Reinforcement Learning for Intensity Control: An Application to Choice-Based Network Revenue Management

Huiling Meng [1], Ningyuan Chen [2], Xuefeng Gao [3]

June 8, 2024

Abstract

Intensity control is a type of continuous-time dynamic optimization problems with many important applications in Operations Research including queueing and revenue management. In this study, we adapt the reinforcement learning framework to intensity control using choice-based network revenue management as a case study, which is a classical problem in revenue management that features a large state space, a large action space and a continuous time horizon. We show that by utilizing the inherent discretization of the sample paths created by the jump points, a unique and defining feature of intensity control, one does not need to discretize the time horizon in advance, which was believed to be necessary because most reinforcement learning algorithms are designed for discrete-time problems. As a result, the computation can be facilitated and the discretization error is significantly reduced. We lay the theoretical foundation for the Monte Carlo and temporal difference learning algorithms for policy evaluation and develop policy gradient based actor critic algorithms for intensity control. Via a comprehensive numerical study, we demonstrate the benefit of our approach versus other state-of-the-art benchmarks.

1 Introduction

Many dynamic optimization problems in Operations Research are intensity controls problems, which is a class of problems with continuous time and a discrete state space. Two notable areas are control problems in queueing (Brémaud 1981, Chen and Yao 1990) and dynamic pricing/assortment problems (Gallego and Van Ryzin 1997, Strauss et al. 2018, Gallego et al. 2019) in revenue management. Although both areas have been studied extensively in the literature, it is fair to say that most problems are still challenging to solve in practice, due to a large number of states. In dynamic pricing and assortment, for example, the possible combinations of the remaining inventory of the products/resources make the state space impractically large and render the exact optimal solutions extremely difficult.

Meanwhile, reinforcement learning (RL) provides a computational framework to solve general dynamic optimization problems that can be formulated as Markov decision processes (MDPs). For a comprehensive introduction to RL, see Sutton and Barto (2018). A prototypical problem that can be solved by RL is tabular MDPs: There is a finite state space, a finite action space, and a discrete time horizon.

[1]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, China. Email: hmeng@se.cuhk.edu.hk.

[2]Rotman School of Management, University of Toronto, Toronto, Canada. Email: ningyuan.chen@utoronto.ca.

[3]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, China. Email: xfgao@se.cuhk.edu.hk.

Faced with intensity control problems, one may be tempted to convert such problems to tabular MDPs and then apply RL algorithms. One conspicuous discrepancy between intensity control and tabular MDPs is whether the time horizon is continuous or discrete. In fact, the continuous time horizon is the defining feature of intensity control problems. For the conversion, one may approximate continuous-time stochastic processes with discrete-time ones. For example, arrivals of customers following a Poisson process are widely used and driving the dynamics of the two areas mentioned above. They can be approximated by a single arrival in a period having a Bernoulli distribution when the time horizon is discretized with a sufficiently refined grid. This type of discretization scheme is usually carried out before the RL algorithm is executed with a uniform and pre-specified grid size.

As for the choice of the discretization grid size, one can clearly see the computational trade-off. On one hand, an accurate approximation of continuous-time processes requires a fine grid. This point can be perfectly illustrated by the approximation of Poisson processes: the grid size Δt needs to be sufficiently small so that it is unlikely to have more than one arrival during a period of Δt. As such, the dynamics can be discretized and the probability of an arrival in a time period has a Bernoulli distribution. On the other hand, if a time step in the discrete-time system corresponds to a minuscule duration in the continuous-time system, then the computational cost is high because of the inflated length of horizon and it may lead to numerical instabilities. To make things worse, there is not guideline on how to choose a proper discretization scheme and the trade-off cannot be evaluated beforehand. In practice, one may experiment RL algorithms on a set of diminishing grid sizes and inspect if the obtained solutions have converged as the gird becomes finer. The computational cost is prohibitively high because a sequence of increasingly challenging problems have to be solved, not to mention that the convergence may not even be warranted at the first place. Indeed, it is known in the RL community that the performance of RL algorithms can be very sensitive with respect to the discreteization grid size; see, e.g., Tallec et al. (2019) in which it is empirically shown that standard Q-learning methods are not robust to changes in time discretization of continuous-time control problems.

In this study, via the classical application of choice-based network revenue management (see Strauss et al. 2018 for a recent review), we provide a framework to implement RL algorithms for intensity control problems, *without the need to discretize the time horizon upfront.* The key insight is that for each sample path of the system generated under a given policy, it is inherently discretized by the jump times associated with the sample path. For example, in the focused application, the state (remaining inventory of the resources) changes and the reward is generated only when a customer arrives. During the time horizon, such arrivals, although occurring at different time points across the sample paths and thus cannot be pre-determined, are finite and typically substantially more coarse (for a given sample path) than what would be required in the naïve discretization scheme mentioned above. Moreover, if the basis functions are chosen properly for the value function and policy function approximations, then the RL algorithm can be implemented without any approximation errors associated with discretization. This offers a huge benefit compared to the naïve discretization: even when the trade-off has been optimally balanced in the latter, the discretization errors cannot be eliminated. We summarize the contribution of the study below.

- We adapt policy evaluation (including Monte Carlo and Temporal Difference methods) and policy gradient in the standard RL framework to the continuous-time intensity control setting without upfront discretization of the time horizon. We then combine them and develop Actor-Critic algorithms. Compared to the naïve procedure that first discretizes the time horizon and then applies the standard RL algorithms for MDPs, our approach has two major strengths. First, leveraging the inherent discretization of the jump points, we show that the adapted RL algorithms can be implemented largely free of discretization errors. This avoids the numerical instability and convergence checking of the naïve discretization. Second, it is more computationally efficient, as the jump points of a sample path are typically more sparse than a refined discretization scheme. In particular, we do not need to consider the union of the jump points of all sample paths.

- We extend the martingale approach for policy evaluation, originally proposed in Jia and Zhou (2022a,b) for entropy-regularized RL in controlled diffusion processes, to intensity control problems with discrete states. In particular, we show that the value function along the state process combined with the accumulated reward and the entropy bonus is a martingale (see Theorem 2). This martingale property not only leads to a natural loss function for Monte Carlo policy evaluation, but also martingale orthogonality conditions that form the basis of continuous-time Temporal Difference methods for intensity control. For policy gradient, we extend the results in Jia and Zhou (2022b) and show that computing the gradient of the value function with respect to a given parameterized stochastic policy is equivalent to a policy evaluation problem with an auxiliary reward function. This allows us to estimate the policy gradient using observable samples and current estimates of value functions for the intensity control problem.

- We conduct a comprehensive numerical experiment to compare the performance of the proposed Actor-Critic algorithm and benchmarks as well as the state-of-the-art algorithms in the literature, including the greedy policy, the CDLP policy (Liu and Van Ryzin 2008), the ADP policy (Zhang and Adelman 2009), and the optimal dynamic programming policy with refined time discretization. We have the following empirical findings: Overall, the performance of the proposed RL algorithm adapted to intensity control is among the best, despite the fact that it is the only policy that does not need to know the environment and has to learn it through simulated samples. The ADP policy from Zhang and Adelman (2009), when the time horizon is discretized properly, has a similar performance. But its performance may be unstable and non-monotone with respect to the decreasing size of the grid. Moreover, the computation time of the RL algorithm doesn't scale rapidly with respect to the problem size. The numerical experiments show encouraging signs of the proposed algorithm to be deployed in practice.

Below we discuss the connection of this work to the literature.

1.1 Literature Review

The network revenue management problem (Gallego and Van Ryzin 1997) is one of the classical problems in revenue management that has been studied by numerous papers. Its choice-based variants have been proposed and studied by Gallego et al. (2004), Talluri and Van Ryzin (2004), Zhang and Cooper (2005), Liu and Van Ryzin (2008), Zhang and Adelman (2009), Zhang (2011) and many subsequent papers. See Strauss et al. (2018) for a review. As a dynamic optimization problem, the discrete-time version can be formulated as an MDP. However, to solve the optimal policy, even numerically, is essentially infeasible due to the exponentially large state and action spaces. The focus of the literature has been to provide efficient algorithms, usually with provable performance guarantees, that solve the problem approximately. The objective of this study is to use the choice-based network revenue management as a case study and show how to adapt the RL framework to the continuous time, because RL algorithms have been shown to have impressive empirical performance for a large class of practical problems. We note that in this literature, a number of studies including Zhang and Adelman (2009), Ma et al. (2020) design algorithms based approximate dynamic programming (ADP), which is an important concept and approach in RL. However, they focus on the discrete-time formulation and value function approximation, while we study the continuous-time formulation and general RL algorithms including exploration and the policy gradient method. Moreover, our theoretical results are not focused on the performance guarantee but the foundation and well-posedness of RL algorithms in the continuous time. In the numerical experiments, we compare our algorithm to two important benchmarks in the literature (Liu and Van Ryzin 2008, Zhang and Adelman 2009).

In terms of methodology, our paper builds on a series of recent studies (Wang et al. 2020, Jia and Zhou 2022b,a) on continuous-time reinforcement learning with *continuous* state and action spaces. In particular, the stochastic processes driving the system are controlled diffusion processes, and the reward is continuously accrued over time in their models. By contrast, we consider continuous-time reinforcement learning for intensity control of point processes with piecewise constant sample paths, where both the state and action spaces are discrete, and the reward is collected only at jump times. This leads to several subtle yet significant differences in our theoretical analysis and algorithm design, which we elaborate below. Firstly, to facilitate a theoretical analysis of value functions under stochastic policies, (Wang et al. 2020, Jia and Zhou 2022b) derived the so-called exploratory state process by applying a law of large number argument to the drift and diffusion coefficients of the controlled diffusion process. This approach does not apply to intensity control, so we instead derive the exploratory dynamics based on analyzing the infinitesimal generator of the sample state process. Second, given that instantaneous rewards in our setting may arise only at times when customers arrive, we aim to generate actions only at these specific times, rather than continuously throughout the time horizon as in Jia and Zhou (2022b). Third, while our general framework extends the martingale approach proposed in Jia and Zhou (2022a,b) to intensity control, there are also some important differences in the specific formulas and implementations for policy evaluation (PE) and policy gradient (PG). A notable distinction arises from the treatment of integrals with respect to dt for functions of time, state and randomized action. These functions have been refined by taking the average over the actions, yielding integrands that depend solely on time and state. This can help reduce the variance of the proposed RL algorithms and avoid generating actions at each point along a pre-determined discretization grid of the horizon as in Jia and Zhou (2022b). Moreover, because the paths of the sample state process in our problem are piecewise constant, we propose an adaptive discretization approach that takes into account the jump times of each sample trajectory to compute the aforementioned integrals. This is in sharp contrast to Jia and Zhou (2022a,b) where integrals are computed by discretizing the horizon uniformly. Our strategy is expected to significantly reduce the approximation errors that often arise with regular uniform discretization scheme. We also mention a concurrent work (Gao et al. 2024) which study RL for general jump-diffusion processes. Their focus is to develop q-learning algorithms, the continuous-time counterpart of Q-learning, for jump-diffusions, whereas our focus is to develop PE and PG based actor critic methods tailored for intensity control. Note that a few recent RL studies have been focusing on the discretization of the (continuous) state space (Sinclair et al. 2020, 2023), whereas this study considers intensity control with a discrete space and the inherent discretization of the (continuous) time horizon by jump points.

The intensity control problem that we study can be viewed as a special class of continuous-time MDPs or the more general semi-Markov decision processes (SMDPs, Puterman (2014)) with discrete state spaces. Several RL algorithms were developed for infinite-horizon continous-time MDPs as well as for SMDPs very early on (Bradtke and Duff 1995, Das et al. 1999). In terms of theoretical results, Gao and Zhou (2022a) recently establish logarithmic regret bounds for learning tabular continuous-time MDPs in the infinite-horizon average-reward setting. Gao and Zhou (2022b) establish regret bounds for continous-time MDPs in the finite-horizon episodic setting. By contrast, we develop model-free RL algorithms for the finite-horizon network revenue management problem without considering regret bounds. Besides RL for continuous-time MDPs with discrete spaces, there is also a surge of interest in studying continuous-time RL for controlled diffusion processes and its applications (mostly in finance), see, e.g., Wang and Zhou (2020), Guo et al. (2022), Wang et al. (2023), Jia and Zhou (2023), Zhao et al. (2024), Wu and Li (2024), Dai et al. (2023). In contrast to these studies, we study continuous-time RL for intensity control problems with discrete state spaces and focus on developing RL algorithms for the choice-base network revenue management problem.

Finally, we mention a growing body of literature on RL algorithms applied to Operations Management. Dai and Gluzman (2022) develop proximal policy optimization methods for queueing network control problems with a long-run average cost objective. Gijsbrechts et al. (2022) demonstrate that the RL algorithm can match the performance of the state-of-the-art policies in inventory management, although tuning the hyperparameters for instances is needed. See Oroojlooyjadid et al. (2022), Li et al. (2023), Azagirre et al. (2024) for other applications.

2 Problem Formulation

We consider the network revenue management problem with m resources and n products. The consumption matrix is given by $A := [a_{ij}]_{m\times n}$. The entry a_{ij} represents the amount of resource i used by selling one unit of product j, making the jth column A^j of A the incidence vector for product j. Let $\mathcal{J} = \{1, \dots, n\}$ be the set of products, with fixed prices denoted by $p = (p_1, \dots, p_n)^\top$.

We consider a continuous-time finite selling horizon $[0, T]$. The initial inventory of the resources is denoted by $c = (c_1, \dots, c_m)^\top$. Consumers arrive according to a Poisson process with rate λ.[4] Upon arrival, based on the assortment offered by the firm $S \subseteq \mathcal{J}$ at the moment, the customer makes a choice $j \in S \cup \{0\}$. For convenience, We denote $\mathcal{A}$ as the collection of all subsets of $\mathcal{J}$ and thus $S \in \mathcal{A}$. We use 0 to represent the no-purchase option. The choice behavior is typically captured by the choice probability $P_j(S) \in [0, 1]$. In other words, the customer purchases product j

[4]Our framework can be easily extended to nonstationary arrival rates. In the study, for simplicity, we focus on stationary arrivals. The same stationary setup is adopted for the choice probabilities introduced below.

with probability $P_j(S)$ when the offered assortment is S. The choice probabilities are fixed over time and satisfy the standard regularity conditions such as $P_j(S) = 0$ for $j \notin S$ and $\sum_{j \in S \cup \{0\}} P_j(S) = 1$, although they may be unknown for the RL algorithms. The firm's decision problem is to find a dynamic policy that offers assortment S_t at time t that maximizes the expected total revenue over the selling horizon $[0, T]$.

We briefly discuss why we focus on the continuous-time setting of the problem at the first place, because many other studies in revenue management (for example, Zhang and Adelman 2009) start with the discrete-time setting in which at most one customer may arrive in a time period. (Note that we are not referring to the naïve discretization of the continuous-time formulation, but the discrete-time formulation of the problem itself.) We choose the continuous-time setting mainly to illustrate the design and implementation of the RL algorithm. Moreover, although it is well expected that the discrete-time formulation is a good approximation of the continuous-time formulation in practice, to our knowledge, there are no theoretical results characterizing the gap between their value functions and optimal policies. Therefore, we believe there are theoretical and practical values in demonstrating how to adapt the RL algorithms to the continuous-time formulation.

2.1 Classical Formulation of Optimal Intensity Control

In this section, we formulate the problem in the language of optimal intensity control. A control process can be represented as $\boldsymbol{S} = \{S_t \in \mathcal{A} : 0 \leq t \leq T\}$, where S_t specifies the firm's offered set at time t. Given a control process $\boldsymbol{S}$, let $N_t^{\boldsymbol{S}} = (N_{1,t}^{\boldsymbol{S}}, \dots, N_{n,t}^{\boldsymbol{S}})^\top$ be a vector of controlled Poisson processes with intensities $(\lambda P_1(S_t), \dots, \lambda P_n(S_t))$, defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_t^{N^\lambda}\}_{t\geq 0})$ along with a Poisson process with rate $\lambda : N^\lambda = \{N_t^\lambda : 0 \leq t \leq T\}$. One can interpret $N^\lambda = \{N_t^\lambda : 0 \leq t \leq T\}$ as the arrival process of all potential consumers with rate λ and $N_t^{\boldsymbol{S}}$ as the cumulative number of the n products sold by time t under the control $\boldsymbol{S}$. The remaining inventory of the resources at time t is represented by $X_t^{\boldsymbol{S}} = c - AN_t^{\boldsymbol{S}}$. Let $\mathcal{S} = \{0, \dots, c_1\} \times \cdots \times \{0, \dots, c_m\}$ be the state space of $X_t^{\boldsymbol{S}}$. Given a control process $\boldsymbol{S}$ generated by a deterministic function z as $S_t = z(t, X_{t-}^{\boldsymbol{S}})$, the process $X_t^{\boldsymbol{S}} \in \mathcal{S}$ is a continuous-time Markov chain. In particular, for $(t, x, S) \in [0, T] \times \mathcal{S} \times \mathcal{A}$, the controlled transition rates of $X_t^{\boldsymbol{S}}$ are given by

$$q(y \mid t, x, S) = \sum_{\{j \in \mathcal{J} : A^j = x - y\}} \lambda P_j(S), \ \forall\, y \neq x; \quad q(x \mid t, x, S) = -\lambda[1 - P_0(S)]. \tag{1}$$

The state x can only transition to state y if a product j consumes an array of resources $A^j = x - y$. Consider $\mathcal{U}$ to be the set of all non-anticipating control processes, which satisfies $\int_0^T A dN_t^{\boldsymbol{S}} \leq c$, $\mathbb{P}$-a.s. Then, for a policy $\boldsymbol{S} \in \mathcal{U}$, the expected total revenue is given by

$$V(0, c; \boldsymbol{S}) := \mathbb{E}^{\mathbb{P}}\left[\int_{(0,T]} p^\top dN_t^{\boldsymbol{S}}\right] = \mathbb{E}^{\mathbb{P}}\left[\int_0^T r(S_t) dt\right],$$

where $r(S) := \lambda \sum_{j=1}^n p_j P_j(S)$ for all $S \in \mathcal{A}$. The value function, denoted by $V(t, x; \boldsymbol{S})$, calculates the expected revenue during the time interval $(t, T]$ given that the vector of remaining inventory at time t is x:

$$V(t, x; \boldsymbol{S}) := \mathbb{E}^{\mathbb{P}}\left[\int_{(t,T]} p^\top dN_s^{\boldsymbol{S}} \mid X_t^{\boldsymbol{S}} = x\right].$$

The goal of this intensity control problem is to find a control $\boldsymbol{S}^* \in \mathcal{U}$ which achieves $V^*(t,x) = \sup_{\boldsymbol{S}\in\mathcal{U}} V(t,x;\boldsymbol{S})$ for all $(t,x) \in [0,T]\times\mathcal{S}$. From the optimal control theory, the optimal value function $V^*(t,x)$ satisfies the following HamiltonJacobiBellman (HJB) equation

$$\begin{cases} \dfrac{\partial V^*}{\partial t}(t,x) + \max\limits_{S\in\mathcal{A}(x)} H(t,x,S,V^*(\cdot,\cdot)) = 0, \quad (t,x)\in[0,T]\times\mathcal{S}, \\ V^*(T,x) = 0, \quad x\in\mathcal{S}, \end{cases} \tag{2}$$

where $\mathcal{A}(x) := \{S \in \mathcal{A} : x \geq A^j \text{ for all } j \in S\}$ denotes the collection of all available assortments at the state x, and the *Hamiltonian* $H : [0,T]\times\mathcal{S}\times\mathcal{A}\times C^{1,0}([0,T]\times\mathcal{S}) \mapsto \mathbb{R}$ is defined as:

$$H(t,x,S,v(\cdot,\cdot)) = r(S) + \sum_{y\in\mathcal{S}} v(t,y)q(y \mid t,x,S). \tag{3}$$

The space $C^{1,0}([0,T]\times\mathcal{S})$ consists of all real-valued functions defined on $[0,T]\times\mathcal{S}$ that are continuously differentiable in t over $[0,T]$ for all $x\in\mathcal{S}$.

We note that optimal control problem (2) is challenging to solve, both analytically and computationally. First, the continuous time horizon generally has to be discretized in order to obtain a numerical solution. The discretization scheme needs to be carefully designed to avoid instability and guarantee convergence. Unfortunately, there are no general guidelines and the practice is rather ad hoc depending on the application. Second, the state and action spaces of the problem are of the sizes $O(\|c\|_1^m)$ and $O(2^n)$, respectively. It is virtually impossible to solve the problem exactly for a medium m or n. Third, in practice, the choice probabilities $P_j(S)$ that determine the transition rates $q(\cdot)$ are typically unknown to the firm and has to be learned through the collected data. In the next section, we restate the problem using the language of RL, which provides a computational framework that mitigates the issues above in practice.

2.2 Formulation of Reinforcement Learning

In this study, we focus on policy-based reinforcement learning. To start, we consider the following policy class, following Definition 2.1 in Chapter 2 of Guo et al. (2009).

Definition 1. *A randomized Markov policy is a real-valued function $\boldsymbol{\pi}(S \mid t,x)$ that satisfies*

(i) For all $(x,S)\in\mathcal{S}\times\mathcal{A}$, the mapping $t\mapsto\boldsymbol{\pi}(S\mid t,x)$ is measurable on $[0,T]$.

(ii) For all $(t,x)\in[0,T]\times\mathcal{S}$, $\boldsymbol{\pi}(\cdot\mid t,x)$ is a probability distribution on the action space $\mathcal{A}$.

Moreover, a randomized Markov policy $\boldsymbol{\pi}(\cdot\mid\cdot,\cdot)$ is called admissible if it further satisfies

(iii) For all $(t,x)\in[0,T]\times\mathcal{S}$, it holds that $\boldsymbol{\pi}(S\mid t,x)=0$ if $S\notin\mathcal{A}(x)$;

(iv) For all $(x,S)\in\mathcal{S}\times\mathcal{A}$, the mapping $t\mapsto\boldsymbol{\pi}(S\mid t,x)$ is continuous on $[0,T]$.

We denote by Π the set of admissible randomized Markov policies. When the context is clear, we simply use $\boldsymbol{\pi}(S\mid t,x)$ to denote the probability of choosing S as the offered assortment in state

(t, x). Note that the action randomization in the stochastic policy $\boldsymbol{\pi}(\cdot \mid \cdot, \cdot)$ is independent of the customer arrival process N^λ discussed in Section 2.1.

In the RL formulation, instead of solving (2) in the hope of obtaining the optimal (deterministic) policy, we consider a class of randomized policies that may choose the offered assortment at t according to some probability distribution over feasible assortments. Such randomized policies encourage exploration of actions and states that are "suboptimal" in the current iteration, which is a key principle of algorithmic design in reinforcement learning. The exploratory policies can help collect data and gradually refine the approximation of the environment. In the meantime, the policy can improve itself and converge to the optimal policy if the RL algorithm is properly designed.

We will specify the choice of the parametric family of policies within the class of admissible randomized Markov policies in Section 6. Generally, the choice needs to satisfy the following two conditions. First, the family is flexible enough so that it should include the optimal policy as a member or at least be able to approximate it. This condition allows the reinforcement learning to converge to a near-optimal policy over time. The crafting of such policy family usually depends on the problem context. Second, the randomness can be tuned by a parameter so that one can control the degree of exploration, depending on the phase of the algorithm. For example, in the end of reinforcement learning, the algorithm can easily turn off or reduce exploration to generate a near-optimal policy from Π.

Given $\boldsymbol{\pi} \in \Pi$, we consider the filtered probability space $(\Omega, \mathcal{F}, \bar{\mathbb{P}}; \{\mathcal{F}_t\}_{t \geq 0})$, where $\mathcal{F}_t = \sigma\{(N_s^{\boldsymbol{\pi}}, S_s^{\boldsymbol{\pi}}) : 0 \leq s \leq t\}$ and the probability measure $\bar{\mathbb{P}}$ is defined on $\mathcal{F}_T$. Let $\mathbb{E}^{\bar{\mathbb{P}}}$ be its corresponding expectation operator. Note that because of the additional randomness introduced in the randomized policy, $\bar{\mathbb{P}}$ is different from $\mathbb{P}$ defined in Section 2.1 and $\bar{\mathbb{P}}$ can be viewed as an extension of $\mathbb{P}$. The value function of $\boldsymbol{\pi}$ is given by

$$V(t, x; \boldsymbol{\pi}) = \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_{(t,T]} p^\top dN_s^{\boldsymbol{\pi}} \mid X_t^{\boldsymbol{\pi}} = x\right]. \tag{4}$$

To encourage exploration, we follow Jia and Zhou (2022b) and introduce the entropy to measure the randomness of a stochastic policy $\boldsymbol{\pi}$. For all $\boldsymbol{\pi} \in \Pi$ and $(t, x) \in [0, T] \times \mathcal{S}$, denote the entropy of $\boldsymbol{\pi}(\cdot \mid t, x)$ by

$$\mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, x)) \coloneqq -\sum_{S \in \mathcal{A}} \boldsymbol{\pi}(S \mid t, x) \log \boldsymbol{\pi}(S \mid t, x).$$

Then, we add the entropy as a bonus to the original value function (4), leading to

$$\begin{aligned} J(t, x; \boldsymbol{\pi}) &= \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_{(t,T]} p^\top dN_s^{\boldsymbol{\pi}} + \gamma \int_t^T \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))ds \mid X_t^{\boldsymbol{\pi}} = x\right] \\ &= \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_t^T r(S_s^{\boldsymbol{\pi}})ds + \gamma \int_t^T \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))ds \mid X_t^{\boldsymbol{\pi}} = x\right], \end{aligned} \tag{5}$$

where $\gamma \geq 0$ is referred to as the temperature parameter and controls the degree of exploration. Such entropy regularization is a commonly used technique to improve exploration in RL, see also Haarnoja et al. (2018).

For the convenience of theoretical analysis, we introduce a Markov process $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$, defined on the original probability space $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_t^{N^\lambda}\}_{t\ge 0})$ that averages out the randomness in the action/policy. The process $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ will be referred to as the exploratory state process. It is equivalent to the sample state process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ (defined on $(\Omega, \mathcal{F}, \bar{\mathbb{P}}; \{\mathcal{F}_t\}_{t\ge 0})$) in the sense that the distribution (law) of $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ under $\mathbb{P}$ is the same as the distribution of $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ under $\bar{\mathbb{P}}$. This allows us to rewrite

$$J(t,x;\boldsymbol{\pi}) = \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_t^T \left\{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, X_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))\right\} ds \mid X_t^{\boldsymbol{\pi}} = x\right] \tag{6}$$

$$= \mathbb{E}^{\mathbb{P}}\left[\int_t^T \left\{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}))\right\} ds \mid \tilde{X}_t^{\boldsymbol{\pi}} = x\right]. \tag{7}$$

Note that by taking expectation with respect to the action randomization, the expectation in (5) reduces to the expectation in (6). In (7), we use the original probability space $\mathbb{P}$ to express the value function, which is easier to work with because the randomness in the policy introduces complexity to the analysis of value functions. Interested readers may find details about the reformulation in Appendix A. We also emphasize that unlike the sample state process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$, the exploratory dynamics $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ is not observable, and hence its trajectories will not be used in our RL algorithm design.

For a given randomized Markov policy $\boldsymbol{\pi}$, its value function $J(t, x; \boldsymbol{\pi})$ can be characterized by a differential equation. The proof of Lemma 1 is deferred to Appendix C.

Lemma 1. *A function $v \in C^{1,0}([0,T] \times \mathcal{S})$ is the value function associated with the stochastic policy $\boldsymbol{\pi}$, i.e., $v(t,x) = J(t,x;\boldsymbol{\pi})$ for all $(t,x) \in [0,T] \times \mathcal{S}$, if and only if it satisfies the following differential equation:*

$$\frac{\partial v}{\partial t}(t,x) + \sum_{S\in\mathcal{A}} H(t,x,S,v(\cdot,\cdot))\boldsymbol{\pi}(S \mid t,x) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid t,x)) = 0, \quad (t,x) \in [0,T) \times \mathcal{S}, \tag{8}$$

with the terminal condition $v(T,x) = 0,\ x \in \mathcal{S}$.

The task of RL is to find a policy $\boldsymbol{\pi}^* \in \Pi$ which attains $J^*(t,x) = \sup_{\boldsymbol{\pi}\in\Pi} J(t,x;\boldsymbol{\pi})$ for all $(t,x) \in [0,T] \times \mathcal{S}$. Compared with the original problem, the optimal policy $\boldsymbol{\pi}^*$ of RL can be characterized by a Boltzmann (or softmax) distribution:

$$\boldsymbol{\pi}^*(S \mid t,x) = \frac{\exp\{\frac{1}{\gamma}H(t,x,S,J^*(\cdot,\cdot))\}}{\sum_{\bar{S}\in\mathcal{A}(x)} \exp\{\frac{1}{\gamma}H(t,x,S,J^*(\cdot,\cdot))\}}, \tag{9}$$

where the Hamiltonian $H(\cdot)$ is introduced in (3). A detailed proof of this characterization is provided in Appendix C. It is clear that even at optimality, the exploration parameter γ encourages offering assortments randomly, although it is more likely to sample the assortments with a higher Hamiltonian in the original problem.

We focus on two objectives of RL in this study. The first objective is policy evaluation (PE): for a given policy $\boldsymbol{\pi} \in \Pi$, PE aims at the employment of a (numerical) procedure to determine

$J(t, x; \boldsymbol{\pi})$ as a function of (t, x) without any knowledge of the customer arrival rate and choice probabilities. This is presented in Section 3. The second objective is policy improvement, in particular, the well-established policy gradient (PG) method: we attempt to estimate the policy gradient $\nabla_\phi J(0, c; \boldsymbol{\pi}^\phi)$ within a suitably chosen parametric family $\{\boldsymbol{\pi}^\phi : \phi \in \Phi\}$. We also require this method to operate solely on observable data, as well as the learned value function of $\boldsymbol{\pi}^\phi$, in the absence of the environmental parameters. This is presented in Section 4. Finally in Section 5, we will combine PE and PG in an iterative manner, which leads to actor-critic algorithms and allows us to leverage the power of RL for the choice-based network revenue management problem.

3 Policy Evaluation

Recall that the objective of PE is to estimate the value function $J(t, x; \boldsymbol{\pi})$ of a given policy $\boldsymbol{\pi}$ using samples, generally without the knowledge of the environment. The samples are the trajectories of the form $\{(t, N_t^{\boldsymbol{\pi}}, X_t^{\boldsymbol{\pi}}, S_t^{\boldsymbol{\pi}}, r_t^{\boldsymbol{\pi}}) : 0 \leq t \leq T\}$ generated under policy $\boldsymbol{\pi}$, where $r_t^{\boldsymbol{\pi}}$ indicates the instantaneous reward at time t, equal to p_j upon the sale of product j and 0 otherwise. In practice, the samples can be obtained from the observed sales data. We aim to evaluate $J(t, x; \boldsymbol{\pi})$ as a function of (t, x). It is often achieved through function approximation, where $J(t, x; \boldsymbol{\pi})$ is approximated by a parametric family of functions $\{J^\theta(t, x) : \theta \in \Theta\}$. The particular form of $\{J^\theta(t, x) : \theta \in \Theta\}$ for the network revenue management problem will be discussed later.

Before delving into the proposed PE method, we refresh the memory of readers by reflecting on how PE is usually conducted in the standard discrete-time RL setting and then drawing the connection by demonstrating how we can discretize the time horizon in our problem and convert it to the discrete-time setting. Specifically, suppose we discretize the time horizon $0 = t_0 < t_1 < \cdots < t_K = T$ with an equal length $\Delta t = \frac{T}{K}$ upfront and denote the corresponding discrete-time model as $\mathcal{M}_{\Delta t}$. For small Δt, it approximately holds that in each discrete period $(t_k, t_{k+1}]$, there is at most one customer arrival: one arrival occurs with probability $\lambda \Delta t$, and no arrivals occur with probability $1 - \lambda \Delta t$. In the PE step, the goal is to evaluate the approximate value function $J_{\Delta t}(t_k, x; \boldsymbol{\pi})$.

For a continuous-time trajectory $\{(t, N_t^{\boldsymbol{\pi}}, X_t^{\boldsymbol{\pi}}, S_t^{\boldsymbol{\pi}}, r_t^{\boldsymbol{\pi}}) : 0 \leq t \leq T\}$, we naturally use $X_{t_k}^{\boldsymbol{\pi}}$ and $S_{t_k}^{\boldsymbol{\pi}}$ as the state and action for the discretized system. For the reward, we denote $r_{(t_k, t_{k+1}]}$ as the realized reward between t_k and t_{k+1} in the continuous system, which also represents the reward at $(t_k, X_{t_k}^{\boldsymbol{\pi}})$ in the discrete system. It allows us to convert the trajectory to a sample path that can be readily used in the discrete-time RL.

In discrete-time RL, Monte Carlo and Temporal Difference (TD) methods are two most common techniques for PE. While the Monte Carlo methods are suited for offline learning, TD methods work both online and offline. We first show the *gradient Monte Carlo* method (see Chapter 9 in (Sutton and Barto 2018)), which updates θ using

$$\theta \leftarrow \theta + \alpha \sum_{k=0}^{K-1} \nabla_\theta J_{\Delta t}^\theta(t_k, X_{t_k}^{\boldsymbol{\pi}}) \left(\sum_{k'=k}^{K-1} \left(r_{(t_{k'}, t_{k'+1}]} + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t_{k'}, X_{t_{k'}}^{\boldsymbol{\pi}})) \Delta t \right) - J_{\Delta t}^\theta(t_k, X_{t_k}^{\boldsymbol{\pi}}) \right), \quad (10)$$

where α is the learning rate and the term $\mathcal{H}(\boldsymbol{\pi}(\cdot \mid t_k, X_{t_k}^{\boldsymbol{\pi}}))$ represents the exploration/entropy bonus. To interpret (10) at a high level, note that the underlying loss function that the gradient

Monte Carlo algorithm seeks to minimize can be formulated as

$$L_{\Delta t}(\theta) = \frac{1}{2}\mathbb{E}^{\bar{\mathbb{P}}}\left[\sum_{k=0}^{K-1}\left(\sum_{k'=k}^{K-1}\Big(r_{(t_{k'},t_{k'+1}]} + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid t_{k'}, X^{\boldsymbol{\pi}}_{t_{k'}}))\Delta t\Big) - J^{\theta}_{\Delta t}(t_k, X^{\boldsymbol{\pi}}_{t_k})\right)^2\right], \tag{11}$$

where the difference term captures the deviation of the estimated value function from the realized reward (and the exploration bonus) aggregated for each time step along the sample path. In contrast to the Monte Carlo methods, which uses the whole trajectory to update θ, the TD methods, when used online, update the estimate of the value function at each discrete time point. For instance, the online TD(0) algorithm (with function approximation) updates θ at every time step k using the following formula:

$$\theta \leftarrow \theta + \alpha\nabla_\theta J^{\theta}_{\Delta t}(t_k, X^{\boldsymbol{\pi}}_{t_k})\Big(r_{(t_k,t_{k+1}]} + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid t_k, X^{\boldsymbol{\pi}}_{t_k}))\Delta t + J^{\theta}_{\Delta t}(t_{k+1}, X^{\boldsymbol{\pi}}_{t_{k+1}}) - J^{\theta}_{\Delta t}(t_k, X^{\boldsymbol{\pi}}_{t_k})\Big), \tag{12}$$

where the TD error characterizes the difference between the estimated value of the current state and the estimated value of the subsequent state plus the realized reward associated with the transition from $X^{\boldsymbol{\pi}}_{t_k}$ to $X^{\boldsymbol{\pi}}_{t_{k+1}}$. For more details, we refer the readers to Chapter 9 in Sutton and Barto (2018).

It is mostly expected and intuitive that as $\Delta t \to 0$, the approximate value function $J_{\Delta t}$ converges to J. While this is the reason why time discretization is so widely used in practice, to our knowledge, there is no framework that universally guarantees the convergence and stability as $\Delta t \to 0$. Moreover, the computation can become an issue when Δt is small and there is no guideline on how to choose Δt considering the trade-off of computational efficiency and convergence. To overcome the challenges, we propose to evaluate $J(t, x; \boldsymbol{\pi})$ for a given policy $\boldsymbol{\pi}$ directly without upfront time discretization. In this study, we introduce two PE methods in the continuous-time setting: one parallels the gradient Monte Carlo algorithm for offline use, and the other corresponds to discrete-time TD algorithms to enable online learning.

3.1 Monte Carlo Methods

In this subsection, our goal is to formulate a valid loss function for the Monte Carlo method. In particular, with the loss function, we can take derivative of $J^\theta(t, x)$ with respect to θ, so that an updating rule similar to (10) can be derived in the continuous time. An ideal loss function is the mean-squared error between the estimated value function $J^\theta(\cdot,\cdot)$ and the true value function $J(\cdot,\cdot;\boldsymbol{\pi})$, which we refer to as the *mean-squared value error* (MSVE):

$$\mathrm{MSVE}(\theta) \coloneqq \frac{1}{2}\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T |J(t, X^{\boldsymbol{\pi}}_t; \boldsymbol{\pi}) - J^\theta(t, X^{\boldsymbol{\pi}}_t)|^2 dt\right]. \tag{13}$$

However, since the true function $J(\cdot,\cdot;\boldsymbol{\pi})$ is not known, minimizing MSVE does not directly produce a feasible algorithm .

Following the loss function (11) designed for discrete-time MDPs, which tracks the error between the estimated value function and the realized reward along sample paths, we propose its continuous-time counterpart $L(\theta)$:

$$L(\theta) = \frac{1}{2}\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \left(\int_{(t,T]} p^\top dN^{\boldsymbol{\pi}}_s + \gamma\int_t^T \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X^{\boldsymbol{\pi}}_{s-}))ds - J^\theta(t, X^{\boldsymbol{\pi}}_t)\right)^2 dt\right]. \tag{14}$$

It is worthwhile to note that our proposed loss function $L(\theta)$ also emerges when replacing $J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})$ in (13) – the expected value-to-go when starting at time t and state $X_t^{\boldsymbol{\pi}}$ – with the reward along the trajectory afterwards. Beyond this observation, we proceed to establish a certain equivalence between $L(\theta)$ and $\text{MSVE}(\theta)$ from a theoretical standpoint.

The next theorem states that minimizing the loss function $L(\theta)$ is equivalent to minimizing MSVE. Indeed, the difference between $L(\theta)$ and $\text{MSVE}(\theta)$ is merely a constant term that does not vary with θ. This establishes the validity of our proposed loss function $L(\theta)$ in the continuous-time setting.

Theorem 1. *It holds that* $\arg\min_\theta L(\theta) = \arg\min_\theta \text{MSVE}(\theta)$.

The proof of Theorem 1 is somewhat delicate, necessitating the martingale property associated with $J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})$, as introduced in (7). To this end, we define

$$\tilde{M}_t := J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + \int_0^t \Big\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \Big\} ds, \quad t \in [0, T], \tag{15}$$

where $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ is the exploratory dynamics introduced earlier for the convenience of theoretical analysis. The stochastic process $\tilde{M}_t$ can be interpreted as the accumulated reward and entropy bonus up to time t and using the value function to take the expectation of the future reward between t and T. The following result establishes the martingality of $\{\tilde{M}_t : 0 \le t \le T\}$, which is sufficient to prove Theorem 1. Furthermore, it provides a martingale characterization of the value function $J(\cdot, \cdot; \boldsymbol{\pi})$, which will form the theoretical basis for the continuous-time TD methods discussed in the next section. The proofs of Theorem 2 and Theorem 1 are deferred to Appendix C.

Theorem 2. *The process* $\{\tilde{M}_t : 0 \le t \le T\}$ *is an* $(\{\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\}_{t \ge 0}, \mathbb{P})$*-martingale. Conversely, if there exists a function* $v \in C^{1,0}([0, T] \times \mathcal{S})$*, such that* $\{\tilde{M}_t^v : 0 \le t \le T\}$ *is a square-integrable* $(\{\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\}_{t \ge 0}, \mathbb{P})$*-martingale, where*

$$\tilde{M}_t^v := v(t, \tilde{X}_t^{\boldsymbol{\pi}}) + \int_0^t \Big\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \Big\} ds,$$

and $v(T, x) = 0$ *for all* $x \in \mathcal{S}$*, then* $v(t, x) = J(t, x; \boldsymbol{\pi})$ *for all* $(t, x) \in [0, T] \times \mathcal{S}$.

With the loss function $L(\theta)$ in (14), one can use the gradient to update θ:

$$\theta \leftarrow \theta + \alpha \int_0^T \nabla J^\theta(t, X_t^{\boldsymbol{\pi}}) \Big(\int_{(t,T]} p^\top dN_s^{\boldsymbol{\pi}} + \gamma \int_t^T \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}})) ds - J^\theta(t, X_t^{\boldsymbol{\pi}}) \Big) dt. \tag{16}$$

Note that (16) is the continuous-time analogue of (10). When there is no additional structure on $X_t^{\boldsymbol{\pi}}$, it is a common practice to approximate the integrals in (16) with numerical procedures such as discretization. However, the piecewise constant nature of the state process $X_t^{\boldsymbol{\pi}}$ enables us to exploit the inherent discretization provided by the jump points, effectively eliminating the need of discretization.

More precisely, note that the integrals in (16) either amount to a finite sum of values at the jump points (such as $\int_{(t,T]} p^\top dN_s^{\boldsymbol{\pi}}$) or share a common general form $\int_0^T u(t, X_t^{\boldsymbol{\pi}})dt$. Let $\{x_t : 0 \le t \le T\}$ be a specific realization of the stochastic process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$. Instead of approximating $\int_0^T u(t, X_t^{\boldsymbol{\pi}})dt$ with

$$\sum_{k=0}^{K-1} u(t_k, x_{t_k})(t_{k+1} - t_k), \tag{17}$$

where $0 = t_0 < t_1 < \cdots < t_K = T$ is a fixed grid on $[0, T]$, we propose an alternative approach. Suppose there are L jumps in the trajectory denoted by $0 = \tau_0 < \tau_1 < \cdots < \tau_{L+1} = T$, we can express $\int_0^T u(t, X_t^{\boldsymbol{\pi}})dt$ as

$$\int_0^T u(t, X_t^{\boldsymbol{\pi}})dt = \sum_{l=0}^{L} \int_{\tau_l}^{\tau_{l+1}} u(t, x_{\tau_l})dt. \tag{18}$$

In contrast to (17), the expression in (18) essentially discretizes the time horizon for each trajectory without any discretization error. Moreover, for each k, $u(t, x_{t_k})$ is a function of t with x_{t_k} held constant. If $u(t, x_{\tau_l})$ takes a simple form in t, such as a polynomial, it allows for exact evaluation of the integral $\int_{\tau_l}^{\tau_{l+1}} u(t, x_{\tau_l})dt$. In this case, we can completely avoid the numerical procedure associated with the scheme (17) and compute the value analytically. Even when $u(t, x_{\tau_l})$ is not integrable analytically, we can employ advanced numerical integration algorithms to obtain a highly accurate approximation of the one-dimensional integral $\int_{\tau_l}^{\tau_{l+1}} u(t, x_{\tau_l})dt$.

Having discussed the general idea, we turn our attention to the network revenue management problem, which exhibits unique characteristics. The practice of approximating the optimal value function using a linear combination of basis functions is well-documented in the literature (see, e.g., Zhang and Adelman 2009, Adelman 2007, Ma et al. 2020). Inspired by this, we propose to take the parametric family $\{J^\theta(\cdot, \cdot) : \theta \in \Theta\}$ as a linear functional space with W basis functions $\varphi_1(\cdot, \cdot), \ldots, \varphi_W(\cdot, \cdot)$, where $J^\theta(t, x) := \sum_{j=1}^{W} \theta_j \varphi_j(t, x)$ for all $(t, x) \in [0, T] \times \mathcal{S}$. As a result, the optimization problem $\arg\min_\theta L(\theta)$ can be considerably simplified. In particular, we show next that we do not have to resort to the gradient method (16) and can compute the optimal solution θ^* explicitly with Monte Carlo.

Let $\varphi(t, x) := (\varphi_1(t, x), \ldots, \varphi_W(t, x))^\top$ and $N_{(t,T]}^{\boldsymbol{\pi}} := \{N_s^{\boldsymbol{\pi}} : t < s \le T\}$. Define $h(t, N_{(t,T]}^{\boldsymbol{\pi}}) := \int_{(t,T]} p^\top dN_s^{\boldsymbol{\pi}} + \gamma \int_t^T \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))ds$. By expanding (14), we obtain

$$\begin{aligned} L(\theta) = \frac{1}{2}\theta^\top \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \varphi(t, X_t^{\boldsymbol{\pi}})\varphi(t, X_t^{\boldsymbol{\pi}})^\top dt\right]\theta - \theta^\top \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \varphi(t, X_t^{\boldsymbol{\pi}}) h(t, N_{(t,T]}^{\boldsymbol{\pi}})dt\right] \\ + \frac{1}{2}\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T [h(t, N_{(t,T]}^{\boldsymbol{\pi}})]^2 dt\right]. \end{aligned}$$

It is easy to see that the matrix $\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \varphi(t, X_t^{\boldsymbol{\pi}})\varphi(t, X_t^{\boldsymbol{\pi}})^\top dt\right]$ is positive semi-definite. Hence, the optimization problem $\arg\min_\theta L(\theta)$ reduces to a simple unconstrained quadratic programming problem. Applying the associated theory of unconstrained quadratic programming, we can assert the existence of a minimizer for $L(\theta)$, though it may not be unique. One such minimizer can be computed as follows:

$$\theta^* = \left(\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \varphi(t, X_t^{\boldsymbol{\pi}})\varphi(t, X_t^{\boldsymbol{\pi}})^\top dt\right]\right)^{(-1)} \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \varphi(t, X_t^{\boldsymbol{\pi}}) h(t, N_{(t,T]}^{\boldsymbol{\pi}}) dt\right], \tag{19}$$

where $D^{(-1)}$ represents the MoorePenrose inverse of matrix D. In the implementation, the expectation can be estimated using Monte Carlo across multiple trajectories. Furthermore, building on the earlier discussion in this section about integrals of the form $\int_0^T u(t, x_t)dt$, the two integrals in (19) can be computed as follows. We denote $b(t_1, t_2, x) := \int_{t_1}^{t_2} \varphi(s, x)ds$, $D(t_1, t_2, x) := \int_{t_1}^{t_2} \varphi(s, x)\varphi(s, x)^\top ds$, and $E(t_1, t_2, x, v(\cdot); \boldsymbol{\pi}) := \int_{t_1}^{t_2} v(s)\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, x))ds$.

While the detail can be found in Section 6, it is clear that b and D have a closed form when the basis functions $\varphi(t, x)$ are polynomials in t. On the other hand, the function E, which depends on the policy parametrization (see (39)), does not admit an analytical expression in general. As a result, we are compelled to use numerical integration methods to approximate the function values for each specific tuple (t_1, t_2, x). Despite this, note that we are only dealing with the numerical integration of scalar functions here, for which we can apply well-developed existing algorithms that are fast and accurate. As discussed earlier, our adaptive discretization method, when paired with one-dimensional numerical integration, is expected to outperform the direct numerical procedure described in (17).

Using the notations b, D and E introduced above, we reformulate the integrals involved in (19) as follows:

$$\int_0^T \varphi(t, X_t^{\boldsymbol{\pi}})\varphi(t, X_t^{\boldsymbol{\pi}})^\top dt = \sum_{l=0}^{L} D(\tau_l, \tau_{l+1}, x_{\tau_l}). \tag{20}$$

In addition, for $t \in [\tau_l, \tau_{l+1})$, $h(t, N_{(t,T]}^{\boldsymbol{\pi}}) = \sum_{l'=l+1}^{L}[p^\top \Delta N_{\tau_{l'}}^{\boldsymbol{\pi}} + \gamma E(\tau_{l'}, \tau_{l'+1}, x_{\tau_{l'}}, \mathbf{1}; \boldsymbol{\pi})] + \gamma E(t, \tau_{l+1}, x_{\tau_l}, \mathbf{1}; \boldsymbol{\pi})$. Then, we have

$$\begin{aligned}\int_0^T \varphi(t, X_t^{\boldsymbol{\pi}}) h(t, N_{(t,T]}^{\boldsymbol{\pi}}) dt = \sum_{l=0}^{L} \Bigg\{ & b(\tau_l, \tau_{l+1}, x_{\tau_l}) \sum_{l'=l+1}^{L} \left[p^\top \Delta N_{\tau_{l'}}^{\boldsymbol{\pi}} + \gamma E(\tau_{l'}, \tau_{l'+1}, x_{\tau_{l'}}, \mathbf{1}; \boldsymbol{\pi})\right] \\ & + \gamma E(\tau_l, \tau_{l+1}, x_{\tau_l}, b(\tau_l, \cdot, x_{\tau_l}); \boldsymbol{\pi}) \Bigg\}.\end{aligned} \tag{21}$$

To conclude this subsection, we have introduced a continuous-time loss function for Monte Carlo method and theoretically justified its validity. Furthermore, we have explored the special case of linear function approximation, where the optimal parameter that minimizes our proposed loss function can be explicitly expressed in a closed form. Concentrating on the closed form that involves integrals along sample trajectories, we have proposed an adaptive discretization procedure to compute the integrals, significantly reducing or even completely avoiding discretization errors. With all the techniques in place, the corresponding Monte Carlo algorithm for PE can be implemented accordingly.

3.2 TD Methods Based on Martingale Orthogonality Conditions

Given that the continuous-time Monte Carlo PE requires the entire sample trajectory over $[0,T]$, this approach is inherently offline and presents challenges when adapting for online use. In this section, we propose the continuous-time version of the TD methods which is suitable for use in both online and offline learning settings. Moreover, we provide a theoretical explanation for the continuous-time TD method from the martingale perspective.

Recall the discrete-time online TD(0) method described in (12). We will first derive the online TD algorithm for the continuous time problem heuristically before providing a theoretical foundation. For realized jump points $\tau_1 < \cdots < \tau_l = t$ so far, we consider updating the parameter only at the jump points, by the following rule

$$\begin{aligned}\theta \leftarrow \theta &+ \alpha \nabla_\theta J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l -})[p^\top \Delta N^{\boldsymbol{\pi}}_{\tau_l} + J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l}) - J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l -})] \\ &+ \int_{\tau_{l-1}}^{\tau_l} \nabla_\theta J^\theta(t, X^{\boldsymbol{\pi}}_{\tau_l -}) \left[\frac{\partial J^\theta}{\partial t}(t, X^{\boldsymbol{\pi}}_{\tau_l -}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, X^{\boldsymbol{\pi}}_{\tau_l -}))\right] dt.\end{aligned} \tag{22}$$

To connect to (12), note that between two jump points we have:

$$J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l}) - J^\theta(\tau_{l-1}, X^{\boldsymbol{\pi}}_{\tau_{l-1}}) = J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l}) - J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l -}) + \int_{\tau_{l-1}}^{\tau_l} \frac{\partial J^\theta}{\partial t}(t, X^{\boldsymbol{\pi}}_{\tau_l -}) dt.$$

Moreover, $r_{(\tau_{l-1}, \tau_l]} = p^\top \Delta N^{\boldsymbol{\pi}}_{\tau_l}$ because the reward is only generated at the jump time. The term $p^\top \Delta N^{\boldsymbol{\pi}}_{\tau_l} + J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l}) - J^\theta(\tau_l, X^{\boldsymbol{\pi}}_{\tau_l -})$ can be interpreted as the shadow price of the product sold at time τ_l. It should be highlighted that, the argument regarding the advantage of exploiting the inherent discretization provided by the jump points, as discussed in the previous section, holds equally true here. Therefore, by directly implementing continuous-time TD method in (22) instead of the discretized approach in (12), we have already reduced the errors associated with time discretization.

To understand the theoretical underpinning of the TD algorithm in the continuous-time setting, we first look at the logic behind the discrete-time TD(0) algorithm. At its most fundamental level, it is based on the Bellman equation: for $k = 0, \ldots, K-1$,

$$J_{\Delta t}(t_k, X^{\boldsymbol{\pi}}_{t_k}; \boldsymbol{\pi}) = \mathbb{E}^{\bar{\mathbb{P}}}\left[r_{(t_k, t_{k+1}]} + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t_k, X^{\boldsymbol{\pi}}_{t_k}))\Delta t + J_{\Delta t}(t_{k+1}, X^{\boldsymbol{\pi}}_{t_{k+1}}; \boldsymbol{\pi}) \mid X^{\boldsymbol{\pi}}_{t_k}\right].$$

A necessary and sufficient condition for the Bellman equation to hold is:

$$\begin{aligned}\mathbb{E}^{\bar{\mathbb{P}}}\Bigg[\sum_{k=0}^{K-1} \eta_k \Big\{\mathbb{E}^{\bar{\mathbb{P}}}\left[r_{(t_k, t_{k+1}]} + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t_k, X^{\boldsymbol{\pi}}_{t_k}))\Delta t + J_{\Delta t}(t_{k+1}, X^{\boldsymbol{\pi}}_{t_{k+1}}; \boldsymbol{\pi}) \mid X^{\boldsymbol{\pi}}_{t_k}\right] \\ - J_{\Delta t}(t_k, X^{\boldsymbol{\pi}}_{t_k}; \boldsymbol{\pi})\Big\}\Bigg] = 0,\end{aligned} \tag{23}$$

for any test function $\eta = (\eta_0, \ldots, \eta_{K-1})^\top$ such that each η_k is $\mathcal{F}^{X^{\boldsymbol{\pi}}}_{t_k}$-measurable and bounded. When the approximate value function $J_{\Delta t}(t_k, x; \boldsymbol{\pi})$ is replaced by $J^\theta_{\Delta t}(t_k, x)$, condition (23) dictates that the Bellman error associated with $J^\theta_{\Delta t}(t_k, x)$ should be orthogonal to the space of test functions

η. Moreover, it follows from the properties of conditional expectation that condition (23) is also equivalent to

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[\sum_{k=0}^{K-1}\eta_k\Big\{r_{(t_k,t_{k+1}]}+\gamma\mathcal{H}(\boldsymbol{\pi}(\cdot\mid t_k,X^{\boldsymbol{\pi}}_{t_k}))\Delta t+J_{\Delta t}(t_{k+1},X^{\boldsymbol{\pi}}_{t_{k+1}};\boldsymbol{\pi})-J_{\Delta t}(t_k,X^{\boldsymbol{\pi}}_{t_k};\boldsymbol{\pi})\Big\}\right]=0. \quad (24)$$

Therefore, the previous orthogonality requirement for the Bellman error has been transformed into one for the TD error. From the perspective of algorithm design, it is natural to select a tractable subspace of test functions, within which the orthogonality condition (24) is required to hold. For instance, if we consider value function approximation $J^{\theta}_{\Delta t}\approx J_{\Delta t}$ and take $\eta_k=\nabla_\theta J^{\theta}_{\Delta t}(t_k,X^{\boldsymbol{\pi}}_{t_k})$, we then aim to solve the following equation for θ :

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[\sum_{k=0}^{K-1}\nabla_\theta J^{\theta}_{\Delta t}(t_k,X^{\boldsymbol{\pi}}_{t_k})\Big\{r_{(t_k,t_{k+1}]}+\gamma\mathcal{H}(\boldsymbol{\pi}(\cdot\mid t_k,X^{\boldsymbol{\pi}}_{t_k}))\Delta t+J^{\theta}_{\Delta t}(t_{k+1},X^{\boldsymbol{\pi}}_{t_{k+1}})-J^{\theta}_{\Delta t}(t_k,X^{\boldsymbol{\pi}}_{t_k})\Big\}\right]=0.$$

Applying a stochastic approximation method (Robbins and Monro 1951) to solve this equation essentially leads to the discrete-time TD(0) algorithm with function approximation. Similarly, taking $\eta_k=\sum_{k'=0}^{k}\lambda^{t_k-t_{k'}}\nabla_\theta J^{\theta}_{\Delta t}(t_{k'},X_{t_{k'}})$ will lead to the TD(λ) algorithm. We next establish a continuous-time analog of condition (24) to characterize the true value function $J(\cdot,\cdot;\boldsymbol{\pi})$.

Theorem 3. *A function $v\in C^{1,0}([0,T]\times\mathcal{S})$ is the value function associated with the policy $\boldsymbol{\pi}$, i.e. $v(t,x)=J(t,x;\boldsymbol{\pi})$ for all $(t,x)\in[0,T]\times\mathcal{S}$, if and only if it satisfies $v(T,x)=0$ for all $x\in\mathcal{S}$, and the following martingale orthogonality condition holds for any bounded process ξ with $\xi_t\in\mathcal{F}^{X^{\boldsymbol{\pi}}}_{t-}$ for all $t\in[0,T]$:*

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T\xi_t\Big\{dv(t,X^{\boldsymbol{\pi}}_t)+p^\top dN^{\boldsymbol{\pi}}_t+\gamma\mathcal{H}(\boldsymbol{\pi}(\cdot\mid t,X^{\boldsymbol{\pi}}_{t-}))dt\Big\}\right]=0. \quad (25)$$

The proof of Theorem 3 relies on the martingality of the process $\{\tilde{M}_t:0\le t\le T\}$ in (15) as discussed in Theorem 2, and the fact that $\{X^{\boldsymbol{\pi}}_t:0\le t\le T\}$ has the same distribution (under $\bar{\mathbb{P}}$) as $\{\tilde{X}^{\boldsymbol{\pi}}_t:0\le t\le T\}$ (under $\mathbb{P}$). See Appendix C.

For the parametric family $\{J^\theta(t,x):\theta\in\Theta\}$, our objective is to find $\theta^*\in\Theta$ such that the estimated value function J^{θ^*} satisfies the orthogonality condition (25). When we denote $M^\theta_t:=J^\theta(t,X^{\boldsymbol{\pi}}_t)+\int_{(0,t]}p^\top dN^{\boldsymbol{\pi}}_s+\gamma\int_0^t\mathcal{H}(\boldsymbol{\pi}(\cdot\mid s,X^{\boldsymbol{\pi}}_{s-}))ds$ and take $\xi_t=\nabla_\theta J^\theta(t,X^{\boldsymbol{\pi}}_{t-})$, the objective then becomes to solve $\mathbb{E}^{\bar{\mathbb{P}}}[\int_0^T\nabla_\theta J^\theta(t,X^{\boldsymbol{\pi}}_{t-})dM^\theta_t]=0$ for a solution $\theta^*\in\Theta$. If it has a unique solution $\theta^*\in\Theta$, the stochastic approximation method can be applied to solve for the unique solution θ^* iteratively. This method updates θ in an offline manner after each whole episode as follows:

$$\theta\leftarrow\theta+\alpha\int_0^T\nabla_\theta J^\theta(t,X^{\boldsymbol{\pi}}_{t-})\Big\{\sum_{j=1}^{n}[J^\theta(t,X^{\boldsymbol{\pi}}_{t-}-A^j)-J^\theta(t,X^{\boldsymbol{\pi}}_{t-})+p_j]dN^{\boldsymbol{\pi}}_{j,t}$$
$$+\left[\frac{\partial J^\theta}{\partial t}(t,X^{\boldsymbol{\pi}}_{t-})+\gamma\mathcal{H}(\boldsymbol{\pi}(\cdot\mid t,X^{\boldsymbol{\pi}}_{t-}))\right]dt\Big\},$$

where Itô's formula has been applied to establish the equality $dJ^\theta(t,X^{\boldsymbol{\pi}}_t)=\sum_{j=1}^{n}[J^\theta(t,X^{\boldsymbol{\pi}}_{t-}-A^j)-J^\theta(t,X^{\boldsymbol{\pi}}_{t-})]dN^{\boldsymbol{\pi}}_{j,t}+\frac{\partial J^\theta}{\partial t}(t,X^{\boldsymbol{\pi}}_{t-})dt$. Inspired by the above offline updating rule, we can also update the parameter θ online at each jump point τ_l (for $l = 1, \ldots, L$). This online updating rule is exactly consistent with our heuristically proposed continuous-time online TD(0) method in (22).

For policy evaluation in the network revenue management problems, we briefly discuss the continuous-time TD(0) method with linear function approximation. We take $J^\theta(t,x) = \theta^\top \varphi(t,x)$, where $\varphi(t,x) = (\varphi_1(t,x), \ldots, \varphi_W(t,x))^\top$ is the vector of basis functions, then there is an unique solution to the system of equations $\mathbb{E}^{\bar{\mathbb{P}}}[\int_0^T \nabla_\theta J^\theta(t, X_{t-}^{\boldsymbol{\pi}}) dM_t^\theta] = 0$ under mild conditions. Indeed, the linear system can be solved explicitly as

$$\theta^* = \left(\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \varphi(t, X_{t-}^{\boldsymbol{\pi}}) d\varphi(t, X_t^{\boldsymbol{\pi}})^\top\right]\right)^{-1} \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \varphi(t, X_{t-}^{\boldsymbol{\pi}})\left\{p^\top dN_t^{\boldsymbol{\pi}} + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, X_{t-}^{\boldsymbol{\pi}}))dt\right\}\right], \tag{26}$$

assuming the existence of the inverse. Moreover, following the adaptive discretization procedure discussed in the previous section, the integrals in (26) can also be computed with reduced errors. Specifically, we have

$$\int_0^T \varphi(t, X_{t-}^{\boldsymbol{\pi}})\left\{p^\top dN_t^{\boldsymbol{\pi}} + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, X_{t-}^{\boldsymbol{\pi}}))dt\right\} = \sum_{l=1}^{L} \varphi(\tau_l, x_{\tau_{l-1}}) r_{\tau_l} + \gamma \sum_{l=0}^{L} E(\tau_l, \tau_{l+1}, x_{\tau_l}, \varphi(\cdot, x_{\tau_l}); \boldsymbol{\pi}). \tag{27}$$

In addition, we denote $F(t_1, t_2, x) := \int_{t_1}^{t_2} \varphi(s,x) \frac{\partial \varphi}{\partial s}(s,x) ds$, where an explicit form of F can be obtained under the selected basis function $\varphi(t,x)$ in Section 6. It follows that

$$\int_0^T \varphi(t, X_{t-}^{\boldsymbol{\pi}}) d\varphi(t, X_t^{\boldsymbol{\pi}})^\top = \sum_{l=1}^{L} \varphi(\tau_l, x_{\tau_{l-1}})[\varphi(\tau_l, x_{\tau_l}) - \varphi(\tau_l, x_{\tau_{l-1}})] + \sum_{l=0}^{L} F(\tau_l, \tau_{l+1}, x_{\tau_l}). \tag{28}$$

On combining these components together, we can then easily obtain an estimate of θ^* in (26) by replacing the expectations with sample averages.

4 Policy Gradient

For a given admissible policy, based on the value function estimate derived from the PE step, we next seek to improve the policy using the PG method. Specifically, consider a parametric family of admissible policies $\{\boldsymbol{\pi}^\phi(\cdot \mid \cdot, \cdot) : \phi \in \Phi\}$, our objective is to determine $\arg\max_{\phi \in \Phi} J(0, c; \boldsymbol{\pi}^\phi)$, which directs our attention to the calculation of the gradient $\nabla_\phi J(0, c; \boldsymbol{\pi}^\phi)$.

For technical purposes, we impose some mild conditions on the parametric family $\{\boldsymbol{\pi}^\phi(\cdot \mid \cdot, \cdot) : \phi \in \Phi\}$:

Assumption 1. *For all $(t, x, S) \in [0, T] \times \mathcal{S} \times \mathcal{A}$, the mapping $\phi \mapsto \boldsymbol{\pi}^\phi(S \mid t, x)$ is smooth on Φ. Moreover, for all $(x, S, \phi) \in \mathcal{S} \times \mathcal{A} \times \Phi$, the mapping $t \mapsto \nabla_\phi \boldsymbol{\pi}^\phi(S \mid t, x)$ is continuous on $[0, T]$.*

According to Lemma 1, the characterization of the value function $J(\cdot, \cdot; \boldsymbol{\pi}^\phi)$ is as follows:

$$\begin{cases} \dfrac{\partial J}{\partial t}(t, x; \boldsymbol{\pi}^\phi) + \displaystyle\sum_{S \in \mathcal{A}} H(t, x, S, J(\cdot, \cdot; \boldsymbol{\pi}^\phi)) \boldsymbol{\pi}^\phi(S \mid t, x) + \gamma \mathcal{H}(\boldsymbol{\pi}^\phi(\cdot \mid t, x)) = 0, \quad (t, x) \in [0, T) \times \mathcal{S}, \\ J(T, x; \boldsymbol{\pi}^\phi) = 0, \quad x \in \mathcal{S}. \end{cases}$$

Taking derivative w.r.t. ϕ on both sides, we obtain a new system of equations satisfied by $g(t, x; \phi) \coloneqq \nabla_\phi J(t, x; \boldsymbol{\pi}^\phi)$:

$$\begin{cases} \frac{\partial g}{\partial t}(t, x; \phi) + \sum_{S \in \mathcal{A}} H(t, x, S, J(\cdot, \cdot; \boldsymbol{\pi}^\phi)) \nabla_\phi \boldsymbol{\pi}^\phi(S \mid t, x) + \gamma \nabla_\phi \mathcal{H}(\boldsymbol{\pi}^\phi(\cdot \mid t, x)) \\ \qquad + \sum_{S \in \mathcal{A}} \left(\sum_{y \in \mathcal{S}} g(t, y; \phi) q(y \mid t, x, S) \right) \boldsymbol{\pi}^\phi(S \mid t, x) = 0, \quad (t, x) \in [0, T) \times \mathcal{S}, \\ g(T, x; \phi) = 0, \quad x \in \mathcal{S}. \end{cases} \tag{29}$$

The following result is a generalization of Lemma 1, except that we allow $R(t, x)$ to be an arbitrary function. The proof is similar to the proof of Lemma 1, and hence omitted.

Lemma 2. *Suppose $R(t, x)$ is a real-valued function defined on $[0, T] \times \mathcal{S}$ that is continuous in $t \in [0, T]$ for all $x \in \mathcal{S}$. For any given policy $\boldsymbol{\pi} \in \Pi$, there exist a unique $\varphi \in C^{1,0}([0, T] \times \mathcal{S})$ satisfying the following equation:*

$$\begin{cases} \frac{\partial \varphi}{\partial t}(t, x) + R(t, x) + \sum_{S \in \mathcal{A}} \left(\sum_{y \in \mathcal{S}} \varphi(t, y) q(y \mid t, x, S) \right) \boldsymbol{\pi}(S \mid t, x) = 0, \quad (t, x) \in [0, T) \times \mathcal{S}, \\ \varphi(T, x) = 0, \quad x \in \mathcal{S}, \end{cases}$$

where $q(y \mid t, x, S)$ is given in (1)*. Moreover, this unique solution φ has a stochastic representation:*

$$\varphi(t, x) = \mathbb{E}^{\bar{\mathbb{P}}} \left[\int_t^T R(s, X_{s-}^{\boldsymbol{\pi}}) ds \mid X_t^{\boldsymbol{\pi}} = x \right], \quad (t, x) \in [0, T] \times \mathcal{S}.$$

For our interest of computing the policy gradient, we denote a special reward function

$$\bar{R}(t, x; \phi) \coloneqq \sum_{S \in \mathcal{A}} H(t, x, S, J(\cdot, \cdot; \boldsymbol{\pi}^\phi)) \nabla_\phi \boldsymbol{\pi}^\phi(S \mid t, x) + \gamma \nabla_\phi \mathcal{H}(\boldsymbol{\pi}^\phi(\cdot \mid t, x)).$$

Then, combining equation (29) and Lemma 2, we have

$$g(t, x; \phi) = \mathbb{E}^{\bar{\mathbb{P}}} \left[\int_t^T \bar{R}(s, X_{s-}^{\boldsymbol{\pi}^\phi}; \phi) ds \mid X_t^{\boldsymbol{\pi}^\phi} = x \right]. \tag{30}$$

Therefore, the computation of PG can be viewed as solving a PE problem with a different reward function. Unlike the earlier described PE task, which requires learning the entire function $g(\cdot, \cdot; \phi)$, the current task is more straightforward as it involves only computing the function value $g(0, c; \phi)$, via (30) along multiple sample trajectories. However, it is important to note that the new reward function, $\bar{R}$, incorporates the Hamiltonian H, which can not be directly observed nor calculated in the absence of knowledge about environmental parameters. The next theorem transforms the representation in (30) into an alternative form which can be estimated based on observations of samples and the learned value function, effectively overcoming the aforementioned challenge.

Theorem 4. *Given an admissible parameterized policy* $\boldsymbol{\pi}^{\phi}$ *satisfying Assumption 1, the policy gradient* $\nabla_{\phi} J(0, c; \boldsymbol{\pi}^{\phi})$ *admits the following representation:*

$$\nabla_{\phi} J(0, c; \boldsymbol{\pi}^{\phi}) = \mathbb{E}^{\bar{\mathbb{P}}}\Bigg[\sum_{j=1}^{n} \int_{(0,T]} \nabla_{\phi} \log \boldsymbol{\pi}^{\phi}(S_t^{\boldsymbol{\pi}^{\phi}} \mid t, X_{t-}^{\boldsymbol{\pi}^{\phi}})[J(t, X_{t-}^{\boldsymbol{\pi}^{\phi}} - A^j; \boldsymbol{\pi}^{\phi}) - J(t, X_{t-}^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi}) + p_j] dN_{j,t}^{\boldsymbol{\pi}^{\phi}} + \gamma \int_0^T \nabla_{\phi} \mathcal{H}(\boldsymbol{\pi}^{\phi}(\cdot \mid t, X_{t-}^{\boldsymbol{\pi}^{\phi}})) dt\Bigg]. \tag{31}$$

For a given policy $\boldsymbol{\pi}^{\phi}$, the PE step yields an optimal estimate $J^{\theta^*}(t, x)$ of the true value function $J(t, x; \boldsymbol{\pi}^{\phi})$ when we consider linear function approximations in the context of network revenue management. With $J(t, x; \boldsymbol{\pi}^{\phi})$ approximated by $J^{\theta^*}(t, x)$, all the terms inside the expectation in (31) become computable from observed trajectories under the current policy $\boldsymbol{\pi}^{\phi}$, and the expectation can be replaced by sample averages in estimating the policy gradient. When dealing with the samples, the computation involves determining a finite sum of values at the jump points and integrating the gradient of the entropy term over $[0, T]$. For the former, we can simply plug in the data $(\tau_l, X_{\tau_l -}, X_{\tau_l}, r_{\tau_l})$ obtained at the jump time τ_l. The latter part – the integration – is addressed using the subsequent technique to reduce the approximation error, as previously discussed in Section 3.1. Denote $G(t_1, t_2, x; \boldsymbol{\pi}^{\phi}) \coloneqq \int_{t_1}^{t_2} \nabla_{\phi} \mathcal{H}(\boldsymbol{\pi}^{\phi}(\cdot \mid s, x)) ds$. Then,

$$\int_0^T \nabla_{\phi} \mathcal{H}(\boldsymbol{\pi}^{\phi}(\cdot \mid t, x_t)) dt = \sum_{l=0}^{L} G(\tau_l, \tau_{l+1}, x_{\tau_l}; \boldsymbol{\pi}^{\phi}). \tag{32}$$

Under the parametrization of $\boldsymbol{\pi}^{\phi}$ to be specified in Section 6, an explicit expression for the scalar integral G is unavailable; a numerical integration algorithm will be used instead for its computation.

Theorem 4 extends the policy gradient formula for controlled diffusion processes in Jia and Zhou (2022b) to the intensity control problem with discrete states. In Jia and Zhou (2022b), the policy gradient formula (with discount factor $\beta = 0$ in their paper) is presented as follows:

$$\nabla_{\phi} J(0, x; \phi) = \mathbb{E}^{\bar{\mathbb{P}}}\Bigg[\int_0^T \nabla_{\phi} \log \boldsymbol{\pi}^{\phi}(a_s^{\boldsymbol{\pi}^{\phi}} \mid t, X_t^{\boldsymbol{\pi}^{\phi}})[dJ(t, X_t^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi}) + r(t, X_t^{\boldsymbol{\pi}^{\phi}}, a_t^{\boldsymbol{\pi}^{\phi}}) dt] - \gamma \int_0^T \nabla_{\phi} \log \boldsymbol{\pi}^{\phi}(a_t^{\boldsymbol{\pi}^{\phi}} \mid t, X_t^{\boldsymbol{\pi}^{\phi}})[\log \boldsymbol{\pi}^{\phi}(a_t^{\boldsymbol{\pi}^{\phi}} \mid t, X_t^{\boldsymbol{\pi}^{\phi}}) + 1] dt \mid X_0^{\boldsymbol{\pi}^{\phi}} = x\Bigg], \tag{33}$$

where $(a_s^{\boldsymbol{\pi}^{\phi}})$ is the action process under policy $\boldsymbol{\pi}^{\phi}$, and $r(\cdot, \cdot, \cdot)$ denotes the continuously accrued running reward in their model. There are some major differences between (31) and (33), which result in crucial modifications to design of the RL algorithms as we elaborate below.

Firstly, the term $dJ(t, X_t^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi})$ in (33) is replaced with

$$\sum_{j=1}^{n} [J(t, X_{t-}^{\boldsymbol{\pi}^{\phi}} - A^j; \boldsymbol{\pi}^{\phi}) - J(t, X_{t-}^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi})] dN_{j,t}^{\boldsymbol{\pi}^{\phi}}. \tag{34}$$

In Jia and Zhou (2022b), where $\{X_t^{\boldsymbol{\pi}^{\phi}} : 0 \le t \le T\}$ is a diffusion process, the term $dJ(t, X_t^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi})$ has to be approximated by the finite difference $J^{\theta}(t + \Delta t, X_{t+\Delta t}^{\boldsymbol{\pi}^{\phi}}) - J^{\theta}(t, X_t^{\boldsymbol{\pi}^{\phi}})$ on a discretized time grid, when implementing the algorithm. As a result, actions must be generated at each time point on the grid to evaluate $\nabla_\phi \log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi} \mid t, X_t^{\boldsymbol{\pi}^\phi})$. However, in our context, we benefit from the inherent characteristics of the jump process $(X_t^{\boldsymbol{\pi}^\phi})$. While Itô's formula establishes the equivalence between $dJ(t, X_t^{\boldsymbol{\pi}^\phi})$ and $\frac{\partial J}{\partial t}(t, X_{t-}^{\boldsymbol{\pi}^\phi})dt + \sum_{j=1}^n [J(t, X_{t-}^{\boldsymbol{\pi}^\phi} - A^j; \boldsymbol{\pi}^\phi) - J(t, X_{t-}^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi)]dN_{j,t}^{\boldsymbol{\pi}^\phi}$, the term $\nabla_\phi \log \boldsymbol{\pi}^\phi(S_t^{\boldsymbol{\pi}^\phi} \mid t, X_{t-}^{\boldsymbol{\pi}^\phi})\frac{\partial J}{\partial t}(t, X_{t-}^{\boldsymbol{\pi}^\phi})$ vanishes after averaging out the randomness from action randomization. Specifically, it follows from $S_t^{\boldsymbol{\pi}^\phi} \sim \boldsymbol{\pi}^\phi(\cdot \mid t, X_{t-}^{\boldsymbol{\pi}^\phi})$ that

$$\begin{aligned}\mathbb{E}^{\bar{\mathbb{P}}}\left[\nabla_\phi \log \boldsymbol{\pi}^\phi(S_t^{\boldsymbol{\pi}^\phi} \mid t, X_{t-}^{\boldsymbol{\pi}^\phi})\frac{\partial J}{\partial t}(t, X_{t-}^{\boldsymbol{\pi}^\phi})\right] &= \mathbb{E}^{\bar{\mathbb{P}}}\left[\frac{\partial J}{\partial t}(t, X_{t-}^{\boldsymbol{\pi}^\phi})\sum_{S\in\mathcal{A}}[\nabla_\phi \log \boldsymbol{\pi}^\phi(S \mid t, X_{t-}^{\boldsymbol{\pi}^\phi})]\boldsymbol{\pi}^\phi(S \mid t, X_{t-}^{\boldsymbol{\pi}^\phi})\right] \\ &= \mathbb{E}^{\bar{\mathbb{P}}}\left[\frac{\partial J}{\partial t}(t, X_{t-}^{\boldsymbol{\pi}^\phi})\nabla_\phi\left(\sum_{S\in\mathcal{A}}\boldsymbol{\pi}^\phi(S \mid t, X_{t-}^{\boldsymbol{\pi}^\phi})\right)\right] = 0.\end{aligned}$$

The refined expression (34) makes it possible to avoid artificially discretizing time in the algorithmic implementation, and instead to solely utilize the information at the jump points in each trajectory of $(X_t^{\boldsymbol{\pi}^\phi})$. Therefore, action randomization is only required at the times when customers arrive.

Secondly, in line with the aim of avoiding successive action randomization at all mesh grid points, we also average out the randomness from the action randomization in the term $-\nabla_\phi \log \boldsymbol{\pi}^\phi(S_t^{\boldsymbol{\pi}^\phi} \mid t, X_{t-}^{\boldsymbol{\pi}^\phi})[\log \boldsymbol{\pi}^\phi(S_t^{\boldsymbol{\pi}^\phi} \mid t, X_{t-}^{\boldsymbol{\pi}^\phi}) + 1]$ in (33). Consequently, we arrive at the term $\nabla_\phi \mathcal{H}(\boldsymbol{\pi}^\phi(\cdot \mid t, X_t^{\boldsymbol{\pi}^\phi}))$, as presented in our theorem. Both of the aforementioned modifications contribute to reducing the variance of the policy gradient method and avoiding unnecessary action randomization.

5 Actor-Critic Algorithms

Combining the PE and PG modules in Sections 3 and 4, we next present two model-free actor-critic algorithms for the network revenue management problem introduced in Section 2. Algorithm 1, which utilizes the Monte Carlo method for PE (Section 3.1), is detailed in the main text. Algorithm 2, adhering to a similar framework but employing the TD(0) method for PE (Section 3.2), is presented in Appendix B.

In both algorithms, we assume a linear parametrization for $\{J^\theta(t,x) : \theta \in \Theta\}$, that is, $J^\theta(t,x) = \sum_{j=1}^W \theta_j \varphi_j(t,x)$, where $\varphi_1, \ldots, \varphi_W$ are the basis functions. We also consider a parameterized family of stochastic policies $\{\boldsymbol{\pi}^\phi(S \mid t, x) : \phi \in \Phi\}$. Our aim is to determine the optimal values for (θ, ϕ) jointly, by alternately updating each parameter. Note that both algorithms are designed for offline learning, where full trajectories are sampled and observed repeatedly during different episodes and (θ, ϕ) are updated after every M episodes, with M defined as the batch size. We tune a function of the iteration number, $\gamma(\cdot)$, to gradually decrease the temperature (exploration level) to zero across the iterations in the algorithm implementation, which is slightly different from our theoretical analysis where γ is held as a constant.

In addition, we employ an environment simulator to generate trajectories under given policies. The environment simulator, denoted as $(t', x', S', r') = \text{Environment}(t, x, \boldsymbol{\pi}^\phi(\cdot \mid \cdot, \cdot))$, operates by first taking current time-state pair (t, x) and the policy $\boldsymbol{\pi}^\phi(\cdot \mid \cdot, \cdot)$ as inputs. It then samples a time interval s from an exponential distribution with rate λ, indicating the duration until the next customer arrival. Using the updated time $t' = t + s$ and the current state x, an offer set S' is sampled from the policy $\boldsymbol{\pi}^{\phi}(\cdot \mid t', x)$. The state transitions to $x - A^j$ with probability $P_j(S')$ and remains at x with probability $P_0(S')$. Based on this distribution, the state at t' is sampled, and the corresponding reward r' is returned. If $x' \neq x$, the simulation outputs the transition time t', new state x', action S' and reward r'. If $x' = x$, with the current time-state pair (t', x'), the process repeats: a new time interval s is drawn from the same exponential distribution, and the above steps are executed until a transition to a different state occurs, at which point it outputs the transition time, along with the corresponding new state, action, and reward at that time.

6 Experimental Setup and Numerical Performance

In this section, we specialize the algorithms to the choice-based network revenue management. We first seek a suitable family of functions $\{J^{\theta}(t, x) : \theta \in \Theta\}$ that can approximate the value function associated with any particular policy within our selected parametric policy family $\{\boldsymbol{\pi}^{\phi} : \phi \in \Phi\}$. We draw inspiration from the form of the approximated optimal value function within a discrete time framework, as suggested by Zhang and Adelman (2009):

$$\tilde{V}_{\Delta t}(t_k, x) = \theta_k + \sum_{i=1}^{m} V_{k,i} x_i, \tag{35}$$

where $V_{k,i}$ estimates the marginal value of a unit resource i in period $(t_k, t_{k+1}]$, and θ_k is a constant offset. For terminal condition, it is assumed that $\theta_K = 0$ and $V_{K,i} = 0$ for all $i = 1, \ldots, m$. We propose to use the following continuous time counterpart for the family $\{J^{\theta}(t, x) : \theta \in \mathbb{R}^{(m+1)\times(d+1)}\}$:

$$J^{\theta}(t, x) \coloneqq \sum_{l=0}^{d} \theta_{(0,l)} (T - t)^l + \sum_{i=1}^{m} \left(\sum_{l=0}^{d} \theta_{(i,l)} (T - t)^l \right) x_i. \tag{36}$$

Here, we replace θ_k and $V_{k,i}$ in the discrete-time approximation (35), which would be infinite-dimensional in the continuous time, with a dth-order polynomial of $(T - t)$. The family $\{J^{\theta}(t, x) : \theta \in \mathbb{R}^{(m+1)\times(d+1)}\}$ constitutes a linear space, with the basis functions

$$\begin{aligned} \varphi(t, x) = (1, T - t, (T - t)^2, \cdots, (T - t)^d; x_1, (T - t)x_1, (T - t)^2 x_1, \cdots, (T - t)^d x_1; \\ \cdots; x_m, (T - t)x_m, (T - t)^2 x_m, \cdots, (T - t)^d x_m)^{\top}. \end{aligned} \tag{37}$$

We now explore the selection of the parametric family of policies $\{\boldsymbol{\pi}^{\phi} : \phi \in \Phi\}$, guided by the family of value functions above. Given an approximation J^{θ} as defined in (36) for the optimal value function J^*, applying the Hamiltonian H to J^{θ} yields the following expression:

$$\begin{aligned} H(t, x, S, J^{\theta}(\cdot, \cdot)) &= \sum_{j=1}^{n} [J^{\theta}(t, x - A^j) - J^{\theta}(t, x) + p_j] \lambda P_j(S) \\ &= -\sum_{l=0}^{d} \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} \theta_{(i,l)} \lambda P_j(S) (T - t)^l + r(S), \end{aligned}$$

Algorithm 1 Actor-Critic Algorithm (PE via Monte Carlo method)

Inputs: initial state c, time horizon T, number of iterations N, batch size M; functional forms of basis functions $\varphi_1, \dots, \varphi_W$, functional form of the policy $\boldsymbol{\pi}^\phi(\cdot \mid \cdot, \cdot)$ and an initial value ϕ_0; a temperature schedule function $\gamma(\cdot)$; initial learning rates α_ϕ and a learning rate schedule function $w(\cdot)$

Required program: an environment simulator $(t', x', S', r') = \mathit{Environment}(t, x, \boldsymbol{\pi}^\phi(\cdot \mid \cdot, \cdot))$

Initialize $\phi = \phi_0$

for $n = 1$ to N **do**

  for $i = 1$ to M **do** ▷ Generate M full trajectories under policy $\boldsymbol{\pi}^\phi$

    Store $(\tau_0^{(i)}, x_0^{(i)}) \leftarrow (0, c)$

    Initialize $l = 0$, $(t, x) = (0, c)$ ▷ Initialize l to count state transitions in each trajectory, and (t, x) to record the time and state right after a transition

    while True **do**

      Apply (t, x) to the environment simulator to get $(t', x', S', r') = \mathit{Environment}(t, x, \boldsymbol{\pi}^\phi(\cdot \mid \cdot, \cdot))$

      if $t' \geq T$ **then**

        break

      end if

      Update $l \leftarrow l + 1$

      Store current transition: $(\tau_l^{(i)}, x_l^{(i)}, S_l^{(i)}, r_l^{(i)}) \leftarrow (t', x', S', r')$

      Update $(t, x) \leftarrow (t', x')$

    end while

    Store $L^{(i)} \leftarrow l$, $\tau_{L^{(i)}+1}^{(i)} \leftarrow T$

  end for

  Evaluate policy $\boldsymbol{\pi}^\phi$: [using formula (19), incorporating techniques (20) and (21)]

$$\theta^* = \left(\frac{1}{M} \sum_{i=1}^{M} \sum_{l=0}^{L^{(i)}} D(\tau_l^{(i)}, \tau_{l+1}^{(i)}, x_l^{(i)}) \right)^{(-1)} \times$$
$$\left(\frac{1}{M} \sum_{i=1}^{M} \sum_{l=0}^{L^{(i)}} \left\{ b(\tau_l^{(i)}, \tau_{l+1}^{(i)}, x_l^{(i)}) \sum_{l'=l+1}^{L^{(i)}} \left[r_{l'}^{(i)} + \gamma(n) E(\tau_{l'}^{(i)}, \tau_{l'+1}^{(i)}, x_{l'}^{(i)}, \mathbf{1}; \boldsymbol{\pi}^\phi) \right] \right.\right.$$
$$\left.\left. + \gamma(n) E(\tau_l^{(i)}, \tau_{l+1}^{(i)}, x_l^{(i)}, b(\tau_l^{(i)}, \cdot, x_l^{(i)}); \boldsymbol{\pi}^\phi) \right\} \right)$$

  Compute policy gradient at ϕ: [using formula (31), incorporating technique (32)]

$$\Delta\phi = \frac{1}{M} \sum_{i=1}^{M} \left(\sum_{l=1}^{L^{(i)}} \nabla_\phi \log \boldsymbol{\pi}^\phi (S_l^{(i)} \mid \tau_l^{(i)}, x_{l-1}^{(i)}) [J^{\theta^*}(\tau_l^{(i)}, x_l^{(i)}) - J^{\theta^*}(\tau_l^{(i)}, x_{l-1}^{(i)}) + r_l^{(i)}] \right.$$
$$\left. + \gamma(n) \sum_{l=0}^{L^{(i)}} G(\tau_l^{(i)}, \tau_{l+1}^{(i)}, x_l^{(i)}; \boldsymbol{\pi}^\phi) \right)$$

  Update ϕ by

$$\phi \leftarrow \phi + \alpha_\phi w(n) \Delta\phi$$

end for

which is also a dth-order polynomial in $(T-t)$. The above discussion, together with (9), motivates us to consider the following parametric family of policies $\{\boldsymbol{\pi}^{\phi} : \phi \in \mathbb{R}^{2^n \times (d+1)}\}$:

$$\boldsymbol{\pi}^{\phi}(S \mid t, x) := \frac{\exp\{\frac{1}{\gamma} h_S(t;\phi)\}}{\sum_{\bar{S}\in\mathcal{A}(x)} \exp\{\frac{1}{\gamma} h_{\bar{S}}(t;\phi)\}}, \quad \text{for } (t,x) \in [0,T]\times\mathcal{S},\ S\in\mathcal{A}(x), \tag{38}$$

where $h_S(t;\phi) := \sum_{l=0}^{d} \phi_{(S,l)}(T-t)^l$. It should be noted that the stochastic policy in (38) involves $2^n \times (d+1)$ parameters $\{\phi_{(S,l)} : S \in \mathcal{A},\ l = 0,\ldots,d\}$, which grows exponentially with n and becomes intractable for even moderate-sized n. To address this challenge, we limit the number of parameters by only capturing the interaction between a pair of products. Specifically, we introduce a set of parameters $\{\phi_{(j,j',l)} : j = 1,\ldots,n;\ j' = 1,\ldots,n;\ l = 0,\ldots,d\}$, yielding a parameter space of $\mathbb{R}^{n\times n\times(d+1)}$. For all $S \in \mathcal{A}$ and $l = 0,\ldots,d$, we let $\phi_{(S,l)}$ be

$$\phi_{(S,l)} = \sum_{1\le j,\,j'\le n} \phi_{(j,j',l)} S^j S^{j'},$$

where the offer set $S \in \mathcal{A}$ is characterized by an n-dimensional binary valued vector $(S^1,\ldots,S^n)$, with $S^j := \mathbf{1}\{j \in S\}$ for $j = 1,\ldots,n$. Intuitively, $\phi_{(j,j',l)}$ captures the interaction between product j and j'. This strategy effectively reduce the number of parameters from $O(2^n)$ to $O(n^2)$. Then, we employ the reduced parametric family $\{\boldsymbol{\pi}^{\phi} : \phi \in \mathbb{R}^{n\times n\times(d+1)}\}$, where $\boldsymbol{\pi}^{\phi}$ is specified as follows: for $(t,x) \in [0,T]\times\mathcal{S}$, $S \in \mathcal{A}(x)$,

$$\boldsymbol{\pi}^{\phi}(S \mid t,x) = \frac{\exp\left\{\frac{1}{\gamma}\sum_{l=0}^{d}\left(\sum_{1\le j,\,j'\le n}\phi_{(j,j',l)}S^jS^{j'}\right)(T-t)^l\right\}}{\sum_{\bar{S}\in\mathcal{A}(x)}\exp\left\{\frac{1}{\gamma}\sum_{l=0}^{d}\left(\sum_{1\le j,\,j'\le n}\phi_{(j,j',l)}\bar{S}^j\bar{S}^{j'}\right)(T-t)^l\right\}}. \tag{39}$$

Having specified $J^{\theta}(t,x)$ and $\boldsymbol{\pi}^{\phi}(S \mid t,x)$ in (36) and (39), we revisit the algorithms to discuss the approximation errors. Firstly, due to the form of the basis functions, as described in (37), the functions $b(\cdot,\cdot)$, $D(\cdot,\cdot)$ and $F(\cdot,\cdot)$ can be evaluated exactly without any approximation errors. However, the softmax form of $\boldsymbol{\pi}^{\phi}$, defined in (39), complicates the derivation of antiderivatives of functions associated with the entropy $\mathcal{H}(\boldsymbol{\pi}^{\phi}(\cdot \mid t,x))$, making it difficult to obtain explicit expressions for $E(t_1,t_2,x,v(\cdot);\boldsymbol{\pi}^{\phi})$ and $G(t_1,t_2,x;\boldsymbol{\pi}^{\phi})$. Instead, the corresponding one-dimensional integrals with respect to time t are addressed using advanced numerical integration techniques with small errors. Therefore, in Algorithm 1 and 2, the errors in calculating θ^* and $\Delta\phi$ arise solely from the replacement of exact expectations with sample averages and the computations involved in the functions E and G, while the latter is minimal.

6.1 Benchmarks

In this section, we introduce the benchmarks that have been studied in the literature to evaluate the performance of our proposed algorithm in experiments.

6.1.1 Optimal Policy from Discretized Dynamic Programming

Given the challenges in solving the HJB equation (2) within a continuous-time framework, one can consider using finite difference approximation to replace derivative, thus deriving the following dynamic programming (DP) problem

$$\begin{cases} V_{\Delta t}^*(t_k, x) = \max_{S \in \mathcal{A}(x)} \left\{ \sum_{j \in S} \lambda \Delta t P_j(S)[p_j + V_{\Delta t}^*(t_{k+1}, x - A^j) - V_{\Delta t}^*(t_{k+1}, x)] \right\} + V_{\Delta t}^*(t_{k+1}, x), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall\, k = 0, \ldots, K-1;\ x \in \mathcal{S}. \\ V_{\Delta t}^*(t_K, x) = 0, \quad \forall\, x \in \mathcal{S}. \end{cases} \tag{40}$$

It is worth noting that the DP problem (40) corresponds exactly to the underlying dynamic program of the discrete-time model $\mathcal{M}_{\Delta t}$ described in Section 3. Therefore, when Δt is sufficiently small, $V_{\Delta t}^*(0, c)$ can serve as an reliable approximation to the true optimal expected revenue $V^*(0, c)$. This makes it a suitable benchmark for evaluating the performance of our proposed policy. When the size of the state space $\mathcal{S}$ is relatively small, we can solve the DP problem (40) recursively to obtain $V_{\Delta t}^*(0, c)$. However, when the size of $\mathcal{S}$ is large, this approach becomes challenging.

6.1.2 Other Policies

In addition, we consider the following policies as benchmarks.

- *Uniform Random Policy.* Denoted by $\boldsymbol{\pi}^{\text{Uniform-Random}}$, this is a stationary policy that selects among all available offer sets at each given state with equal probability. That is, $\boldsymbol{\pi}^{\text{Uniform-Random}}(S \mid x) = \frac{1}{|\mathcal{A}(x)|}$ for all $x \in \mathcal{S}$ and $S \in \mathcal{A}(x)$.
- *Greedy Policy.* The greedy policy is a deterministic and stationary policy that always selects the offer set with the highest expected revenue from all available offer sets at each state, that is, $\arg\max_{S \in \mathcal{A}(x)} \sum_{j \in S} p_j P_j(S)$.
- *CDLP Policy.* This policy is first introduced in Liu and Van Ryzin (2008) based on the following choice-based deterministic linear programming (CDLP) model

$$\begin{aligned} z_{\text{CDLP}} = \max_{h} \ & \sum_{S \in \mathcal{A}} \lambda R(S) h(S) \\ \text{s.t.} \ & \sum_{S \in \mathcal{A}} \lambda Q_i(S) h(S) \le c_i, \quad \forall\, i = 1, \ldots, m. \\ & \sum_{S \in \mathcal{A}} h(S) \le T, \\ & h(S) \ge 0, \quad \forall\, S \in \mathcal{A}, \end{aligned}$$

  where $h(S)$ denotes the total duration that the subset S is offered, $R(S) = \sum_{j \in S} p_j P_j(S)$ denotes the expected revenue from offering S to an arriving customer, and $Q_i(S) = \sum_{j=1}^{n} A_{ij} P_j(S)$ denotes the rate of using a unit of capacity on resource i when S is offered. The CDLP policy executes by sequentially offering each set S for the duration specified by the optimal solution to the CDLP, following the lexicographic order of the variable indices.

  In addition, it has been demonstrated in Liu and Van Ryzin (2008) that the optimal objective function value of the CDLP provides an upper bound on the optimal expected revenue $V^*(0, c)$

in the stochastic problem. It is worth mentioning that although the discussions in Liu and Van Ryzin (2008) are based on a discrete-time model, the conclusion regarding the upper bound property can be readily extended to our continuous-time framework.

- *ADP Policy.* This policy is derived from the approximate dynamic programming (ADP) approach detailed in Zhang and Adelman (2009). Since the ADP approach operates in a discrete-time setting, we apply it to the discrete-time model $\mathcal{M}_{\Delta t}$ described in Section 3. That is, to consider an affine functional approximation $\tilde{V}_{\Delta t}(t_k, x) = \theta_k + \sum_{i=1}^m V_{k,i}x_i$ to the optimal value function $V^*_{\Delta t}(t_k, x)$ and then formulates the dynamic program (40) as a linear program:

$$\begin{aligned}
&\min_{\theta, V} \ \theta_0 + \sum_i V_{0,i}c_i \\
&\text{s.t. } \theta_k - \theta_{k+1} + \sum_i (V_{k,i} - V_{k+1,i})x_i \geq \lambda \Delta t \sum_{j \in S} P_j(S)[p_j - \sum_{i=1}^m V_{k+1,i}A_{ij}], \\
&\qquad\qquad\qquad\qquad\qquad\qquad \forall\, k = 0, \dots, K-1;\ x \in \mathcal{S};\ S \in \mathcal{A}(x), \\
&\quad \theta_K = 0,\ V_{K,i} = 0.
\end{aligned}$$

  With the optimal solution (θ^*, V^*) for the above linear program, the ADP policy is constructed by offering $\arg\max_{S \in \mathcal{A}(x)} \sum_{j \in S} P_j(S)[p_j - \sum_{i=1}^m V^*_{k+1,i}A_{ij}]$ in time period $(t_k, t_{k+1}]$ and state x. Given that the ADP policy varies with different degrees of time discretization, we will refer to the ADP policy under the discrete-time model $\mathcal{M}_{\Delta t}$ as ADP-Δt.

Note that all the benchmarks mentioned above, except for the uniform random policy, require the knowledge of environment, specifically the customer arrival rate λ and the choice probabilities $P_j(S)$. Thus we provide the exact values of the parameters to these policies. Therefore, their performance is slightly inflated compared to our RL algorithm which does not require knowing the values of such parameters.

6.2 Experiment One: A Small Network

We consider a simple example featuring 2 resources and 3 products. The consumption is captured by matrix $A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$. The initial stocks for the 2 resources are set to $c = (5, 5)^\top$ and the price for the 3 products are fixed at $p = (1, 1, 1.5)$. We set the booking horizon to $T = 15$, during which customer arrivals are modeled by a Poisson process with a rate of $\lambda = 0.9$. The choice probabilities are determined by the weights vector $v = (v_0, v_1, v_2, v_3) = (27.8, 42, 42, 55)$ through the multinomial logit (MNL) choice model, specified as

$$P_j(S) \coloneqq \frac{v_j}{v_0 + \sum_{j \in S} v_j}, \qquad \text{for } S \subseteq \mathcal{J}, j \in S.$$

For this example, we present the results of Algorithm 1, which utilizes Monte Carlo method for the PE step. The performance of Algorithm 2, although not presented in the paper, is similar to Algorithm 1. The inputs are configured as follows: we set $c = (5, 5)^\top$, $T = 15$, $N = 12,000$

and $M = 100$. Moreover, the functional forms of basis functions $\varphi(t, x)$ and policy $\boldsymbol{\pi}^{\phi}$ are defined in (37) and (39), respectively, both with $d = 2$. The initial values for all $\phi(j, j', l)$ are set to 0, corresponding to the uniform random policy. The temperature schedule function is defined as $\gamma(n) = 0.1 \times 0.5^{\lfloor \frac{n}{1000} \rfloor}$. Learning rate for ϕ is initialized as $\alpha_{\phi} = 1 \times 10^{-7}$ and decays according to $w(n) = 0.5^{\lfloor \frac{n}{1000} \rfloor}$. Every 100 iterations, we conduct a performance evaluation of the policy $\boldsymbol{\pi}^{\phi}$ obtained at the end of such interval. This evaluation involves simulating a Poisson arrival stream of customers to generate revenues and estimating the expected revenue using 10,000 samples. Figure 1 illustrates the variation in the expected revenue of the Algorithm 1 as the iterations progress. Upon completing the final iteration, iteration 12,000, the simulated average revenue of the resulting policy, denoted as $\boldsymbol{\pi}^{\text{RL-Random}}$, achieves a value of 8.841.

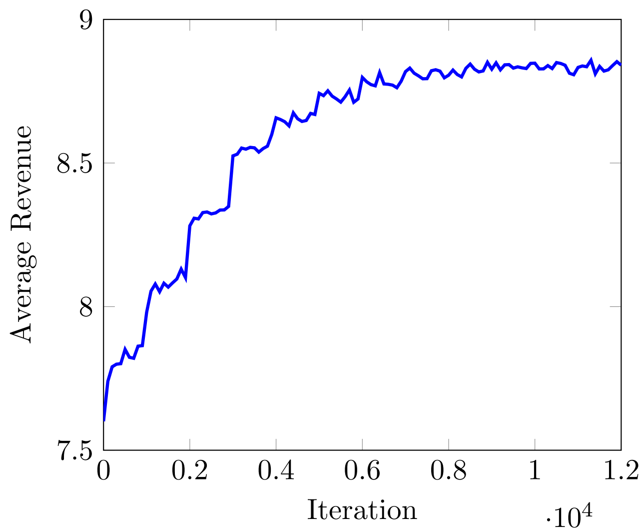

Figure 1: Average revenue of Algorithm 1 as the iterations progress for the example in Section 6.2

Next, we solve the DP problem (40) for this example. By setting $\Delta t = 0.001$, we obtain a highly reliable approximation of the optimal expected revenue, $V^*_{0.001}(0, c) = 8.934$. Moreover, we implement the benchmarks outlined in Section 6.1, including the Uniform-Random, Greedy, CDLP, and ADP policies. In line with the performance evaluations conducted in the RL algorithm, each benchmark policy here is simulated 10,000 times under the same Poisson arrival stream of customers. Table 1 reports the simulated average revenue for our RL policy and the benchmarks, with the 99%-confidence intervals listed in the adjacent column. To make it easy to compare the performance, we also provide the ratio of each simulated average revenue to the optimal value function from dynamic programming, $V^*_{0.001}(0, c) = 8.934$.

The numerical results indicate that, our algorithm achieves 98.96% of the optimal performance, outperforming all other benchmarks other than dynamic programming. Moreover, in the considered small network, ADP policies maintains relatively stable performance across different degrees of time discretization and consistently outperform other benchmarks. Despite this, our RL-random policy still exhibits a slight advantage over the best-performing ADP-0.1 policy.

Table 1: Simulation Results for Selected Policies

Policy	Average Revenue	99%-CI	Ratio(%)
RL-Random	8.841	±0.033	98.96
Uniform-Random	7.599	±0.037	85.06
Greedy	8.473	±0.024	94.84
CDLP	8.577	±0.045	96.00
ADP-1	8.752	±0.041	97.96
ADP-0.5	8.735	±0.042	97.77
ADP-0.2	8.732	±0.042	97.73
ADP-0.1	8.757	±0.042	98.02
ADP-0.05	8.753	±0.042	97.97

6.3 Experiment Two: A Medium-Sized Airline Network

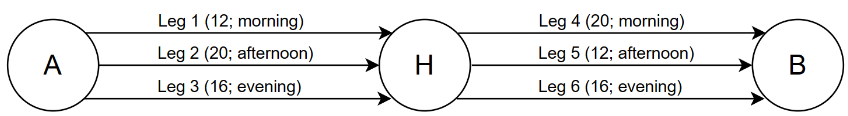

Figure 2: Airline Network

This example considers a medium-sized airline network consisting of 6 flight legs with a total of 9 products (including local and connecting itineraries). The booking horizon is set to $T = 200$ time units. Figure 2 presents the airline network, with each leg labeled, for example, "Leg 1 (12; morning)" represents a morning flight with an initial capacity of 12 seats, and so forth for the remaining legs. Table 2 describes the products. The product set is segmented into 3 disjoint consideration sets, corresponding to 3 customer segments. Segments are defined in terms of their origin-destination market. Table 3 describes precisely the characteristics of each segments. For segments $l = 1, 2, 3$, the second column gives the arrival rates for customers from segment l, denoted by λ_l, so $\lambda = \sum_{l=1}^{3} \lambda_l = 0.8$ is the total arrival rate; the third column outlines the consideration set for each segment, represented by $\mathcal{J}_l$; the fourth column details the preference weights for each product within a segment, expressed as $(v_{lj})_{j \in \mathcal{J}_l}$; the last column presents the no-purchase preference weight for segment l, indicated by v_{l0}. The choice probabilities are determined by employing the mixed multinomial logit choice model as follows: for $j \in \mathcal{J}_l$ for some $l = 1, 2, 3$, $S \subseteq \mathcal{J}$, we have

$$P_j(S) := \begin{cases} \frac{\lambda_l}{\lambda} \cdot \frac{v_{lj}}{\sum_{j \in S \cap \mathcal{J}_l} v_{lj} + v_{l0}}, & j \in S, \\ 0, & j \notin S. \end{cases}$$

Table 2: Product Descriptions and Prices

Product	1	2	3	4	5	6	7	8	9
Legs	1	2	3	4	5	6	$\{1, 4\}$	$\{2, 5\}$	$\{3, 6\}$
Price	8	10	6	8	10	6	9	12	7

Table 3: Segments and Their Characteristics

Segment l	Arrival rate λ_l	Description	Consideration set $\mathcal{J}_l$	Preference weights $(v_{lj})_{j\in\mathcal{J}_l}$	No-purchase weight v_{l0}
1	0.2	A → H	$\{1,2,3\}$	$(5,10,1)$	1
2	0.2	H → B	$\{4,5,6\}$	$(5,10,1)$	1
3	0.4	A → H → B	$\{7,8,9\}$	$(5,1,10)$	5

For this example, we implement Algorithm 1 once again. However, this time we utilize the Adam algorithm for optimization, which is introduced by Kingma and Ba (2014) as an improvement upon the standard SGD algorithm. The inputs are configured as follows: We set $c = (12,20,16,20,12,16)^\top$, $T = 200$, $N = 15,000$ and $M = 10$. Moreover, the functional forms of basis functions $\varphi(t,x)$ and policy $\boldsymbol{\pi}^\phi$ are defined in (37) and (39), respectively, both with $d = 3$. The initial values for all $\phi(j,j',l)$ are set to 0, corresponding to the uniform random policy. The temperature schedule function is defined as $\gamma(n) = 0.1 \times 0.1^{\lfloor \frac{n}{2000} \rfloor}$. Learning rate for ϕ is initialized as $\alpha_\phi = 1 \times 10^{-9}$ and decays according $w(n) = 0.5^{\lfloor \frac{n}{2000} \rfloor}$. Every 100 iterations, we conduct a performance evaluation of the policy $\boldsymbol{\pi}^\phi$ obtained at the end of such interval. Given the higher simulation cost per sample trajectory in this example, we adjust the number of samples for estimating expected revenue from 10,000 to 1,000. Figure 3 illustrates the variation in the expected revenue of the updated policies as the iterations progress. Upon completing the final iteration, iteration 15,000, the simulated average revenue of the resulting policy, denoted as $\boldsymbol{\pi}^{\text{RL-Random}}$, achieves a value of 677.251. We also observe that there is a significant jump in the average revenue around 2,000 iterations. This is because the temperature parameter $\gamma(n)$ changes from 0.1 to 0.01 at $n = 2,000$.

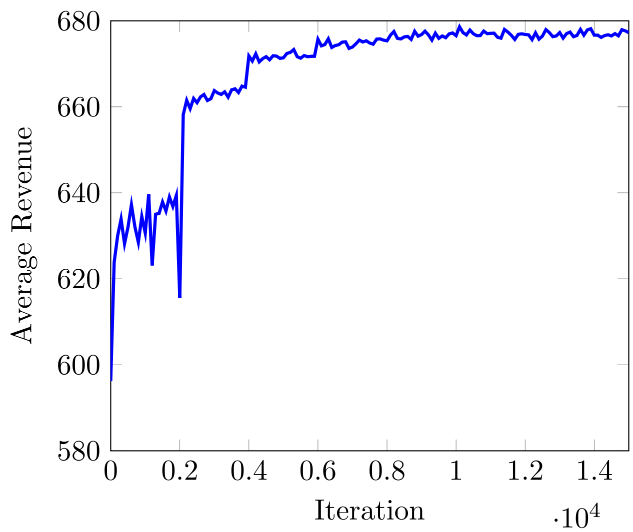

Figure 3: Average revenue of Algorithm 1 as the iterations progress for the example in Section 6.3

Due to the problem's state space being excessively large, approximately 10^6, dynamic programming approaches become infeasible. We therefore implement other benchmarks outlined in Section 6.1 for this example. Each benchmark policy is also simulated 1,000 times under the same Poisson arrival stream of customers. Table 1 reports the simulated average revenue for our RL policy and the benchmarks, with the 99%-confidence intervals listed in the adjacent column. In addition, we also present the relative performance difference between the RL-random policy and each benchmark. The table indicates that our algorithm outperforms simple heuristics such as Uniform-

Table 4: Simulation Results for Selected Policies

Policy	Average Revenue	99%-CI	$\frac{\text{RL-Random}}{*} - 1$ (%)
RL-Random	677.251	±1.822	0
Uniform-Random	595.628	±2.686	13.704
Greedy	605.563	±1.216	11.838
CDLP	654.762	±3.667	3.435
ADP-1	676.337	±2.017	0.135
ADP-0.5	560.520	±4.749	20.825
ADP-0.2	666.219	±3.040	1.656
ADP-0.1	659.389	±3.385	2.709
ADP-0.05	668.200	±3.044	1.355

Random and Greedy by more than 10%, and also shows a 3.435% competitive edge against the more advanced CDLP algorithm. Although the performance of ADP-1 is comparable to that of our algorithm, we observe that for this medium-sized network, the performance of ADP policies is highly sensitive to the level of time discretization. When using a suboptimal discretization level, such as $\Delta t = 0.5$, our algorithm can outperform ADP-0.5 by a margin of up to 20%. This further demonstrates the advantage of our algorithm, as it operates within a continuous-time framework and avoids issues with unfront time discretization to some extent. Moreover, despite the absence of approximated optimal expected revenue values calculated via DP for reference, the CDLP method still offers a theoretical upper bound on the optimal expected revenue, as mentioned in Section 6.1. In this example, the upper bound is 708. Therefore, it can be inferred that the performance of our RL-random policy is within a 5% gap from the performance of the optimal policy.

In terms of the computational cost, the total running times for Experiment 1 and Experiment 2 are 69,103 seconds and 95,818 seconds, respectively. Both experiments were conducted primarily on a NVIDIA Tesla V100-SXM2-16GB GPU, supported by an Intel Xeon Gold 6148 Skylake CPU @ 2.4 GHz. Surprisingly, the most significant computational bottleneck for the proposed algorithm in large-scale problems (hundreds of resources and products) is not the size of the state space, because we have approximated it with basis functions, but the RAM requirement due to the size of the action space. Recall that the firm may offer any one of the 2^n assortments upon the arrival of a customer. Because the RL policy is exploratory and will randomly choose one of the assortments, we use a vector to store the 2^n probabilities. For problems with a limited set of potential assortments, our algorithm can be executed more efficiently. Alternatively, one may truncate the set of assortments to explore or use Markov chain Monte Carlo procedures.

7 Conclusion

In this study, we provide a framework to adapt standard RL algorithms to continuous-time intensity control. Although we have laid the theoretical foundation and the algorithms have been shown to perform well numerically, the study also opens a number of important future directions. First, we have shown that by choosing a proper value function approximation, the discretization error associated with integrating the basis functions over time can be eliminated. It remains unknown what class of policy and function approximation can have the same property. Second, the convergence of the RL algorithms developed in this paper for continuous-time intensity control has yet to be established. Third, many new algorithms have been developed in the RL community (such as proximal policy optimization) and implemented in practice with success, most of which target discrete-time systems. It remains an open question to systematically convert these algorithms to the continuous-time control problems with possibly discrete state spaces (see e.g. Zhao et al. (2024) for a recent study on policy optimization for controlled diffusions).

References

Adelman, D. (2007). Dynamic bid prices in revenue management. *Operations Research 55*(4), 647–661.

Azagirre, X., A. Balwally, G. Candeli, N. Chamandy, B. Han, A. King, H. Lee, M. Loncaric, S. Martin, V. Narasiman, et al. (2024). A better match for drivers and riders: Reinforcement learning at lyft. *INFORMS Journal on Applied Analytics 54*(1), 71–83.

Bradtke, S. J. and M. O. Duff (1995). Reinforcement learning methods for continuous-time markov decision processes. *Advances in Neural Information Processing Systems 7 7*, 393.

Brémaud, P. (1981). Point processes and queues. *Springer*.

Chen, H. and D. D. Yao (1990). Optimal intensity control of a queueing system with state-dependent capacity limit. *IEEE Transactions on Automatic Control 35*(4), 459–464.

Dai, J. G. and M. Gluzman (2022). Queueing network controls via deep reinforcement learning. *Stochastic Systems 12*(1), 30–67.

Dai, M., Y. Dong, and Y. Jia (2023). Learning equilibrium mean-variance strategy. *Mathematical Finance 33*(4), 1166–1212.

Das, T. K., A. Gosavi, S. Mahadevan, and N. Marchalleck (1999). Solving semi-markov decision problems using average reward reinforcement learning. *Management Science 45*(4), 560–574.

Gallego, G., G. Iyengar, R. Phillips, and A. Dubey (2004). Managing flexible products on a network. *Working Paper*.

Gallego, G., H. Topaloglu, et al. (2019). *Revenue management and pricing analytics*, Volume 209. Springer.

Gallego, G. and G. Van Ryzin (1997). A multiproduct dynamic pricing problem and its applications to network yield management. *Operations research 45*(1), 24–41.

Gao, X., L. Li, and X. Y. Zhou (2024). Reinforcement learning for jump-diffusions. *arXiv preprint arXiv:2405.16449*.

Gao, X. and X. Y. Zhou (2022a). Logarithmic regret bounds for continuous-time average-reward markov decision processes. *arXiv preprint arXiv:2205.11168*.

Gao, X. and X. Y. Zhou (2022b). Square-root regret bounds for continuous-time episodic markov decision processes. *arXiv preprint arXiv:2210.00832*.

Gijsbrechts, J., R. N. Boute, J. A. Van Mieghem, and D. J. Zhang (2022). Can deep reinforcement learning improve inventory management? Performance on lost sales, dual-sourcing, and multi-echelon problems. *Manufacturing & Service Operations Management 24*(3), 1349–1368.

Guo, X., O. Hernández-Lerma, X. Guo, and O. Hernández-Lerma (2009). *Continuous-time Markov decision processes*. Springer.

Guo, X., X. Huang, and Y. Huang (2015). Finite-horizon optimality for continuous-time Markov decision processes with unbounded transition rates. *Advances in Applied Probability 47*(4), 1064–1087.

Guo, X., R. Xu, and T. Zariphopoulou (2022). Entropy regularization for mean field games with learning. *Mathematics of Operations research 47*(4), 3239–3260.

Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR.

Jia, Y. and X. Y. Zhou (2022a). Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *Journal of Machine Learning Research 23*(154), 1–55.

Jia, Y. and X. Y. Zhou (2022b). Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *The Journal of Machine Learning Research 23*(1), 12603–12652.

Jia, Y. and X. Y. Zhou (2023). q-learning in continuous time. *Journal of Machine Learning Research 24*(161), 1–61.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, T., C. Wang, Y. Wang, and S. Tang (2023). Deep reinforcement learning for online assortment customization: A data-driven approach. *Working Paper*.

Liu, Q. and G. Van Ryzin (2008). On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management 10*(2), 288–310.

Ma, Y., P. Rusmevichientong, M. Sumida, and H. Topaloglu (2020). An approximation algorithm for network revenue management under nonstationary arrivals. *Operations Research 68*(3), 834–855.

Oroojlooyjadid, A., M. Nazari, L. V. Snyder, and M. Takáč (2022). A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manufacturing & Service Operations Management 24*(1), 285–304.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Robbins, H. and S. Monro (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics 22*(3), 400 – 407.

Sinclair, S., T. Wang, G. Jain, S. Banerjee, and C. Yu (2020). Adaptive discretization for model-based reinforcement learning. *Advances in Neural Information Processing Systems 33*, 3858–3871.

Sinclair, S. R., S. Banerjee, and C. L. Yu (2023). Adaptive discretization in online reinforcement learning. *Operations Research 71*(5), 1636–1652.

Strauss, A. K., R. Klein, and C. Steinhardt (2018). A review of choice-based revenue management: Theory and methods. *European journal of operational research 271*(2), 375–387.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Tallec, C., L. Blier, and Y. Ollivier (2019). Making deep Q-learning methods robust to time discretization. In *International Conference on Machine Learning*, pp. 6096–6104. PMLR.

Talluri, K. and G. Van Ryzin (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science 50*(1), 15–33.

Wang, B., X. Gao, and L. Li (2023). Reinforcement learning for continuous-time optimal execution: actor-critic algorithm and error analysis. *Available at SSRN 4378950*.

Wang, H., T. Zariphopoulou, and X. Y. Zhou (2020). Reinforcement learning in continuous time and space: A stochastic control approach. *J. Mach. Learn. Res. 21*(198), 1–34.

Wang, H. and X. Y. Zhou (2020). Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance 30*(4), 1273–1308.

Wu, B. and L. Li (2024). Reinforcement learning for continuous-time mean-variance portfolio selection in a regime-switching market. *Journal of Economic Dynamics and Control 158*, 104787.

Zhang, D. (2011). An improved dynamic programming decomposition approach for network revenue management. *Manufacturing & Service Operations Management 13*(1), 35–52.

Zhang, D. and D. Adelman (2009). An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science 43*(3), 381–394.

Zhang, D. and W. L. Cooper (2005). Revenue management for parallel flights with customer-choice behavior. *Operations Research 53*(3), 415–431.

Zhao, H., W. Tang, and D. Yao (2024). Policy optimization for continuous reinforcement learning. *Advances in Neural Information Processing Systems 36*.

A The Reformulation of (7) in Section 2.2

For theoretical analysis, we introduce the exploratory dynamics of $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$, which can be interpreted as the average of trajectories of the sample state process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ over infinitely many randomized actions. In the case of controlled diffusions, Wang et al. (2020) derived the exploratory state process by applying a law of large number argument to the drift and diffusion coefficients of the controlled diffusion process. However, the sample state process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ in our setting is a pure jump process, so their approach does not apply. Instead, we derive the infinitesimal generator of the sample state process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$, from which we will identify the dynamics of the exploratory state process.

Given a randomized Markov policy $\boldsymbol{\pi}$, we now derive the generator of the Markov process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$, which is defined on the (enlarged) probability space $(\Omega, \mathcal{F}, \bar{\mathbb{P}}; \{\mathcal{F}_t\}_{t \ge 0})$. Given the time-state pair (t, x), consider M independent copies $S^{(1)}, \ldots, S^{(M)}$ of $S \sim \boldsymbol{\pi}(\cdot \mid t, x)$ and assume the control sampled at t is fixed from t to $t+s$. For all bounded and measurable function $f : \mathcal{S} \to \mathbb{R}$, we have

$$\begin{aligned}
&\lim_{s \to 0} \frac{\mathbb{E}^{\bar{\mathbb{P}}}[f(X_{t+s}^{\boldsymbol{\pi}}) \mid X_t^{\boldsymbol{\pi}} = x] - f(x)}{s} \\
&= \lim_{s \to 0} \frac{\lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}^{\mathbb{P}}[f(X_{t+s}^{S^{(i)}}) \mid X_t^{S^{(i)}} = x] - f(x)}{s} \\
&= \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \lim_{s \to 0} \frac{\mathbb{E}^{\mathbb{P}}[f(X_{t+s}^{S^{(i)}}) \mid X_t^{S^{(i)}} = x] - f(x)}{s} \\
&= \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} \lambda \sum_{j=1}^{n} P_j(S^{(i)}) \cdot [f(x - A^j) - f(x)]. \qquad (41)
\end{aligned}$$

Using the strong law of large numbers, we obtain

$$(41) = \lambda \sum_{j=1}^{n} \sum_{S \in \mathcal{A}} P_j(S) \boldsymbol{\pi}(S \mid t, x) \cdot [f(x - A^j) - f(x)]. \qquad (42)$$

One can observe from (42) that the effect of individually sampled actions has been averaged out. We can construct an ("averaged") exploratory process $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$, defined on the original probability space $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_t^{N^\lambda}\}_{t \ge 0})$, as a continuous-time Markov chain with the generator given by (42) and $\tilde{X}_0^{\boldsymbol{\pi}} = c$. Because the generator and the initial state of $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ and $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ are identical, we infer that the sample state process $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ under $\bar{\mathbb{P}}$ has the same distribution as the distribution of the exploratory state process $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ under $\mathbb{P}$. It follows that the value function (6) is identical to (7).

B Actor-Critic Algorithm with TD(0) for PE

Algorithm 2 Actor-Critic Algorithm (PE via TD(0) method)

Inputs: initial state c, time horizon T, number of iterations N, batch size M; functional forms of basis functions $\varphi_1, \dots, \varphi_W$, functional form of the policy $\boldsymbol{\pi}^{\phi}(\cdot \mid \cdot, \cdot)$ and an initial value ϕ_0; a temperature schedule function $\gamma(\cdot)$; initial learning rates α_ϕ and a learning rate schedule function $w(\cdot)$

Required program: an environment simulator $(t', x', S', r') = \mathit{Environment}(t, x, \boldsymbol{\pi}^{\phi}(\cdot \mid \cdot, \cdot))$

Initialize $\phi = \phi_0$

for $n = 1$ to N **do**

  for $i = 1$ to M **do** ▷ Generate M full trajectories under policy $\boldsymbol{\pi}^{\phi}$

    Store $(\tau_0^{(i)}, x_0^{(i)}) \leftarrow (0, c)$

    Initialize $l = 0$, $(t, x) = (0, c)$ ▷ Initialize l to count state transitions in each trajectory, and (t, x) to record the time and state right after a transition

    while True **do**

      Apply (t, x) to the environment simulator to get $(t', x', S', r') = \mathit{Environment}(t, x, \boldsymbol{\pi}^{\phi}(\cdot \mid \cdot, \cdot))$

      if $t' \geq T$ **then**

        break

      end if

      Update $l \leftarrow l + 1$

      Store current transition: $(\tau_l^{(i)}, x_l^{(i)}, S_l^{(i)}, r_l^{(i)}) \leftarrow (t', x', S', r')$

      Update $(t, x) \leftarrow (t', x')$

    end while

    Store $L^{(i)} \leftarrow l$, $\tau_{L^{(i)}+1}^{(i)} \leftarrow T$

  end for

  Evaluate policy $\boldsymbol{\pi}^{\phi}$: [using formula (26), incorporating techniques (27) and (28)]

$$
\theta^* = \left[\frac{1}{M} \sum_{i=1}^{M} \left(\sum_{l=0}^{L^{(i)}} F(\tau_l^{(i)}, \tau_{l+1}^{(i)}, x_l^{(i)}) + \sum_{l=1}^{L^{(i)}} \varphi(\tau_l^{(i)}, x_{l-1}^{(i)}) [\varphi(\tau_l^{(i)}, x_l^{(i)}) - \varphi(\tau_l^{(i)}, x_{l-1}^{(i)})] \right) \right]^{(-1)} \times
\left[\frac{1}{M} \sum_{i=1}^{M} \left(\sum_{l=1}^{L^{(i)}} \varphi(\tau_l^{(i)}, x_{l-1}^{(i)}) r_{\tau_l}^{(i)} + \gamma(n) \sum_{l=0}^{L^{(i)}} E(\tau_l^{(i)}, \tau_{l+1}^{(i)}, x_l^{(i)}, \varphi(\cdot, x_l^{(i)}); \boldsymbol{\pi}) \right) \right]
$$

  Compute policy gradient at ϕ: [using formula (31), incorporating technique (32)]

$$
\Delta\phi = \frac{1}{M} \sum_{i=1}^{M} \left(\sum_{l=1}^{L^{(i)}} \nabla_\phi \log \boldsymbol{\pi}^{\phi}(S_l^{(i)} \mid \tau_l^{(i)}, x_{l-1}^{(i)}) [J^{\theta^*}(\tau_l^{(i)}, x_l^{(i)}) - J^{\theta^*}(\tau_l^{(i)}, x_{l-1}^{(i)}) + r_l^{(i)}] + \gamma(n) \sum_{l=0}^{L^{(i)}} G(\tau_l^{(i)}, \tau_{l+1}^{(i)}, x_l^{(i)}; \boldsymbol{\pi}^{\phi}) \right)
$$

  Update ϕ by

$$
\phi \leftarrow \phi + w(n) \alpha_\phi \Delta\phi
$$

end for

C Proofs of Statements

Proof of Lemma 1. Let $C^{0,0}([0,T]\times\mathcal{S})$ be the space of all all real-valued functions defined on $[0,T]\times\mathcal{S}$ that are continuous in t over $[0,T]$ for all $x\in\mathcal{S}$. This space becomes a Banach space when endowed with the norm $\|\psi\| := \sup_{(t,x)\in[0,T]\times\mathcal{S}} |\psi(t,x)|$. We denote

$$R(t,x) = \sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid t,x) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid t,x)),$$

and then define an operator $\mathcal{L}$ on $C^{0,0}([0,T]\times\mathcal{S})$ by

$$\mathcal{L}\psi(t,x) := e^{\beta t}\int_t^T \left\{R(s,x) + \sum_{S\in\mathcal{A}}\left(\sum_{y\in\mathcal{S}} e^{-\beta s}\psi(s,y)q(y \mid s,x,S)\right)\boldsymbol{\pi}(S \mid s,x)\right\}ds, \tag{43}$$

where $\beta := 2\lambda+1$. Given that the integrand on the right-hand side of (43) is continuous with respect to s, it follows that the operator $\mathcal{L}$ is well-defined. Then, for $\psi_1,\psi_2\in C^{0,0}([0,T]\times\mathcal{S})$, we obtain

$$\begin{aligned}
|\mathcal{L}\psi_1(t,x) - \mathcal{L}\psi_2(t,x)| &\le e^{\beta t}\int_t^T e^{-\beta s}\left(\sum_{S\in\mathcal{A}}\sum_{y\in\mathcal{S}} |\psi_1(s,y)-\psi_2(s,y)|\cdot|q(y \mid s,x,S)|\,\boldsymbol{\pi}(S \mid s,x)\right)ds \\
&\le e^{\beta t}\int_t^T e^{-\beta s}\left(\|\psi_1-\psi_2\|\cdot\sum_{S\in\mathcal{A}}\left(\sum_{y\in\mathcal{S}}|q(y \mid s,x,S)|\right)\boldsymbol{\pi}(S \mid s,x)\right)ds \\
&\le 2\lambda\,\|\psi_1-\psi_2\|\,e^{\beta t}\int_t^T e^{-\beta s}ds \\
&\le \frac{2\lambda}{\beta}(1-e^{-\beta(T-t)})\,\|\psi_1-\psi_2\| \\
&\le \frac{2\lambda}{\beta}\,\|\psi_1-\psi_2\|.
\end{aligned}$$

Observing that $\frac{2\lambda}{\beta} = \frac{2\lambda}{2\lambda+1} < 1$, we identify $\mathcal{L}$ as a contraction operator on the Banach space $C^{0,0}([0,T]\times\mathcal{S})$. Let $\psi^*\in C^{0,0}([0,T]\times\mathcal{S})$ be the fixed point of $\mathcal{L}$, i.e.,

$$\psi^*(t,x) := e^{\beta t}\int_t^T \left\{R(s,x) + \sum_{S\in\mathcal{A}}\left(\sum_{y\in\mathcal{S}} e^{-\beta s}\psi^*(s,y)q(y \mid s,x,S)\right)\boldsymbol{\pi}(S \mid s,x)\right\}ds. \tag{44}$$

Let $v(t,x) = e^{-\beta t}\psi^*(t,x)$ for all $(t,x)\in[0,T]\times\mathcal{S}$, then v is continuously differentiable in t, i.e. $v\in C^{1,0}([0,T]\times\mathcal{S})$, and satisfies (8). Thus, we establish the existence of the solution to (8).

Suppose $v^*\in C^{1,0}([0,T]\times\mathcal{S})$ is a solution to (8). One can readily see that the transition rates of the Markov process $\tilde{X}^{\boldsymbol{\pi}}$, introduced in Appendix A, at time s is given by

$$\sum_{S\in\mathcal{A}} q(y \mid s,\tilde{X}^{\boldsymbol{\pi}}_{s-},S)\boldsymbol{\pi}(S \mid s,\tilde{X}^{\boldsymbol{\pi}}_{s-}), \quad y\neq\tilde{X}^{\boldsymbol{\pi}}_{s-}.$$

It follows from Theorem 3.1 in Guo et al. (2015) that we have the Dynkin's formula:

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}}\left[v^*(T,\tilde{X}_T^{\boldsymbol{\pi}}) \mid \tilde{X}_t^{\boldsymbol{\pi}} = x\right] - v^*(t,x) \\
&= \mathbb{E}^{\mathbb{P}}\left[\int_t^T \left(\frac{\partial v^*}{\partial s}(s,\tilde{X}_{s-}^{\boldsymbol{\pi}}) + \sum_{y\in\mathcal{S}} v^*(s,y)\Big[\sum_{S\in\mathcal{A}} q(y \mid s,\tilde{X}_{s-}^{\boldsymbol{\pi}},S)\boldsymbol{\pi}(S \mid s,\tilde{X}_{s-}^{\boldsymbol{\pi}})\Big]\right)ds \mid \tilde{X}_t^{\boldsymbol{\pi}} = x\right].
\end{aligned}
$$

Since v^* satisfies Equation (8) with $v^*(T,x)=0$, we then infer that

$$
\begin{aligned}
v^*(t,x) &= \mathbb{E}^{\mathbb{P}}\left[\int_t^T \left\{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s,\tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s,\tilde{X}_{s-}^{\boldsymbol{\pi}}))\right\}ds \mid \tilde{X}_t^{\boldsymbol{\pi}} = x\right] \\
&= J(t,x;\boldsymbol{\pi}), \quad (t,x)\in[0,T]\times\mathcal{S}.
\end{aligned}
$$

The proof is therefore complete. □

Proof of the optimal stochastic policy (9) *in Section 2.2.* Let the space $C^{0,0}([0,T]\times\mathcal{S})$, the endowed norm $\|\cdot\|$, and $R(t,x)$ be as specified in the proof of Lemma 1. We then define an operator $\bar{\mathcal{L}}$ on $C^{0,0}([0,T]\times\mathcal{S})$ by

$$
\bar{\mathcal{L}}\psi(t,x) := e^{\beta t}\int_t^T \sup_{\boldsymbol{\pi}\in\Pi}\left\{R(s,x) + \sum_{S\in\mathcal{A}}\left(\sum_{y\in\mathcal{S}} e^{-\beta s}\psi(s,y)q(y \mid s,x,S)\right)\boldsymbol{\pi}(S \mid s,x)\right\}ds,
$$

where $\beta := 2\lambda + 1$. Using a similar argument as in the proof of Lemma 1, we can identify $\bar{\mathcal{L}}$ as a contraction operator on the Banach space $C^{0,0}([0,T]\times\mathcal{S})$ and then establish the existence of the solution in the space $C^{1,0}([0,T]\times\mathcal{S})$ to the following exploratory HJB equation:

$$
\begin{cases}
\dfrac{\partial v}{\partial t}(t,x) + \sup\limits_{\boldsymbol{\pi}\in\Pi}\left\{\sum\limits_{S\in\mathcal{A}}[H(t,x,S,v(\cdot,\cdot)) - \gamma\log\boldsymbol{\pi}(S \mid t,x)]\boldsymbol{\pi}(S \mid t,x)\right\} = 0, \quad (t,x)\in[0,T)\times\mathcal{S}, \\
v(T,x) = 0, \quad x\in\mathcal{S}.
\end{cases}
\tag{45}
$$

Assume $v^*\in C^{1,0}([0,T]\times\mathcal{S})$ is a solution to (45), then $v^*(T,x)=0$ for all $x\in\mathcal{S}$ and the following inequality holds for all $\boldsymbol{\pi}\in\Pi$ and $(t,x)\in[0,T]\times\mathcal{S}$:

$$
\frac{\partial v^*}{\partial t}(t,x) + \sum_{S\in\mathcal{A}}\left\{H(t,x,S,v^*(\cdot,\cdot)) - \gamma\log\boldsymbol{\pi}(S \mid t,x)\right\}\boldsymbol{\pi}(S \mid t,x) \le 0. \tag{46}
$$

Then, it follows from Theorem 3.1 in Guo et al. (2015) that

$$
\begin{aligned}
v^*(t,x) &= \mathbb{E}^{\mathbb{P}}\left[v^*(T,\tilde{X}_T^{\boldsymbol{\pi}}) \mid \tilde{X}_t^{\boldsymbol{\pi}} = x\right] \\
&\quad - \mathbb{E}^{\mathbb{P}}\left[\int_t^T \left(\frac{\partial v^*}{\partial s}(s,\tilde{X}_{s-}^{\boldsymbol{\pi}}) + \sum_{y\in\mathcal{S}} v^*(s,y)\Big[\sum_{S\in\mathcal{A}} q(y \mid s,\tilde{X}_{s-}^{\boldsymbol{\pi}},S)\boldsymbol{\pi}(S \mid s,\tilde{X}_{s-}^{\boldsymbol{\pi}})\Big]\right)ds \mid \tilde{X}_t^{\boldsymbol{\pi}} = x\right] \\
&\ge \mathbb{E}^{\mathbb{P}}\left[\int_t^T \sum_{S\in\mathcal{A}}\{r(S) - \gamma\log\boldsymbol{\pi}(S \mid t,\tilde{X}_{s-}^{\boldsymbol{\pi}})\}\boldsymbol{\pi}(S \mid s,\tilde{X}_{s-}^{\boldsymbol{\pi}})ds \mid \tilde{X}_t^{\boldsymbol{\pi}} = x\right] \\
&= J(t,x;\boldsymbol{\pi}), \quad \text{for all } \boldsymbol{\pi}\in\Pi,\ (t,x)\in[0,T]\times\mathcal{S}.
\end{aligned}
$$

Moreover, the equality in (46) holds for policy $\boldsymbol{\pi}^*$ defined as follows:

$$\boldsymbol{\pi}^*(S \mid t, x) = \frac{\exp\{\frac{1}{\gamma} H(t, x, S, v^*(\cdot, \cdot))\}}{\sum_{S \in \mathcal{A}(x)} \exp\{\frac{1}{\gamma} H(t, x, S, v^*(\cdot, \cdot))\}}, \quad \text{for } (t, x) \in [0, T] \times \mathcal{S},\ S \in \mathcal{A}(x).$$

Thus, we conclude that $v^*(t, x) = J(t, x; \boldsymbol{\pi}^*) = J^*(t, x)$ for all $(t, x) \in [0, T] \times \mathcal{S}$. This establishes the characterization of the optimal policy in (9). $\square$

Proof of Theorem 2. To show that $\{\tilde{M}_t : 0 \le t \le T\}$ is an $(\{\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\}_{t \ge 0}, \mathbb{P})$-martingale, we first integrate the expression for $J(t, x; \boldsymbol{\pi})$ in (7), yielding

$$\begin{aligned} \tilde{M}_t = \mathbb{E}^{\mathbb{P}} &\left[\int_t^T \left\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \right\} ds \mid \tilde{X}_t^{\boldsymbol{\pi}} \right] \\ &+ \left[\int_0^t \left\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \right\} ds \right]. \end{aligned} \tag{47}$$

Due to the Markov property of $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$, and also note that the second term on the right side of equation (47) is $\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}$-measurable, we obtain

$$\tilde{M}_t = \mathbb{E}^{\mathbb{P}} \left[\int_0^T \left\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \right\} ds \mid \mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}} \right].$$

Since $J(T, x; \boldsymbol{\pi}) = 0$ for all $x \in \mathcal{S}$, we conclude that $\tilde{M}_t = \mathbb{E}^{\mathbb{P}}[\tilde{M}_T \mid \mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}]$, which implies $\{\tilde{M}_t : 0 \le t \le T\}$ is an $(\{\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\}_{t \ge 0}, \mathbb{P})$-martingale. Moreover, since $\tilde{M}_t$ is bounded for any fixed time $t \in [0, T]$, it follows that $\{\tilde{M}_t : 0 \le t \le T\}$ is square-integrable.

Conversely, assume that $\{\tilde{M}_t^v : 0 \le t \le T\}$ is an $(\{\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\}_{t \ge 0}, \mathbb{P})$-martingale and $v(T, x) = 0$ for all $x \in \mathcal{S}$. From this, it follows directly that $\tilde{M}_t^v = \mathbb{E}^{\mathbb{P}}[\tilde{M}_T^v \mid \mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}]$for all $t \in [0, T]$, which is equivalent to

$$\begin{aligned} v(t, \tilde{X}_t^{\boldsymbol{\pi}}) = \mathbb{E}^{\mathbb{P}} &\left[\int_0^T \left\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \right\} ds \mid \mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}} \right] \\ &- \left[\int_0^t \left\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \right\} ds \right], \quad t \in [0, T]. \end{aligned} \tag{48}$$

Since the second term on the right side of Equation (48) is $\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}$-measurable, then it follows from the Markov property of $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ that

$$\begin{aligned} v(t, \tilde{X}_t^{\boldsymbol{\pi}}) &= \mathbb{E}^{\mathbb{P}} \left[\int_t^T \left\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \right\} ds \mid \mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}} \right] \\ &= \mathbb{E}^{\mathbb{P}} \left[\int_t^T \left\{ \sum_{S \in \mathcal{A}} r(S) \boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})) \right\} ds \mid \tilde{X}_t^{\boldsymbol{\pi}} \right] \\ &= J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}), \quad t \in [0, T]. \end{aligned}$$

Therefore, we conclude that $v(t, x) = J(t, x; \boldsymbol{\pi})$ for all $(t, x) \in [0, T] \times \mathcal{S}$. $\square$

Proof of Theorem 1. Denote $U_t \coloneqq \int_{(t,T]} p^\top dN_s^{\boldsymbol{\pi}} - \int_t^T \sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, X_{s-}^{\boldsymbol{\pi}})ds$, then we have

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[U_t \mid X_t^{\boldsymbol{\pi}}\right] = 0. \tag{49}$$

We further denote $M_t \coloneqq J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + \int_0^t \{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, X_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))\}ds$ and $M_t^\theta \coloneqq J^\theta(t, X_t^{\boldsymbol{\pi}}) + \int_0^t \{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, X_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))\}ds$. It follows from $J(T, x; \boldsymbol{\pi}) \equiv 0$ and (49) that

$$\begin{aligned}
2L(\theta) &= \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T (U_t + M_T - M_t^\theta)^2 dt\right] \\
&= \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T [U_t^2 + 2U_t(M_T - M_t^\theta) + (M_T - M_t^\theta)^2] dt\right] \\
&= \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \left[U_t^2 + 2U_t\left(\int_t^T \left\{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, X_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))\right\}ds\right)\right]dt\right] \\
&\quad - \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T 2J^\theta(t, X_t^{\boldsymbol{\pi}})\mathbb{E}^{\bar{\mathbb{P}}}[U_t \mid X_t^{\boldsymbol{\pi}}]dt\right] + \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T (M_T - M_t^\theta)^2 dt\right] \\
&= \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \left[U_t^2 + 2U_t\left(\int_t^T \left\{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, X_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, X_{s-}^{\boldsymbol{\pi}}))\right\}ds\right)\right]dt\right] \\
&\quad + \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T (M_T - M_t^\theta)^2 dt\right].
\end{aligned}$$

Since the first term does not rely on θ, we have

$$\arg\min_\theta L(\theta) = \arg\min_\theta \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T (M_T - M_t^\theta)^2 dt\right]. \tag{50}$$

Next, we denote $\tilde{M}_t^\theta \coloneqq J^\theta(t, \tilde{X}_t^{\boldsymbol{\pi}}) + \int_0^t \left\{\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \gamma\mathcal{H}(\boldsymbol{\pi}(\cdot \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}))\right\}ds$. From Theorem 2 we know that $\{\tilde{M}_t : 0 \le t \le T\}$ in (15) is an $(\{\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\}_{t\ge 0}, \mathbb{P})$-martingale. Therefore,

$$\begin{aligned}
\mathbb{E}^{\mathbb{P}}\left[\int_0^T (\tilde{M}_T - \tilde{M}_t^\theta)^2 dt\right] &= \mathbb{E}^{\mathbb{P}}\left[\int_0^T (\tilde{M}_T - \tilde{M}_t + \tilde{M}_t - \tilde{M}_t^\theta)^2 dt\right] \\
&= \mathbb{E}^{\mathbb{P}}\left[\int_0^T [(\tilde{M}_T - \tilde{M}_t)^2 + (\tilde{M}_t - \tilde{M}_t^\theta)^2 + 2(\tilde{M}_T - \tilde{M}_t)(\tilde{M}_t - \tilde{M}_t^\theta)]dt\right] \\
&= \mathbb{E}^{\mathbb{P}}\left[\int_0^T (\tilde{M}_T - \tilde{M}_t)^2 dt\right] + \mathbb{E}^{\mathbb{P}}\left[\int_0^T (\tilde{M}_t - \tilde{M}_t^\theta)^2 dt\right] \\
&\quad + 2\int_0^T \mathbb{E}^{\mathbb{P}}\left[(\tilde{M}_t - \tilde{M}_t^\theta) \cdot \mathbb{E}^{\mathbb{P}}\left[\tilde{M}_T - \tilde{M}_t \mid \mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\right]\right] dt \\
&= \mathbb{E}^{\mathbb{P}}\left[\int_0^T (\tilde{M}_T - \tilde{M}_t)^2 dt\right] + \mathbb{E}^{\mathbb{P}}\left[\int_0^T |J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - J^\theta(t, \tilde{X}_t^{\boldsymbol{\pi}})|^2 dt\right].
\end{aligned}$$

Noting that the first term does not rely on θ, we obtain

$$\arg\min_\theta \mathbb{E}^{\mathbb{P}}\left[\int_0^T |\tilde{M}_T - \tilde{M}_t^\theta|^2 dt\right] = \arg\min_\theta \mathbb{E}^{\mathbb{P}}\left[\int_0^T |J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - J^\theta(t, \tilde{X}_t^{\boldsymbol{\pi}})|^2 dt\right].$$

Since the distribution of $\{\tilde{X}_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ under $\mathbb{P}$ is the same as the distribution of $\{X_t^{\boldsymbol{\pi}} : 0 \le t \le T\}$ under $\bar{\mathbb{P}}$, together with (50), we conclude that $\arg\min_\theta L(\theta) = \arg\min_\theta \mathrm{MSVE}(\theta)$. □

Proof of Theorem 3. Note that a bounded process ξ with $\xi_t \in \mathcal{F}_{t-}^{X^{\boldsymbol{\pi}}}$ is also predictable w.r.t $\mathcal{F}_t$. On the other hand, for each $j = 1, \cdots, n$, the process $\{N_{j,t}^{\boldsymbol{\pi}} - \int_0^t \lambda P_j(S_s^{\boldsymbol{\pi}}) ds : 0 \le t \le T\}$ is a square-integrable $(\{\mathcal{F}_t\}_{t\ge 0}, \bar{\mathbb{P}})$-martingale. Thus we have

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \xi_t (p^\top dN_t^{\boldsymbol{\pi}})\right] = \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \xi_t \cdot r(S_t^{\boldsymbol{\pi}}) dt\right] = \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \xi_t \Big[\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid t, X_{t-}^{\boldsymbol{\pi}})\Big] dt\right].$$

It suffices to prove that $v(t,x) = J(t,x;\boldsymbol{\pi})$ for all $(t,x) \in [0,T] \times \mathcal{S}$, if and only if v satisfies $v(T,x) = 0$ for all $x \in \mathcal{S}$, and

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T \xi_t \Big\{ dv(t, X_t^{\boldsymbol{\pi}}) + \Big[\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid t, X_{t-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, X_{t-}^{\boldsymbol{\pi}}))\Big] dt \Big\}\right] \tag{51}$$

for any bounded process ξ with $\xi_t \in \mathcal{F}_{t-}^{X^{\boldsymbol{\pi}}}$.

We first establish that $v(t,x) = J(t,x;\boldsymbol{\pi})$ for all $(t,x) \in [0,T] \times \mathcal{S}$, if and only if v satisfies $v(T,x) = 0$ for all $x \in \mathcal{S}$, and

$$\mathbb{E}^{\mathbb{P}}\left[\int_0^T \tilde{\xi}_t \Big\{ dv(t, \tilde{X}_t^{\boldsymbol{\pi}}) + \Big[\sum_{S\in\mathcal{A}} r(S)\boldsymbol{\pi}(S \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}}))\Big] dt \Big\}\right] = 0, \tag{52}$$

for any bounded process $\tilde{\xi}$ with $\tilde{\xi}_t \in \mathcal{F}_{t-}^{\tilde{X}^{\boldsymbol{\pi}}}$. The "only if" part follows immediately from Theorem 2. To establish the "if" part, assume that $v(T,x) = 0$ for all $x \in \mathcal{S}$, and (52) holds for any bounded process $\tilde{\xi}$ with $\tilde{\xi}_t \in \mathcal{F}_{t-}^{\tilde{X}^{\boldsymbol{\pi}}}$. It follows from Theorem 3.1 of Guo et al. (2015) that the process

$$v(t, \tilde{X}_t^{\boldsymbol{\pi}}) - \int_0^t \left(\frac{\partial v}{\partial s}(s, \tilde{X}_{s-}^{\boldsymbol{\pi}}) + \sum_{y\in\mathcal{S}} v(s,y) \Big[\sum_{S\in\mathcal{A}} q(y \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}}, S)\boldsymbol{\pi}(S \mid s, \tilde{X}_{s-}^{\boldsymbol{\pi}})\Big] \right) ds$$

defines an $(\{\mathcal{F}_t^{\tilde{X}^{\boldsymbol{\pi}}}\}_{t\ge 0}, \mathbb{P})$-martingale, and its boundedness at any fixed time $t \in [0,T]$ ensures square-integrability. Then, it follows that

$$\mathbb{E}^{\mathbb{P}}\Big[\int_0^T \tilde{\xi}_t \Big\{ dv(t, \tilde{X}_t^{\boldsymbol{\pi}}) - \Big(\frac{\partial v}{\partial t}(t, \tilde{X}_{t-}^{\boldsymbol{\pi}}) + \sum_{y\in\mathcal{S}} v(t,y) \Big[\sum_{S\in\mathcal{A}} q(y \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}}, S)\boldsymbol{\pi}(S \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}})\Big] \Big) dt \Big\}\Big] = 0, \tag{53}$$

for any bounded process $\tilde{\xi}$ with $\tilde{\xi}_t \in \mathcal{F}_{t-}^{\tilde{X}^{\boldsymbol{\pi}}}$. By taking the difference between equations (52) and (53), we obtain that

$$\mathbb{E}^{\mathbb{P}}\left[\int_0^T \tilde{\xi}_t \Big(\frac{\partial v}{\partial t}(t, \tilde{X}_{t-}^{\boldsymbol{\pi}}) + \sum_{S\in\mathcal{A}} H(t, \tilde{X}_{t-}^{\boldsymbol{\pi}}, S, v(\cdot,\cdot))\boldsymbol{\pi}(S \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}})) \Big) dt\right] = 0 \tag{54}$$

holds for any bounded process $\tilde{\xi}$ with $\tilde{\xi}_t \in \mathcal{F}_{t-}^{\tilde{X}^{\boldsymbol{\pi}}}$. Define the test function $\tilde{\xi}_t$ as follows:

$$\tilde{\xi}_t = \mathrm{sgn}\left(\frac{\partial v}{\partial t}(t, \tilde{X}_{t-}^{\boldsymbol{\pi}}) + \sum_{S\in\mathcal{A}} H(t, \tilde{X}_{t-}^{\boldsymbol{\pi}}, S, v(\cdot,\cdot))\boldsymbol{\pi}(S \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}}) + \gamma \mathcal{H}(\boldsymbol{\pi}(\cdot \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}})) \right),$$

where sgn(·) is the sign function. Then, (54) implies that

$$\frac{\partial v}{\partial t}(t,\tilde{X}^{\boldsymbol{\pi}}_{t-})+\sum_{S\in\mathcal{A}}H(t,\tilde{X}^{\boldsymbol{\pi}}_{t-},S,v(\cdot,\cdot))\boldsymbol{\pi}(S\mid t,\tilde{X}^{\boldsymbol{\pi}}_{t-})+\gamma\mathcal{H}(\boldsymbol{\pi}(\cdot\mid t,\tilde{X}^{\boldsymbol{\pi}}_{t-}))=0,\quad t\in[0,T]. \tag{55}$$

It follows from Lemma 1 that $v(t,x)=J(t,x;\boldsymbol{\pi})$ for all $(t,x)\in[0,T]\times\mathcal{S}$. Hence, we complete the proof of the equivalent condition (52).

Let $D[0,T]$ denote the space of all mappings $f:[0,T]\mapsto\mathcal{S}$. For any process $Y\in D[0,T]$ and fixed $t\in[0,T]$, define the stopped process $Y^{\boldsymbol{\pi}}_{t-\wedge\cdot}\in D[0,T]$ such that $Y^{\boldsymbol{\pi}}_{t-\wedge\cdot}(s)=Y^{\boldsymbol{\pi}}_{s}$ for $s\in[0,t)$, and $Y^{\boldsymbol{\pi}}_{t-\wedge\cdot}(s)=Y^{\boldsymbol{\pi}}_{t-}$ for $s\in[t,T]$. Note that any process ξ with $\xi_t\in\mathcal{F}^{X^{\boldsymbol{\pi}}}_{t-}$ corresponds to a measurable function $\boldsymbol{\xi}:[0,T]\times D([0,T])\mapsto\mathbb{R}$ such that $\xi_t=\boldsymbol{\xi}(t,X^{\boldsymbol{\pi}}_{t-\wedge\cdot})$. Then, $\boldsymbol{\xi}(t,\tilde{X}^{\boldsymbol{\pi}}_{t-\wedge\cdot})$ is $\mathcal{F}^{\tilde{X}^{\boldsymbol{\pi}}}_{t-}$-measurable for each $t\in[0,T]$. Since the distribution of $\{X^{\boldsymbol{\pi}}_t:0\le t\le T\}$ under $\bar{\mathbb{P}}$ is the same as the distribution of $\{\tilde{X}^{\boldsymbol{\pi}}_t:0\le t\le T\}$ under $\mathbb{P}$, it follows that

$$\begin{aligned}&\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T\boldsymbol{\xi}(t,X^{\boldsymbol{\pi}}_{t-\wedge\cdot})\left\{dv(t,X^{\boldsymbol{\pi}}_t)+\Big[\sum_{S\in\mathcal{A}}r(S)\boldsymbol{\pi}(S\mid t,X^{\boldsymbol{\pi}}_{t-})+\gamma\mathcal{H}(\boldsymbol{\pi}(\cdot\mid t,X^{\boldsymbol{\pi}}_{t-}))\Big]dt\right\}\right]\\&=\mathbb{E}^{\mathbb{P}}\left[\int_0^T\boldsymbol{\xi}(t,\tilde{X}^{\boldsymbol{\pi}}_{t-\wedge\cdot})\left\{dv(t,\tilde{X}^{\boldsymbol{\pi}}_t)+\Big[\sum_{S\in\mathcal{A}}r(S)\boldsymbol{\pi}(S\mid t,\tilde{X}^{\boldsymbol{\pi}}_{t-})+\gamma\mathcal{H}(\boldsymbol{\pi}(\cdot\mid t,\tilde{X}^{\boldsymbol{\pi}}_{t-}))\Big]dt\right\}\right].\end{aligned} \tag{56}$$

Conversely, any process $\tilde{\xi}$ with $\tilde{\xi}_t\in\mathcal{F}^{\tilde{X}^{\boldsymbol{\pi}}}_{t-}$ corresponds to a measurable function $\tilde{\boldsymbol{\xi}}:[0,T]\times D([0,T])\mapsto\mathbb{R}$ such that $\tilde{\xi}_t=\tilde{\boldsymbol{\xi}}(t,\tilde{X}^{\boldsymbol{\pi}}_{t-\wedge\cdot})$. Then, $\tilde{\boldsymbol{\xi}}(t,X^{\boldsymbol{\pi}}_{t-\wedge\cdot})$ is $\mathcal{F}^{X^{\boldsymbol{\pi}}}_{t}$-measurable for each $t\in[0,T]$, and equation (56) holds for $\tilde{\boldsymbol{\xi}}(\cdot,\cdot)$. This establishes the equivalence between condition (52) and condition (51). The proof is therefore complete. □

Proof of Theorem 4. By the representation (30) of $g(t,x;\phi)=\nabla_\phi J(t,x;\boldsymbol{\pi}^\phi)$, we have

$$\begin{aligned}\nabla_\phi J(0,c;\boldsymbol{\pi}^\phi)=\mathbb{E}^{\mathbb{P}}\Bigg[&\int_0^T\Big(\sum_{S\in\mathcal{A}}\nabla_\phi\log\boldsymbol{\pi}^\phi(S\mid t,\tilde{X}^{\boldsymbol{\pi}^\phi}_{t-})H(t,\tilde{X}^{\boldsymbol{\pi}^\phi}_{s-},S,J(\cdot,\cdot;\boldsymbol{\pi}^\phi))\boldsymbol{\pi}^\phi(S\mid t,\tilde{X}^{\boldsymbol{\pi}^\phi}_{s-})\Big)dt\\&+\gamma\int_0^T\nabla_\phi\mathcal{H}(\boldsymbol{\pi}^\phi(\cdot\mid t,\tilde{X}^{\boldsymbol{\pi}^\phi}_{t-}))dt\Bigg].\end{aligned} \tag{57}$$

Since the distribution of $\{\tilde{X}^{\boldsymbol{\pi}}_t:0\le t\le T\}$ under $\mathbb{P}$ is the same as the distribution of $\{X^{\boldsymbol{\pi}}_t:0\le t\le T\}$ under $\bar{\mathbb{P}}$, it follows that the second term of (57) equals to the second term of (31).

It remains to show the first term of (57) also equals to the first term of (31). Denote the following expression as $I(t,x,S)$:

$$I(t,x,S):=\nabla_\phi\log\boldsymbol{\pi}^\phi(S\mid t,x)H(t,x,S,J(\cdot,\cdot;\boldsymbol{\pi}^\phi))$$

which is a real-valued function on $\mathcal{S}\times\mathcal{A}$ for fixed $t\in[0,T]$. Then we have

$$\mathbb{E}^{\bar{\mathbb{P}}}\Big[I(t,X^{\boldsymbol{\pi}^\phi}_{t-},S^{\boldsymbol{\pi}^\phi}_t)\Big]=\mathbb{E}^{\mathbb{P}}\left[\sum_{S\in\mathcal{A}}I(t,\tilde{X}^{\boldsymbol{\pi}^\phi}_{t-},S)\boldsymbol{\pi}^\phi(S\mid t,\tilde{X}^{\boldsymbol{\pi}^\phi}_{t-})\right]. \tag{58}$$

By Assumption 1 and the fact that both the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite, we have $\sup_{(t,x,S)\in[0,T]\times\mathcal{S}\times\mathcal{A}} |I(t,x,S)| < \infty$. Then, integrating (58) over $[0,T]$ with respect to t and applying Fubini's theorem gives:

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T I(t, X_{t-}^{\boldsymbol{\pi}^\phi}, S_t^{\boldsymbol{\pi}^\phi})dt\right] = \mathbb{E}^{\mathbb{P}}\left[\int_0^T \left(\sum_{S\in\mathcal{A}} I(t, \tilde{X}_{t-}^{\boldsymbol{\pi}^\phi}, S)\boldsymbol{\pi}^\phi(S \mid t, \tilde{X}_{t-}^{\boldsymbol{\pi}^\phi})\right)dt\right]. \tag{59}$$

Given the definition of H in (3), we have

$$\begin{aligned} H(t,x,S,J(\cdot,\cdot;\boldsymbol{\pi}^\phi)) &= \lambda\sum_{j=1}^n p_j P_j(S) + \sum_{y\neq x}\left(\sum_{\{j\in\mathcal{J}:A^j=x-y\}} \lambda P_j(S)\right)J(t,y;\boldsymbol{\pi}^\phi) - \lambda[1-P_0(S)]J(t,x;\boldsymbol{\pi}^\phi) \\ &= \sum_{j=1}^n [J(t,x-A^j;\boldsymbol{\pi}^\phi) - J(t,x;\boldsymbol{\pi}^\phi) + p_j]\lambda P_j(S). \end{aligned}$$

For each $j = 1,\ldots,n$, we denote $I_j(t,x,S) \coloneqq \nabla_\phi \log \boldsymbol{\pi}^\phi(S \mid t,x)[J(t,x-A^j;\boldsymbol{\pi}^\phi) - J(t,x;\boldsymbol{\pi}^\phi) + p_j]$. This leads to $I(t,x,S) = \sum_{j=1}^n I_j(t,x,S)\lambda P_j(S)$. Then, we claim that for any $j = 1,\ldots,n$, the following equality holds:

$$\mathbb{E}^{\bar{\mathbb{P}}}\left[\int_{(0,T]} I_j(t, X_{t-}^{\boldsymbol{\pi}^\phi}, S_t^{\boldsymbol{\pi}^\phi})dN_{j,t}^{\boldsymbol{\pi}^\phi}\right] = \mathbb{E}^{\bar{\mathbb{P}}}\left[\int_0^T I_j(t, X_{t-}^{\boldsymbol{\pi}^\phi}, S_t^{\boldsymbol{\pi}^\phi})\lambda P_j(S_t^{\boldsymbol{\pi}^\phi})dt\right]. \tag{60}$$

Indeed, for each $j = 1,\ldots,n$, the process $\{N_{j,t}^{\boldsymbol{\pi}^\phi} - \int_0^t \lambda P_j(S_s^{\boldsymbol{\pi}^\phi})ds : 0 \leq t \leq T\}$ is a square-integrable $(\{\mathcal{F}_t\}_{t\geq 0}, \bar{\mathbb{P}})$-martingale. Additionally, the process $\{I_j(t, X_{t-}^{\boldsymbol{\pi}^\phi}, S_t^{\boldsymbol{\pi}^\phi}) : 0 \leq t \leq T\}$ is bounded according to Assumption 1. Thus, the equality (60) holds.

By combining (59) and (60), we can infer that the first term of (57) also equals to the first term of (31). The proof is hence complete. □