Bidding-Based Policy Composition for Dynamic Task Streams

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 (**?**).

Bidding-Based Policy Composition for Dynamic Task Streams

Anonymous authors
Paper under double-blind review

Abstract

We study multi-objective reinforcement learning settings in which objectives appear or disappear at runtime. We propose a modular framework where each objective is supported by a selfish local policy, and coordination is achieved through a novel *auction*-based mechanism: policies bid for the right to execute their actions, with bids reflecting the urgency of the current state. The highest bidder selects the action, enabling a dynamic and interpretable trade-off among objectives. To succeed, each policy must not only optimize its own objective, but also reason about the presence of other goals and learn to produce calibrated bids that reflect relative priority. When objectives change, the system adapts by simply adding or removing the corresponding policies. Moreover, when objectives arise from the same parameterized family—like the class of reachability objectives parameterized by target states—identical copies of a parameterized policy can be deployed. In our implementation, the policies are trained concurrently using proximal policy optimization (PPO). We evaluate on Atari Assault and a gridworld-based path-planning task with dynamic targets. Our method achieves substantially better performance and reduced sample complexity than a single policy trained with PPO.

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.

1.1 Related work

A large body of work in multi-objective reinforcement learning (MORL) relies on *scalarization*, aggregating multiple reward functions into a single scalar objective so that standard single-objective RL algorithms can be applied. The simplest scalarization method is a weighted sum of individual rewards (Gass & Saaty, 1955), though richer nonlinear scalarization functions have also been proposed (Van Moffaert et al., 2013). A key limitation of scalarization is that the relative importance induced by the aggregation function may not align with the designer's true intent. This mismatch can initiate a tedious debugging cycle, particularly in large-scale systems (Hayes et al., 2022). In contrast, our approach achieves a trade-off between reward components without collapsing them into a fixed scalar objective.

Other works pursue trade-offs by fixing a specific optimality criterion. Common choices include Pareto optimality (Van Moffaert & Nowé, 2014) and its approximations (Pirotta et al., 2015), as well as fairness-based criteria across reward functions (Park et al., 2024; Byeon et al., 2025; Siddique et al., 2020). These approaches typically learn a single monolithic policy that satisfies the chosen criterion. By contrast, our objective is to learn independent, selfish local policies for each reward component and compose them at runtime in a principled manner, thereby preserving modularity while still achieving a coherent global trade-off.

Relatively few works study distributed local policies for multiple rewards. A notable example is W-learning (Humphrys, 1995) and its deep RL extension (Rosero et al., 2024), where separate selfish policies are trained alongside meta-policies (W-functions) that assign each state a score reflecting its urgency. At runtime, the policy with the highest score is selected. Other approaches employ alternative aggregation mechanisms, such as ranked voting over actions (Méndez-Hernández et al., 2019), or fixed aggregation rules like summing action values across agents (Russell & Zimdars, 2003). While conceptually related, our approach is technically simpler: it relies on an engineered reward structure that enables the use of standard learning algorithms (e.g., PPO) without additional meta-policies or complex aggregation schemes. Furthermore, to the best of our knowledge, we are the first to introduce the incremental MORL setting, in which reward components can be added or removed at runtime.

The idea of bidding-based selfish policies originates from analogous techniques for multi-objective path planning problems on finite graphs (Avni et al., 2024), as well as from the broader literature on bidding games (Lazarus et al., 1999; Avni et al., 2019; 2025). These works study strategic interaction in finite arenas, where adversarial players bid for the right to determine the next move from a shared action space in pursuit of their objectives. Although these works provide strong theoretical guarantees, they do not naturally extend to infinite arenas. Moreover, players in such games are typically budget-constrained, and the central question concerns the minimum budget required to win. In contrast, we consider infinite arenas and eliminate explicit budget constraints by incorporating bidding rewards and penalties directly into the learning framework.

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 $\mathcal{M} = (S, A, T, \mathbf{R}, \mu_0)$, 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,
- $\mathbf{R} = \{R^i : S \times A \times S \to \mathbb{R}_{\geq 0}\}_{i\in[1;m]}$ 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 $\mathit{val}^{\mathcal{M},i}(\pi) = \mathbb{E}^{\mathcal{M},\pi}[f^i_{\mathrm{ds}}]$. The *optimal* policy for R^i for a given $i \in [1; m]$ is the policy that maximizes the i-value. 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, towards the fulfillment of some required global coordination requirements. 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 non-negative real numbers, and the highest bidder's actions get executed for the following τ consecutive steps, with bidding ties being resolved uniformly at random. The policy whose actions are executed is referred to as the *winning* policy, and it must pay a bidding *penalty* that equals to its bid amount; this is to discourage overbidding. The policies whose actions are not executed are called the *losing* policies, and we consider three different settings for the "payment" they must make:

Loser-Rewarded: the winning policy pays the bidding penalty and the losing policies earn bidding rewards equal to their respective bid values;

Winner-Pays: the winning policy pays the bidding penalty and the losing policies are unaffected (i.e., neither earn bidding rewards nor pay bidding penalties);

All-Pay: all policies pay bidding penalties equal to their respective bid values.

While penalizing the winner discourages overbidding, the situation with the losers is more subtle. In the Loser-Rewarded setting, by rewarding the losers, we encourage policies to bid positively if the current state has some importance to them; this way, if they lose the bidding, they will get some positive reward. In the All-Pay setting, by penalizing all policies, we discourage policies to bid at all unless it is absolutely important. The Winner-Pays setting balances these two: by neither rewarding nor penalizing the losers, we neither encourage nor discourage policies to bid. In Section 3.3, we will see how these three settings induce different kinds of coordination through bidding.

For each policy, the bidding penalty or reward gets, respectively, subtracted or added to the *nominal* reward obtained from the reward functions of the given MO-MDP, and the resulting reward is called the *net* reward.

In summary, through this novel bidding mechanism, each policy can adjust its bid in proportion to the importance for it to execute its action in the current state, and the associated bidding penalty/reward aims to incentivize policies to be truthful. By making the highest bidder active, it is effectively guaranteed that the most important policy is executed. This way, we obtain a purely decentralized scheme to coordinate local policies in a given MO-MDP.

Remark 1 (On the parameter τ). 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...

3.2 The Design Problem and Learning Algorithms

We consider the following learning task for our auction-based compositional framework:

Given an MO-MDP, a constant $\tau > 0$, and $\Delta \in \{$Loser-Rewarded, Winner-Pays, All-Pay$\}$, compute local policies that are optimal for the net rewards obtained in the mode Δ, given that all other local policies behave selfishly towards maximizing their own net rewards.

We will show how the above learning problem boils down to solving a standard learning problem in the multi-agent setting, formalized using a decentralized MDP (DEC-MDP) as defined below. The only difference between a DEC-MDP and an MO-MDP (see Definition 1) is that now each reward function R^i is owned by the Agent i, who now controls a separate set of actions A^i.

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

- S is the set of states,
- $\mathbf{A} = \{A^1, \ldots, A^m\}$ is a set with A^i being the set of Agent i's actions,
- $T : S \times A^1 \times \ldots \times A^m \to \mathbb{D}(S)$ is the transition function mapping a state-action pair to a distribution over the successor states,
- $\mathbf{R} = \{R^i \colon S \times A^1 \times \ldots \times A^m \times S \to \mathbb{R}_{\geq 0}\}_{i \in [1;m]}$ is the set of reward functions, and
- $\mu_0 \in \mathbb{D}(S)$ is the initial state distribution.

The definitions of policies and paths readily extend from MO-MDP to DEC-MDP.

Given a DEC-MDP, the goal is to compute an ensemble of local (memoryless) policies for all individual agents, such that for every $i \in [1;m]$, the i-value cannot be increased by a unanimous change of the local policy π^i. In other words, the goal is to find a set of selfish local policies that are in a Nash equilibrium. This is an extensively studied problem in the literature. **TODO:** Do a little bit of literature survey...

Our focus is not in improved algorithms for DEC-MDP, but rather to show how the local policy synthesis problem for the MO-MDP $\mathcal{M}$ in our auction-based framework reduces to the multi-agent policy synthesis problem in a DEC-MDP $\widetilde{\mathcal{M}}$. Intuitively, for every state s of $\mathcal{M}$, $\widetilde{\mathcal{M}}$ creates two kinds of copies, ones where bidding happens and are represented simply as s, and ones of the form (s, t, i^*) that keeps track of the time t elapsed since the last bidding, and the winner i^* of the last bidding. Furthermore, bidding is facilitated by extending the action space of $\mathcal{M}$ to include all real-valued bids, and each agent in $\widetilde{\mathcal{M}}$ has an identical copy of this extended action space. After bidding in a state s, the winner i^* is selected, and the state moves to $(s, 0, i^*)$. From this point onward, only Agent i^* selects actions $a^0, a^1, \ldots, a^\tau$ to produce the sequence $(s^1, 1, i^*), (s^2, 2, i^*), \ldots, (s^{\tau-1}, \tau - 1, i^*), s^\tau$, after which the next bidding happens, and the process repeats. Finally, the bidding penalties or bidding rewards are only paid during the transition $s \to (s, 0, i^*)$, otherwise, the rewards are inherited from the original MO-MDP.

We formalize this below. Given an MO-MDP $\mathcal{M} = (S, A, T, \mathbf{R}, \mu_0)$, a constant $\tau > 0$, and the mode $\Delta \in \{\textsf{Loser-Rewarded}, \textsf{Winner-Pays}, \textsf{All-Pay}\}$, we define the DEC-MDP $\widetilde{\mathcal{M}} = (\widetilde{S}, \widetilde{\mathbf{A}}, \widetilde{T}, \widetilde{\mathbf{R}}, \widetilde{\mu}_0)$ where

- $\widetilde{S} \coloneqq S \cup S \times [0; \tau - 1] \times [1; m]$,
- $\widetilde{\mathbf{A}} \coloneqq \{\widetilde{A}^i\}_{i \in [1;m]}$ where $\widetilde{A}^i \coloneqq A \cup \mathbb{R}$,
- $\widetilde{\mu}_0 \coloneqq \mu_0 \times \{0\}$,

and for every current state $s \in \widetilde{S}$ and every current action $(b^1, \ldots, b^m) \in \mathbb{R}^m$, writing the highest bidders as $I = \{i \in [1;m] \mid \forall j \in [1;m] \,.\, b^i \geq b^j\}$,

- $\widetilde{T}(s, b^1, \ldots, b^m) \coloneqq \text{Uniform}(\{(s, 0, i)\}_{i \in I})$,
- $\widetilde{R}^i(s, b^1, \ldots, b^m, (s, 0, i^*)) \coloneqq \begin{cases} -b^i & i = i^* \lor \Delta = \textsf{All-Pay}, \\ +b^i & i \neq i^* \land \Delta = \textsf{Loser-Rewarded}, \\ 0 & i \neq i^* \land \Delta = \textsf{Winner-Pays}, \end{cases}$

whereas if the current state is of the form $(s, t, i^*) \in \widetilde{S}$, for every action $(a^1, \ldots, a^m) \in A^m$,

- $\widetilde{T}((s, t, i^*), a^1, \ldots, a^m) \coloneqq \begin{cases} T(s, a^{i^*}) \times ((t+1) \mod \tau) \times \{i^*\} & t < \tau - 1, \\ T(s, a^{i^*}) & t = \tau - 1, \end{cases}$
- $\widetilde{R}^i((s, t, i^*), a^1, \ldots, a^m, (s', t+1, i^*)) \coloneqq R^i(s, a^i, s')$.

KM: A soundness theorem would be good, but what can we say concretely?

3.3 Flavors of Cooperation through Bidding

We provide theoretical insights into the global behavior that emerges out of the auction-based interactions between the local policies. For the sake of theoretical guarantees, and to be able to convey the main essence of our results, we choose the simplest bare bone setting:

Assumption 1. The given MO-MDP has finite state and action spaces, and for every (memoryless) policy, the bottom strongly connected component (BSCC) of the resulting Markov chain (MC) is a sink state where no reward is earned. Furthermore, the time parameter $\tau = 1$, meaning the bidding takes place at each time step before selecting the action.

Firstly, since the MO-MDP is finite, for each individual reward function, *deterministic* memoryless policy suffices. **TODO:** give some citation

The following two types of global behaviors are of particular interest:

Social welfare is the sum (equivalently, the average) of the i-values for all i. We may ask: is the emergent global behavior guaranteed to achieve the maximal social welfare?

Fairness is measured by the disparity between different i-values, i.e., $\max_{i,j\in[1;m]} |val^i - val^j|$. Fairness is maximized when the disparity is minimized. We may ask: is the emergent global behavior guaranteed to achieve the maximal fairness?

Theorem 1. *Suppose the MO-MDP is such that at each state s and for every action a, there exists at most a single $i \in [1;m]$ such that the optimal policy for R^i selects a at s. Then, the Loser-Rewarded setting maximizes the social welfare.*

Proof sketch. First, consider the simple one-shot game, where the agents bid just one time to select an action, and the reward is based on the resulting single probabilistic transition. Suppose for the index $i \in [1;m]$, the expected reward from using the action $a \in A$ is E^i_a, and define $E^i_+ := \max_{a\in A} E^i_a$ and $E^i_- := \min_{a\in A} E^i_a$.

We claim that the optimal bid b^i_* for policy i equals $(E^i_+ - E^i_-)/2$, and upon winning the bidding the optimal action is $a_+ = \arg\max_{a\in A} E^i_a$. Notice that no matter whether policy i becomes the winner or the loser, its net reward is at least $(E^i_+ + E^i_-)/2$: if it wins and chooses a_+, after paying the bidding penalty, the net reward is $E^i_+ - (E^i_+ - E^i_-)/2 = (E^i_+ + E^i_-)/2$; if it loses, no matter what action the opponent chooses, its nominal reward is at least E^i_-, and after the bidding reward, the net reward is $E^i_- + (E^i_+ - E^i_-)/2 = (E^i_+ + E^i_-)/2$. If policy i bids $b^i < b^i_*$, then upon losing, its net reward will be $E^i_- + b^i < E^i_- + b^i_* = (E^i_+ + E^i_-)/2$. If it bids $b^i > b^i_*$, then upon winning, its net reward will be $E^i_+ - b^i < E^i_+ - b^i_* = (E^i_+ + E^i_-)/2$. Therefore, the optimal bid is $b^i_* = (E^i_+ - E^i_-)/2$, which is what each selfish policy is expected to select.

Suppose, policy i is the winner. Then, for every $j \neq i$, $b^i_* \geq b^j_*$, i.e., $(E^i_+ - E^i_-)/2 \geq (E^j_+ - E^j_-)/2$. Simplifying, we get $E^i_+ + E^j_- \geq E^i_- + E^j_+$. It follows that $E^i_+ + \sum_{j\neq i} E^j \geq E^i_+ + \sum_{j\neq i} E^j_- \geq E^i_- + E^k_+ + \sum_{j\neq i,k} E^j$ for every for every $k \neq i$. Since the MO-MDP is purely competitive, there will be at least a single k such that a given action is optimal for k, and therefore the claim follows for the single-shot case.

Now, for the general multi-shot case, we inductively apply the above principle in the Bellman equation, which extends the claim to paths of arbitrary length. The convergence of the Bellman iteration is guaranteed because it is a contraction mapping (since $\gamma < 1$). **KM:** I am not sure about this extension. □

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

Definition 3 (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 4 (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.

5 Implementation and evaluation

5.1 Implementation

Talk about:

1. different bidding mechanisms
2. choice of penalty factor

Table 1: Performance (mean with 95% CI) averaged over the last 5 evaluation checkpoints.

Algorithm	**Gridworld (Min Targets Reached)**	**Assault (Score)**
All-Pay	6.05 [5.74, 6.36]	634.80 [591.14, 678.46]
Winner-Pays	4.07 [3.78, 4.36]	578.04 [521.02, 635.06]
Winner-Pays (Others Rewarded)	4.20 [3.92, 4.48]	662.60 [619.09, 706.11]
Single-Agent	2.31 [2.08, 2.54]	384.72 [343.09, 426.35]

3. action window (remarking that we could additionally allow agents to choose length of action window)
4. use with off-the-shelf RL algorithms

5.2 Environments

5.2.1 MovingTargetsGridworld

Important to mention that we want to maximize $\min(\text{targets reached})$.

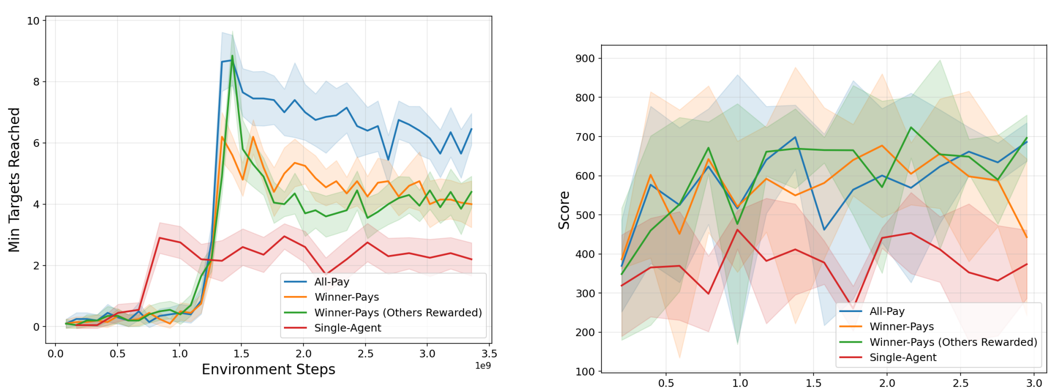
5.2.2 Atari Assault

5.3 Baselines

1. Weighted sum of rewards with standard RL algorithms
2. Deep W learning implemented on top of DQN

5.4 Performance comparison with baselines

Include plots of training steps vs performance of our algorithms vs baselines on both environments



(a) MovingTargetsGridworld. [Placeholder: describe convergence behavior, relative performance of mechanisms, and any notable differences in sample efficiency.]

(b) Atari Assault. [Placeholder: describe convergence behavior, relative performance of mechanisms, and any notable differences in sample efficiency.]

Figure 1: Learning curves for different bidding mechanisms across both environments.

5.5 Interpretability

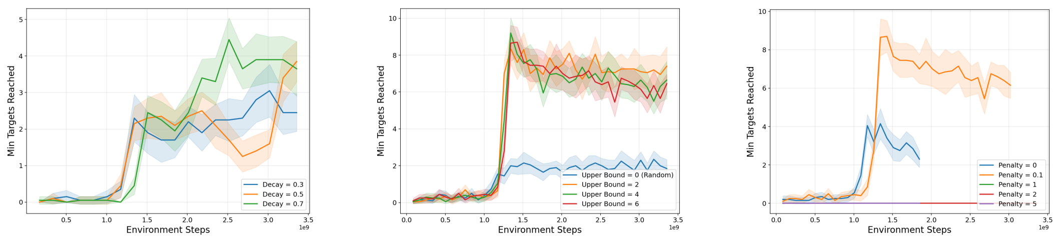
Include plots of distribution of control steps amongst agents, table of average, median, max, min of bids of agents

5.6 Modularity

Plots of performance in gridworld with increasing number of objectives

5.7 Ablations

Impact of max bid, penalty factor



(a) Reward decay. [Placeholder description.]

(b) Bid upper bound. [Placeholder description.]

(c) Bid penalty. [Placeholder description.]

Figure 2: Ablation studies on the MovingTargetsGridworld environment.

References

Guy Avni, Thomas A Henzinger, and Ventsislav Chonev. Infinite-duration bidding games. *Journal of the ACM (JACM)*, 66(4):1–29, 2019.

Guy Avni, Kaushik Mallik, and Suman Sadhukhan. Auction-based scheduling. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 153–172. Springer, 2024.

Guy Avni, Martin Kurečka, Kaushik Mallik, Petr Novotnỳ, and Suman Sadhukhan. Bidding games on markov decision processes with quantitative reachability objectives. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 161–169, 2025.

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

Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.

Woohyeon Byeon, Giseung Park, Jongseong Chae, Amir Leshem, and Youngchul Sung. Multi-objective reinforcement learning with max-min criterion: A game-theoretic approach. *arXiv preprint arXiv:2510.20235*, 2025.

Saul Gass and Thomas Saaty. The computational algorithm for the parametric objective function. *Naval research logistics quarterly*, 2(1-2):39–45, 1955.

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning: Cf hayes et al. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

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

Andrew J Lazarus, Daniel E Loeb, James G Propp, Walter R Stromquist, and Daniel H Ullman. Combinatorial games under auction play. *Games and Economic Behavior*, 27(2):229–264, 1999.

Beatriz M Méndez-Hernández, Erick D Rodríguez-Bazan, Yailen Martinez-Jimenez, Pieter Libin, and Ann Nowé. A multi-objective reinforcement learning algorithm for jssp. In *International Conference on Artificial Neural Networks*, pp. 567–584. Springer, 2019.

Thanh Thi Nguyen, Ngoc Duy Nguyen, Peter Vamplew, Saeid Nahavandi, Richard Dazeley, and Chee Peng Lim. A multi-objective deep reinforcement learning framework. *Engineering Applications of Artificial Intelligence*, 96:103915, 2020.

Giseung Park, Woohyeon Byeon, Seongmin Kim, Elad Havakuk, Amir Leshem, and Youngchul Sung. The max-min formulation of multi-objective reinforcement learning: From theory to a model-free algorithm. *arXiv preprint arXiv:2406.07826*, 2024.

Matteo Pirotta, Simone Parisi, and Marcello Restelli. Multi-objective reinforcement learning with continuous pareto frontier approximation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

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.

Stuart J Russell and Andrew Zimdars. Q-decomposition for reinforcement learning agents. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 656–663, 2003.

Umer Siddique, Paul Weng, and Matthieu Zimmer. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *International Conference on Machine Learning*, pp. 8905–8915. PMLR, 2020.

Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

Kristof Van Moffaert, Madalina M Drugan, and Ann Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pp. 191–199. IEEE, 2013.

Kristof Van Moffaert, Tim Brys, Arjun Chandra, Lukas Esterle, Peter R Lewis, and Ann Nowé. A novel adaptive weight selection algorithm for multi-objective multi-agent reinforcement learning. In *2014 International joint conference on neural networks (IJCNN)*, pp. 2306–2314. IEEE, 2014.

Harm Van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid reward architecture for reinforcement learning. *Advances in neural information processing systems*, 30, 2017.