Specifying Behavior Preference with Tiered Reward Functions

Zhiyuan Zhou [1] **Henry Sowerby** [1] **Michael L. Littman** [1]

Abstract

Reinforcement-learning agents seek to maximize a reward signal through environmental interactions. As humans, our job in the learning process is to express which behaviors are preferable through designing reward functions. In this work, we consider the reward-design problem in tasks formulated as reaching desirable states and avoiding undesirable states. To start, we propose a strict partial ordering of the policy space. We prefer policies that reach the good states faster and with higher probability while avoiding the bad states longer. Then, we propose an environment-independent tiered reward structure and show it is guaranteed to induce policies that are Pareto-optimal according to our preference relation.

1. Introduction

Reinforcement learning (Sutton & Barto, 1998) is concerned with the problem of learning to behave to maximize a reward signal. In biological systems, this reward signal is considered to be the organism's motivational system, using pain and pleasure to modulate behavior. In engineered systems, however, rewards must be selected by the system designer. We view rewards as a kind of programming language—a specification of the agent's target behavior (Littman et al., 2017). As arbiters of correctness in the learning process, humans bear the responsibility of authoring this program.

There are two essential steps in designing reward functions. First, one must decide what kind of behavior is preferable and should be conveyed. Then, there's the choice of reward function that induces such behavior.

In this paper, we look at a specification language that allows for the expression of desirable states (goals and subgoals) and undesirable states (obstacles). Even in this simple setting, providing precise trade-offs is difficult. Is it better for an agent to increase the chance of getting to the goal by 5% if it also incurs 8% higher probability of hitting an obstacle? Is it better to increase by 50% the probability of getting to a goal if the expected time of getting there also increases by 20%? There is no universal preference over behavior and having to explicitly write down all possible trade-offs is challenging. Even if the reward designer has a way of expressing preferences for all possible exchanges, it can be difficult, impossible even, to design a reward function that captures them without prior knowledge of the environment.

Our contribution is twofold: First, we define a preference over the entire policy space via a strict partial ordering on outcomes. Then, we introduce a class of environment-independent tiered reward functions that provably induce Pareto-optimal policies with respect to this preference ordering.

1.1. Favorable Policies

In the goal–obstacle class of tasks we consider, preferences over policies are simplest in the deterministic setting. We imagine all states are either goal states, obstacle states, or neither (background states). All goal states and obstacle states are absorbing. A policy from a fixed known (background) start state will either first reach a goal in g steps, first reach an obstacle in o steps, or run forever remaining in background states without reaching either. We prefer a policy that reaches a goal in g_1 steps to one that reaches a goal in g_2 steps if $g_1 < g_2$. (Reaching a goal faster is better.) We prefer a policy that reaches a goal to one that does not. We prefer a policy that reaches neither a goal nor obstacle to one that reaches an obstacle. And we prefer a policy that reaches an obstacle in o_1 steps to one that reaches an obstacle in o_2 steps if $o_1 > o_2$. (Taking longer to encounter an obstacle is better.) Two different policies that both reach a goal or both reach an obstacle and take the same number of steps to do so are considered equally good. (Steps are indistinguishable.) Thus, in deterministic domains, these preferences form a total order.

As pointed out before, preferences are less clear in a stochastic setting because there can be trade-offs between different outcomes and their probabilities. However, some comparisons are arguably clear cut. Informally, if one policy induces uniformly better outcomes than another—being more

[1]Department of Computer Science, Brown University, Providence RI, United States. Correspondence to: Zhiyuan Zhou <zhouzy@brown.edu>.

The Many Facets of Preference Learning Workshop at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

likely to reach a goal and doing so faster, being less likely to reach an obstacle and getting there more slowly—we prefer such a policy. If the policies can't be directly compared, we propose to be indifferent between them. Thus, we replace the standard reinforcement-learning notion of optimality with Pareto-optimality (Mornati, 2013)—seeking a policy that is either preferred or incomparable to every other policy. Pareto-optimal policies are commonly adopted in the subfield of multiobjective RL (Vamplew et al., 2011).

1.2. Reward Design

Policies in general are hard to express through reward functions (Amodei & Clark, 2016), and some are even impossible to convey with a Markov (state–action-based) reward (Abel et al., 2021). Even when policies are expressible, designing bad reward functions can lead to undesirable or dangerous actions (Amodei & Clark, 2016), easy reward hacking (Amodei et al., 2016), and more. We seek to design good reward functions, which can be characterized by many properties, such as interpretability and learning speed (Devidze et al., 2021). But the most important property a reward function must have is to guarantee the adoption of a desired policy. As we will show later in Section 5, even some intuitively correct reward designs can lead to suboptimal policies. To hedge against this concern, we introduce a tiered reward structure that is guaranteed to induce Pareto-optimal policies. Intuitively, we partition the state space into several tiers, or goodness levels. States in the same tier are associated with the same reward, while states in a more desirable tier are associated with a proportionally higher reward. We prove that these tiered reward structures, with the proper constraints between reward values, induce Pareto-optimal behavior.

2. Related Work

There are many papers that deal with specifying behavior through rewards. Reward machines (Icarte et al., 2018; 2022) are finite state machines that compose reward functions and allow different rewards to be delivered dependent on the agent's trajectory. They reveal the structure of the reward function to the RL agent to support decomposition of complex tasks. Our focus on how to provide incentives for specific outcomes is complementary and the two approaches can be used in concert. Temporal logic based languages (Littman et al., 2017; Camacho et al., 2017; Li et al., 2017; Camacho et al., 2019) have been used to specify behavior. Though these methods can be more expressive, they often lead to intractable planning and learning problems due to state-space explosion issues (Wongpiromsarn et al., 2010). We offer a different expressibility–tractability tradeoff. Preference-based RL methods (Wirth et al., 2017; Brown et al., 2019) learn a reward function based on a dataset of preferences over trajectories. But, as we have shown, preferences can be very difficult to express. Our reward scheme relieves the need for environment-specific preference datasets created by human experts. Multi-objective RL (Vamplew et al., 2011; Toro Icarte et al., 2018) allows for different tasks to be specified through a set of reward functions. Our work proceeds in the orthogonal direction by designing a single reward function to trade off among multiple behaviors.

3. Background

First, we will establish the problem setting. We view a reinforcement-learning environment as a Markov Decision Process (MDP), with state space S, action space A, transition model T, reward function R, and discount factor γ. A policy $\pi : S \times A \to [0, 1]$ is a mapping from the current state to a probability distribution of the action to be taken. The optimal policy starting from some initial state s_0 in the MDP is defined as any reward-maximizing policy $\pi^* \in \text{argmax}_\pi \mathbb{E}[\sum_t \gamma^t r_t | s_0, \pi]$. To make the reward-design problem as simple as possible for designers, we limit the reward function $R : S \to \mathbb{R}$ to be defined solely on states. In goal–obstacle tasks, the goal states and obstacle states are absorbing.

We imagine the state space S as exhibiting a tiered structure, where higher tiers are more desirable than lower tiers, and states within the same tier are equally desirable. We formally define a Tier MDP as:

Definition 3.1. k-Tier Markov Decision Process: A Tier MDP is an MDP with state space S, action space A, transition model $T : S \times A \times S \to \mathbb{R}$, reward function $R : S \to \mathbb{R}$, and discount factor γ. The state space is partitioned into k tiers, where $S = S_1 \cup S_2 \cup \ldots \cup S_k$ and $S_i \cap S_j = \emptyset, \forall i \neq j \in 1, 2, ..., k$. The reward function has the form $R(s) = r_i, \forall s \in S_i, i = 1, 2, ..., k$. In addition, $r_1 < r_2 < \ldots < r_k$.

As an example, the grid world from Russell & Norvig (2010), as illustrated in Figure 1, could be formulated as a 3-Tier MDP—the goal state is one tier (S_3), the lava state one tier (S_1), and all other states reside in the background tier (S_2). It is important to note that we put no constraints on how many states could be in each tier, nor how many tiers there can be. Therefore, the framework has a high degree of generality; any finite MDP with reward defined on states could be formulated as a Tier MDP by placing states with the same reward in the same tier. However, the Tier MDP is most useful when there are clear good and bad states in the state space, such as when there are goal and obstacle states, or even states of intermediate desirability such as subgoal states. In the following sections, we will show how to perform reward design in Tier MDPs.

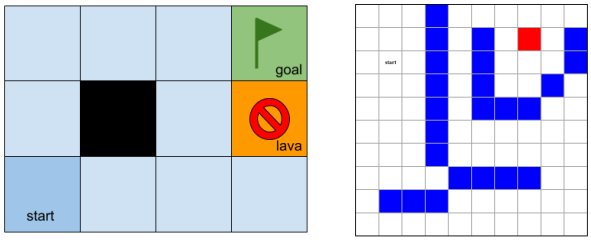

Figure 1. Left: Russell/Norvig grid world. Objective is to reach the goal (green) without first visiting lava (red) ($\gamma = 0.9$). Right: A puddle world. Objective is to reach the goal (red) without crossing any puddles (blue) ($\gamma = 0.99$). For both environments, the agent has probability 0.8 of moving to the specified direction at each step, and probability 0.1 of slipping to either orthogonal side.

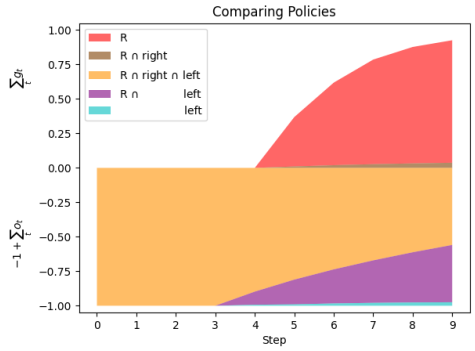

Figure 2. Visualization of the policies of always going left and always going right in the (stochastic) Russell/Norvig grid. The policy R is the same as R in Figure 4. To avoid color overlapping, we separated each policy into disjoint regions visualized by distinct colors. Each colored region in the figure represent the probability-region of one or more policies, joined by $\cap$. For example, the policy "always right" covers the areas in brown and orange.

4. Policy Ordering

A policy can be thought of as inducing a probability distribution over an infinite set of outcomes (specifically the probability of reaching each of the states after t steps, for all t). In goal–obstacle tasks, policies can be characterized by statistics such as probability of reaching the goal and probability of avoiding the obstacle for each possible horizon length.

For the moment, we will limit the problem space to 3-Tier MDPs for simplicity, generalizing to k-Tier MDPs in Section 6. In a 3-Tier MDP, we will call the 3 tiers obstacles (S_1), background (S_2), and goals (S_3), in order of increasing desirability. States in S_1 and S_3 are absorbing. We define o_t to be the probability of being in obstacle S_1 at timestep t, and g_t that of being in a goal S_3 at t.

Given two policies π^A and π^B, we say π^A *dominates* π^B when both of these inequalities hold (and not both being strictly equal at all times):

$$\sum_{i=0}^{t} o_i^A \leq \sum_{i=0}^{t} o_i^B, \quad \forall t = 0, 1, 2, ..., \infty,$$

$$\sum_{i=0}^{t} g_i^A \geq \sum_{i=0}^{t} g_i^B, \quad \forall t = 0, 1, 2, ..., \infty.$$

In words, one policy dominates another if it gets to the goal faster, while delaying encountering obstacles longer. The set of policies that are not dominated by any other policy is the set of *Pareto-optimal* policies. Because there is a finite number of policies and domination is transitive, the set of Pareto-optimal policies is non-empty.

Going back to the example of the Russell/Norvig grid, we can visualize how the probability of reaching the goal (g_t) and reaching lava (o_t) changes over time for different policies. Consider two simple policies on the Russell/Norvig grid—(1) going left from all states ("left") and (2) going right from all states ("right"). We visualize each policy's outcomes as a shaded area upper bounded by $\sum_t g_t$ and lower bounded by $-1 + \sum_t o_t$ in Figure 2. This visualization can be understood as separating the probability space into two, with the goal-reaching probability on the top half of the y-axis in $[0, 1]$ and obstacle-hitting probability in the bottom half of the y-axis in $[-1, 0]$. With this visualization, a Pareto-dominated policy will cover an area that is entirely enclosed by that of a dominating policy because of lower goal-reaching probabilities on the top half and higher obstacle-hitting probabilities on the bottom half. As Figure 2 shows, "right" and "left" do not cover each other, so they are incomparable. Specifically, "right" has a slightly higher probability of reaching the goal (brown), but "left' has a lower probability of reaching the lava (purple and teal).

For comparison, we plot another policy, which we call R, that is state-dependent and moves in the direction of the goal. For policy R, the probability of reaching the target increases with time because each step has a 20% slip probability; agents could slip early on and take longer to reach the goal. Note that area covered by R (red, brown, orange, and purple) completely subsumes that of "right" (brown and orange), demonstrating that "right" is dominated by R. Policy "left", on the other hand, is not dominated by R because it has a lower probability of reaching lava (teal). However, "left" is not Pareto-optimal, because it is dominated by policy G in Figure 4 (not shown in this plot).

Pareto-optimal policies are interesting to consider for two main reasons. First, Pareto-optimal behavior always exists, even when policies that achieve other reasonable things do

not. Take the puddle world environment for example. In this 10×10 grid world, the objective is to reach the goal state without crossing any puddles (Figure 1 right). We use this environment, which is not formulated as a tiered MDP, only to demonstrate that Pareto-optimal policies exist, and so we make the goal state absorbing while puddles and background states are not. Each step has probability p of succeeding, and probability $\frac{1-p}{2}$ of slipping to either sides. However, when $p < 0.87$, there is no state–based reward function that can encourage the agent to circle around on dry land to get to the goal (Littman, 2015). Instead, the agent will either cower in the left corner (e.g., with -1 each step, -70 for puddle, and $+0$ at goal), cross over the first puddle strip and pass through the dry land on top (e.g., with -1 each step, -35 for puddle, and $+0$ at goal), or completely ignore the existence of puddles and get to the goal as directly as possible (e.g., with -1 each step, -0 for puddle, and $+0$ at goal). Even though there are no rewards to express the target objective, Pareto-optimal policies that encourage reaching the goal quickly and obstacles slowly still exist and can be expressed.

Secondly, Pareto-optimality resolves the preference problem by defining a strict partial ordering over the entire policy space. Although the policies on the Pareto frontier are incomparable among themselves, they are all better than the set of Pareto-dominated policies. We simply deem the set of Pareto-optimal policies the desirable behavior, and all others undesirable. Next, we show how to design rewards that guarantee reward–optimal policies are selected from this set.

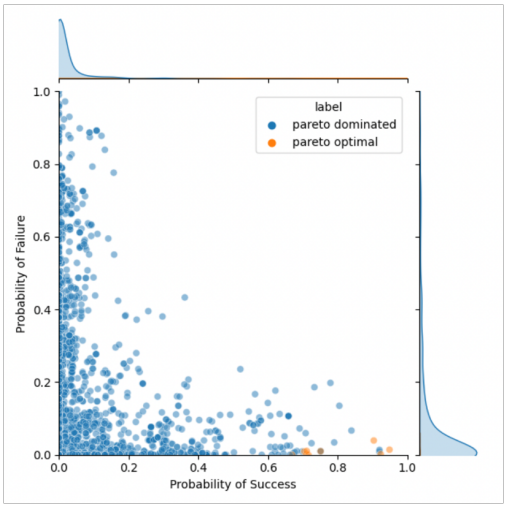

Figure 3. Random policies sampled from the Russell/Norvig grid. Each point in the scatter plot represents one random policy's probability of success (reaching the goal) and failure (reaching the lava), showing that the majority of policies in the policy space is Pareto-dominated.

Even though Pareto-optimal policies are desirable, they are hard to find. As Figure 3 shows, most random policies in the Russell/Norvig grid are not Pareto-optimal. Therefore, we need a reliable way of deriving these policies. In the next section we describe reward functions that induce such policies, allowing reinforcement learning to solve this problem.

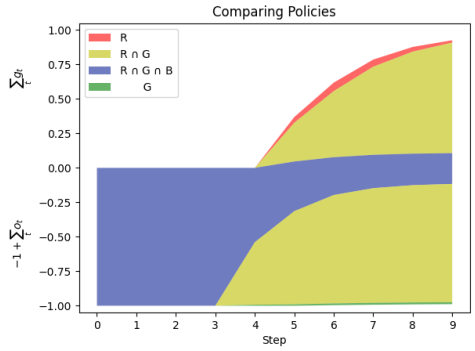

Figure 4. Visualization of three different policies (R, G, B) on Russell/Norvig grid. Visualization scheme is the same as described in Figure 2.

5. Tiered Reward

In this section, we seek a sufficient condition on the reward function so that optimizing expected discounted reward will always result in a Pareto-optimal policy with respect to our preference relation.

Definition 5.1. Pareto-optimal rewards: A reward function $R(s)$ is called *Pareto-optimal* if the policy it induces, $\pi_R \in \operatorname{argmax}_\pi \mathbb{E}[\sum_t \gamma^t r_t | s_0, \pi]$, is Pareto-optimal.

Even some reasonable-sounding reward functions need not be Pareto-optimal. Going back to the Russell/Norvig grid example, an intuitive reward design would be requiring $r_{lava} < r_{back} < r_{goal}$. Consider three example reward functions in Table 1 that satisfy this constraint:

Policy	r_{lava}	r_{back}	r_{goal}
R	-1	-0.1	$+1$
G	-1	0	$+0.5$
B	-1	-0.9	0

Table 1. Three example reward functions of Russell/Norvig grid world.

Both R and G are Pareto-optimal, while B is Pareto-dominated (see Figure 4, where B's areas are entirely enclosed by that of R and of G). Roughly, B doesn't encourage getting to the goal and is also not particularly good at avoiding lava.

In fact, many of the reward functions that satisfy $r_{lava} <$

$r_{back} < r_{goal}$ are not Pareto-optimal. Out of 1000 such rewards that we sampled randomly, 90.5% were Pareto-dominated. Next, we present a simple rule that is sufficient to guarantee environment-independent Pareto-optimal reward functions in 3-Tier MDPs.

Definition 5.2. Tiered Reward: In a 3-Tier Markov Decision Process with discount factor $\gamma \in (0, 1)$, a reward function defined by

$$R(s) = \begin{cases} r_{obs} & \text{if } s \in S_1 \\ r_{back} & \text{if } s \in S_2 \\ r_{goal} & \text{if } s \in S_3 \end{cases}$$

is considered a *Tiered Reward* if

$$r_{obs} < \frac{1}{1-\gamma} r_{back} < r_{goal}.$$

and states in S_1 and S_3 are absorbing.

Theorem 5.3 (Pareto-optimal rewards in 3-Tier MDP)**.** *In a 3-Tier Markov Decision Process, a Tiered Reward is Pareto-optimal.*

Proof. Let π^* be the optimal policy induced with Tiered Reward $R(s)$. Suppose, for the sake of contradiction, there exists some policy π that dominates π^*. Then, by our definition of Pareto dominance,

$$\sum_{i=0}^{t} o_i \leq \sum_{i=0}^{t} o_i^*, \quad \forall t = 0, 1, 2, ..., \infty,$$

$$\sum_{i=0}^{t} g_i \geq \sum_{i=0}^{t} g_i^*, \quad \forall t = 0, 1, 2, ..., \infty,$$

where o_t and g_t are the probabilities of reaching obstacles and goals in exactly t steps following π, and o_t^* and g_t^* are the same for π^*. We can write the value function (of π being evaluated on $R(s)$) as

$$V = \sum_{t=0}^{\infty} g_t(\gamma^t r_{goal} + \sum_{j=0}^{t-1} \gamma^j r_{back}) + o_t(\gamma^t r_{obs} + \sum_{j=0}^{t-1} \gamma^j r_{back}).$$

The value of π^* (V^*) can be written similarly. Denote

$$f_t^g = \gamma^t r_{goal} + \sum_{j=0}^{t-1} \gamma^j r_{back}, \text{ and}$$

$$f_t^o = \gamma^t r_{obs} + \sum_{j=0}^{t-1} \gamma^j r_{back}.$$

That is, f_g^t is the reward obtained on a trajectory that reaches a goal in t steps and f_o^t is the reward obtained on a trajectory that reaches an obstacle in t steps. With $r_{obs} < \frac{1}{1-\gamma} r_{back} < r_{goal}$, we show below that f_t^g is strictly decreasing and f_t^o strictly increasing with respect to t.

Then,

$$\begin{aligned} V - V^* &= \sum_{t=0}^{\infty} (g_t - g_t^*) f_t^g + \sum_{t=0}^{\infty} (o_t - o_t^*) f_t^o \\ &= \sum_{t=0}^{\infty} (\sum_{j=0}^{t} g_j - g_j^*)(f_t^g - f_{t+1}^g) \\ &\quad + \sum_{t=0}^{\infty} (\sum_{j=0}^{t} o_j - o_j^*)(f_t^o - f_{t+1}^o) \quad (*) \\ &> 0 + 0 \\ &= 0. \end{aligned}$$

We have shown, through the value function, that π is strictly better than π^* with respect to the reward function R. But π^* was chosen to optimize R, so that's a contradiction. Since no such π can exist, that means π^* is not dominated by any policy, and is therefore Pareto-optimal. □

We leave some details of the proof in Appendix A and provide some intuition for Tiered Reward here. The middle term in Definition 5.2, $\frac{1}{1-\gamma} r_{back}$, is equal to the cumulative discounted return for infinitely getting a reward in the background tier ($(1 + \gamma + \gamma^2 + ...) r_{back}$). So, in a gross simplification, as long as the reward at the goal is more appealing than infinitely wandering in background states, and the obstacle less appealing, the reward induces behavior that arrives at the goal early and avoids the obstacles. Following this simple constraint, we as reward designers can easily create Pareto-optimal reward functions without requiring knowledge of the transition probabilities in the environment.

6. Generalizing to k Tiers

In Sections 4 and 5, we limited the discussion to 3-Tier MDPs. But MDPs with more than 3 tiers can usefully model important problems such as those with well-defined subgoal states. Specifically, each subgoal region could be its own tier, instead of being grouped into one big background tier. Even though these problems could still be solved as a 3-Tier MDP, more knowledge about the environment could help design better reward functions and accelerate learning. So, in this section, we consider the reward-design problem in Tier MDPs with more than three tiers.

Definition 6.1. Tiered Reward: In a k-Tier($k > 3$) Markov Decision Process with discount factor $\gamma \in (0, 1)$ where the goal tier (k) is absorbing, the reward function R is a Tiered Reward if $R(s) = r_i, \forall s \in S_i, i = 1, 2, ..., k$, for reward values $r_1, r_2, ..., r_k \in \mathbb{R}$, that satisfy

$$r_1 < (\frac{1}{1-\gamma}) r_2 < (\frac{1}{1-\gamma})^2 r_3 < \cdots < (\frac{1}{1-\gamma})^{k-1} r_k \leq 0.$$

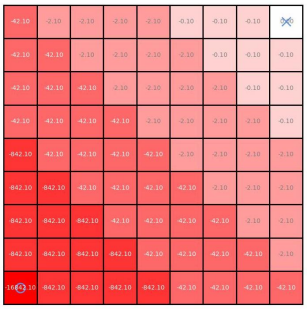

Figure 5. A Tiered Reward in a grid world with 6 tiers. Start state is in the bottom left corner, goal state in top right.

Notice that the Tiered Reward in the k tiers case uses a stricter condition than that of 3 tiers. First, all reward values are non-positive. We enforce this constraint not only because of mathematical convenience (used later in Equation 1), but also because step-wise penalty has been proved to support faster learning (Koenig & Simmons, 1993). Specifically, with a zero-initialized value function, step penalties create an incentive for the agent to try state–action pairs it has never experienced before, resulting in rapid exploration. Secondly, the reward values of higher tiers are exponentially greater than the lower ones. For adjacent tiers i and $i+1$, the reward values always satisfy $r_i < \frac{1}{1-\gamma} r_{i+1} < 0$. One such reward is visualized in Figure 5.

This definition can be understood as a generalization of the 3-Tiered Reward constraint. When the agent resides within tier $i \in \{2, 3, ..., k-2\}$, the k tiers could be partitioned into 3 groups to construct a 3-Tier MDP. In particular, S_1 will include tiers 1 through $i-1$, S_2 is just tier i, and S_3 is tiers $i+1$ to k. Note that we can generalize Theorem 5.3 to allow states in S_1 and S_3 to have any reward values as long as they satisfy the inequality in Definition 5.2 for a fixed reward value in S_2. Namely, denote $r_{low} = \max\{r_1, ..., r_{i-1}\}$ and $r_{high} = \min\{r_{i+1}, ..., r_k\}$, and as a k-Tiered Reward they satisfy

$$r_{low} < (\frac{1}{1-\gamma}) r_i < (\frac{1}{1-\gamma})^2 r_{high}$$

And since $\gamma \in (0, 1)$,

$$r_{low} < (\frac{1}{1-\gamma}) r_i < (\frac{1}{1-\gamma})^2 r_{high} \leq r_{high}. \quad (1)$$

That is, (r_{low}, r_i, r_{high}) is a Tiered Reward function in the 3-Tier MDP with tiers S_1, S_2, and S_3, and therefore induces Pareto-optimal policies (Theorem 5.3). So, at tier i, the policy that optimizes the k-Tiered Reward will push agents to higher tiers as fast as possible and avoid lower tiers, as if they were goals and obstacles, respectively. In the special case that the agent resides within tier $i = 1$, the constraint from Definition 6.1 will treat tiers 2 through k as if they are all goals, pushing the agent towards them. In the case that $i = k$, the agent is already in the "goal tier". So overall, k-Tiered Reward will induce in a ratchet-like policy—go to the higher tiers as fast as possible while not falling back to the lower tiers—that makes learning fast. In fact, it has been shown that a similar increasing-reward profile leads to fast learning (Sowerby et al., 2022). Okudo & Yamada (2021) and Zhai et al. (2022) have also shown that intermediate rewards can accelerate learning and provably improve sample efficiency in goal–reaching tasks.

Besides encouraging early visitation of good tiers, using Tiered Reward also guarantees maximum total visitation of all good tiers. This property is formalized in Theorem 6.2.

Theorem 6.2 (Tiered Reward and Cumulative Tier Visitation)**.** *In a k-Tier Markov Decision Process that has Tiered Reward $R(s)$, the induced optimal policy is π^*. Let $p_t^{*d} \in [0, 1]$ be the probability of being in tier $d \in \{1, 2, ..., k\}$ for the first time at timestep t following policy π^*. Then, there is no policy π, along with its induced probability distribution p_t^d, that satisfies both:*

$$\sum_{i=0}^{t} p_i^1 \leq \sum_{i=0}^{t} p_i^{*1}, \quad \forall t = 0, 1, 2, ..., \infty, \text{ and}$$

$$\sum_{i=0}^{t} p_i^d \geq \sum_{i=0}^{t} p_i^{*d}, \quad \forall d = [2..k], \forall t = 0, 1, 2, ..., \infty.$$

The proof is similar to that of Theorem 5.3 and can be found in Appendix B. To state the theorem in words, if a k-Tier MDP has a Tiered Reward structure, then the resulting policy will visit the worst tier (S_1) for as few times as possible, while visiting all the other good tiers (S_2, ..., S_k) as often as possible, respectively.

7. Conclusion

To resolve the policy-preference problem, we present a strict partial ordering over the policy space using Pareto-optimality. Then, in contrast to standard reward-design solutions that are environment-dependent, we presented Tiered Rewards—a class of environment-independent reward functions that provably leads to (Pareto) optimal behavior. One interesting direction for future work is theoretical guarantees on Tiered Reward leading to asymptotically faster learning.

References

Abel, D., Dabney, W., Harutyunyan, A., Ho, M. K., Littman, M., Precup, D., and Singh, S. On the expressivity of markov reward. *Advances in Neural Information Processing Systems*, 34:7799–7812, 2021.

Amodei, D. and Clark, J. Faulty reward functions in the wild. *URL: https://blog. openai. com/faulty-reward-functions*, 2016.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Brown, D., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.

Camacho, A., Chen, O., Sanner, S., and McIlraith, S. A. Non-markovian rewards expressed in ltl: guiding search via reward shaping. In *Tenth annual symposium on combinatorial search*, 2017.

Camacho, A., Icarte, R. T., Klassen, T. Q., Valenzano, R. A., and McIlraith, S. A. Ltl and beyond: Formal languages for reward function specification in reinforcement learning. In *IJCAI*, volume 19, pp. 6065–6073, 2019.

Devidze, R., Radanovic, G., Kamalaruban, P., and Singla, A. Explicable reward design for reinforcement learning agents. *Advances in Neural Information Processing Systems*, 34:20118–20131, 2021.

Icarte, R. T., Klassen, T., Valenzano, R., and McIlraith, S. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *International Conference on Machine Learning*, pp. 2107–2116. PMLR, 2018.

Icarte, R. T., Klassen, T. Q., Valenzano, R., and McIlraith, S. A. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

Koenig, S. and Simmons, R. G. Complexity analysis of real-time reinforcement learning. In *AAAI*, volume 93, pp. 99–105, 1993.

Li, X., Vasile, C.-I., and Belta, C. Reinforcement learning with temporal logic rewards. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3834–3839. IEEE, 2017.

Littman, M. L. Programming agents via rewards, 2015. URL https://www.youtube.com/watch?v=mN7qIIu7uz8.

Littman, M. L., Topcu, U., Fu, J., Isbell, C., Wen, M., and MacGlashan, J. Environment-independent task specifications via gltl. *arXiv preprint arXiv:1704.04341*, 2017.

Mornati, F. Pareto optimality in the work of pareto. *Revue européenne des sciences sociales. European Journal of Social Sciences*, pp. 65–82, 2013.

Okudo, T. and Yamada, S. Subgoal-based reward shaping to improve efficiency in reinforcement learning. *IEEE Access*, 9:97557–97568, 2021.

Russell, S. J. and Norvig, P. Artificial intelligence (a modern approach), 2010.

Sowerby, H., Zhou, Z., and Littman, M. L. Designing rewards for fast learning. *5th Multidisciplinary Conference on Reinforcement Learning and Decision Making*, 2022.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

Toro Icarte, R., Klassen, T. Q., Valenzano, R., and McIlraith, S. A. Teaching multiple tasks to an rl agent using ltl. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 452–461, 2018.

Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., and Dekker, E. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 84(1):51–80, 2011.

Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J., et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136): 1–46, 2017.

Wongpiromsarn, T., Topcu, U., and Murray, R. M. Receding horizon control for temporal logic specifications. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pp. 101–110, 2010.

Zhai, Y., Baek, C., Zhou, Z., Jiao, J., and Ma, Y. Computational benefits of intermediate rewards for goal-reaching policy learning. *Journal of Artificial Intelligence Research*, 73:847–896, 2022.

A. Details in Proof of Theorem 5.3

Proof that f_t^g is strictly decreasing:

$$\begin{aligned} f_{t+1}^g - f_t^g &= \gamma^{t+1} r_{goal} + \sum_{j=0}^{t} \gamma^j r_{back} - \gamma^t r_{goal} - \sum_{j=0}^{t-1} \gamma^j r_{back} \\ &= \gamma^t(\gamma - 1) r_{goal} + \gamma^t r_{back} \\ &= \gamma^t(1-\gamma)(\frac{1}{1-\gamma} r_{back} - r_{goal}) \\ &< 0 \end{aligned}$$

because $0 < \gamma < 1$ and $\frac{1}{1-\gamma} < r_{goal}$.

Proof that f_t^o is strictly increasing:

$$\begin{aligned} f_{t+1}^o - f_t^o &= \gamma^{t+1} r_{obs} + \sum_{j=0}^{t} \gamma^j r_{back} - \gamma^t r_{obs} - \sum_{j=0}^{t-1} \gamma^j r_{back} \\ &= \gamma^t(\gamma - 1) r_{obs} + \gamma^t r_{back} \\ &= \gamma^t(1-\gamma)(\frac{1}{1-\gamma} r_{back} - r_{obs}) \\ &> 0 \end{aligned}$$

because $0 < \gamma < 1$ and $r_{obs} < \frac{1}{1-\gamma}$.

The pass from the first equality to the second (*) is justified as follows:

$$\begin{aligned} \sum_{t=0}^{\infty}(g_t - g_t^*) f_t^g &= \sum_{t=0}^{\infty}\sum_{j=0}^{t}(g_j - g_j^*) f_t^g - \sum_{t=0}^{\infty}\sum_{j=0}^{t-1}(g_j - g_j^*) f_t^g \\ &= \sum_{t=0}^{\infty}\sum_{j=0}^{t}(g_j - g_j^*) f_t^g - \sum_{t=1}^{\infty}\sum_{j=0}^{t-1}(g_j - g_j^*) f_t^g \\ &= \sum_{t=0}^{\infty}\sum_{j=0}^{t}(g_j - g_j^*) f_t^g - \sum_{t'=0}^{\infty}\sum_{j=0}^{t'}(g_j - g_j^*) f_{t'+1}^g \\ &= \sum_{t=0}^{\infty}(\sum_{j=0}^{t} g_j - g_j^*)(f_t^g - f_{t+1}^g) \end{aligned}$$

Similarly,

$$\sum_{t=0}^{\infty}(o_t - o_t^*) f_t^o = \sum_{t=0}^{\infty}(\sum_{j=0}^{t} o_j - o_j^*)(f_t^o - f_{t+1}^o)$$

B. Proof of Theorem 6.2

Proof. The proof is similar to that of Theorem 5.3. Suppose, for the sake of contradiction, that there exists some such policy π. We can express the value functions as

$$V = \sum_{t=0}^{\infty} \gamma^t \sum_{m=1}^{k} r_m \cdot p_t^m, \text{ and}$$

$$V^* = \sum_{t=0}^{\infty} \gamma^t \sum_{m=1}^{k} r_m \cdot p_t^{*m}.$$

Denote $f_t^m = \gamma^t r_m$. Then, $f_t^m - f_{t+1}^m = r_m \gamma^t (1-\gamma) \leq 0, \forall m$. It's easy to see $f_t^m - f_{t+1}^m$ is strictly increasing in m, so

$$\begin{aligned}
V - V^* &= \sum_{t=0}^{\infty} \gamma^t \sum_{m=1}^{k} r_m (p_t^m - p_t^{*m}) \\
&= \sum_{m=1}^{k} \sum_{t=0}^{\infty} f_t^m (p_t^m - p_t^{*m}) \\
&= \sum_{m=1}^{k} \sum_{t=0}^{\infty} (f_t^m - f_{t+1}^m)(\sum_{j=0}^{t} p_j^m - p_j^{*m}) \\
&> \sum_{m=1}^{k} \sum_{t=0}^{\infty} (f_t^1 - f_{t+1}^1)(\sum_{j=0}^{t} p_j^m - p_j^{*m}) \quad (**) \\
&= \sum_{t=0}^{\infty} (f_t^1 - f_{t+1}^1) \sum_{m=1}^{k} (\sum_{j=0}^{t} p_j^m - p_j^{*m}) \\
&= \sum_{t=0}^{\infty} (f_t^1 - f_{t+1}^1) \cdot \sum_{j=0}^{t} (\sum_{m=1}^{k} p_j^m - \sum_{m=1}^{k} p_j^{*m}) \\
&= \sum_{t=0}^{\infty} (f_t^1 - f_{t+1}^1) \cdot \sum_{j=0}^{t} (1 - 1) \\
&= 0
\end{aligned}$$

Note that the $(**)$ step is justified only because $\sum_{j=0}^{t} p_j^m - p_j^{*m} \geq 0, \forall m = [2..k], \forall t$. The inequalities show that π achieves higher reward than the optimal policy, which is a contradiction. No such π exists. $\square$