An Online Multiobjective Policy Gradient for Long-run Average-reward Markov Decision Process

Rahul Misra*, Manuela L. Bujorianu, and Rafał Wisniewski

***Abstract*— We propose a reinforcement learning (RL) framework for multi-objective decision-making, where the agent seeks to optimize a vector of rewards rather than a single scalar value. The objective is to ensure that the time-averaged reward vector converges asymptotically to a predefined target set. Since standard RL algorithms operate on scalar rewards, we introduce a dynamic scalarization mechanism guided by Blackwell's Approachability Theorem. This theorem enables adaptive updates of the scalarization vector to guarantee convergence toward the target set. Assuming ergodicity, the Markov chain induced by the learned policies admits a stationary distribution, ensuring all states recur with finite return times. Our algorithm exploits this property by defining an inner loop that applies a policy gradient method (with baseline) between successive visits to a designated recurrent state, enforcing Blackwell's condition at each iteration. An outer loop then updates the scalarization vector after each recurrence. We establish theoretical convergence of the long-run average reward vector to the target set and validate the approach through a numerical example.**

I. Introduction

Many real-world sequential decision-making (or control) problems are inherently multiobjective, requiring controllers to optimize several, often conflicting, criteria simultaneously. Traditional Reinforcement learning (RL) frameworks typically assume a scalar reward signal, which simplifies the learning process but fails to capture the complexity of environments where trade-offs between objectives are essential. Thus, in recent years, there has been an increasing focus on applying RL in domains that include multiple objectives or multiple reward signals [1], [2]. The existing literature on multi-objective RL-based control can be divided into two categories. The first approach is to design a scalar reward function that incorporates multiple objectives by scalarizing a vector reward function in a certain user-defined manner, followed by verification using numerical simulations. This can be seen in the papers [2], [3] where the authors design a domain-specific reward function and then adjust the scalarizing weights based on the result of simulations. The second approach is to treat the additional objectives as constraints and learn the solution for the Constrained Markov Decision Process (CMDP) [4]. The drawbacks of the first approach are the following: Firstly, the reward function design is a very domain-specific task and requires thorough knowledge of the domain. Secondly, if the RL task requires more samples (typical for large-scale systems with many states) than the approach of tuning weights after simulations can be computationally very expensive. Lastly and perhaps more importantly, there are no guarantees of optimality when using this approach, as there may exist a better set of scalarizing weights compared to what the user chose by trial and error. The primary drawback of the second approach is as follows: Firstly, Bellman's principle of optimality is not always valid for CMDPs [5], [6], and therefore, only model-based RL methods might be applicable (here, the controller first learns the underlying model and then solves a linear program [7]). Secondly, this method is not robust to adversarial disturbances as the disturbance can be chosen in such a way that the associated Linear program is infeasible [8].

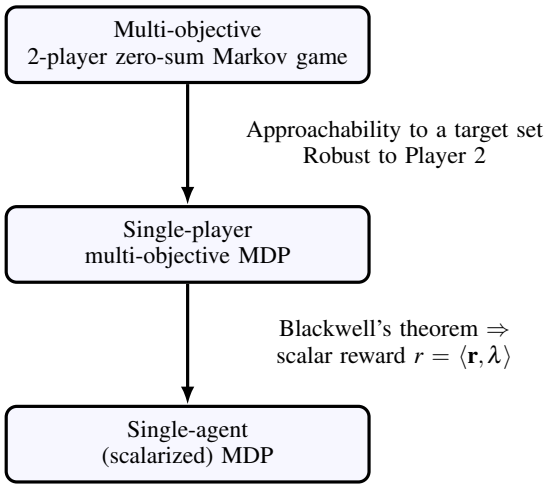
In this paper, we investigate another approach for handling multiple objectives based on Blackwell's Approachability Theorem [9]. The central idea is to steer the long-run time average of the reward vector to a predefined Target set (that can be defined such that it coincides with the Pareto optimal set). The steering is done as follows: First, we calculate the projection of the time-averaged reward vector on the Target set. Thereafter, we calculate the distance between the time-averaged reward vector and its projection on the Target set. This distance is normalized to obtain the unit vector pointing towards the Target set. Blackwell's Approachability Theorem [9] provides the necessary and sufficient condition (in the case of convex sets) that the control policy must satisfy to *Approach* the Target Set despite any adversarial disturbance. This method is related to the concepts of regret minimization, calibration, online optimization, and correlated equilibrium [10], [11]. Despite the usefulness of Blackwell's Approachability Theorem, it should be noted that this Theorem is not directly applicable to systems involving state dynamics, as the Blackwell condition assumes a single-state recurrent system. Shimkin [12] and Mannor [13] extended Blackwell's Approachability Theorem to multiple states in a finite multi-objective Markov Game setting. They assumed that there exists a special recurrent state with finite recurrence time. This allowed them to obtain Approachability guarantees similar in spirit to Blackwell's original result [9]. Milman [14] generalized the work of [12], [13] by removing the recurrence assumption, but the resulting policy computation required knowledge of some constants and was not straightforward. Nevertheless, an RL algorithm has been designed based on Milman's approach in [15]. Another approach in this direction was to restrict the zero-sum game setting to Stackelberg models [16]. Here one player is the leader who reveals his/her move and the second player is the follower who best responds to the leaders

*This work was supported by Poul Due Jensens Fonden project SWIfT

Department of Electronic Systems, Automation and Control, Aalborg University, Fredrik Bajers Vej 7C, 9220 Aalborg East, Denmark rmi@es.aau.dk, lmbu@es.aau.dk, raf@es.aau.dk

preceding move. The resulting Theorem is closer to the original Blackwell's Approachability Theorem without any non-trivial policy computations. However, it is restricted to Stackelberg setting.

Contributions: Our approach in this paper is closely related to [13]. However, it differs in the following ways: Firstly, the RL algorithm of [13] requires running J different RL algorithms in parallel, where J is a possible steering direction. Approachability to the Target set is guaranteed if J is sufficiently large, and this is obviously computationally impractical. In contrast, we show Approachability using a single Policy gradient scheme. Secondly, we extend the method of [13] to systems with continuous state space via function approximation. Our overall approach of going from a multi-objective 2 player zero-sum Markov Game to a single agent Markov decision process is summarized in the following flowchart I. The rest of the paper is organized as follows: In



the next subsection, we introduce some common notation, followed by the precise problem formulation in Section II. The concepts of recurrent times, differential returns, policy gradient theorem, and Blackwell's Approachability Theorem are introduced in Section III. Based on these concepts, the Algorithm is also presented in Section III with convergence guarantee. We simulate the algorithm on a numerical example in Section IV and finally, we provide concluding remarks and possible future work in Section V.

A. Notation

- $\Delta(A)$ denotes a simplex on the finite set $A = \{a_1, \cdots, a_n\}$ i.e. for all $a_i \in A$, $a_i \geq 0$, and $\sum_{i=1}^{n} a_i = 1$.
- The notation $\mathbb{P}_b^a$ represents a probability measure induced by a, b and $\mathbb{E}_b^a$ is the corresponding expectation operator.
- $\langle a, b \rangle$ denotes the standard inner product between the vectors a and b.
- Distance between point and set is defined as $D(a,T) := \inf_{b \in T} \|a - b\|$, where $\|\cdot\|$ is the standard Euclidean norm.
- $\mathbf{r}$ represents a vector reward and r represents a scalar reward.
- π without superscript represents the joint policy i.e. $\pi = (\pi^1, \pi^2)$.
- Π denotes the projection of point a on set B i.e. $\Pi_B(a) = \arg\min_{b \in B} \|a - b\|$.

II. Problem formulation

We consider a Zero-Sum Markov Game from Player 1's perspective (i.e. Player 2's actions can be arbitrary) that is defined as $\mathscr{M} := (X, U^1, U^2, P, R, T, \mu)$, where X is the state space, U^1 and U^2 are the control action spaces, $P : X \times U^1 \times U^2 \to \Delta(X)$ is the transition probability, $R : X \times U^1 \times U^2 \to \mathbb{R}^k$ is the reward vector consisting of k objectives, $T \in \mathbb{R}^k$ is the predefined Target set for Player 1, and $\mu \in \Delta(X)$ is the initial distribution of states. The state space can be a set of finite states or a continuous state space, as we can use parametric representations that allow us to generalize the developed algorithms to possibly infinite states. Similarly, the control action spaces can also be continuous, depending on the parameterization; however, they must be convex and compact in this case. At time $t \in \{0, 1, 2, \cdots\}$, each player observes the state $x_t \in X$, picks a control actions $u_t^1 \in U^1$ (or $u_t^2 \in U^2$). The joint control action taken by the players at time t is denoted by $u_t := (u_t^1, u_t^2)$. The players then observe a sample of the reward vector $\mathbf{r}_t := R(x_t, u_t^1, u_t^2)$, and transition to the new state $x_{t+1} \in X$ based on the transition dynamics P. The functional form of transition dynamics P and the reward function R are unknown to both players; instead, the players observe the history $h_t^i = x_0, u_0^i, x_1, u_1^i, \cdots, x_t$, where $i = \{1, 2\}$ denotes the player. Thus, we have a Reinforcement learning problem as each player's goal is to choose control actions that reinforce the desired outcome (defined in II.1). The set of all possible histories observed by player i up to time t is denoted by $H_t^i := X \times (U^i \times X)^t$. The joint history of the game H_t combines both players' partial histories in the following way, $H_t := X \times (U^1 \times U^2 \times X)^t$. Let $\bar{\mathbf{r}}_t$ be the time-averaged reward vector defined as,

$$\bar{\mathbf{r}}_t := \frac{1}{t} \sum_{n=1}^{t} R(x_n, u_n) \tag{1}$$

The goal of Player 1 is to steer the time-averaged reward vector $\bar{\mathbf{r}}_t$ such that it *Approaches* a desired Target set $T \in \mathbb{R}^k$ (this is made more precise after we define the control policies). We define the control policy for Player 1 (analogously for Player 2) as $\pi_\theta^1 : H_t^1 \to \Delta(U^1)$. Note that π^1 is parameterized by $\theta \in \mathbb{R}^l$. This means that instead of searching for the optimal policy in the entire state space X, we strive to find the best possible optimal policy in the reduced space of dimension l. This technique is referred to as function approximation, as we generally specify the policy as a function that approximates the best possible policy in the reduced l-dimensional space. For the rest of the paper, we will drop θ from the policy π^i for the sake of notational simplicity. The reader should assume that the policies are always parameterized by θ's unless otherwise specified. Each joint control policy $\pi := (\pi^1, \pi^2)$, and initial distribution μ, induces a unique probability measure $\mathbb{P}_\pi^\mu$ (with the corresponding expectation operator $\mathbb{E}_\pi^\mu$) on the

space of (infinite) joint histories $H_\infty = X \times (U^1 \times U^2 \times X)^\infty$ [17]. We will now define the optimality criteria based on the long-run average reward vector. However, before defining the long-run average reward vector, we need an assumption regarding the existence of the stationary distribution of the Markov chain induced by the joint policy π. This assumption will ensure the existence of the limit in (4).

Assumption II.1 (Ergodicity)**.** Every joint policy $\pi = (\pi^1, \pi^2)$ induces a stationary distribution $d^\pi(x)$. The stationary distribution $d^\pi(x)$ is unique and exists for all states $x \in X$, and is defined as follows,

$$d^\pi(x) = \lim_{t\to\infty} \mathbb{P}[x_t = x \mid x_0, \pi] \tag{2}$$

The stationary distribution satisfies the following stationarity property, which is stated next. We begin by defining the simplified notation for transition probability P^π and reward vector R^π while following a joint policy π. For a given joint policy π, we define a state transition matrix P^π between two states x and x' as,

$$P^\pi(x,x') = \sum_{u^1}\sum_{u^2} \pi^1(u^1 \mid x)\pi^2(u^2 \mid x)P(x' \mid x, u^1, u^2).$$

Similarly, we can define $R^\pi(x) := \sum_u \pi(u \mid x)R(x,u)$ as the reward vector generated by following the joint policy π. Due to Assumption II.1, d^π satisfies the following stationarity condition,

$$d^\pi P^\pi = d^\pi. \tag{3}$$

We can now define the long-run average reward vector as follows. Fix an initial distribution μ, then the long-run average reward vector for a given joint policy π is

$$\begin{aligned}\bar{\mathbf{r}}(\pi) &:= \lim_{t\to\infty} \frac{1}{t}\mathbb{E}_\pi^\mu \left[\sum_{n=0}^{t} R(x_n, u_n)\right] \\ &= \lim_{t\to\infty} \frac{1}{t}\sum_{n=0}^{t} \mu \cdot (P^\pi)^n R^\pi \\ &= \mathbb{E}_{d^\pi}\left[R(x,u)\right], \end{aligned} \tag{4}$$

where d^π is the stationary distribution corresponding to the long-run behavior of the Markov chain induced by the joint policy π, and the last equality is because

$$\lim_{t\to\infty} \frac{1}{t}\sum_{n=0}^{t} \mu \cdot (P^\pi)^n = d^\pi,$$

where the limit is in the sense of Cesàro time average limit [18]. In words, asymptotically, the transient effects of the starting states and decisions will be dominated by the long-run behavior characterized by the stationary distribution (3). We can now define Player 1's objective more precisely as,

Definition II.1. Given the initial distribution μ, a policy $\pi^{1\star}$ of Player 1 *Approaches* the target set $T \subset \mathbb{R}^k$ if

$$\lim_{t\to\infty} D(\bar{\mathbf{r}}_t, T) = 0 \quad \mathbb{P}_\pi^\mu \text{ a.s. for any } u^2 \in U^2 \tag{5}$$

Note that the condition (5) must hold for arbitrary control actions of Player 2, which implies that we can focus on designing an algorithm for Player 1 that is robust to bounded adversarial actions.

III. Algorithm for Approaching the Target set

As we consider a vector of rewards (due to the multi-objective nature of our problem), standard Reinforcement learning methods are not applicable as they are designed for scalar rewards. We therefore choose to design an algorithm with two loops. The outer loop will adaptively scalarize the reward vector, while the inner loop will consist of a standard Reinforcement Learning algorithm acting on the scalarized reward. For the inner loop, we will apply a Policy Gradient algorithm, or more precisely, an actor-critic algorithm, since we will also be estimating the Temporal Difference (TD) error. As we study long-run averages, we don't have terminating episodes, in contrast to standard Reinforcement Learning algorithms. Instead, we need to artificially create time instances where we can update the desired parameters. This can be done using recurrence times that indicate when a predefined state $\tilde{x}$ has been observed in the trajectory. Let n denote the number of times the state $\tilde{x}$ has been observed. We define $\tau_0 := 0$ and then corresponding to $n > 0$, let τ_n be the recurrent time for a state $\tilde{x}$ defined as,

$$\tau_n = \min\{t > \tau_{n-1} \mid x_t = \tilde{x}\}, \quad \text{where } n > 0 \tag{6}$$

As a consequence of Assumption II.1, we have finite-time recurrence τ_n for the state $\tilde{x}$, or more precisely

$$\mathbb{E}_\pi[\tau_n] < t' \quad \forall \pi,$$

where t' is a finite number. Note that owing to the stochastic policies and stochastic transition probabilities, τ_n will be a random variable. Furthermore, note that we have dropped μ from the above expectation. This is because for the rest of the paper, we will be focusing on the returns from the recurrent state $\tilde{x}$. The average reward vector up to finite recurrent time τ_n is

$$\eta_n(\pi) := \frac{\mathbb{E}_\pi\left[\sum_{t=0}^{\tau_n - 1} R(x_t, u_t^1, u_t^2)\right]}{\mathbb{E}_\pi[\tau_n]}, \quad \text{where } n > 0. \tag{7}$$

We are now ready to apply the reinforcement learning framework of [19]. Firstly, we need to define the return G_t that quantifies the long-run average performance of the policy. However, we cannot directly use the definition of the standard Differential Returns G_t presented in [19]. This is because [19] considers a scalar reward for a single player, while we are studying a reward vector that is affected by the decisions of both players. Therefore, we need to do some sort of adaptive scalarization of the reward vector and that is presented next. Let $\lambda \in \mathbb{R}^k$ be a unit vector in the reward vector space. We define a projected game at time t as $r := \langle \mathbf{r}, \lambda \rangle$, where r is the scalarized reward in the direction of λ (this unit vector is defined precisely in (18)). Similarly, the scalarized long run average reward for the projected game can be defined as $\bar{r}(\pi) := \langle \bar{\mathbf{r}}(\pi), \lambda \rangle$. Thus, we can now define the Differential return starting at time t, as

$$\begin{aligned} G_t &:= r_{t+1} - \bar{r}(\pi) + r_{t+2} - \bar{r}(\pi) + r_{t+3} - \bar{r}(\pi) + \cdots, \\ &= \sum_{k=t+1}^{\infty} (r_k - \bar{r}(\pi)) \end{aligned} \tag{8}$$

The Differential return G_t quantifies the improvement over the long-run average return $\bar{r}(\pi)$ due to players decisions at time $t+1$ being u^1_{t+1}, u^2_{t+1} with state being x_{t+1}, at time $t+2$ being u^1_{t+2}, u^2_{t+2} with state being x_{t+2}, and so on. Thus, it follows that if $G_t \approx 0$, then no benefit or loss was gained compared to the long-run average reward. Based on the definition of G_t, the Differential Value function for the game is defined as

$$V_\pi(x) := \mathbb{E}_\pi [G_t \mid x_t = x] \tag{9}$$

The Differential value function satisfies a Bellman-like equation [18] (sometimes referred to as the discrete Poisson Equation [20]) or more precisely,

$$V_\pi(x) = \mathbb{E}_\pi \left[r - \bar{r}(\pi) + V_\pi(x') \right] \tag{10}$$

Generally, the term $g := \mathbb{E}_\pi [\bar{r}(\pi)]$ is added on both sides to obtain the Bellman-like equation,

$$V_\pi(x) + g = \mathbb{E}_\pi \left[r + V_\pi(x') \right]. \tag{11}$$

Here, the term $V_\pi(x)$ quantifies the effect of starting at state x (or the bias to state x). Similarly, we can define the Q-function (value function given $u_t = u$) as follows,

$$Q_\pi(x,u) := \mathbb{E}_\pi [G_t \mid x_t = x, u_t = u].$$

Remark 1. Recall that the policy π and control actions u, mentioned in the previous equations, consist of both Player 1's component and Player 2's component, and Player 2 is adversarial, that is, Player 1's policy should be optimal against any control action of Player 2. Thus, we will only consider Player 1's policy from now on, as Player 1 is performing the best response to Player 2's actions. In such a situation, it is possible to apply Policy iteration like algorithms (see Chapter 5 of [18]). In other words, once Player 2's policy is fixed, the Markov Game reduces to a single player Markov Decision Process [21].

Remark 2. For notational simplicity and consistency, we shall still write update in terms of the joint policy $\pi = (\pi^1, \pi^2)$ and joint actions $u = (u^1, u^2)$ unless otherwise stated. As stated in Remark 1 only Player 1 is updating his/her policy π^1 and u^2 is fixed (consequently π^2 is also fixed).

As mentioned earlier, we employ function approximation to extend the algorithm to continuous state spaces. The policy π is parameterized by a vector $\theta \in \mathbb{R}^l$ and depends on a feature representation of state-action pairs. Specifically, each pair (x,u) is mapped to a feature vector $\phi(x,u) \in \mathbb{R}^l$, and the policy uses the inner product $\langle \theta, \phi(x,u) \rangle$ as its score. Intuitively, the feature vector captures the relevant aspects of the state-action space that influence decision-making. A common example is the softmax policy:

$$\pi_\theta(u \mid x) = \frac{\exp(\theta^\top \phi(x,u))}{\sum_{u' \in U} \exp(\theta^\top \phi(x,u'))}, \quad \phi(x,u) \in \mathbb{R}^l, \theta \in \mathbb{R}^l.$$

The parameter θ is updated using the following gradient ascent rule,

$$\theta_{t+1} = \theta_t + \alpha \frac{\partial \bar{r}(\pi)}{\partial \theta}. \tag{12}$$

We can now state the Policy gradient Theorem with compatible function approximation that will allow us to obtain the expression for $\partial \bar{r}(\pi)/\partial \theta$.

Theorem 1 (Policy Gradient Theorem [19], [22])**.** *The gradient of average reward is given by the following expression,*

$$\frac{\partial \bar{r}(\pi)}{\partial \theta} = \sum_x d^\pi(x) \sum_u \frac{\partial \log \pi_\theta(x,u)}{\partial \theta} \delta(x,u), \tag{13}$$

where $\delta(x,u)$ is the TD error calculated at time t as follows,

$$\delta(x_t, u_t) = r(x_t, u_t) - \hat{g}_{t-1} + \hat{V}_{t-1}(x_{t+1}) - \hat{V}_{t-1}(x_t). \tag{14}$$

Here $\hat{V}$ and $\hat{g}$ is approximated via gradient descent with a compatible function approximation that minimizes error (see (16)*,* (17)*).*

This theorem will allow us to sample from the on-policy visited states without knowledge of the stationary distribution. This is because there are no derivatives of the stationary distribution d^π. Instead, we just have an expectation with respect to the unknown distribution. A sample average of the online data between two recurrences of the state $\tilde{x}$ can approximate this expectation. This is due to Kac's Theorem, which is stated next.

Theorem 2 (Kac's Theorem [20])**.** *If the Markov chain induced by the joint policy π is ergodic and admits a state $\tilde{x} \in X$ with positive recurrent time τ_n, then if d^π is the stationary distribution for the induced Markov chain, then*

$$d^\pi(\tilde{x}) = \frac{1}{\mathbb{E}_\pi [\tau_n]}. \tag{15}$$

We will now shed light on how $\hat{V}$ and $\hat{g}$ are computed. Note that, $\hat{V} \approx \langle \rho, \psi(x,u) \rangle$, where $\psi : X \times U \to \mathbb{R}^l$ are the features of the value function and $\rho \in \mathbb{R}^l$ are the corresponding parameters updated via gradient descent in the direction that minimizes the following least squares error,

$$\begin{aligned} \rho_{t+1} &= \rho_t - \beta \nabla_\rho \mathscr{L}(\rho), \\ \mathscr{L}(\rho) &= \frac{1}{t} \sum_{n=1}^{t} (\hat{V}(x_t) - G_t)^2, \end{aligned} \tag{16}$$

where $\hat{V}(x_t) = \langle \rho_t, \psi(x_t, u_t) \rangle$. Substituting (16) in (14) results in the following update rule,

$$\begin{aligned} \hat{V}(x_t) &= \hat{V}(x_t) + \beta \delta_t, \\ \hat{g}_{t+1} &= \hat{g}_t + \beta_g \delta_t, \end{aligned} \tag{17}$$

where β and β_g are step-sizes that satisfy the Robbins Monroe conditions [23]. Thus, we have all the necessary ingredients for calculating the optimal policy in the inner loop. Now, we will focus on the outer loop of our algorithm for which Blackwell's Approachability Theorem for Markov Games is the key ingredient. This theorem gives the necessary condition for optimality in the sense of definition II.1. In the case of convex target sets, the condition (19) given in Blackwell Approachability Theorem 3 is both necessary and sufficient for optimality. Consider the reward space $\mathbb{R}^k$. Let $s \in \mathbb{R}^k$ be any arbitrary point in the reward space. Let

$\Pi_T(s)$ be the projection of s on the Target set T. Then the unit vector λ_s pointing in the direction of the Target set T from s is

$$\lambda_s = \frac{\Pi_T(s) - s}{\|\Pi_T(s) - s\|}. \tag{18}$$

We now have all the ingredients for stating the Approachability Theorem for Markov Games.

Theorem 3 (Approachability Theorem for Markov Games [12])**.** *Given the Assumption II.1. Player* 1 *can* Approach *the set $T \subset \mathbb{R}^k$ as per definition II.1 if the following condition holds: For any point $s \notin T$, there exists a policy π_s^1 such that,*

$$\langle \eta(\pi_s^1, \pi_s^2) - \Pi_T(s), \lambda_s \rangle \geq 0, \text{ for any } u^2 \in U^2. \tag{19}$$

Furthermore, the Approaching *policy is chosen as follows: If $x_t = \tilde{x}$ and the average reward vector upto time t, $\bar{\mathbf{r}}_t \notin T$, play $\pi^1_{\bar{\mathbf{r}}_t}$ until the next visit to state $\tilde{x}$; otherwise play arbitrarily until the next return to state $\tilde{x}$.*

The Blackwell condition (19) can be satisfied in the following way. We strive to choose π^1 in such a way that for any λ, the inner product $\langle \eta(\pi_s^1, \pi_s^2) - \Pi(s), \lambda_s \rangle = 0$, for any $u^2 \in U^2$. Note that η is updated each time we see the state $\tilde{x}$, that is, after recurrence time τ_n. This time is the update time for the outer loop. At time τ_n, we update the scalarizing vector λ by calculating the projection of $\bar{r}$ onto the target set T. This leads us to the inner loop, where Player 1 applies the Policy Gradient algorithm based on the equations (12)-(17) to find the Policy for the given direction λ. We have summarized these steps in the following Algorithm 1. In Algorithm 1, the outer loop runs at timescale n, where n is incremented each time we observe the recurrent state $\tilde{x}$, while the inner loop runs on timescale t that resets after τ_n is reached. Unlike the standard Approachability theorem [9], the Approachability condition (19) needs to be satisfied for a sequence of states.

Proposition 4. *Algorithm 1 ensures convergence of the long-run average reward vector to the Target set T*

Sketch of proof. Fix a recurrent state $\tilde{x}$. The Policy gradient update rule (12) updates policy parameter θ in the direction of maximizing the long run average reward $\langle \bar{\mathbf{r}}(\pi), \lambda \rangle$. The sequence of recurrent times

$$\tau_1, \cdots, \tau_n \to \tau = \frac{1}{d^\pi(\tilde{x})}, \text{ as } n \to \infty,$$

for the recurrent state $\tilde{x}$ due to Kac's Theorem 2. Therefore, $\bar{\mathbf{r}}(\pi) = \eta(\pi)$ as $n \to \infty$. This implies satisfaction of Blackwell condition (19) which can be shown as follows.

Step 1: Without loss of generality, assume that $\eta_0 \notin T$ (as otherwise Player 1 can play arbitrarily as per Theorem 3), therefore the Euclidean distance between η_0 and $\Pi_T(\eta_0)$ is positive.

Step 2: For any policy π, the gradient $\partial \bar{r}(\pi)/\partial\theta$ satisfies Policy Gradient Theorem 1,

$$\frac{\partial \bar{r}(\pi)}{\partial \theta} = \sum_x d^\pi(x) \sum_u \frac{\partial \log \pi_\theta(x,u)}{\partial \theta} \delta(x,u),$$

Algorithm 1 Multi-Objective Policy Gradient

Require: Target set T, Policy and value features ϕ, ψ, Recurrence state $\tilde{x}$, Initial distribution μ, Control action sets U^1, U^2, Step sizes α, β, β_g; Tolerance parameters $\varepsilon_{\text{proj}}$

Ensure: Long-run Average reward Trajectory $\bar{\mathbf{r}}_t \in T$

Initialize: $x_0 \sim \mu$; $u_0^1, u_0^2 \leftarrow 0$; $\bar{\mathbf{r}}_0 \leftarrow R(x_0, u_0)$; $b \leftarrow 0$; $\theta \leftarrow \mathbf{0}$
for $n = 1, \cdots$ **do** ▷ Outer Loop
  $\mathbf{p} \leftarrow \Pi(\bar{\mathbf{r}})$
  $d \leftarrow \|\bar{\mathbf{r}} - \mathbf{p}\|$
  if $d \leq \varepsilon_{\text{proj}}$ **then**
    $\lambda_{\bar{\mathbf{r}}} \leftarrow \mathbf{0}$
  else
    $\lambda_{\bar{\mathbf{r}}} \leftarrow \dfrac{\mathbf{p} - \bar{\mathbf{r}}}{\|\mathbf{p} - \bar{\mathbf{r}}\|}$
  end if
  $G_n \leftarrow 0$
  for $t \leftarrow 1$ **to** τ_n **do** ▷ Inner Loop
    $u_t^1 \sim \pi_t^1$
    Adversery chooses the worst projected point
    $u_t^2 \leftarrow \arg\min_{u^2 \in U^2} \langle \eta(u_t^1, u^2) - \Pi(\bar{\mathbf{r}}), \lambda_{\bar{\mathbf{r}}} \rangle$
    Sample x_t from the environment
    $G_t \leftarrow G_{t-1} + r_t$
    $\delta_t \leftarrow r_t - \hat{g}_{t-1} + \hat{V}_{t-1}(x_{t+1}) - \hat{V}_{t-1}(x_t)$
    $\mathscr{L} \leftarrow (1/t) \sum_{i=1}^t (\hat{V}_{t-1}(x_t) - G_{t-1})^2$
    $\rho_t \leftarrow \rho_{t-1} - \beta \nabla_\rho \mathscr{L}$
    $\hat{V}_t(x_t) \leftarrow \hat{V}_{t-1}(x_t) + \beta \delta_t$
    $\hat{g}_t \leftarrow \hat{g}_{t-1} + \beta_g \delta_t$
    $\theta_t \leftarrow \theta_{t-1} + \alpha \nabla_\theta \log \pi_t^1 \delta_t$
  end for
  $\eta \leftarrow G/\tau_n$
  $n \leftarrow n + 1$
end for

where δ is the TD error (14) for scalarized average reward $\bar{r}(\pi) = \langle \eta(\pi), \lambda_{\eta(\pi)} \rangle$.

Step 3: Gradient ascent on $\bar{r}(\pi)$ increases $\langle \eta(\pi), \lambda_{\eta(\pi)} \rangle$. This can be shown as follows. Let $\eta_{n+1} = \eta(\pi_{\text{new}})$, where π_{new} is obtained at the conclusion of the inner loop in Algorithm 1, and $\eta_n = \eta(\pi_{\text{old}})$, where π_{old} is the previous policy. Then we can show that,

$$\langle \eta_{n+1}, \lambda_{\eta_{n+1}} \rangle \geq \langle \eta_n, \lambda_{\eta_n} \rangle.$$

Thus, after the inner loop, the controller achieves at least the value of the scalarized game in direction $\lambda_{\eta_{n+1}}$:

$$\langle \eta_{n+1}, \lambda_{\eta_{n+1}} \rangle \geq \sup_{\pi^1} \inf_{\pi^2} \langle \eta(\pi^1, \pi^2), \lambda_\eta \rangle.$$

Step 4: Theorem 3 requires that for every $\eta \notin T$, there exists a policy π^1 such that for all adversary policies π^2: $\langle \eta(\pi^1, \pi^2) - \Pi_T(\eta), \lambda_\eta \rangle \geq 0$. Since λ_η is the unit normal from η to $\Pi_T(\eta)$, this condition is equivalent to,

$$\langle \eta_{n+1}, \lambda_{\eta_{n+1}} \rangle \geq \sup_{\pi^1} \inf_{\pi^2} \langle \eta(\pi^1, \pi^2), \lambda_\eta \rangle \geq \langle \Pi_T(\eta), \lambda_\eta \rangle.$$

Thus condition (19) is satisfied. ■

IV. Numerical results

We have applied Algorithm 1 on a simple numerical toy example that simulates temperature ($^\circ C$) and relative humidity (%) in a house similar to the one in [13] for the sake of a direct comparison. The aim of Player 1 (controller) is to control both temperature and relative humidity (states of the system) without any knowledge of the underlying dynamics (modeled as moving averages [24], [25]). However, both temperature and relative humidity are extremely correlated, as increasing or decreasing one will affect the value of the other state. Thus, the problem is multiobjective with the rewards $\mathbf{r} \in \mathbb{R}^2$ that are chosen to be the same as the observed states. The desired target set $T \in \mathbb{R}^2$ is defined based on the comfort of the house occupants. The state space is continuous (the grid in $\mathbb{R}^2$), while the control action space for the controller consists of three pure policies u_0^1, u_1^1, u_2^1. The control action space of the adversary is continuous and can be any point on the lines (corresponding to u_0^1, u_1^1, u_2^1) selected by the controller (see figure 1). From the figure 1, it can be seen that the long-run average reward vector *Approaches* the interior of the Target set T.

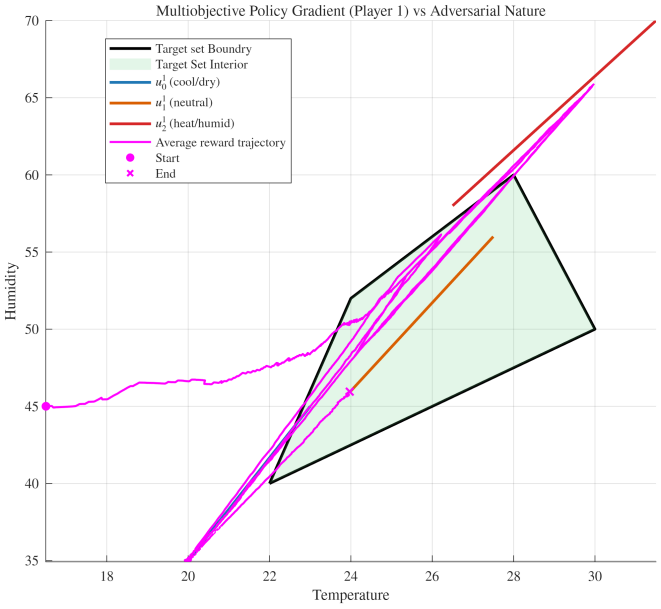


Fig. 1. The long run average reward vector (in pink color) *Approaches* the target set T despite the adversary choosing worst-case points for the policies (u_0^1, u_1^1, u_2^1) selected by Player 1.

V. Conclusions

We have developed an algorithm for multi-objective RL that is robust to adversarial disturbances, and we show that the policy gradient approach can be used for satisfying the Blackwell condition in Proposition 4. The proposed algorithm has been simulated on a numerical example which also demonstrates that the long-run average reward vector converges to the Target set. Future work will involve extending the proposed approach to a more general setting with many players. Another important research direction is to study sample efficiency of the algorithm presented in this paper. As it stands now, the guarantees hold asymptotically.

References

[1] Z. Nagy, G. Henze, S. Dey, J. Arroyo, L. Helsen, X. Zhang, B. Chen, K. Amasyali, K. Kurte, A. Zamzam, *et al.*, "Ten questions concerning reinforcement learning for building energy management," *Building and Environment*, vol. 241, p. 110435, 2023.

[2] J. Li and T. Yu, "Deep reinforcement learning based multi-objective integrated automatic generation control for multiple continuous power disturbances," *IEEE Access*, vol. 8, pp. 156 839–156 850, 2020.

[3] Y. Huang and X. Zhao, "Reinforcement learning-based multiobjective control of grid-connected wind farms," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 5, pp. 7380–7390, 2024.

[4] E. Altman, *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

[5] M. Haviv, "On constrained markov decision processes," *Operations research letters*, vol. 19, no. 1, pp. 25–28, 1996.

[6] R. Misra and R. Wisniewski, "On principle of optimality for safety-constrained markov decision process and p-safe reinforcement learning," *IFAC-PapersOnLine*, vol. 58, no. 17, pp. 338–343, 2024.

[7] E. Altman and A. Shwartz, "Adaptive control of constrained markov chains: Criteria and policies," *Annals of Operations Research*, vol. 28, no. 1, pp. 101–134, 1991.

[8] K. Avrachenkov, J. A. Filar, V. Gaitsgory, and A. Stillman, "Singularly perturbed linear programs and markov decision processes," *Operations Research Letters*, vol. 44, no. 3, pp. 297–301, 2016.

[9] D. Blackwell, "An analog of the minimax theorem for vector payoffs." *Pacific Journal of Mathematics*, vol. 6, no. 4, pp. 1–8, 1956.

[10] V. Perchet, "Approachability, regret and calibration: Implications and equivalences," *Journal of Dynamics and Games*, vol. 1, no. 2, pp. 181–254, 2014.

[11] R. Misra, R. Wisniewski, C. S. Kallesøe, and M. L. Bujorianu, "Robust correlated equilibrium: Definition and computation," *Automatica (To appear)*, 2026.

[12] N. Shimkin *et al.*, "Guaranteed performance regions in markovian systems with competing decision makers," *IEEE Transactions on Automatic Control*, vol. 38, no. 1, pp. 84–95, 1993.

[13] S. Mannor *et al.*, "A geometric approach to multi-criterion reinforcement learning," *The Journal of Machine Learning Research*, vol. 5, pp. 325–360, 2004.

[14] E. Milman, "Approachable sets of vector payoffs in stochastic games," *Games and Economic Behavior*, vol. 56, no. 1, pp. 135–147, 2006.

[15] R. Misra, R. Wiśniewski, and C. S. Kallesoe, "Finding approximate correlated equilibrium for decentralized control," *IEEE Control Systems Letters*, 2025.

[16] D. Kalathil *et al.*, "Approachability in stackelberg stochastic games with vector costs," *Dynamic games and applications*, vol. 7, pp. 422–442, 2017.

[17] A. Jaśkiewicz *et al.*, "Zero-sum stochastic games," *Handbook of dynamic game theory*, pp. 215–272, 2018.

[18] J. Filar *et al.*, *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

[19] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[20] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[21] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *arXiv preprint arXiv:1911.10635*, pp. 321–384, 2019.

[22] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[23] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint: Second Edition*, ser. Texts and Readings in Mathematics. Hindustan Book Agency, 2022. [Online]. Available: https://books.google.dk/books?id=k_ChEAAAQBAJ

[24] G. Mustafaraj, J. Chen, and G. Lowry, "Development of room temperature and relative humidity linear parametric models for an open office using bms data," *Energy and Buildings*, vol. 42, no. 3, pp. 348–356, 2010.

[25] S. Zaharieva, I. Georgiev, S. Georgiev, A. Borodzhieva, and V. Todorov, "A method for forecasting indoor relative humidity for improving comfort conditions and quality of life," *Atmosphere*, vol. 16, no. 3, p. 315, 2025.