Optimal Reward Functions in Distributed Reinforcement Learning

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

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

Abstract

We consider the design of multi-agent systems so as to optimize an overall world utility function when (i) those systems lack centralized communication and control [10, 14], and (ii) each agents runs a distinct Reinforcement Learning (RL) algorithm [3, 11, 13]. A crucial issue in such design problems is to initialize/update each agent's private utility function, so as to induce best possible world utility. Traditional "team game" solutions to this problem sidestep this issue and simply assign to each agent the world utility as its private utility function [6]. In previous work we used the "Collective Intelligence" framework to derive a better choice of private utility functions, one that results in world utility performance up to orders of magnitude superior to that ensuing from use of the team game utility [15, 17, 16, 18]. In this paper we extend these results. We derive the general class of private utility functions that both are easy for the individual agents to learn and that, if learned well, result in high world utility. We demonstrate experimentally that using these new utility functions can result in significantly improved performance over that of our previously proposed utility, over and above that previous utility's superiority to the conventional team game utility.

1 Introduction

In this paper we are interested in multi-agent systems (MAS's) where:

- the agents each run separate reinforcement learning (RL) algorithms;
- there is little centralized, personalized communication or control;
- there is a provided world utility function rating possible histories of the full system.

In such a system, we are confronted with an *inverse* problem: How should we 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 world utility function? Intuitively, we need to provide the agents with utility functions they can learn well, while also ensuring that their doing so won't result in

phenomena like the Tragedy of The Commons (TOC; [8]) or Braess' paradox [15].

This problem is related to work in many other fields, including computational economics, mechanism design, reinforcement learning, computational ecologies, (partially observable) Markov decision processes and game theory. However none of these fields is both applicable in large, real-world problems, and directly addresses the *general* inverse problem. (See [16] for a detailed discussion of the relationship between these fields, involving hundreds of references.) Other previous work does consider the general inverse problem, and does so by employing MAS's in which each agent uses reinforcement learning [5]. However this work simply elects to provide each agent with the world utility function as its private utility function (i.e., implements a "team" game). Unfortunately, as expounded below and in previous work, this approach scales to large problems very poorly. (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.)

In previous work, as an alternative to the team game approach, we used the "COllective INtelligence" (COIN) framework to derive the alternative "Wonderful Life" private utility function (WLU) [16], and demonstrated that it markedly outperforms the team game private utility in several disparate domains [15, 17, 16, 18]. In particular, in [18] we considered an economics "congestion" game, in particular a more challenging variant of Arthur's "El Farol bar attendance problem" [1], also known as the "minority game" [4]. 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. Note the built-in frustration effect that if all agents could predict attendance perfectly, they would all make the same attendance choice, and total enjoyment would be minimal. The goal is to avoid this by designing the reward functions of the attendees so that the total enjoyment across all nights is maximized. Our results indicate that use of the WLU can result in performance *orders of magnitude* superior to that of team game utilities [18].

The WLU has a free parameter (the "clamping parameter"), which we simply set to 0 in our previous work. In this paper we employ a series of approximations to derive a theoretically optimal value of the clamping parameter, and demonstrate the empirical superiority of that value in computer experiments. To derive the optimal value 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. Our experiments involved the Bar problem, whose detailed setup is discussed 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 even further the superiority of the COIN-based approach above that of conventional team-game approaches.

2 Theory of COINs

In this section we summarize that part of the mathematics of COINs 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 RL algorithm "striving to increase" its **private utility**, we write that utility as $\gamma_\eta(\underline{\zeta})$. (The mathematics can readily be generalized beyond such RL-based agents; see [16] 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, are 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, when every agents' private utility is optimized (given the other agents' behavior), world utility is at a critical point (e.g., a local maximum) [16]. In game-theoretic terms, optimal global behavior occurs when the agents' are at a private utility Nash equilibrium [7]. Accordingly, there can be no TOC for a factored system [16, 17, 18].) In addition, off of equilibrium, the private utilities in factored systems are "aligned" with the world utility.

Definition: The $(t = 0)$ **effect set** of node η at $\underline{\zeta}$, $S_\eta^{eff}(\underline{\zeta})$, is the set of all components $\underline{\zeta}_{\eta',t'}$ for which the gradients $\vec{\nabla}_{\underline{\zeta}_{\eta,0}}(C(\underline{\zeta}_{,0}))_{\eta',t'} \neq \vec{0}$. S_η^{eff} with no specification of $\underline{\zeta}$ is defined as $\cup_{\underline{\zeta} \in C} S_\eta^{eff}(\underline{\zeta})$. We will also find it useful to define $\bar{S}_\eta^{eff}$ as the set of all components that are not in S_η^{eff}.

Intuitively, the $t = 0$ effect set of η is the set of all node-time pairs which, under the deterministic dynamics of the system, are affected by changes to η's $t = 0$ state.

Theorem: A system is factored at all $\underline{\zeta} \in C$ iff for all those $\underline{\zeta}$, $\forall \eta$, we can write

$$\gamma_\eta(\underline{\zeta}) = \hat{\Phi}_\eta(\underline{\zeta}_{\cdot S_\eta^{eff}}, G(\underline{\zeta})) \tag{1}$$

for some function $\hat{\Phi}_\eta(.,.)$ such that $\partial_G \hat{\Phi}_\eta(\underline{\zeta}_{\cdot S_\eta^{eff}}, G) > 0$ for all $\underline{\zeta} \in C$ and associated G values (the form of the $\{\gamma_\eta\}$ off of C is arbitrary). (Proof in [16].)

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 the elements of σ to some arbitrary pre-fixed vector $\vec{\kappa}$. Then the (effect set) **Wonderful Life Utility** for node η (at time 0) is $WLU_\eta(\underline{\zeta}) \equiv G(\underline{\zeta}) - G(CL_{S_\eta^{eff}}(\underline{\zeta}))$, where conventionally $\vec{\kappa} = \vec{0}$.

Note the crucial fact that to evaluate the WLU one does *not* need to know how to calculate the system's behavior under counter-factual starting conditions. All that is needed to evaluate WLU_η is the function $G(.)$, the actual $\underline{\zeta}$, and S_η^{eff} (which can often be well-approximated even with little knowledge of C).

Since $G(CL_{S_\eta^{eff}}(\underline{\zeta}))$ is a function only of $\underline{\zeta}_{\cdot S_\eta^{eff}}$, by Thm. 1 we know that WLU is factored. As another example, if $\gamma_\eta = G \ \forall \eta$ (a team game), then the system is factored, in this case regardless of C. However for large systems where G sensitively depends on all components of the system, each agent may experience difficulty discerning the effects of its actions on G. As a consequence, each η may have difficulty achieving high γ_η in a team game. We can quantify this signal/noise effect

by comparing the ramifications on $\gamma_\eta(\underline{\zeta} = C(\underline{\zeta}_{,0}))$ arising from changes to $\underline{\zeta}_{\eta,0}$ with the ramifications arising from changes to $\underline{\zeta}_{\neg\eta,0}$, where $\neg\eta$ represents all nodes *other* than η. We call this quantification **learnability** [16]. A linear approximation to the learnability in the vicinity of $\underline{\zeta}$ is the **differential learnability** $\lambda_{\eta,\gamma_\eta}(\underline{\zeta})$:

$$\lambda_{\eta,\gamma_\eta}(\underline{\zeta}) \equiv \frac{\|\vec{\nabla}_{\underline{\zeta}_{\eta,0}} \gamma_\eta(C(\underline{\zeta}_{,0}))\|}{\|\vec{\nabla}_{\underline{\zeta}_{\neg\eta,0}} \gamma_\eta(C(\underline{\zeta}_{,0}))\|} \quad (2)$$

It can be proven that in many circumstances, especially in large problems, WLU has much higher differential learnability than does the team game choice of private utilities [16]. (Intuitively, this is due to the subtraction occurring in the WLU's removing a lot of the noise.) The result is that convergence to optimal G with WLU is much quicker (up to orders of magnitude so) than with a team game.

However the equivalence class of utilities that are factored for a particular G is not restricted to the associated team game utility and clamp-to-$\vec{0}$ WLU. Indeed, one can consider solving for the utility in that equivalence class that maximizes differential learnability. An approximation to this calculation is to solve for the factored utility that minimizes the expected value of $[\lambda_{\eta,WLR_\eta}]^{-2}$, where the expectation is over the values $\underline{\zeta}_{,0}$ that, while fixed, are not known to the system designer. (As an example, algorithms using pseudo-random number generators are deterministic, strictly speaking, but are effectively stochastic to the system designer.)

A number of further approximations — too long to go through here — have to be made to complete this calculation. The final answer can be approximated as a WLU, where $\vec{\kappa} \neq \vec{0}$, but rather equals the expected S_η^{eff}. Now in the experiments recounted below S_η^{eff} is approximated as the sequence of η's successive actions (i.e., the approximation is made that to first order, η's actions have no effects on the actions of other agents). Furthermore, for simplicity, we do not actually clamp each η separately to its own average action sequence, which would involve modifying WLU_η in an online manner. Rather we clamp all agents to the same average action. We then made the guess that the typical probability distribution over actions is uniform. (Intuitively, we would expect such a choice to be more accurate at early times than at later times in which agents have "specialized".)

3 The Bar Problem

We focus on the following six more general variants of the bar problem investigated in [18]: There are N agents, each picking one out of seven actions every week. Each action corresponds to attending the bar on some particular set of $l \in \{1, 2, 3, 4, 5, 6\}$ out of the seven nights of the current week, i.e., given l, each action is a vertex of the 7-dimensional unit hypercube having l 1's. At the end of the week the agents get their rewards and the process is repeated. For simplicity we chose the attendance profiles of each potential action so that when the actions are selected uniformly the resultant attendance profile across all seven nights is also uniform.

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. (To keep the "congestion" level constant, for l going from 1 to 6 $c = \{3, 6, 8, 10, 12, 15\}$ respectively.) Our choice of $\phi(.)$ means that when either too few or too many agents attend some night in some week world reward R_G is low.

Since we are concentrating on the utilities rather than on the RL algorithms that

use them, we use (very) simple RL algorithms. Each agent η has a 7-dimensional vector giving its estimates of the reward it would receive for taking each possible action. At the beginning of each week, each η picks the night to attend randomly, using a Boltzmann distribution over the seven components of η's estimated rewards vector. For simplicity, temperature does not decay in time. However to reflect the fact that each agent operates in a non-stationary environment, reward estimates are formed using exponentially aged data: in any week t, the estimate 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 period in which all actions by all agents were chosen uniformly randomly, with no learning.

4 Experimental Results

We investigate three choices of $\vec{\kappa}$: $\vec{0}$, $\vec{1} = (1, 1, 1, 1, 1, 1, 1)$, and the "average" action, $\vec{a} = \vec{1}/7$. The associated WLU's are distinguished with a superscript. In the experiments reported here all agents have the same reward function, so from now on we drop the agent subscript from the private utilities. Writing them out, the three WLU's provide the following reward functions:

$$\begin{aligned}
R_{WL^{\vec{0}}}(\underline{\zeta}_{,t}) &\equiv R_G(\underline{\zeta}_{,t}) - R_G(CL^{\vec{0}}_{\eta}(\underline{\zeta}_{,t})) \\
&= \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta}, t)) \; - \; \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta}, t) - 1) \\
R_{WL^{\vec{1}}}(\underline{\zeta}_{,t}) &\equiv R_G(\underline{\zeta}_{,t}) - R_G(CL^{\vec{1}}_{\eta}(\underline{\zeta}_{,t})) \\
&= \sum_{d \neq d_\eta}^{7} \phi_d(x_d(\underline{\zeta}, t)) \; - \; \phi_{d'} x_d(\underline{\zeta}, t) + 1) \\
R_{WL^{\vec{a}}}(\underline{\zeta}_{,t}) &\equiv R_G(\underline{\zeta}_{,t}) - R_G(CL^{\vec{a}}_{\eta}(\underline{\zeta}_{,t})) \\
&= \sum_{d \neq d_\eta}^{7} \phi_d(x_d(\underline{\zeta}, t)) \; - \; \phi_d(x_d(\underline{\zeta}, t) + a_d) \\
&+ \; \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta}, t)) \; - \; \phi_{d_\eta}(x_{d_\eta}(\underline{\zeta}, t) - 1 + a_d)
\end{aligned}$$

where d_η is the night picked by η and $a_d = 1/7$. The team game reward function is simply R_G. 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. Finally, note that when when viewed in attendance space rather than action space, $CL^{\vec{a}}$ is clamping to the attendance vector $\vec{v}_i = \sum_{d=1}^{7} \frac{u_{d,i}}{7}$, where $u_{d,i}$ is the i'th component (0 or 1) of the the d'th action vector. So for example, for $l = 1$, $CL^{\vec{a}}$ clamps to $\vec{v}_i = \sum_{d=1}^{7} \frac{\delta_{d,i}}{7}$, where $\delta_{d,i}$ is the Kronecker delta function.

The results we report in this section are averages over 20 runs with 60 agents, and throughout this article the error bars are too small to depict. Figure 1(a) shows the normalized world reward obtained for the different private utilities as a function of l. $R_{WL^{\vec{a}}}$ performs well for all 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 agents need to select six nights, a situation where the two clamping vectors are very similar ($\vec{1}$ and $\frac{\vec{6}}{7}$, respectively). $R_{WL^{\vec{0}}}$ shows a slight drop in performance when the

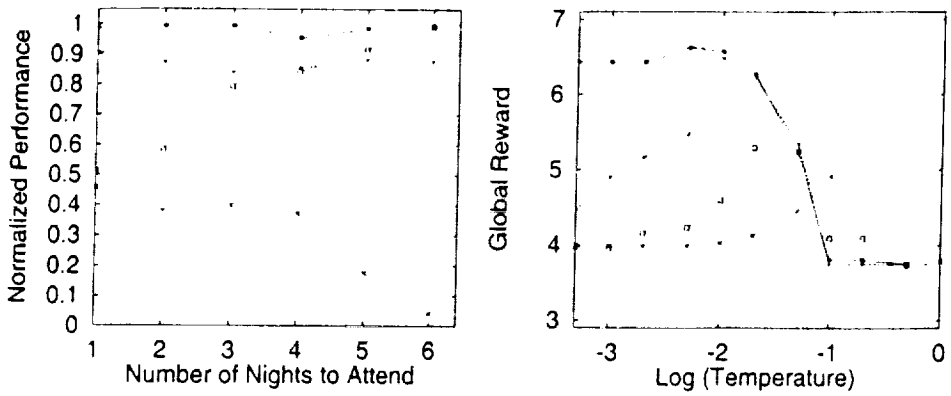

Figure 1: (a) Behavior of different reward function with respect to number of nights to attend. (b) Sensitivity of reward functions to internal parameters. (In both figures, $WL^{\vec{a}}$ is $\diamond$; $WL^{\vec{0}}$ is $+$; $WL^{\vec{1}}$ is $\square$; G is $\times$)

number of nights to attend increases, while R_G shows a much more pronounced drop. Furthermore, in agreement with our previous results [18], despite being factored, the poor signal-to-noise in R_G results in poor performance with it for all problems. (Temperatures varied between .01 and .02 for the three WL rewards, and between .1 and .2 for the G reward, which provided the respective best performances for each.) These results confirm our theoretical prediction of what private utility converges fastest to the world utility maximum.

We also studied the sensitivity of performance to the internal parameters of the learning algorithms. Figure 1(b) presents experiments with $l = 1$ for a set of different temperatures in the RL algorithms. (The two straight lines correspond to the optimal performance, and the "baseline" performance given by uniform occupancies across all nights.) $R_{WL^{\vec{a}}}$ is fairly insensitive to the temperature, until it gets 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.

5 Conclusion

In this article we considered how to configure large multi-agent systems where each agent uses reinforcement learning, and where there is no personalized (agent-specific) centralized communication and control. The inverse problem associated with such systems is how to initialize/update the individual agents' private utility functions so that their collective behavior optimizes a pre-specified world utility function. The mathematics of COINs is specifically designed for this problem, and in previous experiments systems based on it have far outperformed conventional "team game" systems, in which each agent has the world utility as its private utility function. Moreover, the gain in performance grows with the size of the system, typically reaching orders of magnitude for systems that consist of hundred of agents.

In those previous experiments the COIN-based private utilities had a free parameter, which we set to 0. However as we synopsised in this paper, a series of approximations in the mathematics of COINs allows one to derive an optimal value for that

parameter. We then repeated some of our previous computer experiments, only using this new value for the parameter. These experiments confirm that with this new value the system converges to significantly superior world utility values, with less sensitivity to the parameters of the agents' RL algorithms. This makes even stronger the arguments for using a COIN-based system rather than a team-game system. Future work involves improving the approximations needed to calculate the optimal private utility parameter value. In particular, given that that value varies in time, we intend to investigate having it be calculated in an on-line manner.

References

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

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

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

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

[5] 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.

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

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

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

[9] 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.

[10] 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.

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

[12] 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.

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

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

[15] K. Tumer and D. H. Wolpert. Collective intelligence and Braess' paradox. In *Proc. of the 17th Nat. Conf. on Artificial Intelligence*, 2000. to appear.

[16] D. H. Wolpert and K. Tumer. An Introduction to Collective Intelligence. In J. M. Bradshaw, editor, *Handbook of Agent technology*. AAAI/MIT Press, 2000. (available from `http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`).

[17] 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.

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