Figure 5.

6. TAYLOR SERIES METHOD

To address the more general nonlinear problem, we consider a slightly different framework. In this case, each agent

is assigned a real-valued number r_η. The state of the system $\vec{\zeta}_t$ is a vector with these numbers as components. Each agent can choose among three actions which results in r_η being modified by $\{\pm\Delta, 0\}$. Nonlinear update rules $\vec{\zeta}_t = \vec{c}(\vec{\zeta}_0)$ are functions of these real-valued variables.

Construction of factored utilities

$$g_\eta(\vec{\zeta}_t) = G(\vec{c}(\vec{\zeta}_0)) - G(\vec{c}(\hat{C}\hat{L}\vec{\zeta}_0)). \qquad (19)$$

requires that $\vec{c}(\vec{\zeta}_0)$ be independent of η choice of action. One way to accomplish this to clamp (apply $\hat{C}\hat{L}$) to $\vec{\zeta}_0$ and reevolve the system. To avoid re-evolving the system, we approximate $\vec{C}(\hat{C}\hat{L}\vec{\zeta}_0)$ with a Taylor Series expansion about the unclamped $\vec{\zeta}_0$ initial state.

$$\vec{c}(\hat{C}\hat{L}\vec{\zeta}_0) = \vec{c}(\vec{\zeta}_0) + \Delta(\vec{\zeta}_0 - \hat{C}\hat{L}\vec{\zeta}_0) \cdot \vec{\nabla}\vec{c}(\vec{\zeta}_0) \qquad (20)$$

Varying Δ provides us a small parameter to control the expansion. It should be noted that while this method requires that $\vec{c}(\vec{\zeta})$ be differentiable, the world utility G need not be.

Figure 6 presents results for a quadratic update rule with randomly generated coefficients $\vec{c}(\vec{\zeta}_0) = \sum_{i,j} a_{i,j} \zeta_0^i \zeta_0^j$. The agents are given a random initial starting point with $-1 < r_\eta < 1$. Because $\vec{c}$ is quadratic, $G(\vec{\zeta}_t)$ is a quartic polynomial in N dimensions. Since the coefficients $\{a_{i,j}\}$ have random signs, the function G has as many increasing directions as it decreasing directions. The goal of the system is to traverse this high dimensional surface, find an increasing direction, and then follow that direction to infinity.

In light of the central equation we plot the average intelligences of the agents. For three possible actions, the best action has an intelligence of 1 while the worst choice gives 0.33. A random walk (no learning) gives a value of 0.67 on average. We find that a team game has the same intelligence as a random walk. The TAU utility g displays a much higher intelligence which is also reflected in better performance.

It is interesting to adjust the ratio of $\pm$ signs in the coefficients of the polynomials. If we introduce, for example, more negative coefficients than positive, we expect the surface to preferentially turn down. The task for the agents becomes more challenging. We find, in fact, that three of the limited observability utilities perform worse over time (i.e. their world utility decreases). The team game also performs worse over time. In fact, not only does the team game give poor performance, but it fails altogether. The lowest noise TAU utilities g and $g_{nf}^{75\%}$ still give robust performance.
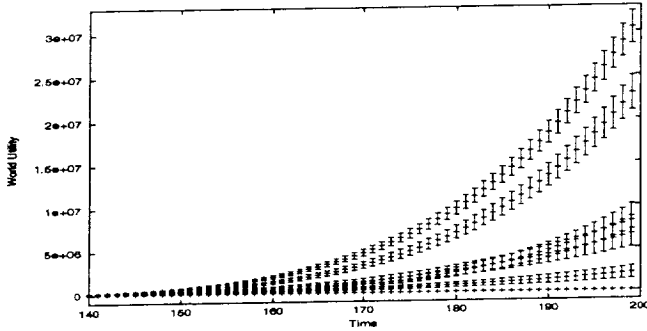
Figure 6: System performance for $N = 50$ agents using the Taylor Series method. The dynamics is governed by a quadratic function of the agents' "positions". The world utility G is a quartic in N dimensions. (upper two graphs are g and $g_{nf}^{75\%}$; middle two are $g_{nf}^{25\%}$ and $g^{75\%}$; lower two are $g^{25\%}$ and a team game G.) The initial training period is not shown.
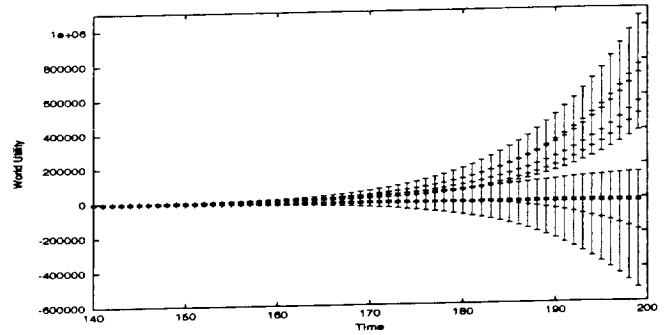

Figure 8: Taylor Series method where the quadratic coefficients have more $-$ than $+$ signs. (graphs: upper pair are g and $g_{nf}^{75\%}$; middle three are $g^{75\%}$, $g^{25\%}$, and the team game; lower is $g_{nf}^{25\%}$.) In this case, three of the limited observability utilities and the team game perform worse over time (i.e. their world utilities decrease).
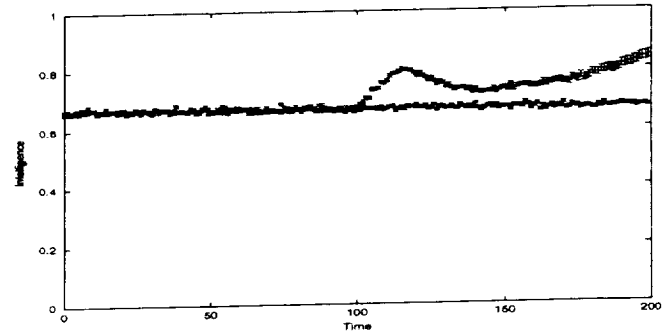

Figure 9: Percentile intelligence for agents using TAU g_η (upper graph) versus a team game (lower graph). when the surface preferentially turns down. The degradation in intelligence as compared to Figure 7 reflects the greater difficulty of the problem.
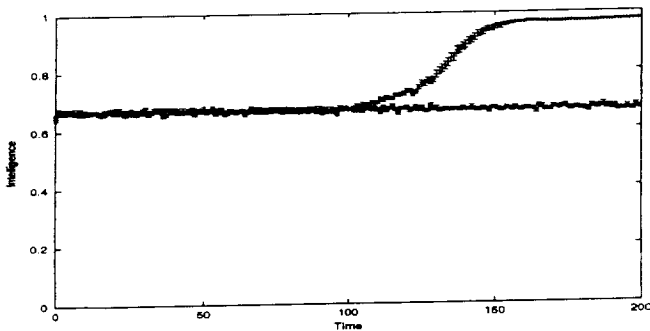
To further study this dramatic difference in performance, we compared the average intelligence of the agents for g and the team game. The results are show in Figure 9. In the case of the team game, again, there is no appreciable change in intelligence from the initial training period to when the agents are invoking learning algorithms. Conversely, for the g utility, the agents perform at a higher intelligence than the team game albeit lower than the situation in Figure 7.

Figure 7: Percentile intelligence for agents using TAU g (upper graph) versus a team game (lower graph). For three actions, a random walk (no learning) would give an average intelligence of 67 %.

7. CONCLUSION

We present a detailed extension of the $COIN$ framework to systems the undergo Markovian evolution. We find consistent, robust improvement of performance as compared to the corresponding team game. The approach is applied to systems with linear and quadratic (nonlinear) update rules. Results from numerical simulations are presented. This framework also naturally includes the case of limited observability. We found that even COIN-based utility functions constrained by limited observability often outperformed conventional team game utilities having full observability. We also found a new class of nonfactored utilities that consistently outperformed their factored counterpart, due to

improved signal-to-noise characteristics.

To address the general nonlinear case, we developed a Taylor Series method. In this case, the system of agents can be imagined to traverse an N-dimensional surface. We find that the system's performance can depend on the characteristics of the surface being optimized. We show that in some situations a team game will fail altogether (i.e. its performance will degrade over time) while the corresponding TAU utility continues to perform well.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] W. B. Arthur. Complexity in economic theory: Inductive reasoning and bounded rationality. *The American Economic Review*, 84(2):406–411, May 1994.

[2] C. Boutilier. Multiagent systems: Challenges and opportunities for decision theoretic planning. *AI Magazine*, 20:35–43, winter 1999.

[3] C. Boutilier, Y. Shoham, and M. P. Wellman. Editorial: Economic principles of multi-agent systems. *Artificial Intelligence Journal*, 94:1–6, 1997.

[4] J. M. Bradshaw, editor. *Software Agents*. MIT Press, 1997.

[5] G. Caldarelli, M. Marsili, and Y. C. Zhang. A prototype model of stock exchange. *Europhysics Letters*, 40:479–484, 1997.

[6] D. Challet and Y. C. Zhang. On the minority game: Analytical and numerical studies. *Physica A*, 256:514, 1998.

[7] C. Claus and C. Boutilier. The dynamics of reinforcement learning cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, Madison, WI, June 1998.

[8] 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.

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

[10] J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 242–250, June 1998.

[11] B. A. Huberman and T. Hogg. The behavior of computational ecologies. In *The Ecology of Computation*, pages 77–115. North-Holland, 1988.

[12] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, 1998.

[13] N. F. Johnson, S. Jarvis, R. Jonson, P. Cheung, Y. R. Kwong, and P. M. Hui. Volatility and agent adaptability in a self-organizing market. preprint cond-mat/9802177, February 1998.

[14] L. P. Kaelbing, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[15] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163, 1994.

[16] T. Sandholm and R. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166, 1995.

[17] T. Sandholm, K. Larson, M. Anderson, O. Shehory, and F. Tohme. Anytime coalition structure generation with worst case guarantees. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 46–53, 1998.

[18] S. Sen. *Multi-Agent Learning: Papers from the 1997 AAAI Workshop (Technical Report WS-97-03.* AAAI Press, Menlo Park, CA, 1997.

[19] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[20] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[21] K. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.

[22] K. Tumer and D. H. Wolpert. Collective intelligence and Braess' paradox. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 104–109, Austin, TX, 2000.

[23] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3/4):279–292, 1992.

[24] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. In *Journal of Artificial Intelligence Research*, 1993.

[25] D. H. Wolpert. Bounded-rationality game theory. pre-print, 2001.

[26] D. H. Wolpert. The mathematics of collective intelligence. pre-print, 2001.

[27] D. H. Wolpert and K. Tumer. An Introduction to Collective Intelligence. Technical Report NASA-ARC-IC-99-63, NASA Ames Research Center, 1999. URL:http://ic.arc.nasa.gov/ic/projects/coin_pubs.html. To appear in Handbook of Agent Technology, Ed. J. M. Bradshaw, AAAI/MIT Press.

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

[29] D. H. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, pages 952–958. MIT Press, 1999.

[30] D. H. Wolpert, K. Wheeler, and K. Tumer. General principles of learning-based multi-agent systems. In *Proceedings of the Third International Conference of Autonomous Agents*, pages 77–83, 1999.

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

[32] Y. C. Zhang. Modeling market mechanism with evolutionary games. *Europhysics Letters*, March/April 1998.