Reinforcement Learning in Distributed Domains: An Inverse Game theoretic Approach

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

Kagan Tumer
NASA Ames Research Center
Mailstop 269-3
Moffett Field, CA 94035
kagan@ptolemy.arc.nasa.gov

Abstract

We consider the design of multi-agent systems (MAS) so as to optimize an overall world utility function when each agent in the system runs a Reinforcement Learning (RL) algorithm based on own its private utility function. Traditional game theory deals with the "forward problem" of determining the state of a MAS that will ensue from a specified set of private utilities of the individual agents. Accordingly, it can be used to predict what world utility would be induced by any such set of private utilities if each agent tried to optimize its utility by using RL algorithms (under appropriate assumptions concerning rationality of those algorithms, information sets, etc.)

In this work we are interested instead in the inverse problem, of how to design the private utilities to induce as high a value of world utility as possible. To ground the analysis in the real world, we investigate this problem in the context of minimizing the loss of importance-weighted communication data traversing a constellation of communication satellites. In our scenario the actions taken by the agents are the introduction of virtual "ghost" traffic into the decision-making of a (pre-fixed, non-learning) distributed routing algorithm. The idea is that judiciously chosen, such ghost traffic can "mislead" the routing algorithm in a way that overcomes deficiencies in that algorithm and thereby improves global performance. The associated design problem is to determine private utilities for the agents that will lead them to introduce precisely that desired ghost traffic. We show in a set of computer experiments that by using inverse game theory it is possible to solve this design problem, i.e., to assign private utilties that lead the agents to introduce ghost traffic that does indeed improve global performance.

1 Introduction

In this paper we are interested in multi-agent systems (MAS's [15; 19; 20]) having the following characteristics:
- the agents each run reinforcement learning (RL) algorithms;
- there is little to no centralized communication or control;
- there is a provided world utility function that rates the possible histories of the full system.

These kinds of problems may well be most readily addressed by having each agent run a Reinforcement Learning (RL) algorithm. In such a system, we are confronted with the inverse problem of how to initialize/update the agents' individual utility functions to ensure that the agents do not "work at cross-purposes", so that their collective behavior maximizes the provided global utility function. Intuitively, we need to provide the agents with utility functions they can learn well, while ensuring that their doing so won't result in economics phenomena like the Tragedy of The Commons (TOC; [12]), liquidity trap or Braess' paradox [21].

This problem is related to work in many other fields, including computational economics, mechanism design, reinforcement learning for adaptive control, statistical mechanics, computational ecologies, and game theory, in particular, evolutionary game theory. However none of these fields directly addresses the inverse problem. (This is even true for the field of mechanism design; see [24] for a detailed discussion of the relationship between these fields, involving several hundred references.)

Other previous work involves MAS's where agents use reinforcement learning [3; 9], and/or where agents model the behavior of other agents [13]. Typically this work simply elects to provide each agent with the global utility function as its private utility function, in a so-called "exact potential" or "team" game. Unfortunately, as expounded below, this can result in very poor global performance in large problems. Intuitively, the difficulty is that each agent can have a hard time discerning the echo of its behavior on the global utility when the system is large.

In previous work we used the COIN framework to derive the alternative "Wonderful Life Utility" (WLU) [24], a utility that generically avoids the pitfalls of the team game utility. In some of that work we used the WLU for distributed control of network packet routing [25]. Conventional approaches to packet routing have each router run a shortest path algorithm (SPA), i.e., each router routes its packets in the way that it expects will get those packets to their destinations most quickly. Unlike with a COIN, with SPA-based routing the routers have no concern for the possible deleterious side-effects of their routing decisions on the global goal (e.g., they have no concern for whether they induce bottlenecks). We ran simulations that demonstrated that a COIN-based routing system has substantially better throughputs than does the best possible SPA-based system [25], even though that SPA-based system has information denied the COIN system. In related work we have shown that use of the WLU automatically avoids the infamous Braess' paradox, in which adding new links can actually decrease throughput — a situation that readily ensnares SPA's.

Finally, in [26] 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" [8]. In this problem, agents have 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 attendees drops. Our goal is to design the reward functions of the attendees so that the total enjoyment across all nights is maximized. In this previous work of ours we showed that use of the WLU can result in performance *orders of magnitude* superior to that of team game utilities.

The WLU has a free parameter (the "clamping parameter"), which we simply set to 0 in our previous work. To determine the optimal value of that parameter we must employ some of the mathematics of COINs, whose relevant concepts we review in the next section. We next use those concepts to sketch the calculation deriving the optimal clamping parameter. To facilitate comparison with previous work, we chose to conduct our experimental investigations of the performance with this optimal clamping parameter in variations of the Bar Problem. We present those variations in Section 3. Finally we present the results of the experiments in Section 4. Those results corroborate the predicted improvement in performance when using our theoretically derived clamping parameter. This extends the superiority of the COIN-based approach above conventional team-game approaches even further than had been done previously.

2 Theory of COINs

In this section we summarize that part of the theory of COINs presented in [25; 24; 26] that is relevant to the study in this article. We consider the state of the system across a set of consecutive time steps, $t \in \{0, 1, ...\}$. Without loss of generality, all relevant characteristics of agent η at time t — including its internal parameters at that time as well as its externally visible actions — are encapsulated by a Euclidean vector $\underline{\zeta}_{\eta,t}$, the *state* of agent η at time t. $\underline{\zeta}_{,t}$ is the set of the states of all agents at t, and $\underline{\zeta}$ is the system's worldline, i.e., the state of all agents across all time.

World utility is $G(\underline{\zeta})$, and when η is an ML algorithm "striving to increase" its **private utility**, we write that utility as $\gamma_\eta(\underline{\zeta})$. (The mathematics can readily be generalized beyond such ML-based agents; see [23] for details.) Here we restrict attention to utilities of the form $\sum_t R_t(\underline{\zeta}_{,t})$ for **reward functions R_t**.

We are interested in systems whose dynamics is deterministic. (This covers in particular any system run on a digital computer, even one using a pseudo-random number generator to generate apparent stochasticity.) We indicate that dynamics by writing $\underline{\zeta} = C(\underline{\zeta}_{,0})$. So all characteristics of an agent η at $t = 0$ that affects the ensuing dynamics of the system, including its private utility, must be included in $\underline{\zeta}_{\eta,0}$.

Definition: A system is **factored** if for each agent η individually,

$$\gamma_\eta(C(\underline{\zeta}_{,0})) \geq \gamma_\eta(C(\underline{\zeta}'_{,0})) \quad \Leftrightarrow \quad G(C(\underline{\zeta}_{,0})) \geq G(C(\underline{\zeta}'_{,0})) ,$$

for all pairs $\underline{\zeta}_{,0}$ and $\underline{\zeta}'_{,0}$ that differ only for node η.

For a factored system, the side effects of changes to η's $t = 0$ state that increase its private utility cannot decrease world utility. If the separate agents have high values of their private utilities, by luck or by design, then they have not frustrated each other, as far as G is concerned. (We arbitrarily phrase this paper in terms of changes at time 0; the formalism is easily extended to deal with arbitrary times.)

The definition of factored is carefully crafted. In particular, it does *not* concern changes in the value of the utility of agents other than the one whose state is varied. Nor does it concern changes to the states of more than one agent at once. Indeed, consider the following alternative desideratum to having the system be factored:

any change to $\underline{\zeta}_{,0}$ that simultaneously improves the ensuing values of all the agents' utilities must also improve world utility. Although it seems quite reasonable, there are systems that obey this desideratum and yet quickly evolve to a *minimum* of world utility ([26]).

For a factored system, when every agents' private utility is optimized (given the other agents' behavior), world utility is at a critical point [24]. In game-theoretic terms, optimal global behavior occurs when the agents' are at a private utility Nash equilibrium [11]. Accordingly, there can be no TOC for a factored system.

As a trivial example, if $\gamma_\eta = G \; \forall\eta$, then the system is factored, regardless of C. However there exist other, often preferable sets of $\{\gamma_\eta\}$, as we now discuss.

Definition: The ($t = 0$) **effect set** of node η at $\underline{\zeta}$, $C^{eff}_\eta(\underline{\zeta})$, is the set of all components $\underline{\zeta}_{\eta',t'}$ for which the gradients $\vec{\nabla}_{\underline{\zeta}_{\eta,0}}(C(\underline{\zeta}_{,0}))_{\eta',t'} \neq \vec{0}$. C^{eff}_η with no specification of $\underline{\zeta}$ is defined as $\cup_{\underline{\zeta} \in C} C^{eff}_\eta(\underline{\zeta})$.
Intuitively, the effect set of η is the set of all node-time pairs affected by changes to η's $t = 0$ state.

Definition: Let σ be a set of agent-time pairs. $\mathrm{CL}_\sigma(\underline{\zeta})$ is $\underline{\zeta}$ modified by "clamping" the states corresponding to all elements of σ to some arbitrary pre-fixed value, here taken to be $\vec{0}$. The **wonderful life utility** (WLU) for σ at $\underline{\zeta}$ is defined as:

$$WLU_\sigma(\underline{\zeta}) \equiv G(\underline{\zeta}) - G(\mathrm{CL}_\sigma(\underline{\zeta})) \; . \qquad (1)$$

In particular, the WLU for the effect set of node η is $G(\underline{\zeta}) - G(\mathrm{CL}_{C^{eff}_\eta}(\underline{\zeta}))$.

A node η's effect set WLU is analogous to the change world utility would undergo had node η "never existed". (Hence the name of this utility - cf. the Frank Capra movie.) However CL(.) is a purely "fictional", counterfactual mapping, in that it produces a new $\underline{\zeta}$ without taking into account the system's dynamics. The sequence of states produced by the clamping operation in the definition of the WLU need not be consistent with the dynamical laws embodied in C. This is a crucial strength of effect set WLU. It means that to evaluate that WLU we do *not* try to infer how the system would have evolved if node η's state were set to $\vec{0}$ at time 0 and the system re-evolved. So long as we know G and the full $\underline{\zeta}$, and can accurately estimate what agent-time pairs comprise C^{eff}_η, we know the value of η's effect set WLU — even if we know nothing of the details of the dynamics of the system.

Theorem 1: A COIN is factored if $\gamma_\eta = WLU_{C^{eff}_\eta} \; \forall\eta$ (proof in [24]).

If our system is factored with respect to some $\{\gamma_\eta\}$, then each $\underline{\zeta}_{\eta,0}$ should be in a state with as high a value of $\gamma_\eta(C(\underline{\zeta}_{,0}))$ as possible. So for such systems, our problem is determining what $\{\gamma_\eta\}$ the agents will best be able to maximize while also causing dynamics that is factored with respect to those $\{\gamma_\eta\}$.

Now regardless of $C(.)$, both $\gamma_\eta = G \; \forall\eta$ and $\gamma_\eta = WLU_{C^{eff}_\eta} \; \forall\eta$ are factored systems. However since each agent is operating in a large system, it may experience difficulty discerning the effects of its actions on G when G sensitively depends on all components of the system. Therefore each η may have difficulty learning how to achieve high γ_η when $\gamma_\eta = G$. This problem can be obviated by using effect set WLU, since the subtraction of the clamped term removes some of the "noise" of the activity of other agents, leaving only the underlying "signal" of how agent η affects its utility.

We can quantify this signal/noise effect by comparing the ramifications on the private utilities arising from changes to $\underline{\zeta}_{\eta,0}$ with the ramifications arising from changes to $\underline{\zeta}_{\hat{\eta},0}$, where $\hat{\eta}$ represents all nodes *other* than η. We call this quantification the **learnability** of those utilities at the point $\underline{\zeta} = C(\underline{\zeta}_{\eta,0})$ [24]. A linear approximation to the learnability in the vicinity of the worldline $\underline{\zeta}$ is the **differential learnability** $\lambda_{\eta,\gamma_\eta}(\underline{\zeta})$:

$$\lambda_{\eta,\gamma_\eta}(\underline{\zeta}) \equiv \frac{\|\vec{\nabla}_{\underline{\zeta}_{\eta,0}} \gamma_\eta(C(\underline{\zeta}_{,0}))\|}{\|\vec{\nabla}_{\underline{\zeta}_{\hat{\eta},0}} \gamma_\eta(C(\underline{\zeta}_{,0}))\|} \; . \qquad (2)$$

Differential learnability captures the signal-to-noise advantage of the WLU in the following theorem:

Theorem 2: Let σ be a set containing C^{eff}_η. Then

$$\frac{\lambda_{\eta,WLU_\sigma}(\underline{\zeta})}{\lambda_{\eta,G}(\underline{\zeta})} = \frac{\|\vec{\nabla}_{\underline{\zeta}_{\hat{\eta},0}} G(C(\underline{\zeta}_{,0}))\|}{\|\vec{\nabla}_{\underline{\zeta}_{\hat{\eta},0}} G(C(\underline{\zeta}_{,0})) - \vec{\nabla}_{\underline{\zeta}_{\hat{\eta},0}} G(\mathrm{CL}_\sigma(C(\underline{\zeta}_{,0})))\|}$$

(proof in [24]). This ratio of gradients should be large whenever σ is a small part of the system, so that the clamping won't affect G's dependence on $\underline{\zeta}_{\hat{\eta},0}$ much, and therefore that dependence will approximately cancel in the denominator term. In such cases, WLU is factored, just as G is, but far more learnable. The experiments presented below illustrate the power of this fact in the context of the bar problem, where one can readily approximate effect set WLU and therefore use a utility for which the conditions in Thm.'s 1 and 2 should hold.

3 The Bar Problem

Arthur's bar problem [1] can be viewed as a problem in designing COINs. Loosely speaking, in this problem at each time t each agent η decides whether to attend a bar by predicting, based on its previous experience, whether the bar will be too crowded to be "rewarding" at that time, as quantified by a reward function R_G.

The greedy nature of the agents frustrates the global goal of maximizing R_G at t. This is because if most agents think the attendance will be low (and therefore choose to attend), the attendance will actually be high, and vice-versa. We modified Arthur's original problem to be more general, and since we are not interested here in directly comparing our results to those in [1; 8], we use a more conventional ML algorithm than the ones investigated in [1; 7; 8].

There are N agents, each picking one of seven nights to attend a bar in a particular week, a process that is then repeated for the following weeks. In each week, each agent's pick is determined by its predictions of the associated rewards it would receive if it made that pick. Each such prediction in turn is based solely upon the rewards received by the agent in those preceding weeks in which it made that pick.

The world utility is $G(\underline{\zeta}) = \sum_t R_G(\underline{\zeta}_{,t})$, where $R_G(\underline{\zeta}_{,t}) \equiv \sum_{k=1}^{7} \phi(x_k(\underline{\zeta}, t))$, $x_k(\underline{\zeta}, t)$ is the total attendance on night k at week t, $\phi(y) \equiv y \exp(-y/c)$; and c is a real-valued parameter. Our choice of $\phi(.)$ means that when too few agents attend some night in some week, the bar suffers from lack of activity and therefore the world reward is low. Conversely, when there are too many agents the bar is overcrowded and the reward is again low.

Since we are concentrating on the choice of utilities rather than the RL algorithms that use them, we use simple RL algorithms. Each agent η has a 7-dimensional vector representing its estimate of the reward it would receive for attending each night of the week. At the beginning of each week, to trade off exploration and exploitation, η picks the night to attend randomly using a Boltzmann distribution over the seven components of η's estimated rewards vector. For simplicity, temperature did not decay in time. However to reflect the fact that each agent perceives an environment that is changing in time, the reward estimates were formed using exponentially aged data: in any week t, the estimate agent η makes for the reward for attending night i is a weighted average of all the rewards it has previously received when it attended that night, with the weights given by an exponential function of how long ago each such reward was.

To form the agents' initial training set, we had an initial training period in which all actions by all agents were chosen uniformly randomly, and the associated rewards recorded by all agents. After this period, the Boltzmann scheme outlined above was "turned on".

This simple RL algorithm works with rewards rather than full-blown utilities. So formally speaking, to apply the COIN framework to it it is necessary to extend that framework to encompass rewards in addition to utilities, and in particular to concern effect set wonderful life reward (WLR), whose value at moment t for agent η is $R_G(\underline{\zeta}_{,t}) - R_G(CL_{C_\eta^{eff}}(\underline{\zeta}_{,t}))$. To do this one uses Thm. 1 to prove that, under some mild assumptions, if we have a set of private rewards that are factored with respect to world rewards, then maximizing those private rewards also maximizes the full world utility. In terms of game theory, a Nash equilibrium of the single-stage game induces a maximum of the world utility defined over the entire multi-stage game. (Intuitively, this follows from the fact that the world utility is a sum of the world rewards.) In addition, one can show that the WLR is factored with respect to the world reward, and that it has the same advantageous learnability characteristics that accrue to the WLU. Accordingly, just as the COIN framework recommend we use WLU when dealing with utility-based RL algorithms, it recommends that we use WLR in the bar problem when dealing with reward-based RL algorithms. See [23].

Example: It is worth illustrating how the WLR is factored with respect to the world reward in the context of the bar problem. Say we're comparing the action of some particular agent going on night 1 versus that agent going on night 2, in some pre-fixed week. Let x'_1 and x'_2 be the total attendances of everybody *but* our agent, on nights 1 and 2 of that week, respectively. So $WLR(1)$, the WLR value for the agent attending night 1, is given by $\phi(x'_1 + 1) - \phi(x'_1 + CL_1) + \phi(x'_2) - \phi(x'_2 + CL_2) + \sum_{i>2}[\phi(x_i) - \phi(x_i + CL_i)]$, where "$CL_i$" is the i'th component of our clamped vector. Similarly, $WLR(2) = \phi(x'_1) - \phi(x'_1 + CL_1) + \phi(x'_2 + 1) - \phi(x'_2 + CL_2) + \sum_{i>2}[\phi(x_i) - \phi(x_i + CL_i)]$.
Combining, $sgn(WLR(1) - WLR(2)) = sgn(\phi(x'_1 + 1) - \phi(x'_1) - \phi(x'_2 + 1) + \phi(x'_2))$. On the other hand, $R_G(1)$, the G value for the agent attending night 1, is $\phi(x'_1 + 1) + \phi(x'_2) + \sum_{i>2} \phi(x_i)$. Similarly, $R_G(2)$ is $\phi(x'_1) + \phi(x'_2 + 1) + \sum_{i>2} \phi(x_i)$. Therefore $sgn(R_G(1) - R_G(2)) = sgn(\phi(x'_1 + 1) + \phi(x'_2) - \phi(x'_2 + 1) - \phi(x'_1))$.
So $sgn(WLR(1) - WLR(2) = sgn(R_G(1) - R_G(2))$. This is true for any pair of nights, and any attendances $\{x_i\}$, and any clamping vector. This establishes the claim that WLR is factored with respect to the world reward, for the bar problem.

When using the WLR we are faced with the question of setting the clamping parameter, i.e., of determining the best values to which to clamp the components C_η^{eff} of $\underline{\zeta}$. One way to do this is to solve for those values that maximize differential learnability. An approximation to this calculation is to solve for the clamping parameter that minimizes the expected value of $[\lambda_{\eta, WLR_\eta}]^{-2}$, where the expectation is over the values $\underline{\zeta}_{,t}$ and associated rewards making up η's training set.

A number of approximations have to be made to carry out this calculation. The final result is that η should clamp to its (appropriately data-aged) empirical expected average action, where that average is over the elements in its training set [23]. Here, for simplicity, we ignore the data-aging stipulation of this result. Also for simplicity, we do not actually make sure to clamp each η separately to its own average action, a process that involves η modifying what it clamps to in an online manner. Rather we choose to clamp all agents to the same vector, where that vector is an initial guess as to what the average action of a typical agent will be. Here, where the initial training period has each agent choose its action uniformly randomly, that guess is just the uniform average of all actions. The experiments recounted in the next section illustrate that even using these approximations, performance with the associated clamping parameter is superior to that of using the WL reward with clamping to $\vec{0}$, which in turn exhibits performance significantly superior to use of team game rewards.

4 Experiments

4.1 Single Night Attendance

Our initial experiments compared three choices of the clamping parameter: Clamping to "zero" i.e., the action vector given by $\vec{0} = (0,0,0,0,0,0,0)$, as in our original work; clamping to "ones" i.e., the action vector $\vec{1} = (1,1,1,1,1,1,1)$; and clamping to the (ideal) "average" action vector for the agents after the initial training period, denoted by $\vec{a}$. Intuitively, the first clamping is equivalent to the agent "staying at home," while the second option corresponds to the agent attending every night. The third option is equivalent to the agents attending partially on all nights in proportions equivalent to the overall attendance profile of all agents across the initial training period. (If taken, this "action" would violate the dynamics of the system, but because it is a fictional action as described in Section 2, it is consistent with COIN theory.)

In order to distinguish among the different clamping operators, we will include the action vector to which the agents are clamped as a subscript (e.g., $CL^{\vec{0}}$ will denote the operation where the action is clamped to the zero vector). Because all agents have the same reward function in the experiments reported here, we will drop the agent subscript from the reward function.

We compared performance with these three WLR's and the team game reward, R_G. Writing them out, those three WLR reward functions are:

$$\begin{aligned} R_{WL_{\vec{0}}}(\underline{\zeta}_{,t}) &\equiv R_G(\underline{\zeta}_{,t}) - R_G(CL^{\vec{0}}_{\eta}(\underline{\zeta}_{,t})) \\ &= \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta},t)) \; - \; \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta},t) - 1) \\ R_{WL_{\vec{1}}}(\underline{\zeta}_{,t}) &\equiv R_G(\underline{\zeta}_{,t}) - R_G(CL^{\vec{1}}_{\eta}(\underline{\zeta}_{,t})) \end{aligned}$$

$$\begin{aligned} &= \sum_{d \neq d_\eta}^{7} \phi_d(x_d(\underline{\zeta},t)) \; - \; \phi_d(x_d(\underline{\zeta},t) + 1) \\ R_{WL_{\vec{a}}}(\underline{\zeta}_{,t}) &\equiv R_G(\underline{\zeta}_{,t}) - R_G(CL^{\vec{a}}_{\eta}(\underline{\zeta}_{,t})) \\ &= \sum_{d \neq d_\eta}^{7} \phi_d(x_d(\underline{\zeta},t)) \; - \; \phi_d(x_d(\underline{\zeta},t) + a_d) \\ &+ \; \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta},t)) \; - \; \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta},t) - 1 + a_d) \end{aligned}$$

where d_η is the night picked by η, and a_d is the component of $\vec{a}$ corresponding to night d.

The team game reward, R_G, results in the system meeting the desideratum of factoredness. However, because of Theorem 2, we expect R_G to have poor learnability, particularly in comparison to that of $R_{WL_{\vec{0}}}$; (see [24] for details). Note that to evaluate $R_{WL_{\vec{0}}}$ each agent only needs to know the total attendance on the night it attended. In contrast, R_G and $R_{WL_{\vec{a}}}$ require centralized communication concerning all 7 nights, and $R_{WL_{\vec{1}}}$ requires communication concerning 6 nights.

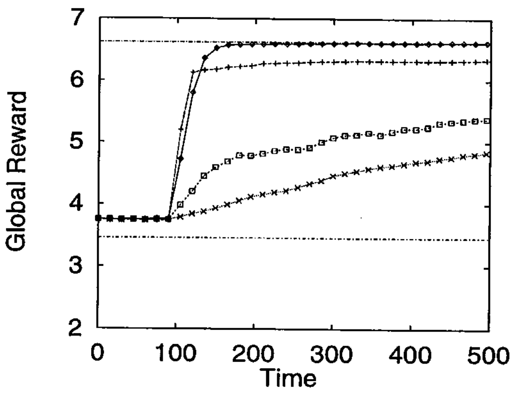

Figure 1: Reward function comparison when agents attend one night. ($WL_{\vec{a}}$ is ◇ ; $WL_{\vec{0}}$ is + ; $WL_{\vec{1}}$ is □ ; G is ×)

Figure 1 graphs world reward against time, averaged over 100 runs, for 60 agents and $c = 3$. (Throughout this paper, error bars are too small to depict.) The two straight lines correspond to the optimal performance, and the "baseline" performance given by uniform occupancies across all nights. Systems using $WL_{\vec{a}}$ and $WL_{\vec{0}}$ rapidly converged to optimal and to quite good performance, respectively. This indicates that for the bar problem the "mild assumptions" mentioned above hold, and that the approximations in the derivation of the optimal clamping parameter are valid.

In agreement with our previous results, use of the reward R_G converged very poorly in spite of its being factored. The same was true for the $WL_{\vec{1}}$ reward. This behavior highlights the subtle interaction between factoredness and learnability. Because the signal-to-noise was higher for these reward functions, it was very difficult for individual agents to determine what actions would improve their private utilities and therefore had difficulties in finding good solutions.

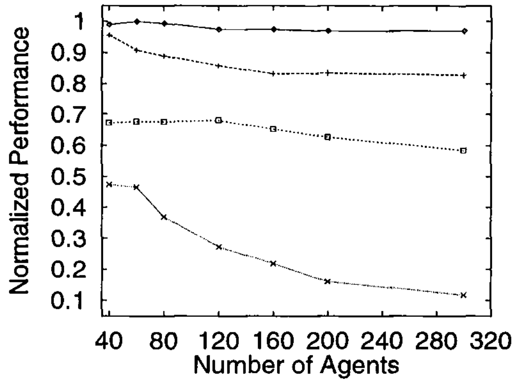

Figure 2: Scaling properties of the different reward function. ($WL_{\vec{a}}$ is ◇ ; $WL_{\vec{0}}$ is + ; $WL_{\vec{1}}$ is □ ; G is ×)

Figure 2 shows how $t = 500$ performance scales with N for each of the reward signals. For comparison purposes the performance is normalized — for each utility U we plot $\frac{R_U - R_{base}}{R_{opt} - R_{base}}$, where R_{opt} and R_{base} are the optimal performance and a canonical baseline performance given by uniform attendance across all nights, respectively. Systems using R_G perform adequately when N is low. As N increases however, it becomes increasingly difficult for the agents to extract the information they need from R_G. Because of their superior learnability, systems using the WL rewards overcome this signal-to-noise problem to a great extent. Because the WL rewards are based on the *difference* between the actual state and the state where one agent is clamped, they are much less affected by the total number of agents. However, the action vector to which agents are clamped also affects the scaling properties.

4.2 Multiple Night Attendance

In order to study the relationship between the clamping parameter and the resulting world utility in more detail, we now modify the bar problem as follows: Each week, each agents picks **three** nights to attend the bar. So each of the seven possible actions now corresponds to a different attendance pattern. (Keeping the number of candidate actions at 7 ensures that the complexity of the RL problem faced by the agents is roughly the same.) Here those seven attendance profiles were attending the first three nights, attending nights 2 through 4, ..., attending on nights 7, 1 and 2.

Figure 3 shows world reward value as a function of time for this problem, averaged over 100 runs, for all four reward functions. For these simulations $c = 8$, and there were 60 agents. Optimal and baseline performance are plotted as straight lines. Note that in the experiments of the previous section $CL^{\vec{a}}$ clamps to the attendance vector v with components $v_i = \sum_{d=1}^{7} \frac{\delta_{d,i}}{7}$, where $\delta_{d,i}$ is the Kronecker delta function. Now however it clamps to $v_i = \sum_{d=1}^{7} \frac{u_{d,i}}{7}$, where $u_{d,i}$ is the i'th component (0 or 1) of the the d'th action vector, so that for each d it contains three 1's and four 0's.

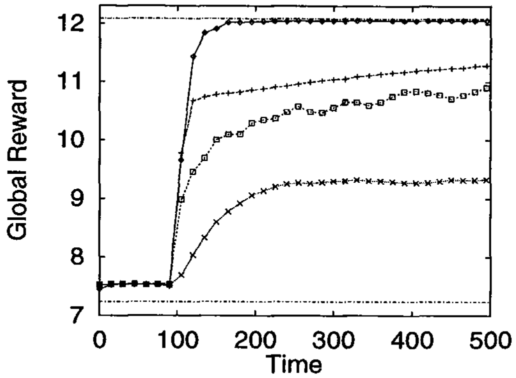

Figure 3: Reward function comparison when agents attend three nights. ($WL_{\vec{a}}$ is ◇ ; $WL_{\vec{0}}$ is + ; $WL_{\vec{1}}$ is □ ; G is ×)

As in the previous case, the reward obtained by clamping to the average action $R_{WL_{\vec{a}}}$ performs near optimally. $R_{WL_{\vec{0}}}$ on the other hand shows a slight drop-off compared to the previous problem. $R_{WL_{\vec{1}}}$ now performs almost as well as $R_{WL_{\vec{0}}}$. All three WL rewards still significantly outperform the team game reward. What is noticeable though is that as the number of nights to attend increases, the difference between $R_{WL_{\vec{0}}}$ and $R_{WL_{\vec{1}}}$ decreases, illustrating how changing the problem can change the relative performances of the various WL rewards.

4.3 Sensitivity to Clamping

The results of the previous section shows that the action vector to which agents clamp has a considerable impact on the global performance. In this section we study how that dependence varies with changes in the problem formulation.

We considered four additional variants of the bar problem just like the one described in the previous subsection, only with four new values for the number of nights each agent attends. As in the previous section, we keep the number of actions at seven and map those actions to correspond to attending particular sets of nights. Also as in the previous section, we choose the attendance profiles of each potential action so that when the actions are selected uniformly the resultant attendance profile is also uniform. We also modify c to keep the "congestion" level of the problem at a level similar to the original problem. (For the number of nights attended going from one to six, $c = \{3, 6, 8, 10, 12, 15\}$ respectively.)

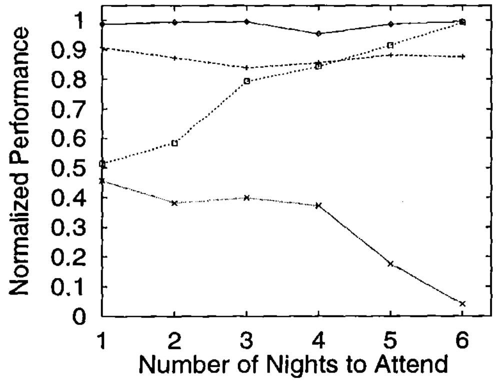

Figure 4: Behavior of different reward function with respect to number of nights to attend. ($WL_{\vec{a}}$ is ◇ ; $WL_{\vec{0}}$ is + ; $WL_{\vec{1}}$ is □ ; G is ×)

Figure 4 shows the normalized world reward obtained for the different rewards as a function of the number of nights each agent attends. $R_{WL_{\vec{a}}}$ performs well across the set of problems. $R_{WL_{\vec{1}}}$ on the other hand performs poorly when agents only attend on a few nights, but reaches the performance of $R_{WL_{\vec{a}}}$ when agent need to select six nights, a situation where the two clamped action vectors are very similar. $R_{WL_{\vec{0}}}$ shows a slight drop in performance when the number of nights to attend increases, while R_G shows a much more pronounced drop. These results reinforce the conclusion obtained in the previous section that the clamped action vector that best matches the aggregate empirical attendance profile results in best performance.

4.4 Sensitivity to Parameter Selection

The final aspect of these reward functions we study is the sensitivity of the associated performance to the internal parameters of the learning algorithms. Figure 5 illustrates experiments in the original bar problem presented in Figures 1 and 2, for a set of different temperatures in the Boltzamnn distribution. $R_{WL_{\vec{a}}}$ is fairly insensitive to the temperature, until it is so high that agents' actions are chosen almost randomly. $R_{WL_{\vec{0}}}$ depends more than $R_{WL_{\vec{a}}}$ does on having sufficient exploration and therefore has a narrower range of good temperatures. Both $R_{WL_{\vec{1}}}$ and R_G have more serious learnability problems, and therefore have shallower and thinner performance graphs.

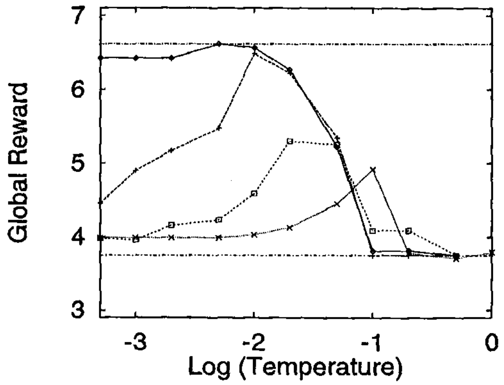

Figure 5: Sensitivity of reward functions to internal parameters. ($WL_{\vec{a}}$ is ◇ ; $WL_{\vec{0}}$ is + ; $WL_{\vec{1}}$ is □ ; G is ×)

5 Conclusion

In this article we consider how to configure large multi-agent systems where each agent uses reinforcement learning. To that end we summarize relevant aspects of COIN theory, focusing on how to initialize/update the agents' private utility functions so that their collective behavior optimizes a global utility function.

In traditional "team game" solutions to this problem, which assign to each agent the global utility as its private utility function, each agent has difficulty discerning the effects of its actions on its own utility function. We confirmed earlier results that if the agents use the alternative "Wonderful Life Utility" with clamping to $\vec{0}$, the system converges to significantly superior world reward values than do that associated team game systems. We then demonstrated that this wonderful life utility also results in faster convergence, better scaling, and less sensitivity to parameters of the agents' learning algorithms. We also showed that optimally choosing the action to which agents clamp (rather than arbitrarily choosing $\vec{0}$) provides significant further gains in performance, according to all of these performance measures. Future work involves investigating various ways of having the agents determine their optimal clamping vectors dynamically.

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] E. Baum. Manifesto for an evolutionary economics of intelligence. In C. M. Bishop, editor, *Neural Networks and Machine Learning*. Springer–Verlag, 1998.

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

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

[5] J. A. Boyan and A. W. Littman. Learning evaluation functions for global optimization and boolean satisfiability. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence.* AAAI Press, 1998.

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

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

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

[9] 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.

[10] 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.

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

[12] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[13] 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.

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

[15] 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.

[16] 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.

[17] M. L. Littman and J. Boyan. A distributed reinforcement learning scheme for network routing. In *Proceedings of the 1993 International Workshop on Applications of Neural Networks to Telecommunications*, pages 45–51, 1993.

[18] 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.

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

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

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

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

[23] D. H. Wolpert and K. Tumer. The mathematics of collective intelligence. pre-print, 2000.

[24] D. H. Wolpert and K. Tumer. An Introduction to Collective Intelligence. In J. M. Bradshaw, editor, *Handbook of Agent technology.* AAAI Press/MIT Press, to appear. Available as tech. rep. NASA-ARC-IC-99-63 from `http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`.

[25] 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.

[26] 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.

[27] W. Zhang and T. G. Dietterich. Solving combinatorial optimization tasks by reinforcement learning: A general methodology applied to resource-constrained scheduling. *Journal of Artificial Intelligence Reseach*, 2000.

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