Divide and Coordinate: A Multi-Policy Framework for Multi-Objective Reinforcement Learning

Anonymous authors
Paper under double-blind review

Keywords: Multi-Objective RL, Multi-Agent RL

Summary

The summary appears on the cover page. Although it can be identical to the abstract, it does not have to be. One might choose to omit the stated contributions in the Summary, given that they will be stated in the box below. The original abstract may also be extended to two paragraphs. The authors should ensure that the contents of the cover page fit entirely on a single page. The cover page does **not** count towards the 8–12 page limit.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contribution(s)

1. Provide a succinct but precise list of the contribution(s) of the paper. Use contextual notes to avoid implications of contributions more significant than intended and to clarify and situate the contribution relative to prior work (see the examples below). If there is no additional context, enter “None”. Try to keep each contribution to a single sentence, although multiple sentences are allowed when necessary. If using complete sentences, include punctuation. If using a single sentence fragment, you may omit the concluding period. A single contribution can be sufficient, and there is no limit on the number of contributions. Submissions will be judged mostly on the contributions claimed on their cover pages and the evidence provided to support them. Major contributions should not be claimed in the main text if they do not appear on the cover page. Overclaiming can lead to a submission being rejected, so it is important to have well-scoped contribution statements on the cover page.
   Context: None
2. The submission template for submissions to RLJ/RLC 2025
   Context: Built from previous RLC/RLJ, ICLR, and TMLR submission templates
3. *[Example of one contribution and corresponding contextual note for the paper “Policy gradient methods for reinforcement learning with function approximation” (**?**).]*
   This paper presents an expression for the policy gradient when using function approximation to represent the action-value function.
   Context: Prior work established expressions for the policy gradient without function approximation (**?**).

Divide and Coordinate: A Multi-Policy Framework for Multi-Objective Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

1 Introduction

TODO: I am putting this here, it will go at the end of the introduction.

Prior works proposed a composition technique based on Q-learning. Each local policy π^i for each individual reward function R^i would be designed using a Q-learning agent that disregards all reward functions other than its own. Along the Q-function, it also learns a W-function which maps every state to a numeric importance score (Humphrys, 1995). Intuitively, if $W(s)$ is high, then it is highly important for the local policy to be able to execute its action in the state s. The composition of policies happens at runtime, when at each state s, if $W^i(s)$ is the W-value of the i-th local policy, for $i \in [1; m]$, and if $i^* = \arg\max_i W^i(s)$, then we select the action proposed by the policy π^{i^*} at the current state s. It has been demonstrated that, interestingly, W-learning generates selfish local policies that end up cooperating in practice. Subsequently, this framework has been extended to deep learning and applied to realistic applications (Rosero et al., 2024).

A limitation of W-learning is that it assumes that all local policies will be honest while broadcasting their W-values: if any of the policies is dishonest, i.e., emits a higher W-value than the actual, then it will get undue advantages in executing its actions, potentially compromising the global performance. To put it in game theoretic terminologies, the local policies are not "strategyproof." This could be a serious issue if, e.g., the local policies are obtained through different third-party vendors.

2 Preliminaries: Multi-Objective MDPs

Mainstream RL algorithms consider Markov decision processes (MDP) equipped with a *single* reward function, pertaining to a single task or *objective* for the system. In reality, a majority of real-world applications of RL requires satisfying multiple, partly contradictory objectives. We model such multi-objective decision-making problems using multi-objective MDPs (MO-MDP), as formally defined below. Intuitively, an MO-MDP has the exact same syntax as a regular MDP, except that it now has multiple reward functions pertaining to the different objectives. We formalize MO-MDP below. We will use the notation $\mathbb{D}(\Sigma)$ to represent the set of all probability distributions over a given alphabet Σ.

Definition 1 (MO-MDP). *A multi-objective Markov decision process (MO-MDP) with $m \in \mathbb{Z}_{>0}$ objectives is specified by a tuple $\mathcal{M} = (S, A, T, R, \mu_0)$, where*

- *S is the set of states,*
- *A is the set of actions,*
- *$T : S \times A \to \mathbb{D}(S)$ is the transition function mapping a state-action pair to a distribution over the successor states,*

- $\{R^i : S \times A \times S \to \mathbb{R}_{\geq 0}\}_{i\in[1;m]}$ *is the set of reward functions, and*
- $\mu_0 \in \mathbb{D}(S)$ *is the initial state distribution.*

The notions of policies and paths induced by them are exactly the same as in classical MDPs, which we briefly recall below. First, we introduce some notation. Given an alphabet Σ, we will write Σ^* and Σ^ω to denote the set of every finite and infinite word over Σ, respectively, and will write $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. Given a word $w = \sigma_0\sigma_1 \ldots \in \Sigma^\infty$, and given a $t \geq 0$ that is not larger than the length of w, we will write w_t and $w_{0:t}$ to denote respectively the t-th element of w, i.e., $w_t = \sigma_t$, and the prefix of w up to the t-th element, i.e., $w_{0:t} = \sigma_0 \ldots \sigma_t$.

A *policy* in an MO-MDP $\mathcal{M}$ is a function $\pi : (S \times A)^* \times S \to \mathbb{D}(A)$ that maps a history of state-action pairs and the current state to a distribution over actions. A *path* on $\mathcal{M}$ induced by π is a sequence $\rho = (s_0, a_0)(s_1, a_1), \ldots \in (S \times A)^\infty$ such that for every $t \geq 0$, (1) the probability that the action a_{t+1} is picked by π based on the history is positive, i.e., $\pi(\rho_{0:t}, s_{t+1})(a_{t+1}) > 0$, and (2) the probability of moving to the state s_{t+1} from s_t due to action a_t is positive, i.e., $T(s_t, a_t)(s_{t+1}) > 0$. A path can be either finite or infinite, and we will write $\mathit{Paths}(\mathcal{M}, \pi)$ to denote the set of all infinite paths fo $\mathcal{M}$ induced by π. Given a finite path $\rho = (s_0, a_0) \ldots (s_t, a_t)$, the probability that ρ occurs is given by: $\mu_0(s_0) \cdot \prod_{k=0}^{t-1} T(s_k, a_k)(s_{k+1}) \cdot \pi(\rho_{0:k}, s_{k+1})(a_{k+1})$. This can be extended to a probability measure over the set of all infinite paths in $\mathcal{M}$ using standard constructions, which can be found in the literature (Baier & Katoen, 2008). Given a measurable set of paths Ω and a function $f \colon \mathit{Paths}(\mathcal{M}, \pi) \to \mathbb{R}$, we will write $\mathbb{P}^{\mathcal{M},\pi}[\Omega]$ and $\mathbb{E}^{\mathcal{M},\pi}[f]$ to denote, respectively, the probability measure of Ω and the expected value of f evaluated over random infinite paths.

We will use the standard discounted reward objectives, where we fix $\gamma \in [0, 1]$ as a given discounting factor. Let $\rho = (s_0, a_0)(s_1, a_1), \ldots \in \mathit{Paths}(\mathcal{M}, \pi)$ be an infinite path induced by π. Define the discounted sum function, mapping ρ to the discounted sum of the associated rewards: $f^i_{\mathrm{ds}}(\rho) := \sum_{t=0}^{\infty} \gamma^t \cdot R^i(s_t, a_t)$. The *$i$-value* of the policy ρ for $\mathcal{M}$ is the expected value of the discounted sum of the i-th reward we can secure by executing ρ on $\mathcal{M}$, written as $\mathit{val}^{\mathcal{M},i}(\pi) = \mathbb{E}^{\mathcal{M},\pi}[f^i_{\mathrm{ds}}]$. When the reward index i is unimportant, we will refer to every element of the set $\{\mathit{val}^{\mathcal{M},i}\}_{i\in[1;m]}$ as a *value component*.

When the MO-MDP $\mathcal{M}$ is clear from the context, we will drop it from all notation and will simply write $\mathit{Paths}(\pi)$, $\mathbb{P}^\pi$, $\mathbb{E}^\pi$, and val^i.

It is known that *memoryless* (aka, stationary) policies suffice for maximizing single discounted reward objectives, where a policy π is called memoryless if the proposed action only depend on the current state. In other words, given every pair of finite paths ρ, ρ' both ending at the same state, the probability distributions $\pi(\rho)$ and $\pi(\rho')$ are identical.

Unlike classical single-objective MDPs, the optimal policy synthesis problem for MO-MDP requires fixing one of many possible optimality criteria. Many possibilities exist, including pareto optimality, requiring a solution where none of the value components could be unanimously improved without hurting the others; weighted social welfare, requiring a weighted sum of the value components be maximized; and fairness, requiring the minimum attained value by any value component is maximized. **TODO:** Give some citations for each category.

3 Auction-Based Compositional RL on Multi-Objective MDPs

We consider the compositional approach to policy synthesis for MO-MDPs, where we will design a selfish, *local* policy maximizing each individual value component, and the composition of all local policies gives rise to some globally optimal solution. The main crux is in the composition process, where each local policy may propose a different action, but the composition must decide one of the actions that will be actually executed. Importantly, the composition must be implementable in a distributed manner, meaning we will *not* use any global policy that would pick an action by analyzing all local policies and their reward functions. **TODO:** running example

3.1 The Framework

We present a novel *auction*-based RL framework for compositional policy synthesis for MO-MDPs. In our framework, not only do the local policies emit actions, but also they *bid* for the privilege of executing their actions for a given number of time steps $\tau \in \mathbb{N}_{>0}$ in future. The bids are all nonnegative real numbers, and the highest bidder's actions get executed for the subsequent τ steps, with ties being resolved uniformly at random. The policy whose actions are executed is referred to as the *active* policy while the rest are called *idle* policies. To discourage overbidding, we require the active policy to pay a one-time price—modeled as a negative reward or a *penalty*—equal to its bid value. This way, it is against the interest of a policy to bid more than the total reward it would earn if it is active in the next τ steps; otherwise, the net earning would be negative, while bidding the zero amount would secure non-negative earnings. Through bidding, each policy can communicate the importance for it to execute its actions, and the composition mechanism guarantees that the most important policy is executed.

The parameter τ controls how frequently the agent changes its policies. In practice, if τ is too small, the switching could be too frequent for any of the objectives to be fulfilled. For example, **TODO:** running example...

We introduce three different variations of the compositional framework, based on three kinds of reward systems for the idle policies:

1. Each idle policy earns a reward of the same amount as its bid, in addition to the default reward from the transitions in the MO-MDP.
2. Idle policies only get the default reward specified by the original MO-MDP, i.e., they neither earn a reward nor pay a penalty associated to the bidding.
3. Just like the active policy, each idle policy pays a penalty of the same amount as its bid.

Each of the three variations have their own benefits and pitfalls, which we will describe subsequently in Section 3.3.

3.2 Learning Local Policies

We show that computing each individual local policy in our framework reduces to finding the optimal policy in a stochastic game, for which we could use any standard off-the-shelf learning framework. In particular, given an MO-MDP and given the objective $i \in [1; m]$, we construct a stochastic game that captures all possible interactions of policy i against the other policies.

Suppose we are given the MO-MDP $\mathcal{M} = (S, A, T, R, \mu_0)$.

3.3 A Comparative Study of the Three Variations

4 A Multi-Agent Bidding Approach for Multi-Objective RL

Definition 2 (MO-MDP)**.** *A multi-objective Markov decision process (MO-MDP) with $m \in \mathbb{Z}_{>0}$ objectives is specified by a tuple $\mathcal{M} = (S, A, T, R, \mu_0)$, where*

- *S is the set of states,*
- *A is the set of actions,*
- *$T : S \times A \to \mathbb{D}(S)$ is the transition function mapping a state-action pair to a distribution over states,*
- *$R : S \times A \times S \to \mathbb{R}^m$ is the reward function with each output component corresponding to the different objectives, and*
- *$\mu_0 \in \mathbb{D}(S)$ is the initial state distribution.*

A *policy* in an MO-MDP $\mathcal{M}$ is a function $\pi : (S \times A)^* \times S \to \mathbb{D}(A)$ that maps a history of state-action pairs and the current state to a distribution over actions.

Definition 3 (MAB-MDP)**.** *Let $\mathcal{M} = (S, A, T, R, \mu_0)$ be an MO-MDP with m objectives and let $b \in \mathbb{Z}_{>0}$ be the bid upper bound. Also, define $M = \{1, \ldots, m\}$ be indices of the m agents corresponding to the m objectives along with $\perp$ representing a null agent. Lastly, let $B = \{0, \ldots, b\}$ be the range of bids and $\rho > 0$ be the bid penalty factor. We define the multi-agent bidding Markov decision process (MAB-MDP) as a tuple $\mathcal{B}_{\mathcal{M}} = (\hat{S}, \hat{A}, \hat{T}, P, \hat{R}, \hat{\mu}_0)$ where*

- *$\hat{S} = M \times S$ is the new state space augmented with the index of the agent that won the previous round of bidding,*
- *$\hat{A} = A^m \times B^m$ represents the action space of the m agents in which each agent selects an action from A and a bid from B,*
- *$\hat{T} : \hat{S} \times \hat{A} \to \mathbb{D}(\hat{S})$ is the new transition function defined as,*

$$\hat{T}((\_, s), (\mathbf{a}, \mathbf{b})) \coloneqq \frac{1}{|B_{\max}|} \sum_{i \in B_{\max}} (T(s, a_i), i)$$

  where $B_{\max} \coloneqq \{i \mid b_i = \max\{b_1, \ldots, b_m\}\}$ is the set of agents with maximal bids. The tuple $(T(s, a_i), i)$ represents the distribution over $\hat{S}$ induced by the original transition function T such that the second component is fixed, and the weighted sum represents taking the weighted sums of the distributions over $\hat{S}$.
- *$P : \hat{A} \times M \to \mathbb{R}^m$ is the bidding penalty for the m agents and the second component is the index of the agent that won the bidding.*
- *$\hat{R} : \hat{S} \times \hat{A} \times \hat{S} \to \mathbb{R}^m$ is the reward function for the m agents with*

$$\hat{R}_k((\_, s_0), (\mathbf{a}, \mathbf{b}), (i, s)) \coloneqq R_k(s_0, a_i, s) - P_k((\mathbf{a}, \mathbf{b}), i)$$

  where $i \in M$ is the index of the agent that won the bid and chose the action.
- *$\hat{\mu}_0 \coloneqq (\mu_0, 1)$ is the initial state distribution over $\hat{S}$ induced by μ_0 and the second component is fixed to be 1 without loss of generality.*

Given an MAB-MDP $\mathcal{B}_{\mathcal{M}}$, a *policy* for each agent indexed by $i \in \{1, \ldots, m\}$ takes a similar form: $\pi_i : (\hat{S} \times \hat{A})^* \times \hat{S} \to \hat{A}$. Intuitively, a state $(i, s) \in \hat{S}$ encodes the agent that won the bidding and chose the action to reach s in the previous step. At each step, each of the agents choose an action and a bid, and an action amongst the set of highest bidders is chosen uniformly at random. The reward function includes a penalty term that captures the desired bidding mechanism.

References

Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

Mark Humphrys. W-learning: Competition among selfish q-learners. 1995.

Juan C Rosero, Nicolás Cardozo, and Ivana Dusparic. Multi-objective deep reinforcement learning optimisation in autonomous systems. In *2024 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, pp. 97–102. IEEE, 2024.