This is the preprint of the contribution published as:

The publisher's version is available at:

Off-Policy Temporal Difference Learning for Perturbed Markov Decision Processes: Theoretical Insights and Extensive Simulations

Ali Forootani, *Senior Member, IEEE*, Raffaele Iervolino, *Senior Member, IEEE*, Massimo Tipaldi, and Mohammad Khosravi *Member, IEEE*

Abstract

Dynamic Programming suffers from the curse of dimensionality due to large state and action spaces, a challenge further compounded by uncertainties in the environment. To mitigate these issue, we explore an off-policy based Temporal Difference Approximate Dynamic Programming approach that preserves contraction mapping when projecting the problem into a subspace of selected features, accounting for the probability distribution of the perturbed transition probability matrix. We further demonstrate how this Approximate Dynamic Programming approach can be implemented as a particular variant of the Temporal Difference learning algorithm, adapted for handling perturbations. To validate our theoretical findings, we provide a numerical example using a Markov Decision Process corresponding to a resource allocation problem.

Index Terms

Reinforcement Learning, Markov Decision Processes, Temporal Difference Learning, Perturbed Probability Transition Matrix.

I. INTRODUCTION

IN large-scale Markov Decision Processes (MDPs), Dynamic Programming (DP) faces the curse of dimensionality, where the state and action spaces grow exponentially, making computation infeasible [1]. To tackle these challenges, Approximate Dynamic Programming (ADP) methods are employed in the literature [2], [3]. ADP techniques such as Temporal Difference (TD) use function approximations and simulations to simplify computational complexity [4], [5].

On-policy and off-policy TD methods rely on three main methodologies to estimate cost functions: gradient-based solutions, which use stochastic gradient descent to minimize the prediction error incrementally (e.g., [6], [7]); least-squares minimization techniques, which estimate cost functions by minimizing the least-squares error between the estimated and true returns (e.g., [8], [1], [9]); and probabilistic approaches, which employ Bayesian techniques to model the uncertainty in value estimates, adjusting predictions based on the confidence in the current estimates (e.g. [10], [11]). Off-policy TD is a powerful technique within the broader field of ADP that aims to estimate cost functions using data generated by a policy different from the one being improved [1]. Unlike on-policy methods, which require following the same policy being evaluated, off-policy TD learning offers greater flexibility by leveraging experience from a variety of policies or even simulated environments [12]. In particular, off-policy TD approaches enable more extensive exploration of the environment, potentially resulting in better policies [13]. In [14], generalized advantage estimation was proposed, which smooths the trade-off by reducing variance in policy gradient

updates, allowing more stable learning in high-dimensional control tasks. Exploration strategies, pivotal in Reinforcement Learning (RL), were demonstrated effectively in their greedy approach within Deep Q-Learning to achieve human-level control in `Atari` games [15]. Further advanced exploration-exploitation trade-offs with Proximal Policy Optimization (PPO) was introduced in [12], which uses a clipped surrogate objective to ensure stable policy updates while balancing exploration and exploitation, making it robust for continuous and discrete action spaces.

MDP leverages simulations to generate experience and approximate optimal cost functions that lead to effective policies. However, inaccuracies in the model and dynamic changes in the environment can introduce challenges. If the model does not accurately represent the environment or if the environment changes over time, the effectiveness of a policy based on the model may be compromised. In this context, it becomes crucial to understand how deviations between the model and the real environment impact the optimal cost function and the performance of the perturbed policy, associated with *perturbed transition probability matrices*, which means *perturbed* MDPs. Perturbation analysis in RL examines how changes in the environment or model parameters impact policy performance and value functions. In [10], a TD Bayesian approach for value function approximation that models the uncertainty of predictions was proposed, providing robustness against model perturbations.

This paper proves that sufficiently small perturbation in the environment (transition probability matrices) will lead to restricted bound between the optimal cost-to-go function and non-optimal one. In particular, we examine how transition probability matrix perturbations can be account as off-policy TD learning, specifically Q-learning [16], by investigating the impact of weighted combinations of exploration and optimal actions on efficient learning [16]. In addition, we analyze the variations of cost functions in the context of policy perturbations and provide the corresponding upper bounds for discounted MDPs. More precisely, let the transition matrices $\mathcal{P}^*$ and $\mathcal{Q}$ correspond to the optimal policy π^* and an exploratory policy $\pi_{\mathcal{Q}}$, respectively. Also, let the perturbed transition matrix $\bar{\mathcal{P}}$ be defined as a weighted combination of these two matrices, i.e., for $\bar{\mathcal{P}}$, we have

$$\bar{\mathcal{P}} = (I - \mathcal{A})\mathcal{Q} + \mathcal{A}\mathcal{P}^*, \tag{1}$$

where I denotes the identity matrix and $\mathcal{A}$ is a diagonal matrix with $\alpha \in]0, 1[$ on its diagonal, i.e. $\mathcal{A} = \alpha I$ and α is the discounted factor, specifying the trade-off between the use of the optimal policy π^* and the exploratory policy $\pi_{\mathcal{Q}}$. $\mathcal{A} = \alpha I$ represents a uniform trade-off between the exploratory policy $\pi_{\mathcal{Q}}$ and the optimal policy π^*, but a more general choice of $\mathcal{A}$ can enable state-dependent perturbations. For instance, by allowing $\mathcal{A}$ to have different diagonal entries for each state, perturbations can be adjusted adaptively, reflecting varying levels of confidence in the estimated transition dynamics or incorporating state-dependent exploration-exploitation trade-offs. This formulation also aligns well with frameworks that introduce state-dependent learning rates or adaptive exploration strategies, making it particularly useful in reinforcement learning settings where uniform perturbation assumptions may not hold.

Our framework explicitly links the perturbation of transition matrices to the exploration-exploitation trade-off through a convex combination of $\mathcal{Q}$ and $\mathcal{P}^*$. While similar trade-offs exist in algorithms like PPO and TRPO [12], [17], the novelty lies in our matrix-based formulation and the explicit derivation of convergence guarantees under perturbations, which is less explored in MDP research. We characterize the performance resulting from the utilization of suboptimal policies associated with $\bar{\mathcal{P}}$. In particular, we discuss the projection on the subspace of specifically selected features based on such perturbations and the convergence properties for off-policy TD methods. Furthermore, following the work of [18], we ensure that the perturbed cost function remains bounded over an infinite time horizon, provided that for any pairs of stationary policy (including $\pi = \pi^*$) it is $\|\mathcal{P} - \bar{\mathcal{P}}\|_\infty \leq 1 - \alpha$.

Non-stationary MDPs have been extensively studied in the RL literature, particularly in settings where transition dynamics or reward functions evolve over time. Prior works such as [19]–[29] focus on different aspects of non-stationarity, including sliding-window-based estimations [21], [23], [30], variation budget constraints [27], [29], and kernel-based approaches [26]. Most of these works, however, primarily address settings where the transition kernel evolves in an arbitrary or adversarial manner, often requiring

explicit tracking mechanisms for adapting to changes. In contrast, our formulation introduces a structured perturbation model where the perturbed transition matrix is represented as a convex combination of the optimal and exploratory transition matrices, as given in (1). This representation provides a principled approach to modeling controlled deviations in transition dynamics, allowing for a smooth interpolation between optimal and exploratory policies, which aligns with exploration-exploitation trade-offs studied in optimistic RL methods [31]–[33]. Furthermore, unlike previous works that rely on tabular settings or parametric drift assumptions [27], [34], our framework generalizes to off-policy learning scenarios by explicitly linking policy perturbations to transition matrix perturbations, offering a novel perspective on policy optimization under controlled non-stationarity.

This article is organized as follows. Section II provides some preliminaries on DP and outlines the cost function approximation problem. The on-policy ADP approach is discussed in Section III. Section IV analyzes the perturbed transition probability matrices and their impacts on cost functions. Section V explores the connection between perturbed transition probability matrices and Q-learning. In Section VI, the proposed off-policy TD approach is assessed using a resource allocation problem. Section VII concludes the paper.

II. Preliminaries

Let us consider an infinite-horizon DP problem for an MDP. Such a problem often is solved by employing Bellman's principle of optimality recursively backwards in time [35]. In this regard, a shorthand notation is introduced for Bellman operator as $\mathcal{F}^* : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$ for any cost function vector $J \in \mathbb{R}^{|\mathcal{X}|}$ which can be expressed as [1]:

$$(\mathcal{F}^* J)(x) = \min_{u \in \mathcal{U}} \left[\mathcal{R}(x) + \alpha \sum_{x' \in \mathcal{X}} \mathcal{P}_{xx'}(u) J(x') \right], \tag{2}$$

where $\mathcal{X}$ is the MDP state space with cardinality $|\mathcal{X}|$, with x and x' being its two generic elements. Moreover, $u \in \mathcal{U}$ is a generic element of finite set of control actions $\mathcal{U}$, $\mathcal{R} \in \mathbb{R}^{|\mathcal{X}|}$ the vector of instant cost with element $\mathcal{R}(x)$[2], $\mathcal{P} : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to [0, 1]$ the state transition probability matrix, with generic element $\mathcal{P}_{xx'}(u)$, and $\alpha \in]0, 1[$ the discount factor. In matrix form, (2) simplifies to: $\mathcal{F}^* J = \mathcal{R} + \alpha \mathcal{P}^* J$. As noted in [13], this shorthand notation is common for simplifying complex expressions. Applying the sequence of optimal decision functions $\{\mu^*, \mu^*, \dots\}$, where $\mu^* : \mathcal{X} \to \mathcal{U}$, as time horizon goes to infinity results in optimal stationary policy π^* with the associate steady state probability distribution $\epsilon^* \in \mathbb{R}^{|\mathcal{X}|}$(we denote by ϵ_x^* an element of this vector corresponding to the state x). The Bellman equation for the optimal cost function satisfies: $J^* = \mathcal{R} + \alpha \mathcal{P}^* J^*$, where $J^* \in \mathbb{R}^{|\mathcal{X}|}$ is the vector of optimal costs, and $\mathcal{P}^* \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ is the optimal transition probability matrix. Any non-optimal stationary policy is shown by π with its associated shorthand notation $\mathcal{F}_\pi$ and associated cost function J_π [1], [35]. The cost function $J^*(x)$ is usually approximated by a parametric model $\tilde{J}^*(x, r)$, where $r \in \mathbb{R}^\psi$ is the parameter vector to be optimized, and $\psi \in \mathbb{N}^+$ is a selected number of features. This approximation, $\tilde{J}^* : \mathcal{X} \times \mathbb{R}^\psi \to \mathbb{R}^{|\mathcal{X}|}$, uses a low-dimensional linear function $J^* \approx \Phi r^*$, where $\Phi \in \mathbb{R}^{|\mathcal{X}| \times \psi}$ is a feature matrix with linearly independent columns which we show by $\phi(x)$ its row corresponding to state x, and $r^* \in \mathbb{R}^\psi$ is the parameter vector to compute. Substituting the approximation into the Bellman equation yields the projected equation: $\Phi r^* = \Pi(\mathcal{R} + \alpha \mathcal{P}^* \Phi r^*)$, where $\Pi = \Phi(\Phi^\top \Theta^* \Phi)^{-1} \Phi^\top \Theta^*$ is the projection operator, and Θ^* is the diagonal matrix with ϵ^* on the diagonal and zero elsewhere [36]. Here, Φr^* approximates J^*, making the equation solvable. This approach relies on three key assumptions for any stationary policy π (including the optimal π^*) and related cost function J_π: (i) the corresponding finite Markov chain is regular, with the stochastic matrix $\mathcal{P}$ having a unique steady-state probability distribution ϵ, with strictly positive elements ϵ_x, i.e., $\epsilon_x > 0, \forall x \in \mathcal{X}$; (ii) the feature matrix Φ is full rank with rank ψ; (iii) the feature matrix Φ captures

---

[1]This approach can be applied similarly for value function optimization by replacing min function by max function. See the example in Section VI.

[2]We omit the dependence of $\mathcal{R}$ on a specific control action.

the key characteristics of the cost function J_π, allowing Φr to closely approximate it. In this context, the dimensionality of the feature space (ψ) plays a crucial role in determining computational feasibility.

III. ADP METHODS BASED ON ON-POLICY FRAMEWORK

Before delving into the main results of this article, we present introductory material on the on-policy ADP approach to prepare for the discussion of off-policy methods. In a more general framework, TD methods can be classified into on-policy and off-policy approaches [16]. In on-policy learning, the goal is to learn the cost function $\tilde{J}_\pi(x)$, which represents the approximated expected long-term discounted costs from state x when following a target policy π. The learning process occurs while the agent actively follows the same policy π. On-policy methods are particularly effective in ensuring good convergence, especially when using linear function approximations. To find the optimal parameter vector r^* in the on-policy case for the optimal policy π^*, we solve the following optimization problem:

$$r^* = \arg\min_r \|\Phi r - (\mathcal{R} + \alpha \mathcal{P}^* \Phi r)\|_{\epsilon^*}^2, \tag{3}$$

which minimizes the square of the weighted Euclidean norm[3] [4] of the error between Φr and the projected Bellman equation, where the weight ϵ^* is the steady-state probability distribution associated to π^*. By differentiating (3) and setting the gradient to zero, we obtain: $\Phi^\top \Theta^* (\Phi r^* - \mathcal{R} - \alpha \mathcal{P}^* \Phi r^*) = 0$, where Θ^* is a diagonal matrix with distribution ϵ^* along its diagonal. Solving this equation gives:

$$r^* = \left(\Phi^\top \Theta^* \Phi - \alpha \Phi^\top \Theta^* \mathcal{P}^* \Phi\right)^{-1} \left(\Phi^\top \Theta^*\right) \mathcal{R}.$$

In this formulation, the term $\left(\Phi^T \Theta^* \Phi - \alpha \Phi^\top \Theta^* \mathcal{P}^* \Phi\right)$ represents the matrix involved in solving for r^*, incorporating both the feature matrix and the transition dynamics. This approach provides a way to approximate the optimal cost function in infinite-horizon problems using a lower-dimensional representation, leveraging the structure of the Bellman equation and the projection framework to find an effective solution. Defining $\mathcal{Z}^* = \Phi^\top \Theta^* (I - \alpha \mathcal{P}^*) \Phi$, and $d^* = \Phi^\top \Theta^* \mathcal{R}$, then, the optimal parameter vector is: $r^* = \mathcal{Z}^{*-1} d^*$. For large state spaces, computing r^* directly is infeasible, so iterative methods are employed. The update rule can be derived as follows:

$$r_{k+1}^* = \arg\min_r \|\Phi r - (\mathcal{R} + \alpha \mathcal{P}^* \Phi r_k)\|_{\epsilon^*}^2.$$

Taking the gradient and setting it to zero:

$$(\Phi^\top \Theta^* \Phi) r_{k+1}^* = \Phi^\top \Theta^* (\mathcal{R} + \alpha \mathcal{P}^* \Phi r_k^*),$$

and rearranging and simplifying:

$$r_{k+1}^* = r_k^* - (\Phi^\top \Theta^* \Phi)^{-1} (\mathcal{Z}^* r_k^* - d^*). \tag{4}$$

This iterative update formula is related to Least Squares Temporal Difference (LSTD) methods [13]. Note that (4) is applicable to any generic target policy π when considering the associated steady state probability distribution ϵ (whose elements will be the diagonal elements of a diagonal matrix Θ). In off-policy learning, the objective remains the same: to estimate the cost function $J_\pi(x)$ for the target policy π. However, the actions taken during the learning process follow a different behavior policy $\bar{\pi}$. Even though the agent follows $\bar{\pi}$ during learning, the focus is still on accurately estimating the cost function for the target policy π (see [6] and reference therein). The mechanism to compute iteratively the parameter vector is analogous to the one in (4) and is based on perturbation analysis, which is discussed in the next section.

---

[3] Weighted Euclidean norm on $\mathbb{R}^{|\mathcal{X}|}$ for any vector $J \in \mathbb{R}^{|\mathcal{X}|}$ is defined as $\|J\|_{\epsilon^*} = \sqrt{\sum_{\forall x \in \mathcal{X}} \epsilon_x^* \left(J(x)\right)^2}$.

[4] In this article, we make use of standard definitions of norms for matrices and vectors, e.g., $\|\mathcal{P}\|_\infty$ or $\|J\|_\infty$.

IV. Perturbation Analysis of stochastic matrices and Cost function bounds

In order to apply off-policy methods, which require perturbation of the transition probability matrices (for exploration purposes), it is important to understand how perturbations in transition dynamics affect cost functions. This section establishes bounds for the differences between cost functions from perturbed and unperturbed dynamics, using properties of stochastic matrices and positive definiteness. These results provide insights into the resilience of MDPs under uncertain transition probabilities, aiding ADP methods in developing robust decision strategies. Using the results reported in [37] (Lemma 6.3.1), the following Lemma 1 and Remark 1 guarantee the convergence of on-policy TD approaches and accordingly recursive iteration (4). This result will be extended to show the convergence property in the case of off-policy TD approach as well.

Lemma 1. *For any stochastic matrix $\mathcal{P}$ corresponding to irreducible and regular Markov chain with associated stationary probability distribution ϵ, whose elements are arranged along the diagonal of a diagonal matrix Θ, the matrix $\Theta(I - \alpha\mathcal{P})$ is positive definite, where $\alpha \in]0,1[$.*

Proof: To prove that $\Theta(I - \alpha\mathcal{P})$ is positive definite, we want to show that for any non-zero vector $J \in \mathbb{R}^{|\mathcal{X}|}$: $J^\top\Theta(I - \alpha\mathcal{P})J > 0$. We have: $J^\top\Theta J = \sum_x \epsilon_x J_x^2$ and: $J^\top\Theta\mathcal{P}J = \sum_x \epsilon_x J_x \sum_{x'} \mathcal{P}_{xx'}J_{x'}$, where the term $\sum_{x'} \mathcal{P}_{xx'}J_{x'}$ is the expected value of $J_{x'}$ given x, denoted by $\mathbb{E}[J_{x'} \mid x]$. Since ϵ is a stationary distribution we can rewrite: $\sum_x \epsilon_x J_x \sum_{x'} \mathcal{P}_{xx'}J_{x'} = \mathbb{E}_\epsilon\left[J_x \cdot \mathbb{E}[J_{x'} \mid x]\right]$. Applying the Cauchy-Schwarz inequality to the expectation $\mathbb{E}_\epsilon\left[J_x \cdot \mathbb{E}[J_{x'} \mid x]\right]$, we get:

$$\mathbb{E}_\epsilon\left[J_x \cdot \mathbb{E}[J_{x'} \mid x]\right] \leq \sqrt{\mathbb{E}_\epsilon[J_x^2]} \cdot \sqrt{\mathbb{E}_\epsilon\left[(\mathbb{E}[J_{x'} \mid x])^2\right]}.$$

By Jensen's inequality for convex functions, we also have:

$$\mathbb{E}\left[(\mathbb{E}[J_{x'} \mid x])^2\right] \leq \mathbb{E}[J_{x'}^2].$$

Thus,

$$\sqrt{\mathbb{E}_\epsilon\left[(\mathbb{E}[J_{x'} \mid x])^2\right]} \leq \sqrt{\mathbb{E}_\epsilon[J_x^2]}.$$

Therefore, $\mathbb{E}_\epsilon\left[J_x \cdot \mathbb{E}[J_{x'} \mid x]\right] \leq \mathbb{E}_\epsilon[J_x^2]$. Or, equivalently, $\sum_x \epsilon_x J_x \sum_{x'} \mathcal{P}_{xx'}J_{x'} \leq \sum_x \epsilon_x J_x^2$. Being $\alpha \in]0,1[$, this gives us the desired result: $\sum_x \epsilon_x J_x^2 > \alpha \sum_x \sum_{x'} \epsilon_x \mathcal{P}_{xx'}J_x J_{x'}$. ∎

Remark 1. *From the results of Lemma 1 and full rank assumption of matrix Φ we know $\mathcal{Z} = \Phi^\top\Theta(I - \alpha\mathcal{P})\Phi$ is positive definite. Moreover $(\Phi^\top\Theta\Phi)$ is symmetric positive definite, therefore invertible (see Lemma 5.5 in [38]). From Proposition 6.3.3 and Lemma 6.3.2 in [37] the recursive iteration (4) directly converges to the solution of the projected equation since the matrix $I - (\Phi^\top\Theta\Phi)^{-1}\mathcal{Z}$ has eigenvalues strictly within the unit circle (see also Theorem 5.6 in [38]).*

Remark 1 implies that necessary and sufficient condition for the convergence of any on-policy TD algorithm depends on the positive definiteness of matrix $\mathcal{Z}$, and hence of $\Theta(I - \alpha\mathcal{P})$. This concept can be extended to the case of off-policy TD algorithm. In the next Lemma, we investigate such property for the case of perturbed transition probability matrix $\bar{\mathcal{P}}$.

Lemma 2. *For any stochastic matrix $\mathcal{P}$ corresponding to an irreducible and regular Markov chain, consider a stationary probability distribution $\bar{\epsilon}$, whose elements are arranged along the diagonal of a diagonal matrix $\bar{\Theta}$, associated with the perturbed matrix $\bar{\mathcal{P}} = (I - \mathcal{A})\mathcal{Q} + \mathcal{A}\mathcal{P}$, being $\mathcal{A}$ a diagonal matrix with $\alpha \in]0,1[$ on its diagonal, and $\mathcal{Q}$ another stochastic matrix. Then, the matrix $\bar{\Theta}(I - \alpha\mathcal{P})$ is positive definite.*

Proof: We need to show that for any non-zero vector $J \in \mathbb{R}^{|\mathcal{X}|}$, $J^\top\bar{\Theta}(I - \alpha\mathcal{P})J > 0$. To this end, consider the matrix $\bar{\Theta}(I - \alpha\bar{\mathcal{P}})$, and apply the results of Lemma 1. Following the same proof steps, we can state that:

$$\sum_x \bar{\epsilon}_x J_x^2 \geq \sum_{x,x'} \bar{\epsilon}_x \bar{\mathcal{P}}_{xx'}J_x J_{x'}, \tag{5}$$

and also $\sum_x \bar{\epsilon}_x J_x^2 - \alpha \sum_x \bar{\epsilon}_x \sum_{x'} \frac{\bar{P}_{xx'}}{\alpha} J_x J_{x'} \geq 0$. Since $\bar{P}_{xx'} = \alpha \mathcal{P}_{xx'} + (1-\alpha)\mathcal{Q}_{xx'}$, and being $\mathcal{Q}$ another stochastic matrix, by substitution in (5), it is: $\sum_x \bar{\epsilon}_x J_x^2 - \alpha \sum_x \bar{\epsilon}_x \sum_{x'} \mathcal{P}_{xx'} J_x J_{x'} > 0$, which completes the proof. ∎

Now we are ready to consider the impact of perturbation introduced by the matrix $\bar{\mathcal{P}}$, defined as $\bar{\mathcal{P}} = (I - \mathcal{A})\mathcal{Q} + \mathcal{A}\mathcal{P}^*$, on the projected Bellman equation. This perturbation can be incorporated into the projected Bellman equation by considering the perturbed transition matrix $\bar{\mathcal{P}}$ instead of $\mathcal{P}^*$. The projected Bellman equation with perturbation is: $\Phi\bar{r} = \bar{\Pi}(\mathcal{R} + \alpha\mathcal{P}^*\Phi\bar{r})$, where $\bar{\Pi} = \Phi(\Phi^\top\bar{\Theta}\Phi)^{-1}\Phi^\top\bar{\Theta}$ is the projection operator. Substituting $\bar{\Pi}$ and rearranging, we get: $(\Phi^\top\bar{\Theta}\Phi - \alpha\Phi^\top\bar{\Theta}\mathcal{P}^*\Phi)\bar{r} = \Phi^\top\bar{\Theta}\mathcal{R}$. Defining $\bar{\mathcal{Z}} = \Phi^\top\bar{\Theta}(I - \alpha\mathcal{P}^*)\Phi, \bar{d} = \Phi^\top\bar{\Theta}\mathcal{R}$, so the solution for $\bar{r}$ is: $\bar{r} = \bar{\mathcal{Z}}^{-1}\bar{d}$. This formulation allows us to solve for the parameter vector $\bar{r}$ that approximates the cost function under the perturbed policy, providing a way to assess the impact of deviations from the optimal policy within the framework of ADP. Similar to previous case, for large state spaces, solving directly for $\bar{r}$ is impractical. Instead, iterative methods like TD are used. The iterative update formula in the off-policy case is:

$$\bar{r}_{k+1} = \bar{r}_k - (\Phi^\top\bar{\Theta}\Phi)^{-1}(\bar{\mathcal{Z}}r_k - \bar{d}), \tag{6}$$

whose convergence property can be derived from the following theorem.

Theorem 3. *Consider a Markov Decision Process (MDP) with state space $\mathcal{X}$, action space $\mathcal{U}$, reward vector $\mathcal{R}$, and optimal transition probability matrix $\mathcal{P}^*$. Let the perturbed transition probability matrix be defined as: $\bar{\mathcal{P}} = (I - \mathcal{A})\mathcal{Q} + \mathcal{A}\mathcal{P}^*$, where $\mathcal{A}$ is a diagonal matrix with the discount factor $\alpha \in]0,1[$ along its diagonal, and $\mathcal{Q}$ is a stochastic matrix associated with an exploratory policy. Define: $\bar{\mathcal{Z}} = \Phi^\top\bar{\Theta}(I - \alpha\mathcal{P}^*)\Phi, \quad \bar{d} = \Phi^\top\bar{\Theta}\mathcal{R}$, where $\Phi \in \mathbb{R}^{|\mathcal{X}|\times m}$ is a feature matrix, and $\bar{\Theta}$ is a diagonal matrix with the stationary probability distribution $\bar{\epsilon}$ on its diagonal. The recursive iteration for the parameter vector $\bar{r}_k \in \mathbb{R}^\psi$ is given by: $\bar{r}_{k+1} = \bar{r}_k - (\Phi^\top\bar{\Theta}\Phi)^{-1}(\bar{\mathcal{Z}}\bar{r}_k - \bar{d})$. Then, the iterative update $\bar{r}_k$ converges to the solution $\bar{r}$ of the projected Bellman equation: $\Phi\bar{r} = \bar{\Pi}(\mathcal{R} + \alpha\mathcal{P}^*\Phi\bar{r})$, where $\bar{\Pi} = \Phi(\Phi^\top\bar{\Theta}\Phi)^{-1}\Phi^\top\bar{\Theta}$.*

Proof: From the results of Lemma 2 for the case $\mathcal{P} = \mathcal{P}^*$, we know $\bar{\mathcal{Z}} = \Phi^\top\bar{\Theta}(I - \alpha\mathcal{P}^*)\Phi$ is positive definite. Moreover $(\Phi^\top\bar{\Theta}\Phi)$ is symmetric positive definite, therefore invertible. Since the convergence of (6) is directly related to the positive definiteness of the term $\bar{\Theta}(I - \alpha\mathcal{P}^*)$, then we can guarantee the convergence of projected based ADP with perturbed transition matrices. ∎

Remark 2. *The recursive iteration (6) can be computed using the Monte Carlo simulation mechanism. This procedure involves producing two sequence of state visits via off-policy TD methods, where the sequence $\{\bar{x}(0), \bar{x}(1), \bar{x}(2), \dots\}$ is generated using the transition matrix $\bar{\mathcal{P}}$ (behavior policy) or a steady-state distribution $\bar{\epsilon}$, and we also generate an additional sequence of independent transitions $\{(\bar{x}(0), x(0)), (\bar{x}(1), x(1)), \dots\}$ according to an original transition matrix $\mathcal{P}$ not necessarily optimal (target policy).*

An important metric in analyzing perturbed MDPs is the upper bound for the difference between cost functions derived from perturbed and unperturbed transition probabilities matrices. This ensures consistent policy performance under non-stationary or perturbed conditions in MDPs and helps manage approximation errors in value iteration and ADP methods, thereby maintaining optimal policy quality [12].

The next theorem provides an upper bound for the difference between optimal cost function and its perturbed counterpart. This result is particularly useful when the optimal policy is not known and we want to obtain a bound on the norm of the error between the estimated cost function and optimal one.

Theorem 4. *Consider an MDP with $\mathcal{X}, \mathcal{U}, \mathcal{R}$, and two different transition probability matrices, $\mathcal{P}^*$ and $\bar{\mathcal{P}}$, with associated cost functions J^* and $\bar{J}$, respectively. Let $\bar{\mathcal{P}} = (I - \mathcal{A})\mathcal{Q} + \mathcal{A}\mathcal{P}^*$, where $\mathcal{A}$ is a diagonal matrix with $\alpha \in]0,1[$ on the diagonal, and $\mathcal{Q}$ is a stochastic matrix. If $\|\mathcal{P}^* - \bar{\mathcal{P}}\|_\infty \leq (1-\alpha)$, the difference between J^* and $\bar{J}$ is bounded by: $\|J^* - \bar{J}\|_\infty \leq \frac{\alpha\|\mathcal{R}\|_\infty}{1-\alpha}$.*

Proof: Using the Bellman optimality equations and the fixed point mapping of the DP for J^* and $\bar{J}$

[35], we can write the following:

$$\left|J^*(x) - \bar{J}(x)\right| = \left|\mathcal{R}(x) + \alpha \sum_{x' \in \mathcal{X}} \mathcal{P}^*_{xx'} J^*(x') - \mathcal{R}(x) - \alpha \sum_{x' \in \mathcal{X}} \bar{\mathcal{P}}_{xx'} \bar{J}(x')\right|. \tag{7}$$

This simplifies to:

$$\left|J^*(x) - \bar{J}(x)\right| = \alpha \left|\sum_{x' \in \mathcal{X}} \mathcal{P}^*_{xx'} J^*(x') - \sum_{x' \in \mathcal{X}} \bar{\mathcal{P}}_{xx'} \bar{J}(x')\right|. \tag{8}$$

We now decompose the sum:

$$\sum_{x' \in \mathcal{X}} \mathcal{P}^*_{xx'} J^*(x') - \sum_{x' \in \mathcal{X}} \bar{\mathcal{P}}_{xx'} \bar{J}(x) = \sum_{x' \in \mathcal{X}} \left(\mathcal{P}^*_{xx'} J^*(x') - \bar{\mathcal{P}}_{xx'} J^*(x')\right) + \sum_{x' \in \mathcal{X}} \bar{\mathcal{P}}_{xx'} (J^*(x') - \bar{J}(x')). \tag{9}$$

Substituting (9) into (8) we have:

$$\left|J^*(x) - \bar{J}(x)\right| = \alpha \left|\sum_{x' \in \mathcal{X}} \left(\mathcal{P}^*_{xx'} J^*(x') - \bar{\mathcal{P}}_{xx'} J^*(x')\right) + \sum_{x' \in \mathcal{X}} \bar{\mathcal{P}}_{xx'} (J^*(x') - \bar{J}(x'))\right|,$$

Using triangle inequality we get:

$$\left|J^*(x) - \bar{J}(x)\right| \leq \alpha \sum_{x' \in \mathcal{X}} \left|\mathcal{P}^*_{xx'} - \bar{\mathcal{P}}_{xx'}\right| \left|J^*(x')\right| + \alpha \sum_{x' \in \mathcal{X}} \left|\bar{\mathcal{P}}_{xx'}\right| \left|J^*(x') - \bar{J}(x')\right|, \tag{10}$$

where the first term contains the difference between the transition matrices $\mathcal{P}^*$ and $\bar{\mathcal{P}}$, while the second term accounts for the difference in cost functions. Using the assumption that $\left\|\mathcal{P}^* - \bar{\mathcal{P}}\right\|_\infty < (1 - \alpha)$, the properties of infinite norm, and the fact that (10) holds for any norm, we bound the first term:

$$\left\|J^* - \bar{J}\right\|_\infty \leq \alpha(1 - \alpha)\|J^*\|_\infty + \alpha \sum_{x' \in \mathcal{X}} \left|\bar{\mathcal{P}}_{xx'}\right| \left|J^*(x') - \bar{J}(x')\right|.$$

Next, we use the fact that $\sum_{x' \in \mathcal{X}} \bar{\mathcal{P}}_{xx'} = 1$, since $\bar{\mathcal{P}}$ is a stochastic matrix. Therefore, we have:

$$\left\|J^* - \bar{J}\right\|_\infty \leq \alpha(1 - \alpha)\left\|J^*\right\|_\infty + \alpha\left\|J^* - \bar{J}\right\|_\infty,$$

Rearranging the inequality to isolate $\|J^* - \bar{J}\|_\infty$, we get: $\left\|J^* - \bar{J}\right\|_\infty \leq \frac{\alpha(1-\alpha)\left\|J^*\right\|_\infty}{1-\alpha}$. Since we know from the Bellman equation that $\|J^*\|_\infty \leq \frac{\|\mathcal{R}\|_\infty}{1-\alpha}$, by substituting this bound, we obtain: $\|J^* - \bar{J}\|_\infty \leq \frac{\alpha\|\mathcal{R}\|_\infty}{1-\alpha}$, which completes the proof. ∎

Remark 3. *The norm condition $\|\mathcal{P}^* - \bar{\mathcal{P}}\|_\infty \leq (1 - \alpha)$ in Theorem 4 is satisfied whenever it is $\|\mathcal{P}^* - \mathcal{Q}\|_\infty \leq 1$. Indeed, the deviation from the optimal policy introduces a perturbation matrix $\mathcal{P}^* - \bar{\mathcal{P}} = (I - \mathcal{A})(\mathcal{P}^* - \mathcal{Q})$, whose norm is bounded as follows: $\|\mathcal{P}^* - \bar{\mathcal{P}}\|_\infty \leq (1 - \alpha)\|\mathcal{P}^* - \mathcal{Q}\|_\infty$.*

Remark 4. *While Theorems 3 and 4 analyze the impact of perturbations on the optimal transition matrix $\mathcal{P}^*$, the proposed algorithm is applicable to any general target transition matrix $\mathcal{P}$, corresponding to an arbitrary stationary policy, whether optimal or suboptimal. Since off-policy TD learning is independent of whether $\mathcal{P}$ is optimal, our framework extends beyond optimal policies and remains valid for exploratory and intermediate policies as well.*

Algorithm 1 summarizes the off-policy implementation using TD learning and perturbed transition matrix $\bar{\mathcal{P}}$ for the general target transition probability matrix $\mathcal{P}$.

Algorithm 1 Off-Policy `TD` Learning with $\bar{\mathcal{P}}$

1: Initialize parameter vector $\bar{r}_0$, discount factor α, feature representation $\phi(x)$ for each state x, and behavior policy transition matrix $\bar{\mathcal{P}}$
2: **for** each episode **do**
3:     **for** each time step k **do**
4:         Generate the state $\bar{x}(k)$ using $\bar{\mathcal{P}}$
5:         Generate the state $x(k)$ from the transition pair $\big(\bar{x}(k), x(k)\big)$ using $\mathcal{P}$
6:         Observe the immediate cost $\mathcal{R}(x(k))$ associated with the target policy
7:         Compute the `TD` error:
$$\delta_k = \mathcal{R}(x(k)) + \alpha\phi(x(k))^\top \bar{r} - \phi(\bar{x}(k))^\top \bar{r}$$

8:         Update parameter vector: $\bar{r} \leftarrow \bar{r} + \delta_k \phi(x_k)$
9:     **end for**
10: **end for**

V. Analogy Between the Perturbed Transition Matrix and Q-Learning

In this section, we discuss how the analysis on perturbed transition matrix $\bar{\mathcal{P}}$ relates to Q-learning, where the optimal policy π^* is combined with an exploratory policy, say $\pi_{\mathcal{Q}}$. The mixture of these two policies can be expressed as a weighted combination, similar to the exploration-exploitation trade-off in Q-learning. In Q-learning, the policy is a mixture of an *exploratory policy* and an *optimal policy*, in the sense that the action-selection strategy in Q-learning combines exploitation of the optimal policy π^* and exploration using a stochastic policy $\pi_{\mathcal{Q}}$.

This is mathematically described by the policy mixture [16]:

$$\pi_\xi(u|x) = \xi \cdot \pi_{\mathcal{Q}}(u|x) + (1 - \xi) \cdot \pi^*(u|x),$$

where $\pi^*(u|x)$ is the optimal policy that chooses action u in state x based on the optimal Q-function $\mathcal{Q}^*(x, u)$, $\pi_{\mathcal{Q}}(u|x)$ is an exploratory policy, and $\xi \in]0, 1[$ is the exploration parameter controlling the balance between exploration and exploitation. Under this mixed policy π_ξ, the transition dynamics can be expressed as a combination of the transition matrices corresponding to π^* and $\pi_{\mathcal{Q}}$: $P_{\pi_\xi} = \xi P_{\pi_{\mathcal{Q}}} + (1 - \xi) P_{\pi^*}$. We now observe that the perturbed transition matrix $\bar{\mathcal{P}}$ can be seen as a similar mixture of policies, with α playing the role of $1 - \xi$ in Q-learning:

$$\bar{\mathcal{P}} = (I - \mathcal{A})\mathcal{Q} + \mathcal{A}\mathcal{P}^*.$$

This formulation shows that $\bar{\mathcal{P}}$ models the transition dynamics of a perturbed policy $\bar{\pi}$, which can be described as a probabilistic mixture of the optimal policy π^* and the exploratory policy $\pi_{\mathcal{Q}}$: $\bar{\pi}(u|x) = (1 - \alpha)\pi_{\mathcal{Q}}(u|x) + \alpha\pi^*(u|x)$[5]. Thus, α acts similarly to $1 - \xi$, determining how much the system follows the optimal policy $\mathcal{P}^*$ versus exploring the exploratory transitions defined by $\mathcal{Q}$. Since $\bar{\mathcal{P}}$ is a weighted combination of the optimal transition matrix $\mathcal{P}^*$ and the exploratory matrix $\mathcal{Q}$, the deviation from the optimal transition matrix is given by: $\|\mathcal{P}^* - \bar{\mathcal{P}}\|_\infty = (1 - \alpha)\|\mathcal{P}^* - \mathcal{Q}\|_\infty$.

Assuming that $\|\mathcal{P}^* - \mathcal{Q}\|_\infty \leq 1$, this deviation is bounded by: $\|\mathcal{P}^* - \bar{\mathcal{P}}\|_\infty \leq 1 - \alpha$. This expression is analogous to the temporal difference error in Q-learning, where α controls the trade-off between exploration and exploitation [16]. As α decreases, the deviation from the optimal transition dynamics increases, reflecting a greater degree of exploration. This trade-off allows the system to balance between following the optimal policy π^* and exploring alternative policies through $\pi_{\mathcal{Q}}$.

The deviation from the optimal transition matrix $\mathcal{P}^*$ is controlled by α, similar to the exploration parameter in Q-learning. While the principles of perturbation in Q-learning and the presented approach share similarities, they are *not interchangeable* in general. Q-learning excels in model-free and discrete setups, while our approach is better suited for structured, large-scale, or robust policy optimization tasks. These differences make them complementary tools in the `RL` framework.

---

[5]The policy $\bar{\pi}$ deviates from π^* to $\pi_{\mathcal{Q}}$ with probability $1 - \alpha$.

VI. NUMERICAL SIMULATIONS

In this section, we compute an estimation of the optimal value function J^* through $\bar{J}$, and hence the parameter vector $\bar{r}$, utilizing the off-policy `TD` approach discussed earlier for an `MDP` related to a resource allocation problem [1], [2]. In summary, the challenge of managing resource pricing for dynamic allocation is tackled by utilizing parallel stochastic Markov chains. In this model, customers request a resource with a probability of λ_i and release it with a probability of μ_i, both of which correspond to the price c_i. The decision-maker's objective is to maximize profit over an infinite time horizon, which is analogous to minimizing costs in the dual formulation of the problem. Denoting by $x_i(k)$ the state of Markov chain (here number of customers) associated to c_i, we can compute the value function as follows:

$$J^*(x) = \max_{u(k) \in \mathcal{U}} \lim_{T \to \infty} \mathbb{E}\left[\sum_{k=0}^{T} \alpha^k \sum_{i=1}^{m} c_i x_i(k)\right],$$

where $\mathcal{U} = \{c_1, c_2, \ldots, c_m\}$. This resource allocation problem exhibits the curse of dimensionality as the number of available resources N and the possible price choice m increase (see [1] for details).

To provide more information and to elaborate more (see [18], [39], [40]), by exploiting Bellman's principle of optimality, the optimal value function for the tail subproblem at time k can be written as

$$J_k^*\big(x(k)\big) = \max_{u(k) \in \mathcal{U}} \mathbb{E}\left[\sum_{k=0}^{T} \sum_{i=1}^{m} \alpha^k c_i x_i(k)\right]$$

$$= \max_{u(k) \in \{c_1, c_2, \ldots, c_m\}} \mathbb{E}\left[\sum_{i=1}^{m} c_i \, x_i(k) + \alpha J_{k+1}^*\big(x(k+1)\big)\right].$$

More specifically, by applying Bellman's recursive backward procedure and starting from the terminal condition

$$J_T\big(x(T)\big) = \sum_{i=1}^{m} c_i \, x_i(T),$$

the optimal value function for such a tail subproblem can be expressed as

$$J_k^*\big(x(k)\big) = \max_{u(k) \in \{c_1, c_2, \ldots, c_m\}} \mathbb{E}\left\{\sum_{i=1}^{m} c_i \, x_i(k)\right.$$

$$\left. + \alpha J_{k+1}^*\big(x(k+1)\big)\right\} = \sum_{i=1}^{m} c_i \, x_i(k) +$$

$$\alpha \max_{u(k) \in \{c_1, c_2, \ldots, c_m\}} \mathbb{E}\left\{J_{k+1}^*\big(x(k+1)\big)\right\}$$

$$= \sum_{i=1}^{m} c_i \, x_i(k) +$$

$$\alpha \max\left[\mathbb{E}\left\{J_{k+1}^*\big(x(k+1)\big) \mid u(k) = c_1\right\}, \ \ldots,\right.$$

$$\left.\mathbb{E}\left\{J_{k+1}^*\big(x(k+1)\big) \mid u(k) = c_m\right\}\right].$$

Finally, $J_0\big(x(0)\big)$, generated at the last step, is equal to the optimal value function $J^*\big(x(0)\big)$ computed for the entire finite time horizon T. Fig. 1 shows an example of the `MDP` associated to a resource allocation problem for a given policy.

As illustrated in Table I as the number of resources N and the number of actions m increase, the cardinality of the state space grows exponentially. For instance, if $m = 5$ and $N = 20$, performing just one

iteration of Bellman's recursion to solve the exact Dynamic Programming (DP) problem entails evaluating 5×53130 action-state pairs (u, x) and computing their associated transition probabilities $\mathcal{P}_{xx'}(u)$, where $\mathcal{P} \colon \mathcal{X} \times \mathcal{U} \times \mathcal{X}$.
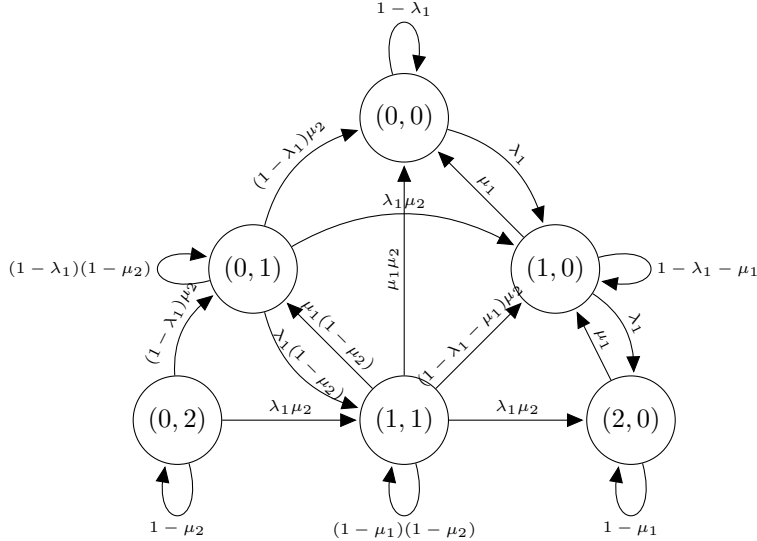

Fig. 1. The graph shows the MDP state space and the corresponding state transition probabilities for a resource allocation problem with $N = 2$ and $m = 2$, and for the control input $u(k) = c_1$. The system states are $(x_1, x_2) \in \{(0,0), (0,1), (1,0), (1,1), (0,2), (2,0)\}$, where x_1 and x_2 are associated to c_1 and c_2, respectively.

TABLE I
DEMONSTRATING THE CURSE OF DIMENSIONALITY IN RESOURCE ALLOCATION PROBLEM

Example of state space explosion [3]		
m	N	*Number of States*
2	10	66
3	20	1 771
5	20	53 130
6	20	230 230
4	50	316 251
5	50	3 478 761

Consider a resource allocation problem with $m = 4$ price levels and $N = 20$ resources. In this case, the cardinality of the state space is $|\mathcal{X}| = 10626$, highlighting the curse of dimensionality in DP problems. In our simulation, each DP iteration requires evaluating 4×10626 state-action pairs and dealing with transition probability matrices of size $10626 \times 4 \times 10626$. This is in addition to saving each iteration's results in a lookup table and comparing different actions u for each state x at every time step k.

Moreover, from the fixed-point equation of the DP operator in the infinite-horizon setting, we have

$$J^* = \mathcal{R} + \alpha \, \mathcal{P}^* J^*.$$

Solving this equation often requires matrix inversion on the order of 10626×10626, illustrating the significant computational burden. As a result, Monte Carlo simulation is used to avoid direct matrix inversion and multiplication for such large-scale problems. Consider a discount factor of $\alpha = 0.9$, with prices $c = [0.8, 0.9, 1, 1.1]$, arrival probabilities $\lambda = [0.090596, 0.048632, 0.015657, 0.005088]$, and service probabilities $\mu = [0.483723, 0.444019, 0.024843, 0.335103]$. These probabilities satisfy the condition: $\max_{c_i, c_j \in \mathcal{U}} \sum_{x'} \left| \mathcal{P}_{xx'}(c_i) - \mathcal{P}_{xx'}(c_j) \right| < (1 - \alpha)$, for all $i, j = 1, \ldots, m$, ensuring stability across all pairs of stationary policies (including the optimal one). We consider 5 features for each state x of MDP, as $\phi_1(x) = 1, \phi_i(x) = x_i, \ i = 1, \ldots, \ m$, hence the parameter vector $r \in \mathbb{R}^5$. We ran 100 Monte Carlo

simulations each starting from an arbitrary initial state with the length of 50000 iterations. In particular for Monte Carlo simulations we employ the procedure that briefly narrated in Remark 2 and Algorithm 1.

The results of these simulations are presented in Fig. 2, where, for simplicity, we have depicted the behavior of $\|\bar{r}_k\|_2$ of parameter vectors across 10 experiments, showing the convergence of the proposed off-policy TD learning (see Lemma 2 and Remark 2). By averaging the parameter vectors resulted from these experiments, we have $\bar{r} = [0.0212,\ 1.9179,\ 2.2853,\ 8.9394,\ 3.3748]^\top$.

After computing $\bar{r}$, one can use the results of the Theorem 4 to approximate J^* for a given state x. In Fig. 2, for the horizontal axis we employ logarithmic scale since the alteration rates of the curves in linear scale are not adequately detectable through iterations.
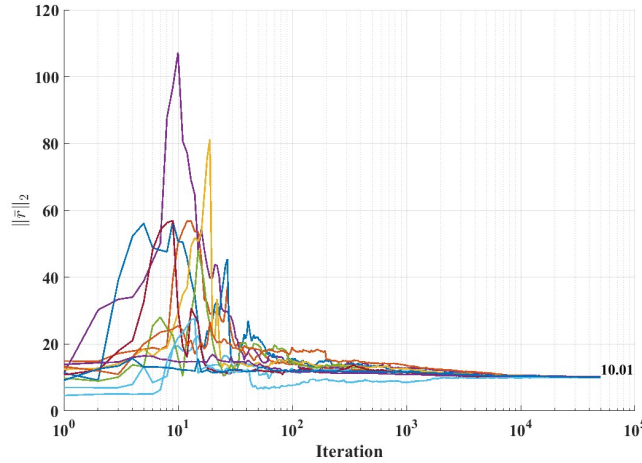

Fig. 2. The behavior of parameter vector r through off-policy TD approach for the resource allocation with $m = 4$ prices, $N = 20$ resources and $|\mathcal{X}| = 10626$.

In the second round of the simulation, we assess the behavior of parameter vector $\bar{r}$ under different target policies as follows:

- *Greedy target policy:* The system selects the action that maximizes the instantaneous revenue, ensuring the highest immediate return at each decision step, i.e. the highest income.
- *Fair target policy:* The system chooses the action that minimizes the instantaneous revenue, i.e. the lowest income.
- *Random target policy:* A decision is selected randomly from the available options, introducing stochasticity into the control process to explore different state transitions.
- *Hybrid Heuristic target policy:* The system uses a probabilistic approach, where with 60% probability, it selects the decision with the lowest value of μ (potentially minimizing service rate impact), and in the remaining 40%, it selects a random action to maintain exploration.
- *Modulo-Based target policy:* The decision is determined based on the modulo operation of the sum of the current state values, enforcing a cyclic or structured selection process that balances system state transitions over time.

The results of these simulations, as depicted in Fig. 3, illustrate the impact of different target policies on the evolution of the parameter vector $\bar{r}$ in the off-policy TD learning process. The plot shows the ℓ_2-norm of $\bar{r}$ over iterations, providing insights into the convergence behavior under varying decision-making strategies. In particular:

(i) Greedy Target Policy (black dashed line): This policy consistently produces the highest parameter values throughout the iterations. Since it always selects the action that maximizes immediate revenue, it leads to a more aggressive exploitation of high-reward states, resulting in higher values of $\|\bar{r}\|_2$. The shaded region represents the performance bounds within which other policies operate.
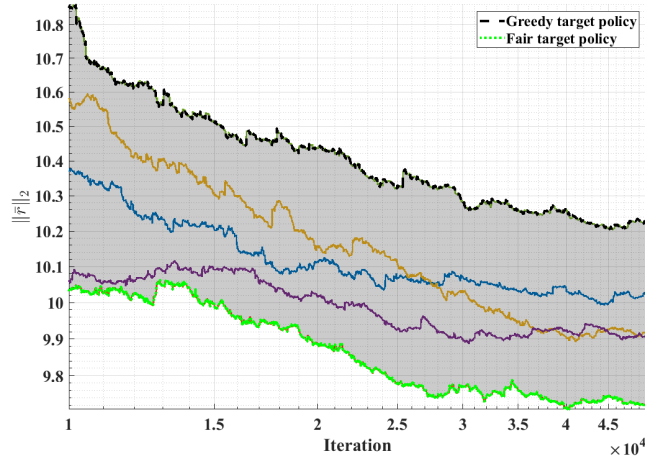
Fig. 3. Comparing different target policies on the parameter vector $\bar{r}$ through off-policy TD approach.

(ii) Fair Target Policy (green dotted line): In contrast, the Fair target policy, which minimizes immediate revenue, exhibits the lowest values of $\|\bar{r}\|_2$ across iterations. This outcome aligns with expectations, as this policy prioritizes decisions that yield lower short-term returns, leading to a more conservative parameter estimation.

Other Policies (colored solid lines), i.e. the Random, Hybrid Heuristic, and Modulo-Based target policies, fall within the bounds set by the Greedy and Fair policies. These policies introduce varying degrees of stochasticity and structure in decision-making: the Random policy (blue) exhibits moderate behavior, as its random selection of actions balances between high and low revenue choices; the Hybrid Heuristic policy (purple) maintains a controlled exploration strategy, sometimes choosing low-service-rate decisions while preserving randomness, leading to a curve closer to the middle range; the Modulo-Based policy (brown) follows a structured cyclic decision rule, causing its trajectory to fluctuate but still remain within the expected range.

Overall, the results of this simulation confirms that the Greedy policy leads to the most aggressive growth in parameter values, while the Fair policy results in the lowest values. The remaining policies provide intermediate trade-offs between exploration and exploitation.

VII. CONCLUSION

In this article, we analyzed the impact of perturbations on optimal policy in DP problems and computed the resulting cost function's deviation from its optimal value. In particular, we handled perturbations in state transition probability matrices using an off-policy TD projection-based approach, which also allowed to address the curse of dimensionality in large-scale MDPs. We also provided the necessary and sufficient conditions for the convergence of the associated Monte Carlo based simulations algorithm. Finally, we validated the presented theoretical results via a suitable numerical example.

APPENDIX

It is worth to highlight that a MATLAB package has been provided to support different implementation phase of this article which is available on https://github.com/Ali-Forootani/Off_Policy_TD_Algorithm.

REFERENCES

[1] A. Forootani, M. Tipaldi, M. G. Zarch, D. Liuzza, and L. Glielmo, "A least-squares temporal difference based method for solving resource allocation problems," *IFAC Journal of Systems and Control*, vol. 13, p. 100106, 2020.

[2] A. Forootani, M. Tipaldi, R. Iervolino, and S. Dey, "Enhanced exploration least-squares methods for optimal stopping problems," *IEEE Control Systems Letters*, vol. 6, pp. 271–276, 2021.

[3] R. Iervolino, M. Tipaldi, and A. Forootani, "A Lyapunov-based version of the value iteration algorithm formulated as a discrete-time switched affine system," *International Journal of Control*, vol. 96, no. 3, pp. 577–592, 2023.

[4] Z. Lin, J. Ma, J. Duan, S. E. Li, H. Ma, B. Cheng, and T. H. Lee, "Policy iteration based approximate dynamic programming toward autonomous driving in constrained dynamic environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5003–5013, 2023.

[5] A. Forootani, R. Iervolino, M. Tipaldi, and S. Baccari, "A kernel-based approximate dynamic programming approach: Theory and application," *Automatica*, vol. 162, p. 111517, 2024.

[6] R. S. Sutton, A. R. Mahmood, and M. White, "An emphatic approach to the problem of off-policy temporal-difference learning," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2603–2631, 2016.

[7] R. S. Sutton, C. Szepesvári, and H. R. Maei, "A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation," *Advances in neural information processing systems*, vol. 21, no. 21, pp. 1609–1616, 2008.

[8] D. P. Bertsekas, "Temporal difference methods for general projected equations," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2128–2139, 2011.

[9] D. Choi and B. Van Roy, "A generalized kalman filter for fixed point approximation and efficient temporal-difference learning," *Discrete Event Dynamic Systems*, vol. 16, no. 2, pp. 207–239, 2006.

[10] M. Geist and O. Pietquin, "Kalman temporal differences," *Journal of artificial intelligence research*, vol. 39, pp. 483–532, 2010.

[11] Y. Engel, S. Mannor, and R. Meir, "Reinforcement learning with gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, pp. 201–208, 2005.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] D. Bertsekas, *A course in Reinforcement Learning*. Athena Scientific, 2023.

[14] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

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

[17] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, PMLR, 2015.

[18] A. Forootani, R. Iervolino, and M. Tipaldi, "Applying unweighted least-squares based techniques to stochastic dynamic programming: Theory and application," *IET Control Theory & Applications*, vol. 13, no. 15, pp. 2387–2398, 2019.

[19] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *Journal of Machine Learning Research*, vol. 11, no. 51, pp. 1563–1600, 2010.

[20] P. Gajane, R. Ortner, and P. Auer, "A sliding-window algorithm for markov decision processes with arbitrarily changing rewards and transitions," *arXiv preprint arXiv:1805.10066*, 2018.

[21] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Hedging the drift: Learning to optimize under non-stationarity," *arXiv preprint arXiv:1903.01461*, 2019.

[22] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Non-stationary reinforcement learning: The blessing of (more) optimism," *Available at SSRN 3397818*, 2019.

[23] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism," *arXiv preprint arXiv:2006.14389*, 2020.

[24] W. Mao, K. Zhang, R. Zhu, D. Simchi-Levi, and T. Başar, "Near-optimal regret bounds for model-free rl in non-stationary episodic MDPs," 2020.

[25] R. Ortner, P. Gajane, and P. Auer, "Variational regret bounds for reinforcement learning," in *Uncertainty in Artificial Intelligence*, pp. 81–90, PMLR, 2020.

[26] O. D. Domingues, P. Ménard, M. Pirotta, E. Kaufmann, and M. Valko, "A kernel-based approach to non-stationary reinforcement learning in metric spaces," *arXiv preprint arXiv:2007.05078*, 2020.

[27] H. Zhou, J. Chen, L. R. Varshney, and A. Jagmohan, "Nonstationary reinforcement learning with linear function approximation," *arXiv preprint arXiv:2010.04244*, 2020.

[28] A. Touati and P. Vincent, "Efficient learning in non-stationary linear markov decision processes," *arXiv preprint arXiv:2010.12870*, 2020.

[29] C.-Y. Wei and H. Luo, "Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach," in *Conference on Learning Theory*, pp. 4300–4354, PMLR, 2021.

[30] H. Zhong, Z. Yang, Z. Wang, and C. Szepesvári, "Optimistic policy optimization is provably efficient in non-stationary MDPs," *arXiv preprint arXiv:2110.08984*, 2021.

[31] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan, "Provably efficient reinforcement learning with linear function approximation," *arXiv preprint arXiv:1907.05388*, 2019.

[32] A. Ayoub, Z. Jia, C. Szepesvari, M. Wang, and L. F. Yang, "Model-based reinforcement learning with value-targeted regression," *arXiv preprint arXiv:2006.01107*, 2020.

[33] D. Zhou, J. He, and Q. Gu, "Provably efficient reinforcement learning for discounted MDPs with feature mapping," *arXiv preprint arXiv:2006.13165*, 2020.

[34] Y. Fei, Z. Yang, Z. Wang, and Q. Xie, "Dynamic regret of policy optimization in non-stationary environments," *arXiv preprint arXiv:2007.00148*, 2020.

[35] D. Bertsekas, *Reinforcement learning and optimal control*, vol. 1. Athena Scientific, 2019.

[36] D. Bertsekas, "Multiagent reinforcement learning: Rollout and policy iteration," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 249–272, 2021.

[37] D. P. Bertsekas *et al.*, "Dynamic programming and optimal control 3rd edition, volume ii," *Belmont, MA: Athena Scientific*, vol. 1, 2011.

[38] A. Forootani, R. Iervolino, M. Tipaldi, and S. Dey, "Transmission scheduling for multi-process multi-sensor remote estimation via approximate dynamic programming," *Automatica*, vol. 136, p. 110061, 2022.

[39] A. Forootani, M. Tipaldi, M. G. Zarch, D. Liuzza, and L. Glielmo, "A least-squares temporal difference based method for solving resource allocation problems," *IFAC Journal of Systems and Control*, vol. 13, p. 100106, 2020.

[40] A. Forootani, M. Tipaldi, M. Ghaniee Zarch, D. Liuzza, and L. Glielmo, "Modelling and solving resource allocation problems via a dynamic programming approach," *International Journal of Control*, vol. 94, no. 6, pp. 1544–1555, 2021.