On Separating Agent Designer Goals from Agent Goals: Breaking the Preferences–Parameters Confound

Satinder Singh
Computer Science & Engineering
University of Michigan
baveja@umich.edu

Richard L. Lewis
Psychology
University of Michigan
rickl@umich.edu

Jonathan Sorg
Computer Science & Engineering
University of Michigan
jdsorg@umich.edu

Andrew G. Barto
Computer Science
University of Massachusetts
barto@cs.umass.edu

Akram Helou
Computer Science & Engineering
University of Michigan
akramhel@umich.edu

***Abstract*—Designers of artificial agents have goals and purposes that implicitly define a preference over possible agent behaviors. Nevertheless, it is rare that the designer knows the most preferred behavior in a form that allows it to be simply programmed into the agent. Instead, in building autonomous agents it is often more robust and useful for the designer to build an internal representation of goals into the agent itself. In particular, in reinforcement learning (RL) approaches the agent's goals are captured as maximization of cumulative reward. From the designer's point of view, the agent's goals (the reward function in RL) are parameters of agent design. It is standard to assume that the artificial agent's goals should be the same as the agent designer's goals. In this paper we show that this view unnecessarily and detrimentally confounds *preferences* of the agent designer with *parameters* of the artificial agent. We provide a formal framework that breaks the confound and defines a new problem in RL agent design: the search for the best reward function. We empirically demonstrate several implications of breaking the confound, the most important of which is the possibility of building RL agents that significantly outperform agents built with the confounded notion of reward. We briefly discuss how the new framework offers distinct advantages over other approaches to modifying reward functions.**

INTRODUCTION

In most artificial intelligence (AI) approaches to building agents, the agent designer's goals and purposes implicitly define a preference over possible behaviors of the agent in its environment: given two different behaviors, the designer is able to decide which (if any) of the two is preferred. Axiomatic decision theory shows that (under mild conditions) this assumption is equivalent to a utility function that maps entire behavior traces to scalar values, such that a behavior trace with a larger associated utility value is preferred to a behavior trace with a lower associated utility value [1]. Without loss of generality, then, we identify the goals of an agent designer with such utility functions.

Having such a utility function leaves open the question of how to design the agent. It is rare that the agent designer knows the most preferred or optimal behavior in a form that allows him or her to simply program it into the agent. More often the designer must produce an agent able to function in a complex environment, and must do so without access to a complete and accurate model of that environment. Additionally, the agent should work well in that environment despite having limited computational resources. In this paper we are interested in approaches to building *autonomous* agents in which the agent designer builds goals and purposes into the artificial agent itself, with the result that the agent's behavior in its environment is driven at least in part by its own internal representation of goals.

From the agent designer's point of view, the agent's goals are parameters of agent design: changing agent goals changes agent behavior. This is useful and powerful because it allows the agent designer to define what the agent is to do without having to worry about how it is to do it. This leads to more robust and adaptive behavior than does an attempt to program a fixed way of behaving. We focus here on reinforcement learning (RL) approaches that capture an agent's goals as the maximization of some cumulative measure of a *received* scalar signal or *reward*. We adopt the view that any agent with this formulation of goals is an RL agent (whether the agent learns through online interaction with its environment or plans offline with a model is irrelevant to this paper's discussion).

Note that there are at least two notions of goals (represented by distinct mathematical functions): the *agent designer's goals* and the *artificial autonomous agent's goals*. Should the two notions of goals be the same? This is the question of central interest to this paper. To the best of our knowledge, RL theory and more generally decision-theoretic and other AI approaches to building autonomous agents ignore this question by focusing on what the artificial agent should do to achieve its goals—however they came to be defined. This has served the very useful purpose of allowing great progress in algorithm development.

In practice, however, there is always a designer with his or her own goals, and these have to be translated somehow into the agent's goals. Indeed, in applying RL to practical problems, designers typically experiment with a variety of reward functions and other agent parameters to try to best achieve their design objectives. However, the adjustment of rewards is usually viewed by the designer as refining *one's own* goal because RL does not recognize a distinction between the designer's and the agent's goals: it implicitly or explicitly *confounds* them by assuming that the artificial agent's goals should be the same as the goals of the agent designer. Indeed, it is perhaps hard to see intuitively how it could be otherwise. But in this paper we will show that it is in fact otherwise. More specifically, our objective is to (a) clearly define the confound, (b) clearly define a meta-problem in agent design (a search for good agent reward functions) that results from breaking the confound, and (c) empirically demonstrate the *benefit* of solving the meta-problem. Put negatively, we will empirically demonstrate that the confound can be *detrimental*—that is, assuming equivalence between the two kinds of goals can lead to poorer performance (as assessed by the designer's goals) than permitting their separation.

Although for concreteness we focus our subsequent formalisms, results, and empirical investigations on RL agents, we expect that the need for separating agent goals from agent-designer goals holds for most if not all approaches to building autonomous agents (where we define an autonomous agent as one with its own internal goals.)

Separating Behavior-Preferences and Agent-Parameters in RL

We first briefly define the conventional framing of the RL problem and then present a new framing that separates the goals of the agent designer and the goals of the agent itself.

RL Problem Formulation

Dynamics: At each time step, an agent $\mathcal{G}$ receives an observation o from its environment $\mathcal{M}$, takes an action a, and repeats this process until a time horizon. The state at time step k, denoted h_k, is the history of interaction $o_1 a_1 \cdots o_{k-1} a_{k-1} o_k$ to time step k (subscripts denote time). The transition dynamics define a probability distribution over possible next states h_{k+1} as a function of h_k and a_k. Although representations of state more compact than full histories are used in practice, for this paper it is convenient to use the history as state for complete generality.

Reward Function: The reward function R maps states to scalar rewards. Associated with R is a cumulative measure that defines a return function, denoted $\mathcal{F}_R$, that maps a trajectory of states to a cumulative reward. For example: $\mathcal{F}_R(h_k) = \sum_{i=1}^{k} \gamma^i R(h_k^i)$, where h_k^i is the i^{th} state in a history of length $k \geq i$, and $0 \leq \gamma < 1$ is a discount factor. Our results hold for undiscounted formulations of return as well.

The Dual Role: The reward function, R, specifies preferences over agent behaviors; a history h_k is preferred over history h'_k if and only if $\mathcal{F}_R(h_k) > \mathcal{F}_R(h'_k)$. R also serves as agent parameters as follows. Fixing R and all the other agent parameters (denoted θ) makes the agent $\mathcal{G}(R;\theta)$ a fixed algorithm for extending every history by choosing actions, possibly stochastically, i.e., $\mathcal{G}(R;\theta) : \mathcal{H} \rightarrow P_A$, where $\mathcal{H}$ is the set of all histories and P_A is a probability distribution over the set of available actions A. Any optimization or learning specified by R is internal to the agent, and from the agent designer's point of view, agent $\mathcal{G}(R;\theta)$ just produces a fixed distribution over nonstationary and possibly adaptive behaviors. Thus, the conventional RL formulation confounds behavior-preferences and agent-parameters, i.e., it invests both of these roles into the same function R of state (hence the name "preferences–parameters" confound).

Breaking the Confound

We break the confound by (1) reserving the reward function R for its role as an agent parameter, and (2) defining a separate *objective utility function* to capture behavior preferences. To emphasize this, we relabel the reward function as an agent-internal[1] reward function denoted R_I.

Objective Utility Function: We denote the objective utility function $\mathcal{E}$, which maps a set of complete histories ($\mathcal{H}^c$) to scalar utilities, i.e., $\mathcal{E} : \mathcal{H}^c \rightarrow \Re$. What constitutes a complete history is part of the definition of the designer's goals. Common choices include a fixed (small or large) time horizon, an indefinite but finite horizon determined by some event such as the environment transitioning to a terminal state, or perhaps an infinite horizon. Note that this definition is very general. In particular, this does not require (but allows) there to be the kind of compositional additive form often assumed for reward.

The most important consequence of the separation afforded by our new RL problem formulation is this: the agent designer, who has access to the objective utility function, now faces the *meta-problem* of designing the internal reward function for the RL agent, so that in maximizing its expected cumulative internal reward, the agent achieves the highest possible utility (for the designer) as measured by the objective utility function.

Optimal Internal Rewards

The interaction between environment $\mathcal{M}$ and agent $\mathcal{G}(R_I;\theta)$ will produce a distribution, denoted μ, over the set of complete histories, such that for any $h \in \mathcal{H}^c$, $\mu(h|\mathcal{M}, \mathcal{G}(R_I;\theta))$ is the probability of obtaining sequence

[1]Note that the use of the adjective "internal" does not imply that the reward function is somehow modifiable by the agent itself; it remains immutable to the agent. Thus, all progress on RL algorithms remains available under our new framework proposed here.

h from the interaction. Presumably, the histories are the result of the RL agent somehow attempting to optimize internal returns defined by R_I (and the associated cumulative measure) under the architectural constraints imposed by θ. For example, the agent could be a Q-learning agent that uses a linear function approximator for storing the Q-value function, uses a step-size of 0.1, an exploration rate of 0.1, etc. These details can be complex and their impact hard to analyze, but fortunately for our purposes it is possible to abstract away from these details as follows.

How good is an agent $\mathcal{G}(R_I;\theta)$ in environment $\mathcal{M}$ for the agent designer with objective utility function $\mathcal{E}$? The expected utility to the designer is denoted $\mathbb{E}[\mathcal{E}|R_I]$, where for ease of exposition we have dropped the explicit dependence on the other quantities, $\mathcal{M}$ and θ, and is defined as follows:

$$\mathbb{E}[\mathcal{E}|R_I] = \sum_{h\in\mathcal{H}^c} \mathcal{E}(h)\mu(h|\mathcal{M},\mathcal{G}(R_I;\theta)). \tag{1}$$

The **meta-reward-optimization** problem faced by the agent designer is to find the optimal internal reward function, defined as

$$\begin{aligned} R_I^* &= \arg\max_{R_I\in R} \mathbb{E}[\mathcal{E}|R_I] \\ &= \arg\max_{R_I\in\mathcal{R}} \sum_{h\in\mathcal{H}^c} \mathcal{E}(h)\mu(h|\mathcal{M},\mathcal{G}(R_I;\theta)), \end{aligned} \tag{2}$$

where $\mathcal{R}$ is the space[2] of possible R_I. For simplicity, the optimal internal reward's dependence on the objective utility function and the other agent parameters is suppressed in the notation. Note that although we focus on the unusual aspect of optimizing the internal reward function parameters in this paper, in general the agent designer faces the joint optimization problem of optimizing all the agent parameters, including θ.

Comparing agents based on optimal and confounded reward functions

We can now consider comparing the performance of agents developed by solving the meta-reward-optimization problem to agents developed under the (conventional) confounded view of reward. Such comparisons—either theoretical or empirical—require specifying the **confounded reward function** that results from adopting the assumption that agent goals and designer's goals must be equivalent. We denote the confounded reward function by $R_\mathcal{E}$, to emphasize its direct dependence on the objective utility function $\mathcal{E}$. In some domains (such as our example below) $\mathcal{E}$ may already be defined in an additive form making the formulation of $R_\mathcal{E}$ transparent, in more general cases $\mathcal{E}$ would have to be transformed (perhaps approximately) into an additive form as required by the RL formulation. The following theorem follows immediately from the definitions above.

[2]In general, this will be determined by the constraints imposed by the agent architecture.

Theorem. *For any decision process $\mathcal{M}$, any objective utility $\mathcal{E}$, and any class of agents $\mathcal{G}(\cdot;\theta)$, if R_I^* is the optimal internal reward function found by solving the meta-problem as above from a reward search space $\mathcal{R}$ that includes $R_\mathcal{E}$, then $\mathbb{E}[\mathcal{E}|R_I^*] \geq \mathbb{E}[\mathcal{E}|R_\mathcal{E}]$.*

The theorem states that provided the search space of internal-reward functions, $\mathcal{R}$, includes the confounded reward function $R_\mathcal{E}$, the agent $\mathcal{G}(R_I^*;\theta)$ defined by solving the meta-problem under the new view of reward achieves at least as large an expected objective utility measured by $\mathcal{E}$ as the agent $\mathcal{G}(R_\mathcal{E};\theta)$ defined under the confounded view of reward.

While the proof of the theorem is straightforward, the implications of the theorem are not. In particular it shows that breaking the confound never hurts, at least in principle. This provides the foundation for further work on internal rewards.

When do we expect the inequality in the theorem to be a strict inequality, and when do we expect it to be an equality? Intuitively, when the agent's time and resources are "unbounded" we expect to obtain equality between the confounded and optimal internal rewards. For example, consider pairing a Markov environment, in which the agent designer's utility is defined additively as the average confounded reward over an infinite horizon, with a Q-learning agent with provably convergent settings of learning and exploration rates and a large enough discount factor and one that has enough memory to represent the Q-function as a look-up table. In such cases, we would not expect to benefit from breaking the confound, because it is known that such a Q-learning agent will converge to the optimal behavior with respect to the confounded reward and with an infinite horizon the slow rate of convergence will be overcome in the limit.

However, whenever the agent is "bounded" in some fashion, either through time or resource constraints or both, we expect to obtain a strict inequality—and in most interesting and real-world problems, agents will be bounded by constraints. We explore this intuition empirically here and leave its formalization to future work.

Empirical Illustration

We present empirical results to highlight three consequences of breaking the confound. For some agent and environment pairs:

1) Breaking the confound yields an agent that outperforms the agent which uses the confounded reward function ($\mathbb{E}[\mathcal{E}|R_I^*] > \mathbb{E}[\mathcal{E}|R_\mathcal{E}]$).
2) The optimal reward function R_I^* depends on the objective utility function $\mathcal{E}$ — but that dependence is more complex and interesting than might initially be thought.
3) The optimal reward function R_I^* depends on the agent parameters θ, i.e., all else being equal, the best reward

for one agent may not be the best reward for a different agent.

Although it can be demonstrated that breaking the confound can help many types of RL agents—including Q-learning agents, model building and planning agents, policy-gradient agents, and others—due to a lack of space, we focus here on a single agent–environment pair that illustrates each of the consequences listed above.

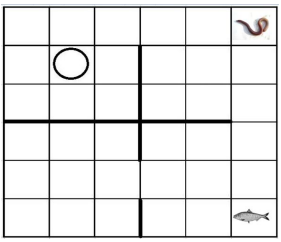


Figure 1. Fish-or-Bait Domain. See text for details

Fish-or-Bait Domain with a Learning Agent: Figure 1 shows a representation of a 6×6 grid world we call the Fish-or-Bait Domain. In the top right corner of the grid is an inexhaustible supply of worms (bait), and at the bottom right corner is an inexhaustible supply of fish. The thick lines represent barriers that have to be navigated around. The agent is shown as a circle. In each location the agent can move deterministically `North`, `South`, `East` or `West`. Any action that takes the agent off the grid or crosses a barrier fails with no resulting movement. In addition to the movement actions, the agent has actions `pick-up` and `eat`. At the worm location, `pick-up` results in the agent carrying a worm, and `eat` results in the agent eating a worm. The agent cannot carry more than one worm at a time. If the agent executes `eat` at the fish location, it succeeds in eating one fish, but only if it is carrying a worm; otherwise the action fails. At any other location, `eat` results in the agent consuming the worm if it is carrying one. The agent is *not-hungry* for one time step after eating a fish, *medium-hungry* for one time step after eating a worm, and *hungry* at all other times. The agent observes its (x, y) grid-location, whether it is carrying a worm, and its level of hunger; thus, the agent faces an MDP and its observations comprise the state.

The agent uses look-up-table-based Q-learning with an initial Q-value function of zero, a learning step-size of 0.1, and behaves according to an ϵ-greedy selection of actions (with probability $1 - \epsilon$ the agent selects the greedy action with respect to the Q-value function, and with probability ϵ selects a random exploratory action). Unless stated otherwise, we use $\epsilon = 0.1$.

The objective utility function is of additive form: the utility for a history is 0.04 times the number of worms eaten (being in state *medium-hungry*) plus 1.0 times the number of fish eaten (being in state *not-hungry*) in that history. Correspondingly, the confounded reward function $R_{\mathcal{E}}$ assigns the small reward of 0.04 to being in state *medium-hungry* and a large reward of 1.0 to being in state *not-hungry*.[3] We examined the effect of the choice of horizon by exploring the range of horizons from $1,000$ to $50,000$ in steps of $1,000$. Note that longer horizons are not extensions of smaller horizons, i.e., each horizon value corresponds to a distinct objective utility function and thus requires an independent experiment.

The search space of internal reward functions, $\mathcal{R}$, was defined as the set of all mappings from each of the three mutually disjoint hunger levels to scalars. Without loss of generality, we restricted each scalar to lie between -1.0 and 1.0. At each time step, the agent received reward equal to the scalar corresponding to its current hunger level. Note that the theorem's requirement that the reward search space contain $R_{\mathcal{E}}$ was satisfied by our choice of $\mathcal{R}$.

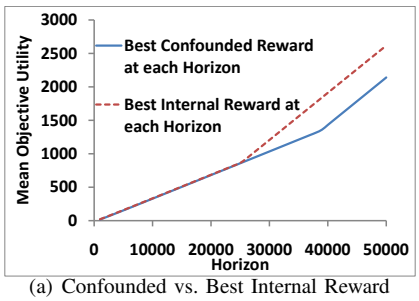
We solved the meta-reward-optimization problem by discretized brute-force search[4], although to show the advantage conferred by breaking the confound we do not need to find the optimal internal reward function, just one that outperforms the confounded reward. We denote the result of our brute-force search $\hat{R}^*_I$ to emphasize its approximate nature.

The six panels in Figure 2 show our main results. Figure 2(a) plots for each horizon value the mean (over 200 runs) objective utility for the agent with the best internal reward function for that horizon and the mean objective utility for the agent with the best[3] confounded reward for that horizon. We emphasize that the two curves in Figure 2(a) are not standard learning curves because the x-axis is not time steps; each point on the x-axis corresponds to a separate experiment with a distinct horizon in the objective utility function and we have connected the separate points into a curve for visual clarity.

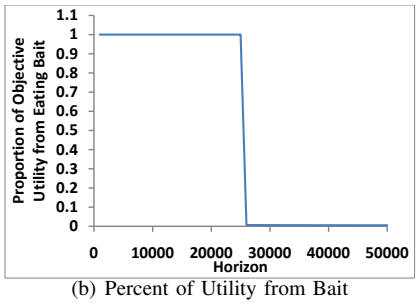
Dominance of the internal rewards: The best internal reward agent outperforms the best confounded reward agent for all values of the horizon. Although visually the difference between the two is small for the first 25,000 horizon values, the difference is strictly in favor of the best internal reward by an absolute amount more than 3.0. This validates our main empirical claim that a strict advantage is attained by breaking the confound.

[3] More precisely, we define $R_{\mathcal{E}}$ by fixing the reward for being not-hungry, medium-hungry and hungry to to $1.0 + c$, $0.04 + c$, and c, respectively, and we optimize via search the value of $c \in [-1, 1]$ separately for each finite horizon. We do this optimization to not unnecessarily put $R_{\mathcal{E}}$ at a disadvantage. Shifting the rewards by c is equivalent to initializing the value function to $-c/(1-\gamma)$. This effectively allows for optimistic initialization of the value function (which is initialized to 0 in all our experiments).

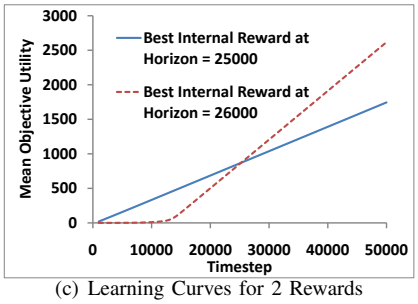
[4] For each level of hunger we searched over the following: the point 0, points in the range $\pm[0.01, 0.1]$ in increments of 0.01, and points in the range $\pm[0.2, 1]$ in increments of 0.1. These intervals and increments were determined through several iterations of the experiment.



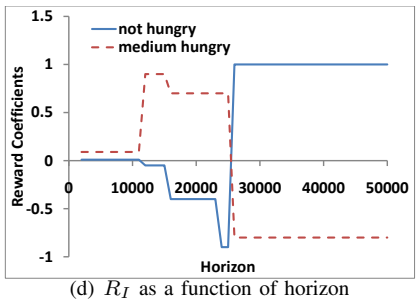
(a) Confounded vs. Best Internal Reward



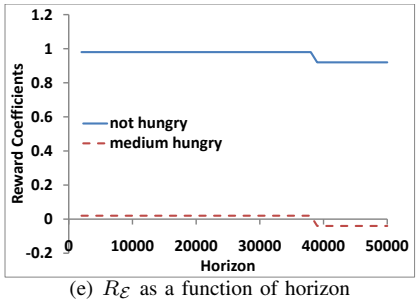
(b) Percent of Utility from Bait



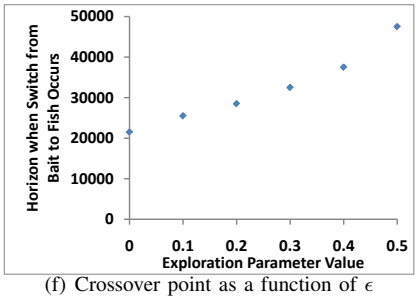
(c) Learning Curves for 2 Rewards



(d) R_I as a function of horizon



(e) $R_\mathcal{E}$ as a function of horizon



(f) Crossover point as a function of ϵ

Figure 2. Results for the Fish-or-Bait Domain. See text for details.

Sensitivity to the horizon: Figure 2(a) has another interesting property: the low constant initial advantage is followed by a sharp rise in the advantage at a horizon of 26,000, which is followed by a roughly constant absolute advantage of about 450 after a horizon of 40,000. This results from an interesting intuitive property of the Fish-or-Bait domain. Specifically, the behavior to eat bait is rather simple—find the worm location and then stay there eating. The behavior to eat fish is more complex—find the worm location, pick up a worm and carry it to the fish location without eating the worm at any time, eat the fish, and then go back to the worm location to repeat the cycle. Thus, it is far easier for the agent to learn to eat bait than to learn to fish. However, the longer the horizon, the more it becomes worth it for the agent to learn to fish because there is time within the horizon both to learn to fish and to exploit the greater reward that comes from fishing. Figure 2(b) shows this effect of horizon. It plots the proportion of mean objective utility obtained by eating bait as a function of horizon. For horizons of 25,000 or less almost all the utility comes from eating bait. For horizons of 26,000 and above almost all the utility comes from eating fish. Solving the meta-reward-optimization problem thus specializes the internal reward function to detailed properties of the objective utility function, in this case the horizon. To illustrate that this is indeed what is happening, Figure 2(c) compares the performance of two specific agents. The first uses the $\hat{R}_I^*$ found for a horizon of 25,000, i.e., just *before* the shift from eating bait to fishing. The second uses the $\hat{R}_I^*$ found for a horizon of 26,000, i.e., just *after* the shift from eating bait to fishing. As can be seen in Figure 2(a), the first agent quickly learns to find the bait location and steadily eat there to achieve a constant reward increment of 0.04, while the second agent takes some time (about 12,000 steps) to learn how to fish and then achieves a constant rate of reward by fishing that is higher than that of eating bait. By time 26,000 the second agent's increased rate of reward helps it overtake the total reward of the first agent. Another effect of the relative difficulty of learning to fish versus learning to eat bait is seen in Figure 2(a): the agent based on breaking the confound can learn to eat fish and gain the higher rate of return much sooner (at a horizon of 26,000) than an agent that preserves the confound (at a horizon of 40,000).

Violation of monotonicity: Figure 2(d) plots as a function of horizon the optimal reward coefficients for the two hunger-level conditions of not-hungry (when the agent has just eaten fish) and medium-hungry (when the agent has just eaten bait). For the 12,000 steps it takes the agent to learn to first get fish, the best internal reward gives a small positive reward to medium-hungry (eating bait) and no reward to not-hungry (eating fish) because there is no point in rewarding eating-fish up until this horizon.

Something very interesting happens for horizons between 12,000 and 25,000, when it is possible to learn to eat fish but there isn't enough time to exploit that learning. The best internal reward makes it very rewarding to eat bait and very costly to eat fish. This ensures that for this range of horizons the agent eats bait and avoids fish.

From 26,000 steps onwards this polarity reverses and the agent highly rewards eating fish and highly negatively rewards eating bait. The latter negative reward serves to

make sure that the agent is not distracted from the goal of learning to fish. Note that in the middle range of horizons the best internal reward in effect reverses the desirability of bait versus fish expressed in the objective utility function.

The general point here is that the *optimal reward function need not be a monotone transform of* $\mathcal{E}$—that is, the optimal reward may not even preserve all preference orderings over states imposed by the objective utility. For comparison to Figure 2(d), which plots the optimal reward coefficients, Figure 2(e) plots as a function of horizon the same two coefficients for the best[3] confounded reward, which by definition preserves the relative ordering of fish and bait in the objective utility function. The switch from slight positive reward for being medium-hungry to slight negative reward for being not-hungry at about 40,000 steps is crucial because as explained above prior to that horizon the best confounded reward agent should eat bait and after that horizon should avoid eating bait.

Sensitivity to agent parameters: Finally, Figure 2(f) shows the sensitivity of $\hat{R}_I^*$ to other parameters of the agent, in this case the exploration rate parameter ϵ. The larger the ϵ, the more often the agent will do a random action. This makes it more difficult for the agent to learn to fish because it has to avoid eating the bait for the many steps it takes to carry the bait to the fish pond. As seen in Figure 2(f), the horizon at which the optimal-reward induced crossover from eating bait to eating fish occurs increases as a function of ϵ. As it gets harder to learn to fish, the best internal reward focuses on the bait for longer horizons.

Summary: Our results for the Fish-or-Bait domain illustrate the three consequences of breaking the confound that we set out at the beginning of this section. First, using an internal-reward function that solves the meta-reward-optimization problem can lead to significantly better agents as measured by the objective utility function when compared to agents using the confounded reward function, i.e., agents using the objective utility function transformed into reward. Second, the best internal reward function depends on the details of the objective utility function, but the nature of this dependence is not necessarily simple: the best reward function need not be a straightforward (perhaps monotonic) transformation of the objective utility function. Third, the best internal reward function depends on the internal parameters of the agent.

Finally, we note that the experimental results reported here were designed to illustrate the existence and interesting properties of simple forms of good internal reward functions. The reward functions were much simpler, for example, than reward functions that would more directly provide the optimal value function or an optimal policy to the agent to speed up learning. The derivation of efficient search algorithms for finding good and simple internal rewards and compact representations for expressing internal rewards is left to future work.

Related Work

Previous work has recognized that the reward function itself is an important element of an agent's design, e.g., work on curious agents [2], Dyna [3], exploration bonuses [4], reward shaping [5], Rmax [6], PAC-MDP [7] intrinsically motivated RL [8], and contributions from the developmental robotics community [9], [10], [11], [12]. While each of these approaches modify what we termed the confounded reward to improve performance, they differ from our framework in that they do not make explicit a fully independent representation of the agent designer's goals, do not identify the preferences–parameters confound, and do not make explicit the resulting meta-optimization problem. As a result, the common objective of the alternative approaches referenced above is to help accelerate the achievement of asymptotic behavior (what the agent with confounded reward function would do in the limit). This is most clearly seen in the reward shaping work which provides conditions under which a modified reward function will provably lead to the same asymptotic behavior as a given confounded reward function, with the hope that this will be achieved faster. In our framework, on the other hand, the form of a good internal reward function is *unconstrained* by the form of the objective utility function. A consequence of this is that the behavior learned by an agent on the basis of its internal reward function can be qualitatively very different from the asymptotic behavior that would be learned via a confounded reward function, as was seen with the monotonicity violation in the Fish-or-Bait domain for objective utility functions with small horizons.

In the most closely related prior work, Singh, Lewis, and Barto ([13]) provided an evolutionary perspective on the origin of rewards in natural agents faced with distributions of environments. However, they did not explicitly identify the confound central to the present paper which instead focuses on the machine learning perspective of designing an artificial agent, possibly for a single environment, thereby yielding the theoretical framework and results reported here.

Conclusion

In this paper we identify the need to separate autonomous agent goals from agent designer goals in general, and the confounded nature of reward in conventional RL in particular, as well as provide a new framework that puts into sharp focus the search for good reward functions as a well-defined computational problem in agent design whose solution has significant implications for performance. The results reported here are early steps toward understanding this problem. But they do suggest that the relationship between good internal reward functions on the one hand, and domains, agent architectures, and designer goals on the other, is sufficiently rich and of sufficient practical import to warrant much further theoretical and empirical work.

REFERENCES

[1] A. Mas-Colell, M. D. Whinston, and J. R. Green, Microeconomic Theory. Oxford University Press, 1995.

[2] J. Schmidhuber, "Curious Model-Building Control Systems," in IEEE International Joint Conference on Neural Networks, 1991, pp. 1458–1463.

[3] R. S. Sutton, "Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming," in The Seventh International Conference on Machine Learning. Morgan Kaufmann, 1990, pp. 216–224.

[4] P. Dayan and T. J. Sejnowski, "Exploration bonuses and dual control," Machine Learning, vol. 25, no. 1, pp. 5–22, 1996.

[5] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in Proceedings of the 16th International conference on Machine Learning, Bled, Slovenia, 1999, pp. 278–287.

[6] R. I. Brafman and M. Tennenholtz, "R-max - a general polynomial time algorithm for near-optimal reinforcement learning," Journal of Machine Learning Research, vol. 3, pp. 213–231, 2003.

[7] J. Asmuth, M. L. Littman, and R. Zinkov, "Potential-based Shaping in Model-based Reinforcement Learning," in Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08), 2008, pp. 604–609.

[8] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically Motivated Reinforcement Learning," in Proceedings of Advances in Neural Information Processing Systems 17, 2005.

[9] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic Motivation Systems for Autonomous Mental Development," IEEE Transactions on Evolutionary Computation, vol. 11, no. 2, pp. 265–286, April 2007.

[10] M. Schembri, M. Mirolli, and G. Baldassarre, "Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot," in Proceedings of the 6th International Conference on Development and Learning, 2007.

[11] E. Uchibe and K. Doya, "Finding intrinsic rewards by embodied evolution and constrained reinforcement learning," Neural Networks, vol. 21, no. 10, pp. 1447–1455, 2008.

[12] D. H. Ackley and M. Littman, "Interactions between learning and evolution," Artificial Life II, SFI Studies in the Sciences of Complexity, vol. X, 1991.

[13] S. Singh, R. L. Lewis, and A. G. Barto, "Where Do Rewards Come From?" in Proceedings of the Annual Conference of the Cognitive Science Society, 2009, pp. 2601–2606.