Structured Strategies for Learning and Exploration in Sequential Decision Making

by

Yijia Wang

B.S. in Logistics Engineering, Tianjin University, 2013

M.S. in Management Science, Tianjin University, 2016

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Yijia Wang

It was defended on

April 5, 2022

and approved by

Daniel Jiang, Ph.D, Assistant Professor, Department of Industrial Engineering

Lisa Maillart, Ph.D, Professor, Department of Industrial Engineering

Jayant Rajgopal, Ph.D, Professor, Department of Industrial Engineering

Matthias Poloczek, Ph.D, Principal Scientist, Amazon

Jeffrey Kharoufeh, Ph.D, Professor, Department of Industrial Engineering, Clemson University

Dissertation Director: Daniel Jiang, Ph.D, Assistant Professor, Department of Industrial Engineering

Structured Strategies for Learning and Exploration in Sequential Decision Making

Yijia Wang, PhD

University of Pittsburgh, 2022

Solving Markov decision processes (MDPs) efficiently is challenging in many cases, for example, when the state space or action space is large, when the reward function is sparse and delayed, and when there is a distribution of MDPs. Structures in the policy, value function, reward function, or the state space can be useful in accelerating the learning process. In this thesis, we exploit structures in MDPs to solve them effectively and efficiently. First, we study problems with concave value function and basestock policy, and leverage these two structures to propose an approximate dynamic programming (ADP) algorithm. Next, we study the exploration problem in unknown MDPs, introduce structured intrinsic reward to the problem, and propose a Bayes-optimal algorithm for learning the intrinsic reward. Finally, we move to problems with structured state space (slow and fast state), build a hierarchical model which exploits the structure, and propose ADP algorithms for the hierarchical model.

Keywords: Markov decision processes; Approximate dynamic programming; Reinforcement learning.

Table of Contents

List of Tables

List of Figures

1.0 Introduction

In this thesis, we study the problem of leveraging structure in Markov decision processes (MDPs) to solve them efficiently. Specifically, we focus on structures in the policy space, value space, reward space, and state space. We examine the benefits of such structures by studying the following problems:

1. We study the problem of controlling inventory and dispensing the inventory to groups of people in Chapter 2. In this problem, we consider demand nonstationarity and limited storage capacity. The optimal policy for this problem is difficult to find exactly when the state space is large, when the stochastic models are unknown, or when data on demand is collected slowly over time. We propose a data-driven method that leverages structure in both the policy and the value function ((state-dependent basestocks and concavity, respectively), and show that this method can discover near-optimal policies.

2. We then study the problem of exploration in a distribution of unknown MDPs with sparse and delayed rewards in Chapter 3. Finding the optimal policy is expensive and time-consuming. We introduce subgoals with an intrinsic shaped reward to the problem. The structured rewards provided by well-designed subgoals can efficiently guide the exploration process by stimulating the agent to explore more in the important states. We propose a one-step Bayes-optimal algorithm that iteratively finds the optimal subgoal design.

3. In Chapter 4, we consider infinite horizon MDPs with *fast-slow* structure, meaning that certain parts of the state space move "fast" (and are more influential) while other parts of the state space transition more slowly (and are less influential). We propose hierarchical value iteration algorithms based on the idea of "freezing" the slow states, solving a set of finite-horizon MDPs, and applying value iteration to an auxiliary MDP that transitions on a slower timescale.

1.1 Structured Actor-Critic for Managing Public Health Points-of-Dispensing

Public health organizations face the problem of dispensing treatments (i.e., vaccines, antibiotics, and others) to groups of affected populations through "points-of-dispensing" (PODs) during emergency situations, typically in the presence of complexities like demand stochasticity, heterogenous utilities (e.g., for vaccine distribution, certain segments of the population may need to be prioritized), and limited storage. We formulate a hierarchical MDP model with two levels of decisions (and decision-makers): the upper-level decisions come from an inventory planner that "controls" a lower-level dynamic problem, which optimizes dispensing decisions that take into consideration the heterogeneous utility functions of the random set of PODs. We then derive structural properties of the MDP model and propose an approximate dynamic programming (ADP) algorithm that leverages structure in both the *policy* and the *value* space (state-dependent basestocks and concavity, respectively). The algorithm can be considered an *actor-critic* method; to our knowledge, this chapter is the first to jointly exploit policy and value structure within an actor-critic framework. We prove that the policy and value function approximations each converge to their optimal counterparts with probability one and provide a comprehensive numerical analysis showing improved empirical convergence rates when compared to other ADP techniques. Finally, we show how an aggregation-based version of our algorithm can be applied in a realistic case study for the problem of dispensing naloxone (an overdose reversal drug) via first responders amidst the ongoing opioid crisis.

1.2 Subgoal-based Exploration via Bayesian Optimization

Policy optimization in unknown, sparse-reward environments with expensive and limited interactions is challenging, and poses a need for effective exploration. Motivated by complex navigation tasks that require real-world training (when cheap simulators are not available), we consider an agent that faces an unknown distribution of environments and must decide on an exploration strategy, through a series of training environments, that can benefit policy

learning in a test environment drawn from the environment distribution. Most existing approaches focus on fixed exploration strategies, while the few that view exploration as a meta-optimization problem tend to ignore the need for *cost-efficient* exploration. We propose a cost-aware Bayesian optimization approach that efficiently searches over a class of dynamic subgoal-based exploration strategies. The algorithm adjusts a variety of levers — the locations of the subgoals, the length of each episode, and the number of replications per trial — in order to overcome the challenges of sparse rewards, expensive interactions, and noise. Our experimental evaluation demonstrates that, when averaged across problem domains, the proposed algorithm outperforms the meta-learning algorithm MAML by 19%, the hyperparameter tuning method Hyperband by 23%, BO techniques EI and LCB by 24% and 22%, respectively. We also provide a theoretical foundation and prove that the method asymptotically identifies a near-optimal subgoal design from the search space.

1.3 Frozen-State Approximate Value Iteration for Fast-Slow Markov Decision Processes

In this chapter, we consider infinite horizon MDPs with *fast-slow* structure, meaning that certain parts of the state space move "fast" (and are more influential) while other parts of the state space transition more "slowly" (and are less influential). Such structure is common in important real-world problems where sequential decisions need to be made at high frequencies, yet information that varies at a slower timescale also influences the optimal policy. Examples include: (1) service allocation for a multi-class queue with (slowly varying) stochastic costs, (2) energy demand response, where both day-ahead and real-time prices play a role in the firm's revenue, and (3) joint multi-product inventory control and pricing, where the expected demand of some products is low and the expected demand of the other products are high. Models that fully capture these problems often result in MDPs with large state spaces and large effective time horizons (due to frequent decisions), rendering them computationally intractable. We propose an algorithmic framework based on the idea of "freezing" the slow states, solving a set of simpler finite-horizon MDPs (the *lower-level*

MDPs), and applying value iteration (VI) to an auxiliary MDP that transitions on a slower timescale (the *upper-level* MDP). We also show how this technique can be applied in an ADP setting, where a feature-based approximation is used. On the theoretical side, we analyze the expected regret incurred by the policies obtained via our frozen-state approach, provide explicit guidance on the number of periods to use in the lower-level MDP, and derive error bounds for the ADP approach. Finally, we give empirical evidence that the frozen-state approach generates effective policies using just a fraction of the computational cost.

2.0 Structured Actor-Critic for Managing Public Health PODs

Public health organizations manage "points-of-dispensing" (PODs), operated by first responders or first receivers [1], for distributing critical medical supplies during emergency situations (e.g., the ongoing opioid crisis, the COVID-19 pandemic, the 2009 H1N1 influenza pandemic, meningitis outbreaks). In this chapter, we consider the hierarchical and sequential problem of optimizing inventory control and making dispensing decisions for multiple PODs. Our problem setting is specifically motivated by the ongoing opioid overdose *harm reduction* efforts of public health organizations in cities across the U.S., where the opioid epidemic was declared a public health emergency in 2017. In particular, our modeling is motivated by the Naloxone for First Responders Program (NFRP), a statewide naloxone distribution initiative in Pennsylvania. Unfortunately, despite the efforts of these organizations, there are often shortages of naloxone [2, 3]. Meanwhile, the severity of the opioid crisis has worsened during the COVID-19 pandemic [4]. This intersection of naloxone shortages with the increasing numbers of overdose incidents suggests that the careful management and dispensing of naloxone inventory is a particularly relevant problem.

Our setup contains *hierarchical decisions* in order to model the interplay between two decision-makers: the "upper-level" central inventory manager and a "lower-level" dispensing coordinator. The NFRP is an example of an organization that operates in this manner through the use of a centralized coordinating entity (CCE) that manages dispensing. Both decision-makers make sequential and non-myopic decisions (at different timescales) and are modeled using Markov decision processes (MDPs). Another novel point of emphasis for our model is the notion that the effectiveness of the public health intervention can vary across different groups of the affected population [5] and across different locations. Therefore, instead of modeling demand in a static and homogeneous manner, we consider the case where at each period, new demand information is revealed sequentially as a POD attribute and demand. The dispensing decisions are made according to the arrivals of PODs with a goal of maximizing total utility. The essential trade-off considered by the dispensing

coordinator is: *should we satisfy a lower-priority demand now, or save the inventory for a possible higher-priority demand in the future?*

The model we develop, however, is quite general and useful for related problems in public health as well where hierarchical decisions and demand heterogeneity may be an issue (e.g., vaccine distribution, where certain segments of the population are more susceptible and where higher-level inventory and lower-level dispensing decisions might be made separately). Other important characteristics of this problem include demand nonstationarity and the potential for limited storage capacity.

Exact computation of the optimal policy for this model is difficult when the number of states is large, when the stochastic models are unknown, or when demand data is collected slowly over time. The main methodological contribution of the chapter addresses these issues through a *structured actor-critic* algorithm; our proposed method exploits structure in both the *policy* and the *value function* and can discover near-optimal policies in a fully data-driven way. Our algorithm uses several gradient updates on each iteration and thus is highly suitable for the situation where data arrives in an ongoing fashion and online updates are desired. In other words, a large batch of historical data is not required for our algorithm and the policy can be learned over time. We now give five examples of public health problems for which our model and algorithm are applicable.

Example 2.0.1 (Opioid Overdose Epidemic)**.** *The rate of opioid overdose deaths tripled between 2000 and 2014 in the United States [6]. More recently, in July 2017, it was estimated that there are 142 American deaths each day due to overdose [7]. Naloxone is a drug that has the ability to reverse overdoses within seconds to minutes. To save lives amidst the current opioid epidemic, it is critical for naloxone to be widely distributed. Indeed, many harm reduction programs such as NFRP are undertaking the challenge by distributing naloxone free of charge to first responders. The NFRP program is run by Pennsylvania Commission on Crime and Delinquency (PCCD), who dispenses naloxone to eligible first responders through centralized, local hubs in each county or region First responders include emergency medical services, law enforcement, fire fighters, public transit drivers and so on. One challenge facing these organizations is that the utility of naloxone varies across different types of first responders. [8, 9] emphasize the importance of law enforcement officers, who are "often a community's*

first contact with opioid overdose victims after 9-1-1 services have been summoned." The utility of naloxone also varies across regions due to the varying levels of opioid usage in different populations. The West Virginia Department of Health and Human Resources (DHHR) purchased about 34,000 doses of naloxone; in addition to distributing to the state police, fire departments, and emergency medical services, DHHR additionally planned to distribute 1,000 doses of naloxone to each of the eight high priority counties, including Berkeley, Cabell, Harrison, Kanawha, Mercer, Monongalia, Ohio, and Raleigh [10]. Therefore, the prioritization of certain "demand classes" is an important consideration when naloxone is expensive or when quantities are limited; see, e.g., [11] for a report on rationing practices in Baltimore.

Example 2.0.2 (COVID-19)**.** *By the end of February 2021, COVID-19 has resulted in 28,409,727 cases and 511,903 deaths in the US [12]. Compared with 5-17 age group, the rate of death is 1100 times higher in 65-74 age group, 2800 times higher in 75-84 age group, and 7900 times higher in 85 and older age group [13]. According to the COVID-19 vaccination recommendations by CDC [14], in phase 1a, healthcare personnel and long-term care facility residents are offered vaccination first. Subsequently, the 75 and older age group and the 65-74 age group are vaccinated in phases 1b and 1c.*

Example 2.0.3 (Influenza)**.** *The need for distinct demand classes was also observed for the case of vaccine distribution during the 2009 H1N1 influenza pandemic. The H1N1 influenza virus first emerged in Mexico and California in April 2009 [15] and the pandemic lasted until August 2010 [16]. Children and young adults were disproportionately affected when compared to older adults [17]: during April 15 and May 5, 2009, among the 642 confirmed infected patients in the U.S. (ranging from 3 months to 81 years old), 60% were 18 years old or younger [18]. The reported H1N1 cases from April 15 to July 24, 2009, show that the infected rate (number of cases per 100,000 population) of 0 to 4 age group is 17.6 times of the infected rate of 65 and older age group, and the rate of 5 to 24 age group is 20.5 times of the rate of 65 and older age group [19]. The Advisory Committee on Immunization Practice (ACIP) recommended a priority group (about 159 million Americans), in which there was a subset with highest priority (about 62 million Americans) [20]. Patients aged 65 and older were only considered for vaccination once the demand amongst younger groups were met [21].*

Example 2.0.4 (Hepatitis A)**.** *Hepatitis A outbreaks began in 2016 and are currently (as of August 2021) ongoing in 36 states across the U.S. Recent data from August 26, 2021 shows 5098 cases (77 deaths) in Florida, 5077 cases (64 deaths) in Kentucky, 920 cases (30 deaths) in Michigan, 3148 cases (28 deaths) in Tennessee, and 3754 cases (16 deaths) in Ohio [22]. This outbreak largely affects the homeless, drug users, and their direct contacts [22]. Center for Disease Control (CDC) guidelines suggest that vaccine inventory be conducted monthly to ensure adequate supplies and that the vaccine order decisions take into account projected demand and storage capacity [23], two important aspects of our model. The CDC also recommends against overstocking, which presents the risk of wastage and outdated vaccines.*

Our Results. The main contributions of this chapter are summarized below.

- In this chapter, we first develop and analyze a hierarchical, finite-horizon Markov decision process (MDP) model that abstracts the above problems into a single framework. The upper-level problem (the "upper-level MDP") is an inventory model that controls a lower-level dispensing problem (the "lower-level MDP"). Here, we consider the setting where the utilities of PODs differ across regions due to the varying intervention effects on patients in different populations. The demand and POD-attribute distributions at each period depend on an information process, which can represent past demand realizations or other external information.

- We then analyze the structural properties. The MDP features basestock-like structure in a discrete state setting and discretely-concave value functions; both of these properties depend on the discrete-concavity observed in the lower-level problem. The motivation for a discrete state formulation comes from the naloxone distribution application, where demand quantities are relatively small; this is not an ideal setting for use of a continuous state approximation.

- Next, we propose a new actor-critic algorithm that exploits the structural properties of the MDP. More specifically, the algorithm tracks both policy and value function approximations (an identifying feature of an "actor-critic" method) and utilizes the structure to improve the empirical convergence rate. Moreover, the algorithm is suited for a setting

where data arrive continually and the policy is updated over time. This algorithm (and its general idea) is potentially of broader interest beyond the public health application.

- Finally, we present a case study for the problem of dispensing naloxone. We show how an aggregation-based version of the algorithm can be applied in a setting with continuous information states. In addition to computing approximations to the optimal inventory management and dispensing strategies, we also conduct a sensitivity analysis to understand the impact of various model parameters.

The chapter is organized as follows. A literature review is provided in Section 2.1. We introduce the hierarchical MDP model in Section 2.2 and derive its structural properties in Section 2.3. The proposed *stuctured actor-critic* algorithm is given and discussed in Section 2.4. In Section 2.5, we conduct numerical experiments. We propose an aggregation-based version of the algorithm in Section A.3 and finally present the naloxone case study in Section 2.6.

2.1 Literature Review

In this section, we provide a brief review of related literature. The upper-level replenishment decisions in this chapter are closely related to both lost-sales and perishable inventory models. In the lost-sales case, [24] constructs simple myopic approximations for three variations of the classical model with lead time. [25] studies a single-item, make-to-stock production model with several demand classes and lost sales and constructs stock-rationing levels for the optimal policy. [26] focuses on random supply interruptions in lost-sales inventory systems with positive lead times. [27] finds that the standard base-stock policy performs poorly compared to some other heuristic policies. We also refer readers to [28] for a detailed review. Our public health application is also somewhat related to the problems studied in perishable inventory models [29], even though our motivating application does not require us to explicitly model age.

Related to our hierarchical model is the case of multi-echelon systems, where, for example, an upper echelon (e.g. a central warehouse) replenishes the inventory of a lower echelon (e.g. a retailer) that serves demand [30]. [31] studies the optimal ordering and allocation policies for the upper echelon and [32] constructs an allocation policy for the multi-echelon system. In the model of [33], each retailer is allowed to replenish once from the warehouse during an ordering cycle. [34] shows the optimality of base-stock policies and derives newsvendor-type equations for the optimal base-stock levels. [35] studies a multi-product multi-echelon inventory system. [36] aims to optimize the reorder points of both echelons given fixed order quantities. Our model expands upon this literature in that the optimization problems for the two echelons are modeled as two nested MDPs (or a "hierarchical" MDP). We show the concavity of the value function of both the upper-level and lower-level, which is then utilized to derive the structured actor-critic algorithm.

Our proposed actor-critic method falls under the class of approximate dynamic programming (ADP) or reinforcement learning (RL) algorithms [37, 38, 39]. Possibly the most well-known RL technique is Q-learning [40], a model-free approach that uses stochastic approximation (SA) to learn state-action value function (or "Q-function"). In some cases, convexity of the value function is known a priori and can be exploited; see, e.g., [41, 42, 43, 44, 45, 46]. The updates used in the value function approximation part of our algorithm most closely resemble [42] and [43].

Related to the policy function approximation part of our algorithm, [47] proposes a stochastic approximation method to compute basestock levels in continuous state inventory problems. Our method also utilizes two types of basestock structure, but it does so in a different way from [47] due to our focus on discrete-valued inventory states. The primary feature of an actor-critic algorithm is that it approximates both the policy and value function [48, 49, 50, 51]. The "actor" is the policy function approximation (for selecting actions) and the "critic" represents the value function approximation used to "criticize" the actions selected by the actor. The novelty of our method is due to its use of the structure in *both the value function and the policy*. Our experimental results show significant advantages of exploiting this policy-value structure. Further, differing from most actor-critic methods, we do not use stochastic policy, reducing the number of policy parameters to be learned.

In addition, state aggregation is a commonly used method to deal with large dynamic problems [52, 53, 54], including inventory management [55, 56, 57, 58, 59]. Error bounds for these types of approximations can be found in [60], [61], and [62]. Our results in Section 2.6 make use of partial aggregation of the state space.

Due to the discrete inventory states used in our model, we make use of the concept of $L^\natural$-convexity (concavity) as a tool in the analysis. This theory was first developed in [63] for discrete convex analysis and then extended to continuous variables by [64]. Closely related concepts are l-convexity and submodularity. It turns out that these ideas are useful in understanding the structures of optimal policies in the field of inventory management, as introduced by [65] in an assemble-to-order multi-item system. [66] uses $L^\natural$-convexity in some variations of the basic multiperiod lost-sales model with lead time and [67] extend the results to lost-sales serial inventory systems. [68] use similar ideas to analyze inventory-pricing systems with lead time, and [69] study finite capacity systems with both manufacturing and remanufacturing. See [70] for a survey of applications utilizing the theory of $L^\natural$-convexity.

As for the utility in our model, the quality-adjusted life-year (QALY) is widely used to in healthcare to measure the value of medical interventions. The QALY was originally developed for cost-effectiveness analysis to decide scarce resource allocation across competing healthcare programs [71, 72, 73], and has been endorsed by the US Panel on Cost-Effectiveness in Health and Medicine as a standardized methodological approach in cost-effectiveness analyses [74]. The QALY has been used in naloxone distribution research to evaluate the cost-effectiveness of distributing naloxone to heroin users [75], distributing naloxone to adults at risk of heroin overdose in UK [76], and one-time versus biannual distribution [77].

2.2 Model Formulation

As discussed above, our MDP model is motivated by the hierarchical structure of public health organizations, such as Pennsylvania's NFRP, which distributes naloxone to a CCE that, in turn, coordinates the dispensing of naloxone to first responders in various counties.

We assume that the central inventory manager makes *replenish-up-to* and *dispense-down-to* decisions to the central storage periodically. Then, naloxone is distributed to the dispensing coordinator, who makes dispensing decisions to sequentially and randomly arriving PODs. Given an initial allotment of inventory, the dispensing decisions to PODs are made with the goal of maximizing cumulative utility of the satisfied naloxone demand within the dispensing period. (The trade-off here considers, for example, the number of naloxone kits that should be provided to first responders in a neighborhood with high drug overdose death rate versus the first responders in a neighborhood with low drug overdose death rates.) The timing of events during each period is as follows: (1) the central inventory manager decides the replenish-up-to and dispense-down-to levels, (2) the dispensing coordinator receives naloxone, and (3) based on POD demands, POD attributes, and the level of available inventory, the dispensing coordinator dispenses naloxone in order to maximize utility. Figure 1 gives an illustration of the timing of these events. In this section, we first discuss the lower-level dispensing problem and then illustrate the upper-level inventory control model.

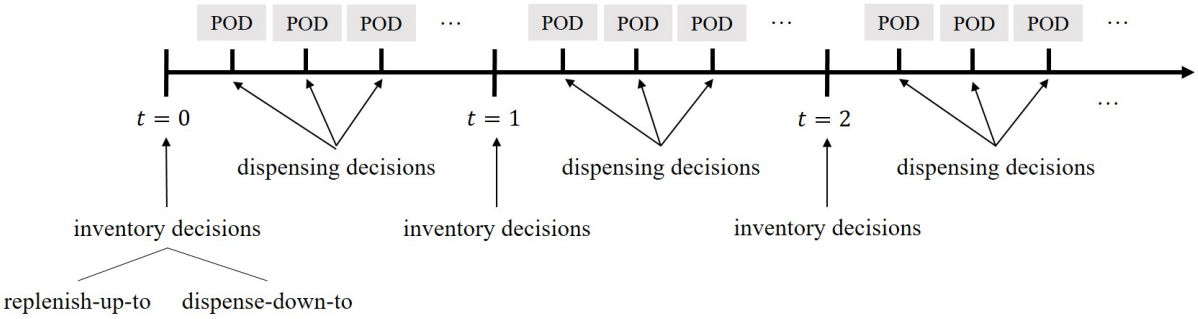


Figure 1: Sequence of events.

2.2.1 The Dispensing MDP

We first discuss the *lower-level* MDP for making dispensing decisions within each period t. After the central inventory manager makes the replenish-up-to and dispense-down-to decisions, the dispensing coordinator receives a sequence of POD demands to satisfy, starting with an initial inventory allotment based on the dispense-down-to decision. The dispensing model contains n sub-periods. In sub-period i, the arriving POD is represented by an *at-*

tribute $\Xi_{t,i}$ which is interpreted as the arriving POD's attributes. When there is no arriving POD in sub-period i, $\Xi_{t,i} = 0$. The distribution of $\Xi_{t,i}$ depends on an exogenous information process $\{W_t\}$ that transitions according to the upper-level timescale (and thus stays fixed at a particular realization w for all sub-periods in the dispensing problem; a full description of this process will be given in Section 2.2.2). Given realizations w_t and $\xi_{t,i}$ of the exogenous information W_t and attribute $\Xi_{t,i}$, we consider an increasing expected utility function $u_{w_t}(\cdot, \xi_{t,i})$, whose argument is the number of inventory units y_i dispensed to the arriving POD in sub-period i. For the remainder of this section, we omit the subscript t in for convenience.

These utility functions should be interpreted as parameters specified by the public health organization. The motivation for modeling heterogeneous utilties for the case of naloxone dispensing is primarily due to varying severity of the epidemic across different regions and populations (first responders in regions with more opioid users should have higher priority). To model this heterogeneity in demand, our model allows region and other related information to be encoded within the attribute ξ_i, which then determines the utility.

The dispensing coordinator aims to maximize the total utility subject to the initial inventory allotment x_0. In sub-period i, given exogenous information w, available inventory level x_i and attribute state ξ_i about the arriving POD, a dispensing decision y_i is made. Let $\{\mu_{w,0}, \mu_{w,1}, \ldots, \mu_{w,n-1}\}$ be lower-level dispensing policy for exogenous information w and suppose $\mathcal{M}_w$ is the set of all feasible policies that satisfy $\mu_{w,i}(x_i, \xi_i) \leq x_i$. The objective on the lower-level is given by

$$U_{w,0}(x_0, \xi_0) = \max_{\mu_w \in \mathcal{M}_w} \mathbf{E}\Big[\sum_{i=0}^{n-1} u_w\big(\mu_{w,i}(X_i, \Xi_i), \Xi_i\big) \,\Big|\, X_0 = x_0, \Xi_0 = \xi_0, W = w\Big],$$

where the transition follows $X_{i+1} = x_i - \mu_{w,i}(x_i, \xi_i)$. The optimum is attained by an optimal policy μ_w^*. We now write the Bellman optimality equation for the objective:

$$U_{w,i}(x_i, \xi_i) = \max_{y_i \leq x_i} u_w\big(y_i, \xi_i\big) + \mathbf{E}_w\big[U_{w,i+1}(X_{i+1}, \Xi_{i+1})\big] \tag{2.1}$$

for $i = 0, 1, \ldots, n-1$, and $U_{w,n}(x_n, \xi_n) = 0$, where $\mathbf{E}_w$ is being used as shorthand for the expected value conditioned on $\{W_t = w\}$.

2.2.2 The Inventory Control MDP

The sequential inventory control aspect of the model contains T planning periods. In each period, there are two decisions to be made: the replenish-up-to level and the dispense-down-to level. In the first period $t = 0$, the initial resource level $R_0 = 0$. In the last period $t = T$, no decision is made and the remaining inventory R_T is either worthless or charged a disposal cost (controlled by a parameter $b \geq 0$). Let $\{W_t\}$ be the aforementioned exogenous information process, which may contain information regarding past POD demands, current disease trends, or other dynamic information related to the public health situation. As discussed above, the information state W_t influences the distribution of the attributes $\Xi_{t,i}$ of the arriving PODs for sub-periods $i = 1, 2, \ldots, n$ of the lower-level problem in period t. We assume that W_t takes values in a finite set $\mathcal{W}$ and that it is a Markov process.

Let $R_{\max}$ be the capacity of the central storage facility. At the end of each period t, the central inventory manager makes a replenish-up-to decision z_t^{rep} based on the available resource level $R_t \in \{0, 1, \ldots, R_{\max}\}$ and the exogenous information $W_t \in \mathcal{W}$. After this, the central inventory manager makes a dispense-down-to decision z_t^{dis} based on the replenish-up-to decision z_t^{rep} and W_t.

We will often refer to particular values of the resource level R_t and exogenous information W_t using the notations r and w, respectively. Let $\bar{\mathcal{Z}}(r) = \{r, r+1, \ldots, R_{\max}\}$ be the set of feasible replenish-up-to decisions if the current inventory level is r, so that $z_t^{\text{rep}} \in \bar{\mathcal{Z}}(R_t)$ in period t. This means the central inventory manager orders $z_t^{\text{rep}} - R_t$ units of inventory with a per-unit ordering cost $c_w \geq c_{\min}$ (note that we allow this ordering cost to depend on the exogenous information w), where $c_{\min}$ is a positive scalar.

Let $\underline{\mathcal{Z}}(z^{\text{rep}}) = \{0, 1, \ldots, z^{\text{rep}}\}$ be the set of feasible dispense-down-to decisions if the current resource level is z^{rep}, so that $z_t^{\text{dis}} \in \underline{\mathcal{Z}}(z_t^{\text{rep}})$ in period t. This means the central inventory manager delivers $z_t^{\text{rep}} - z_t^{\text{dis}}$ units of inventory to the dispensing coordinator, and $z_t^{\text{rep}} - z_t^{\text{dis}}$ serves as the initial inventory allotment in the lower-level dispensing MDP problem. The transition to the next inventory state R_{t+1} is given by:

$$R_{t+1} = z_t^{\text{dis}}. \tag{2.2}$$

Each unit of leftover inventory after applying the transition (2.2) is charged a holding cost $h < c_{\min}$.

A policy $\{\pi_0, \pi_1, \ldots, \pi_{T-1}\}$ is a sequence of mappings $\pi_t = (\pi_t^{\text{rep}}, \pi_t^{\text{dis}})$ from states (R_t, W_t) to replenish-up-to levels and dispense-down-to levels. Let Π be the set of all feasible policies that satisfy $\pi_t^{\text{rep}}(R_t, W_t) \geq R_t$ and $\pi_t^{\text{dis}}(R_t, W_t) \leq \pi_t^{\text{rep}}(R_t, W_t)$. Our objective is given by:

$$\max_{\pi \in \Pi} \mathbf{E}\Big[\sum_{t=0}^{T-1}\big(-hR_t - c_{W_t}\big(\pi_t^{\text{rep}}(R_t, W_t) - R_t\big) + U_{W_t,0}\big(\pi_t^{\text{rep}}(R_t, W_t) - \pi_t^{\text{dis}}(R_t, W_t),\, \Xi_{t,0}\big)\Big] - b\, R_T,$$

where R_t transitions according to (2.2) for $(z_t^{\text{rep}}, z_t^{\text{dis}}) = \pi_t(R_t, W_t)$, and the gap between the two decisions of the upper-level problem, $\pi_t^{\text{rep}}(R_t, W_t) - \pi_t^{\text{dis}}(R_t, W_t)$, serves as the initial resource level of the lower-level problem. We now write a preliminary set of Bellman optimality equations for the objective above. Let $V_T(r, w) = -b\, r$ be the terminal value (note: b is zero if there is no disposal cost). For $t < T$, we have

$$\begin{aligned} V_t(r, w) = \max_{z^{\text{rep}} \in \bar{\mathcal{Z}}(r),\, z^{\text{dis}} \in \underline{\mathcal{Z}}(z^{\text{rep}})} & (c_w - h)\, r - c_w z^{\text{rep}} \\ & + \mathbf{E}_w\big[U_{w,0}\big(z^{\text{rep}} - z^{\text{dis}}, \Xi_{t,0}\big) + V_{t+1}\big(z^{\text{dis}}, W_{t+1}\big)\big]. \end{aligned} \tag{2.3}$$

So far, we have considered z^{rep} and z^{dis} as being made simultaneously in each period, but we can equivalently view the dispense-down-to decision z^{dis} to be taken after the replenish-up-to decision z^{rep} (this reflects the reality and also is useful for our analysis and algorithm). The set $\underline{\mathcal{Z}}(z^{\text{rep}})$ of feasible dispense-down-to decisions is dependent on the replenish-up-to decision z^{rep}. Therefore, the value function in each period can be broken into two steps:

$$V_t^{\text{rep}}(r, w) = (c_w - h)\, r + \max_{z^{\text{rep}} \in \bar{\mathcal{Z}}(r)} \big\{-c_w z^{\text{rep}} + V_t^{\text{dis}}(z^{\text{rep}}, w)\big\}, \tag{2.4}$$

$$V_t^{\text{dis}}(z^{\text{rep}}, w) = \max_{z^{\text{dis}} \in \underline{\mathcal{Z}}(z^{\text{rep}})} \mathbf{E}_w\big[U_{w,0}\big(z^{\text{rep}} - z^{\text{dis}}, \Xi_{t,0}\big) + V_{t+1}^{\text{rep}}\big(z^{\text{dis}}, W_{t+1}\big)\big], \tag{2.5}$$

with $V_T^{\text{rep}}(r, w) = -b\, r$. Similarly, there are two postdecision value functions in each period corresponding to the replenish-up-to decision and the dispense-down-to decision respectively:

$$\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w) = -c_w z^{\text{rep}} + V_t^{\text{dis}}(z^{\text{rep}}, w), \tag{2.6}$$

$$\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w) = \mathbf{E}_w\big[V_{t+1}^{\text{rep}}\big(z^{\text{dis}}, W_{t+1}\big)\big]. \tag{2.7}$$

The optimal policy can be written as follows

$$\pi_t^{\text{rep},*}(r, w) \in \underset{z^{\text{rep}} \in \bar{\mathcal{Z}}(r)}{\arg\max}\ \tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w). \tag{2.8}$$

$$\pi_t^{\text{dis},*}(z^{\text{rep}}, w) \in \underset{z^{\text{dis}} \in \underline{\mathcal{Z}}(z^{\text{rep}})}{\arg\max}\ \mathbf{E}_w\big[U_{w,0}\big(z^{\text{rep}} - z^{\text{dis}}, \Xi_{t,0}\big)\big] + \tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w), \tag{2.9}$$

where, with a slight abuse/reuse of notation, $\pi_t^{\text{dis},*}(z^{\text{rep}}, w)$ is the optimal dispense-down-to policy when the replenish-up-to level is z^{rep}. Combining (2.4)-(2.9), we obtain equivalent formulations of the optimality equation written using $\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$, $\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$, $\pi_t^{\text{rep},*}(r, w)$, and $\pi_t^{\text{dis},*}(z^{\text{rep}}, w)$:

$$\begin{aligned}\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w) = -c_w z^{\text{rep}} + \mathbf{E}_w\big[U_{w,0}\big(z^{\text{rep}} - \pi_t^{\text{dis},*}(z^{\text{rep}}, w),\, \Xi_{t,0}\big)\big] \\ + \tilde{V}_t^{\text{dis}}\big(\pi_t^{\text{dis},*}(z^{\text{rep}}, w), w\big),\end{aligned} \tag{2.10}$$

$$\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w) = \mathbf{E}_w\big[(c_{W_{t+1}} - h)\, z^{\text{dis}} + \tilde{V}_{t+1}^{\text{rep}}\big(\pi_{t+1}^{\text{rep},*}(z^{\text{dis}}, W_{t+1}), W_{t+1}\big)\big], \tag{2.11}$$

with $\tilde{V}_{T-1}^{\text{dis}}(z^{\text{dis}}, w) = -b\, z^{\text{dis}}$.

Our proposed algorithm will make use of the convenient formulations of $\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$ and $\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$ as expectations in (2.10) and (2.11). These formulations are useful for ADP for two reasons: (1) the maximization is within the expectation, so a data- or sample-driven method is easier to incorporate and (2) knowledge about the policies $\pi_t^{\text{rep},*}$ and $\pi_t^{\text{dis},*}$ can be used within a value function approximation procedure. Indeed, our actor-critic algorithm will make use of the interplay between the greedy policy functions (2.8) and (2.9) and the optimal value functions (2.6) and (2.7).

2.3 Structural Properties

In this section, we analyze the structure properties of the postdecision value functions $\tilde{V}_t^{\text{rep}}$ and $\tilde{V}_t^{\text{dis}}$ and the optimal policies $\pi_t^{\text{rep},*}$ and $\pi_t^{\text{dis},*}$. We remind the reader that our model uses discrete inventory states. As opposed to the standard continuous inventory state approximation, this modeling decision was made in order to accomodate the public health setting, where resources are potentially scarce. Our structural analysis makes use the properties of $L^\natural$-concave functions, an approach used often in inventory models [70].

Definition 2.3.1 ($L^\natural$-concave function)**.** *A function $g : \mathbb{Z}^d \to \mathbb{R} \cup \{+\infty\}$ with $\operatorname{dom} g \neq \emptyset$ is $L^\natural$-concave if and only if it satisfies discrete midpoint concavity:*

$$g(p) + g(q) \leq g\Big(\Big\lceil \frac{p+q}{2} \Big\rceil\Big) + g\Big(\Big\lfloor \frac{p+q}{2} \Big\rfloor\Big) \tag{2.12}$$

for all $p, q \in \mathbb{Z}^d$, where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are the ceiling *and* floor *functions, respectively.*

For the one-dimensional case, $g : \mathbb{Z} \to \mathbb{R}$, the condition (2.12) can be reduced to the simpler statement: $g(p) - g(p-1) \geq g(p+1) - g(p)$ for all $p \in \mathbb{Z}$, and $L^\natural$-concavity is equivalent to discrete concavity [78]. Throughout the rest of the chapter, we will use *discretely concave* to refer to one-dimensional functions that satisfy this condition.

Assumption 2.3.1. *For any w and ξ, the expected utility function $u_w(x, \xi)$ is discretely concave in x.*

Proposition 2.3.1. *Suppose Assumption 2.3.1 is satisfied. Then, for each information state w, POD attribute ξ, and sub-period i, the lower-level value function $U_{w,i}(x, \xi)$ is discretely concave in the inventory state x.*

Proposition 2.3.2. *Suppose Assumption 2.3.1 is satisfied. Then, the following properties hold:*

1. *For each t and information state w, the postdecision value function $\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$ is discretely concave in z^{rep} and $\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$ is discretely concave in z^{dis}.*

2. *For each t and state (r, w), the optimal policy $\pi_t^{\text{rep},*}(r, w)$ can be written as a series of state-dependent, discrete basestock policies, with thresholds $l_t^{\text{rep}}(w) \in \{0, 1, \ldots, R_{\max}\}$:*

$$\pi_t^{\text{rep},*}(r, w) = \max\{r, l_t^{\text{rep}}(w)\}.$$

It is optimal to replenish the inventory level as close as possible to $l_t^{\text{rep}}(w)$.

3. *For each t and state (z^{rep}, w), the optimal policy $\pi_t^{\text{dis},*}(z^{\text{rep}}, w)$ can be written as a series of state-dependent, discrete basestock policies, with thresholds $l_t^{\text{dis}}(z^{\text{rep}}, w) \in \{0, 1, \ldots, R_{\max}\}$ that*

$$\pi_t^{\text{dis},*}(z^{\text{rep}}, w) = \min\{z^{\text{rep}}, l_t^{\text{dis}}(z^{\text{rep}}, w)\}.$$

Proof. See Appendix A.1.2 for the proof of Part 1. Parts 2 and 3 then follow directly from (2.8) and (2.9) respectively. □

We remark that the state-dependency of the replenish-up-to thresholds $l_t^{\text{rep}}(w)$ in Proposition 2.3.2 refers only to the exogenous information state W_t, while the dispense-down-to thresholds $l_t^{\text{dis}}(z^{\text{rep}}, w)$ are dependent on both the inventory and information states (z^{rep}, w). In the former case, if $r < l_t^{\text{rep}}(w)$, it is optimal to replenish up to $l_t^{\text{rep}}(w)$, while if $r_t \geq l_t^{\text{rep}}(w)$, it is optimal not to replenish. The quantity ordered is given by $\pi_t^{*,\text{rep}}(r, w) - r$. In the latter case, if $z^{\text{rep}} > l_t^{\text{dis}}(z^{\text{rep}}, w)$, it is optimal to dispense down to $l_t^{\text{dis}}(z^{\text{rep}}, w)$, while if $z^{\text{rep}} \leq l_t^{\text{dis}}(z^{\text{rep}}, w)$, it is optimal to dispense down to z^{rep}.

For algorithmic reasons, we define $v_t^{\text{rep}}(z^{\text{rep}}, w) = \Delta\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$ and $v_t^{\text{dis}}(z^{\text{dis}}, w) = \Delta\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$ to be the "slopes" of postdecision state values $\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$ and $\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$ respectively, where

$$\Delta\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w) = \tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w) - \tilde{V}_t^{\text{rep}}(z^{\text{rep}} - 1, w),$$

$$\Delta\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w) = \tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w) - \tilde{V}_t^{\text{dis}}(z^{\text{dis}} - 1, w),$$

and $\tilde{V}_t^{\text{rep}}(-1, w) = \tilde{V}_t^{\text{dis}}(-1, w) \equiv 0$. It holds that $\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w) = \sum_{z'=0}^{z^{\text{rep}}} v_t^{\text{rep}}(z', w)$, where $v_t^{\text{rep}}(0, w) \equiv \tilde{V}_t^{\text{rep}}(0, w)$. Proposition 2.3.2 implies that $v_t^{\text{rep}}(z, w) \geq v_t^{\text{rep}}(z', w)$ for all $0 < z \leq z'$. The same is true for $\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$ and $v_t^{\text{dis}}(z^{\text{dis}}, w)$.

2.4 The Structured Actor-Critic Method

In this section, we focus on the upper-level inventory control and dispensing problem and introduce the structured actor-critic (S-AC) algorithm. The goal of the algorithm is to approximate the postdecision value functions $\tilde{V}^{\text{rep}}$ and $\tilde{V}^{\text{dis}}$ and the optimal (basestock) policies $\pi^{\text{rep},*}$ and $\pi^{\text{dis},*}$ by exploiting structure for both. For the lower-level dispensing problem, we use backward induction to solve the dynamic programming exactly, and apply the optimal lower-level dispensing policy μ_w^* to each of the arrived PODs.

2.4.1 Overview of the Main Idea

Our algorithm is based on the recursive relationship of (2.6) and (2.7) and the properties of the problem as described in Proposition 2.3.2. The basic structure is a time-dependent version of the actor-critic method, which makes use of the interaction between the value approximations and the policy approximations in each iteration. The "actor" refers to the policy approxmations $\{\bar{\pi}^{\text{rep},k}\}$ and $\{\bar{\pi}^{\text{dis},k}\}$, and the "critic" refers to the value approximations $\{\bar{V}^{\text{rep},k}\}$ and $\{\bar{V}^{\text{dis},k}\}$. If the optimal policy is known, then the postdecision values can be calculated by (2.10) and (2.11); similarly, if the value function is known, the optimal policies can be calculated by (2.8) and (2.9). The proposed algorithm applies these two relationships in an alternating fashion.

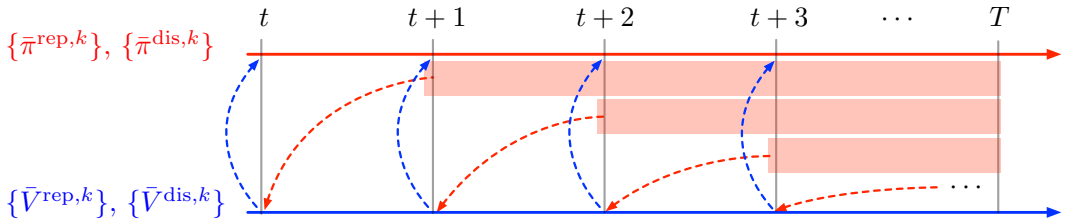


Figure 2: An illustration of how value and policy functions interact under the S-AC algorithm.

We represent the replenish-up-to policy by approximate basestock thresholds $\{\bar{l}^{\text{rep},k}\}$, where $\bar{l}_t^{\text{rep},k}(w)$ is the approximation to $l_t^{\text{rep}}(w)$ at iteration k. Note that compared to a standard actor-critic implementation which tracks a stochastic policy for each state [38], this is a significant reduction in the number of parameters needed to be learned. We represent the

dispense-down-to policy as approximations $\{\bar{\pi}_t^{\text{dis},k}(z^{\text{rep}}, w)\}$. As for the values, we represent them as approximations $\{\bar{v}^{\text{rep},k}\}$ and $\{\bar{v}^{\text{dis},k}\}$, where $\bar{v}_t^{\text{rep},k}(z^{\text{rep}}, w)$ and $\bar{v}_t^{\text{dis},k}(z^{\text{dis}}, w)$ approximate the discrete slopes $v_t^{\text{rep}}(z^{\text{rep}}, w) = \Delta\tilde{V}_t^{\text{rep}}(z^{\text{rep}}, w)$ and $v_t^{\text{dis}}(z^{\text{dis}}, w) = \Delta\tilde{V}_t^{\text{dis}}(z^{\text{dis}}, w)$, respectively. According to Proposition 2.3.2, if the approximations of the slopes are non-increasing in z^{rep} and z^{dis}, respectively, then the approximate value function is discretely concave in each of the decisions.

These approximations are iteratively updated via a stochastic approximation method [79, 80]. At each iteration, the algorithm has three steps. In the first step, we observe an exogenous information sequence and the attribute-request vectors for the whole planning horizon. In the second step, we observe the value of the current state under the current policy approximations, subject to the observed attribute-demand vectors. This value is used to update the value approximations. Finally, in the third step, we use the *implied basestock threshold* from the latest value function to update our approximate policy. The interactions between the policy and value approximations are shown in Figure 2.

Throughout the rest of the chapter, we use *bar* notation (e.g., $\bar{v}^{\text{rep},k}$ or $\bar{l}^{\text{rep},k}$) to denote approximations tracked by the algorithm at iteration k. On the other hand, we use *hat* notation (e.g., $\hat{V}_t^{\text{rep},k}$ or $\hat{v}_t^{\text{rep},k}$) to denote observed values at iteration k (these are one-time observations used to update the tracked approximations).

2.4.2 Algorithm Description

First, let us give some notation. The observed trajectory of the exogenous information process $\{W_t\}$ at iteration k is denoted $\{w_0^k, w_1^k, \ldots, w_{T-1}^k\}$ and the initial postdecision replenished resource level at period 0 is $z_0^{\text{rep,k}}$. The corresponding attribute $\{\xi_{t,1}^k\}$ observed at iteration k is assumed to follow the conditional distributions given w_t^k. Similarly, let $\mathbf{Z}_t^k(w)$ be an independent realization of the process $(W_\tau, \xi_{\tau,1})_{\tau=t}^{T-1}$ conditioned on $W_t = w$. This sequence of realizations is used to obtain an observation of the value of policy approximation starting at t and $W_t = w$ and we denote its elements by

$$\mathbf{Z}_t^k(w) = \big\{(\check{w}_\tau^k, \check{\xi}_{\tau,1}^k) : \tau = t, \ldots, T-1\big\},$$

where $\check{w}_t^k = w$. Define $\tilde{\pi}^{\mathrm{rep},k}$ and $\tilde{\pi}^{\mathrm{dis},k}$ as the rounded policies, i.e.

$$\tilde{\pi}^{\mathrm{rep},k}(r, w) = \texttt{round}[\bar{\pi}^{\mathrm{rep},k}(r, w)]$$

for all (r, w), $\tilde{\pi}^{\mathrm{dis},k}(z^{\mathrm{rep}}, w) = \texttt{round}[\bar{\pi}^{\mathrm{dis},k}(z^{\mathrm{rep}}, w)]$ for all (z^{rep}, w), where $\texttt{round}[x]$ returns the nearest integer to $x \in \mathbb{R}$. This is necessary because our approximate thresholds will not be integers. Let $f_t^{\mathrm{rep}}(\tilde{\pi}^{\mathrm{rep},k-1}, \tilde{\pi}^{\mathrm{dis},k-1}; \mathbf{Z}_t^k(w_t), r_t)$ be the Monte Carlo estimates of the replenish-up-to postdecision value starting in period t under the current policy approximations and an initial state (r_t, w_t):

$$\begin{aligned} &f_t^{\mathrm{rep}}\big(\tilde{\pi}^{\mathrm{rep},k-1}, \tilde{\pi}^{\mathrm{dis},k-1}; \mathbf{Z}_t^k(w_t), r_t\big) \\ &\quad = \sum_{\tau=t}^{T-2}\big[-c_{\check{w}_\tau^k}\tilde{z}_\tau^{\mathrm{rep}} + U^{\mu^*}_{\check{w}_\tau^k,0}\big(\tilde{z}_\tau^{\mathrm{rep}} - \tilde{z}_\tau^{\mathrm{dis}}, \check{\xi}^k_{\tau,0}\big) + (c_{\check{w}_{\tau+1}^k} - h)\tilde{z}_\tau^{\mathrm{dis}}\big] \\ &\qquad - c_{\check{w}_{T-1}^k}\tilde{z}_{T-1}^{\mathrm{rep}} + U^{\mu^*}_{\check{w}_{T-1}^k}\big(\tilde{z}_{T-1}^{\mathrm{rep}} - \tilde{z}_{T-1}^{\mathrm{dis}}, \check{\xi}^k_{T-1,0}\big) - b\,\tilde{z}_{T-1}^{\mathrm{dis}}, \end{aligned} \tag{2.13}$$

where for all $\tau \geq t$, $\mu^* = \mu^*_{\check{w}_\tau^k}$, $\tilde{z}_\tau^{\mathrm{rep}} = \tilde{\pi}_\tau^{\mathrm{rep},k-1}(r_\tau, \check{w}_\tau^k)$, $\tilde{z}_\tau^{\mathrm{dis}} = \tilde{\pi}_\tau^{\mathrm{dis},k-1}(\tilde{z}_\tau^{\mathrm{rep}}, \check{w}_\tau^k)$. Let $f_t^{\mathrm{dis}}(\tilde{\pi}^{\mathrm{rep},k-1}, \tilde{\pi}^{\mathrm{dis},k-1}; \mathbf{Z}_t^k(w_t), z_t^{\mathrm{rep}})$ be the Monte Carlo estimates of the dispense-down-to post-decision value starting in period t under the current policy approximations and an initial state $(z_t^{\mathrm{rep}}, w_t)$:

$$\begin{aligned} &f_t^{\mathrm{dis}}\big(\tilde{\pi}^{\mathrm{rep},k-1}, \tilde{\pi}^{\mathrm{dis},k-1}; \mathbf{Z}_t^k(w_t), z_t^{\mathrm{rep}}\big) \\ &\quad = \sum_{\tau=t}^{T-2}\big[(c_{\check{w}_{\tau+1}^k} - h)\tilde{z}_\tau^{\mathrm{dis}} - c_{\check{w}_{\tau+1}^k}\tilde{z}_{\tau+1}^{\mathrm{rep}} + U^{\mu^*}_{\check{w}_{\tau+1}^k,0}\big(\tilde{z}_{\tau+1}^{\mathrm{rep}} - \tilde{z}_{\tau+1}^{\mathrm{dis}}, \check{\xi}^k_{\tau+1,0}\big)\big] - b\,\tilde{z}_{T-1}^{\mathrm{dis}}, \end{aligned} \tag{2.14}$$

where $\tilde{z}_t^{\mathrm{dis}} = \tilde{\pi}_t^{\mathrm{dis},k-1}(z_t^{\mathrm{rep}}, \check{w}_t^k)$, and for all $\tau \geq t+1$, $\mu^* = \mu^*_{\check{w}_\tau^k}$, $\tilde{z}_\tau^{\mathrm{rep}} = \tilde{\pi}_\tau^{\mathrm{rep},k-1}(r_\tau, \check{w}_\tau^k)$, $\tilde{z}_\tau^{\mathrm{dis}} = \tilde{\pi}_\tau^{\mathrm{dis},k-1}(\tilde{z}_\tau^{\mathrm{rep}}, \check{w}_\tau^k)$. The replenish-up-to policy is

$$\bar{\pi}_\tau^{\mathrm{rep},k}(r_\tau, \check{w}_\tau^k) = \max\{r_\tau, \bar{l}_\tau^{\mathrm{rep},k}(\check{w}_\tau^k)\}.$$

Although there is substantial notation used in defining f_t^{rep} and f_t^{dis}, we remark that they are simply Monte Carlo observations of the policy's postdecision values respectively corresponding to the replenish-up-to and dispense-down-to decisions.

At each period t, to compute the approximate slopes, we use f_t^{dis} to observe values $\hat{V}_t^{\mathrm{rep},k}(z_t^{\mathrm{rep},k}, w_t^k)$ and $\hat{V}_t^{\mathrm{rep},k}(z_t^{\mathrm{rep},k} - 1, w_t^k)$, and f_{t+1}^{rep} to observe values $\hat{V}_t^{\mathrm{dis},k}(z_t^{\mathrm{dis},k}, w_t^k)$ and

$\hat{V}_t^{\text{dis},k}(z_t^{\text{dis},k}-1, w_t^k)$, where f_t^{dis} and f_{t+1}^{rep} are implied by the current policies $\bar{\pi}^{\text{rep},k-1}$ and $\bar{\pi}^{\text{dis},k-1}$; specifically, for $z^{\text{rep}}, z^{\text{dis}} \geq 0$, the observations $\hat{V}_t^{\text{rep},k}(z^{\text{rep}}, w_t^k)$ and $\hat{V}_t^{\text{dis},k}(z^{\text{dis}}, w_t^k)$ are

$$\begin{aligned}\hat{V}_t^{\text{rep},k}(z^{\text{rep}}, w_t^k) = &- c_{w_t^k} z^{\text{rep}} + U^{\mu^*}_{\check{w}_t^k,0}\big(z^{\text{rep}} - \tilde{\pi}_t^{\text{dis},k-1}(z^{\text{rep}}, w_t^k), \check{\xi}^k_{t,0}\big) \\ &+ f_t^{\text{dis}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), z^{\text{rep}}\big),\end{aligned} \tag{2.15}$$

and

$$\hat{V}_t^{\text{dis},k}(z^{\text{dis}}, w_t^k) = (c_{w_{t+1}} - h) z^{\text{dis}} + f_{t+1}^{\text{rep}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_{t+1}^k(w_{t+1}), z^{\text{dis}}\big), \tag{2.16}$$

where w_{t+1} is sampled from the distribution $W_{t+1} \,|\, W_t = w_t^k$. The approximate slopes $\hat{v}_t^{\text{rep},k}$ and $\hat{v}_t^{\text{dis},k}$ are given by:

$$\hat{v}_t^{\text{rep},k} = \hat{V}_t^{\text{rep},k}(z_t^{\text{rep},k}, w_t^k) - \hat{V}_t^{\text{rep},k}(z_t^{\text{rep},k}-1, w_t^k), \tag{2.17}$$

$$\hat{v}_t^{\text{dis},k} = \hat{V}_t^{\text{dis},k}(z_t^{\text{dis},k}, w_t^k) - \hat{V}_t^{\text{dis},k}(z_t^{\text{dis},k}-1, w_t^k), \tag{2.18}$$

where we define $\hat{V}_t^{\text{rep},k}(-1, w_t^k) = \hat{V}_t^{\text{dis},k}(-1, w_t^k) \equiv 0$. By doing so, the value assigned to $\hat{v}_t^{\text{rep},k}$ when $z_t^{\text{rep},k} = 0$ is actually $\hat{V}_t^{\text{rep},k}(0, w_t^k)$. This also applies to $\hat{v}_t^{\text{dis},k}$. We now summarize the structured actor-critic method; the full details of the approach are given in Algorithm 1.

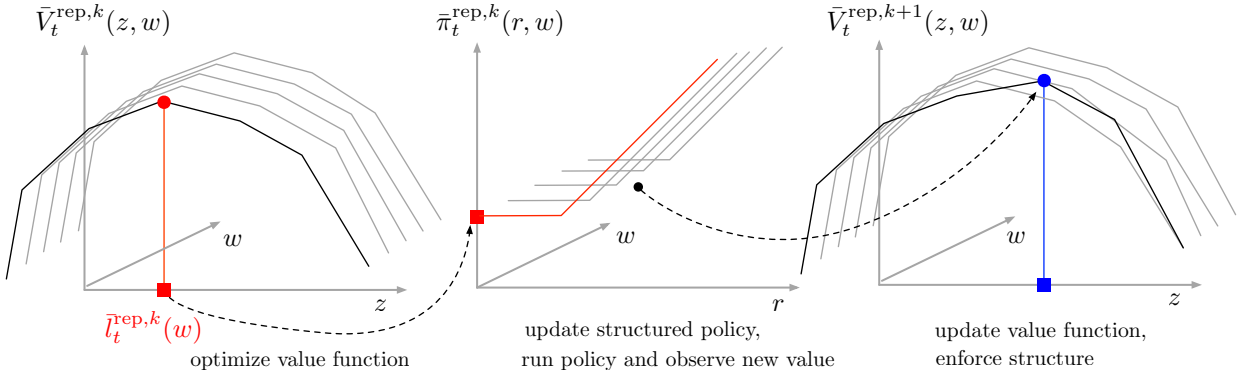


Figure 3: An illustration of the sequence of updates used in the S-AC algorithm.

- The inputs of Algorithm 1 are a random initial basestock policy $\bar{l}^{\text{rep},0}$, and concave, piecewise linear value function approximations $\bar{v}^{\text{rep},0}$ and $\bar{v}^{\text{dis},0}$.

Algorithm 1: Structured Actor-Critic Method

Input: Lower-level optimal policy μ^* (learned from backward dynamic programming). Initial policy estimates $\bar{l}^{\text{rep},0}$ and $\bar{\pi}^{\text{dis},0}$, and value estimates $\bar{v}^{\text{rep},0}$ and $\bar{v}^{\text{dis},0}$ (nonincreasing in z^{rep} and z^{dis} respectively). Stepsize rules $\tilde{\alpha}_t^k$ and $\tilde{\beta}_t^k$ for all t, k.

Output: Approximations $\bar{l}^{\text{rep},k}$, $\bar{\pi}^{\text{dis},k}$, $\bar{v}^{\text{rep},k}$, and $\bar{v}^{\text{dis},k}$.

for $k = 1, 2, \ldots$ **do**

  Sample initial states $z_0^{\text{rep},k}$ and $z_0^{\text{dis},k}$.

  for $t = 0, 1, \ldots, T-1$ **do**

    Observe w_t^k and $\xi_{t,1}^k$, then $\hat{v}_t^{\text{rep},k}$ and $\hat{v}_t^{\text{dis},k}$ according to (2.17) and (2.18).

    Perform SA step:

    $\tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) = \big(1 - \alpha_t^k(z^{\text{rep}}, w)\big)\, \bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) + \alpha_t^k(z^{\text{rep}}, w)\, \hat{v}_t^{\text{rep},k}$,

    $\tilde{v}_t^{\text{dis},k}(z^{\text{dis}}, w) = \big(1 - \alpha_t^k(z^{\text{dis}}, w)\big)\, \bar{v}_t^{\text{dis},k-1}(z^{\text{dis}}, w) + \alpha_t^k(z^{\text{dis}}, w)\, \hat{v}_t^{\text{dis},k}$.

    Perform the concavity projection operation: $\bar{v}_t^{\text{rep},k} = \Pi_{z_t^{\text{rep},k}, w_t^k}(\tilde{v}_t^{\text{rep},k})$, $\bar{v}_t^{\text{dis},k} = \Pi_{z_t^{\text{dis},k}, w_t^k}(\tilde{v}_t^{\text{dis},k})$.

    Observe $\hat{l}_t^{\text{rep},k}$ according to (2.8) and update the replenish-up-to threshold: $\bar{l}_t^{\text{rep},k}(w) = \big(1 - \beta_t^k(w)\big)\, \bar{l}_t^{\text{rep},k-1}(w) + \beta_t^k(w)\, \hat{l}_t^{\text{rep},k}$.

    Observe $\hat{\pi}_t^{\text{dis}}$ according to (2.9) and update the dispense-down-to policy:

    for $z_t^{\text{rep}} = 0, 1, \ldots, R_{\max}$ **do**

      $\bar{\pi}_t^{\text{dis},k}(z^{\text{rep}}, w) = \big(1 - \alpha^k(z^{\text{rep}}, w)\big)\, \bar{\pi}_t^{\text{dis},k-1}(z^{\text{rep}}, w) + \alpha^k(z^{\text{rep}}, w)\, \hat{\pi}_t^{\text{dis}}$.

    end

    If $t < T-1$, take $z_{t+1}^{\text{rep},k}$ and $z_{t+1}^{\text{dis},k}$ according to the ϵ-greedy exploration policy.

  end

end

- Each iteration k consists of a loop through the time periods t.
- At period t, the approximate slopes are updated in Lines 4–8. Based on $z_t^{\text{rep},k}$, $z_t^{\text{dis},k}$ and $\mathbf{Z}_t^k(w_t^k)$, we first observe the sequences of the predecision resource $\{r_{t+1}, r_{t+2}, \ldots, r_T\}$ and the postdecision resources $\{z_t^{\text{rep},k}, z_{t+1}^{\text{rep}}, \ldots, z_{T-1}^{\text{rep}}\}$ and $\{z_t^{\text{dis},k}, z_{t+1}^{\text{dis}}, \ldots, z_{T-1}^{\text{dis}}\}$. These are computed according to (2.2), and the equations $z_\tau^{\text{rep}} = \tilde{\pi}_\tau^{\text{rep},k-1}(r_\tau, w_\tau^k)$, and $z_\tau^{\text{dis}} = \arg\max_{z^{\text{dis}} \in \underline{\mathcal{Z}}(z_\tau^{\text{rep}})} U^{\mu^*}_{\check{w}_\tau^k,0}\big(z_\tau^{\text{rep}} - z^{\text{dis}}, \check{\xi}_{\tau,0}^k\big) + \bar{V}_\tau^{\text{dis},k-1}\big(z^{\text{dis}}, \check{w}_\tau^k\big)$ for all $\tau \geq t+1$. In the following illustration, let us take the value slope and policy corresponding to the replenish-up-to decision as an example, those corresponding to the dispense-down-to decision are similar.
- The observation of the slope $\hat{v}_t^{\text{rep},k}$ *implied by the policies* $\tilde{\pi}^{\text{rep},k-1}$ and $\tilde{\pi}^{\text{dis},k-1}$ is computed using (2.15) and (2.17) and used to calculate the smoothed slopes $\tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w)$ in Line 5, where $\alpha_t^k(z^{\text{rep}}, w) = \tilde{\alpha}_t^k \mathbb{1}\{z^{\text{rep}} = z_t^{\text{rep},k}\}\mathbb{1}\{w = w_t^k\}$. Thus, only the state $(z_t^{\text{rep},k}, w_t^k)$ is updated.
- A concavity projection operation in Line 8 is performed on the slopes $\tilde{v}_t^{\text{rep},k}$, resulting in a new set of slopes $\Pi_{z_t^{\text{rep},k}, w_t^k}(\tilde{v}_t^{\text{rep},k})$, in order to avoid violation of concavity. The component of $\Pi_{z_t^{\text{rep},k}, w_t^k}(\tilde{v}_t^{\text{rep},k})$ at state (z^{rep}, w) is

$$\Pi_{z_t^{\text{rep},k}, w_t^k}(\tilde{v}_t^{\text{rep},k})[z^{\text{rep}}, w]$$
$$= \begin{cases} \tilde{v}_t^{\text{rep},k}(z_t^{\text{rep},k}, w_t^k) & \text{if } w = w_t^k,\ z^{\text{rep}} < z_t^{\text{rep},k},\ \tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) < \tilde{v}_t^{\text{rep},k}(z_t^{\text{rep},k}, w_t^k) \\ & \text{or } w = w_t^k,\ z^{\text{rep}} > z_t^{\text{rep},k},\ \tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) > \tilde{v}_t^{\text{rep},k}(z_t^{\text{rep},k}, w_t^k), \\ \tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) & \text{otherwise.} \end{cases}$$

- The approximate replenish-up-to policy is updated in Lines 9. The observation $\hat{l}_t^{\text{rep},k}$ is the maximum point of $\bar{V}_t^{\text{rep},k}(\cdot, w_t^k)$ inside the set $\mathcal{Z}(0)$, which is the implied replenish-up-to basestock threshold from the value function approximation. Given the observation, the policy is updated with stepsize $\beta_t^k(w) = \tilde{\beta}_t^k \mathbb{1}\{w = w_t^k\}$.
- The approximate dispense-down-to policy is updated in Lines 11–13. For each z^{rep}, we can observe $\hat{\pi}_t^{\text{dis}}$ according to (2.9). The policy is updated with the observation and stepsize $\alpha_t^k(z^{\text{rep}}, w) = \tilde{\alpha}_t^k \mathbb{1}\{z^{\text{rep}} = z_t^{\text{rep}}\}\mathbb{1}\{w = w_t^k\}$.

- Finally, the next replenish-up-to decision follows an ϵ-greedy policy, which is to select $z_{t+1}^{\text{rep},k} = \tilde{\pi}_\tau^{\text{rep},k-1}(r_\tau, w_\tau^k)$ with probability $1-\epsilon$, or take $z_{t+1}^{\text{rep},k}$ randomly from $\mathcal{Z}(r_{t+1}^k)$ with probability ϵ. In our numerical experiments, ϵ is chosen to be 0.1.

Figure 3 illustrates how the replenish-up-to value function and policy approximations interact with each other. The first two panels together show that given a structured value function, its maximizer (red square) is used to update the structured policy. Panels two and three together show that an observation of the current policy's value (blue circle) is in turn used to update the structured value function (where a projection step occurs to enforce structure). The process then repeats with the new maximizer (blue square).

2.4.3 Convergence Analysis

In this section, we give some theoretical assumptions and then state the convergence of Algorithm 1; in particular, the convergence of both the value function approximations $\bar{v}^{\text{rep},k}$ and $\bar{v}^{\text{dis},k}$ and the basestock policies $\bar{l}^{\text{rep},k}$ and $\bar{\pi}^{\text{dis},k}$. Let $\{\bar{v}_t^{\text{rep},k}\}_{k\geq 0}$ and $\{\bar{v}_t^{\text{dis},k}\}_{k\geq 0}$ be the sequences of slopes, let $\{\bar{l}_t^{\text{rep},k}\}_{k\geq 0}$ and $\{\bar{\pi}_t^{\text{dis},k}\}_{k\geq 0}$, be the sequences of policies generated by the algorithm. For period T, we assume $v_T^{\text{rep}}(z^{\text{rep}}, w) = 0$ for all iterations $k \geq 0$ and all possible postdecision states (z^{rep}, w), as we only need to learn the policy and slopes up to period $T-1$. We work on a probability space $(\Omega, \mathcal{F}, \mathbf{P})$, where $\mathcal{F} = \sigma\{(r_t^k, z_t^{\text{rep},k}, z_t^{\text{dis},k}, w_t^k, \boldsymbol{\xi}_t^k, \mathbf{D}_t^k, \hat{v}_t^k),\, t \leq T,\, k \geq 0\}$, where $\boldsymbol{\xi}_t^k = (\xi_{t,1}^k, \xi_{t,2}^k, \ldots, \xi_{t,n_k}^k)$, $\mathbf{D}_t^k = (D_{t,1}^k, D_{t,2}^k, \ldots, D_{t,n_k}^k)$. Moreover, we define

$$\begin{aligned}\mathcal{F}_t^k = \sigma\big\{&\{(r_\tau^{k'}, z_\tau^{\text{rep},k'}, z_\tau^{\text{dis},k'}, w_\tau^{k'}, \boldsymbol{\xi}_\tau^{k'}, \mathbf{D}_\tau^{k'}, \hat{v}_\tau^{k'}), k' < k, \tau \leq T\} \\ &\cup \{(r_\tau^k, z_\tau^{\text{rep},k}, z_\tau^{\text{dis},k}, w_\tau^k, \boldsymbol{\xi}_\tau^k, \mathbf{D}_\tau^k, \hat{v}_t^k), \tau \leq t\}\big\},\end{aligned}$$

for $t \leq T-1$ and $k \geq 1$, with $\mathcal{F}_t^0 = \{\emptyset, \Omega\}$ for all $t \leq T$. Their relationships are $\mathcal{F}_t^k \subseteq \mathcal{F}_{t+1}^k$ for $t \leq T-1$ and $\mathcal{F}_T^k \subseteq \mathcal{F}_0^{k+1}$.

Assumption 2.4.1. *For any z and w, suppose the stepsizes $\{\alpha_t^k(z^{\text{rep}}, w)\}$, $\{\alpha_t^k(z^{\text{dis}}, w)\}$, and $\{\beta_t^k(w)\}$ satisfy the following conditions:*

(i) For $\text{x} \in \{\text{rep}, \text{dis}\}$, $\alpha_t^k(z^{\text{x}}, w) = \tilde{\alpha}_t^k \mathbb{1}\{z^{\text{x}} = z_t^{\text{x},k}\}\mathbb{1}\{w = w_t^k\}$ for some $\tilde{\alpha}_t^k \in \mathbb{R}$ that is $\mathcal{F}_t^k$-measurable,

(ii) $\beta_t^k(w) = \tilde{\beta}_t^k \mathbb{1}\{w = w_t^k\}$ *for some* $\tilde{\beta}_t^k \in \mathbb{R}$ *that is* $\mathcal{F}_t^k$*-measurable,*

(iii) *For* $\mathrm{x} \in \{\mathrm{rep}, \mathrm{dis}\}$, $\sum_{k=0}^{\infty} \alpha_t^k(z^{\mathrm{x}}, w) = \infty$, $\sum_{k=0}^{\infty} \left(\alpha_t^k(z^{\mathrm{x}}, w)\right)^2 < \infty$ *almost surely,*

(iv) $\sum_{k=0}^{\infty} \beta_t^k(w) = \infty$, $\sum_{k=0}^{\infty} \left(\beta_t^k(w)\right)^2 < \infty$ *almost surely.*

Assumption 2.4.1(i) and (ii) ensures that only the slope and threshold for the observed state is updated in Line 5 of Algorithm 1; the ones corresponding to unobserved states are kept the same until the projection step. Parts (iii) and (iv) are standard conditions on the stepsize. To keep the convergence results clean, we also assume the state-dependent basestock thresholds are unique (this assumption can be easily relaxed).

Assumption 2.4.2. *There is a unique optimal solution to* $\max_{z \in \mathcal{Z}(0)} \tilde{V}_t^{\mathrm{rep}}(z, w)$*, which implies that there is a single optimal replenishment basestock threshold for each* w*. The unique optimal solution assumption also applies to* $\tilde{V}_t^{\mathrm{dis}}$*.*

Assumptions (2.3.1)-(2.4.2) are used for the next two results. The primary novel aspect of our analysis is to connect the approximate policies with the approximate value functions through the structural properties of the problem. Before stating the main convergence result, Theorem 2.4.1, we introduce a lemma that illustrates the crucial mechanism for convergence.

Lemma 2.4.1. *The following hold:*

1. *For any fixed period* t*, suppose that the policies* $\bar{\pi}_\tau^{\mathrm{rep},k} \to \pi_\tau^{\mathrm{rep}}$ *almost surely for* $\tau \geq t+1$*, and* $\bar{\pi}_\tau^{\mathrm{dis},k} \to \pi_\tau^{\mathrm{dis}}$ *almost surely for* $\tau \geq t$*. Then it holds that* $\bar{v}_t^{\mathrm{rep},k}(z^{\mathrm{rep}}, w) \to v_t^{\mathrm{rep}}(z^{\mathrm{rep}}, w)$ *almost surely.*
2. *For any fixed period* t*, suppose that the policies* $\bar{\pi}_\tau^{\mathrm{rep},k} \to \pi_\tau^{\mathrm{rep}}$ *and* $\bar{\pi}_\tau^{\mathrm{dis},k} \to \pi_\tau^{\mathrm{dis}}$ *almost surely for* $\tau \geq t+1$*. Then it holds that* $\bar{v}_t^{\mathrm{dis},k}(z^{\mathrm{dis}}, w) \to v_t^{\mathrm{dis}}(z^{\mathrm{dis}}, w)$ *almost surely.*

Sketch of Proof. Let us show part (1) of the lemma. The proof for part (2) is similar. We first construct two deterministic sequences $\{G^m\}$ and $\{I^m\}$ such that $G^0 = v^{\mathrm{rep}} + v_{\max}^{\mathrm{rep}}$ and $I^0 = v^{\mathrm{rep}} - v_{\max}^{\mathrm{rep}}$ with

$$G^{m+1} = \frac{G^m + v^{\mathrm{rep}}}{2} \quad \text{and} \quad I^{m+1} = \frac{I^m + v^{\mathrm{rep}}}{2},$$

where $|v_t^{\mathrm{rep}}(z^{\mathrm{rep}}, w)| \leq v_{\max}^{\mathrm{rep}}$ for all t, z^{rep}, and w. These sequences have been previously used in [37]. Lemma 2.4.1 is proved if we have

$$I_t^m(z^{\mathrm{rep}}, w) \leq \bar{v}_t^{\mathrm{rep},k-1}(z^{\mathrm{rep}}, w) \leq G_t^m(z^{\mathrm{rep}}, w), \tag{2.19}$$

for any m and sufficiently large k. The proof proceeds by showing the following.

1. Define noise terms $\epsilon_t^k(z_t^{\text{rep},k}, w_t^k) = \mathbf{E}\big[\hat{v}_t^{\text{rep},k}\big] - v_t^{\text{rep}}(z_t^{\text{rep},k}, w_t^k)$ and $\varepsilon_t^k(z_t^{\text{rep},k}, w_t^k) = \hat{v}_t^{\text{rep},k} - \mathbf{E}\big[\hat{v}_t^{\text{rep},k}\big]$. Recall that $\hat{v}_t^{\text{rep},k} = \hat{V}_t^{\text{rep},k}(z_t^{\text{rep},k}, w_t^k) - \hat{V}_t^{\text{rep},k}(z_t^{\text{rep},k} - 1, w_t^k)$, where

$$\begin{aligned}\hat{V}_t^{\text{rep},k}(z^{\text{rep}}, w_t^k) = &- c_{w_t^k} z^{\text{rep}} + U^{\mu^*}_{\check{w}_t^k,0}\big(z^{\text{rep}} - \tilde{\pi}_t^{\text{dis},k-1}(z^{\text{rep}}, w_t^k), \check{\xi}_{t,0}^k\big) \\ &+ f_t^{\text{dis}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), z^{\text{rep}}\big).\end{aligned}$$

   From the assumption that $\bar{\pi}_\tau^{\text{rep},k} \to \pi_\tau^{\text{rep}}$ and $\bar{\pi}_\tau^{\text{dis},k} \to \pi_\tau^{\text{dis}}$ almost surely for all $\tau \geq t+1$, and the fact that $f_t^{\text{dis}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_t^k(w), z^{\text{rep}}\big)$ depends on the replenish-up-to policies for periods $t+1$ onward and the dispense-down-to policies for periods t onward, we conclude that

$$\mathbf{E}_w\big[f_t^{\text{dis}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_t^k(w), z^{\text{rep}}\big)\big] \to \tilde{V}_t^{\text{dis}}\big(\pi_t^{\text{dis},*}(z^{\text{rep}}, w), w\big)$$

   almost surely. Therefore, $\epsilon_t^k(z_t^{\text{rep},k}, w_t^k)$ converges to zero almost surely and $\varepsilon_t^k(z_t^{\text{rep},k}, w_t^k)$ is unbiased.

2. We partition the state space $\mathcal{S}$ into two parts: (1) states $(z^{\text{rep}}, w) \in \mathcal{S}_t^-$ and (2) states $(z^{\text{rep}}, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$, where $\mathcal{S}_t^-$ is a random set of states that are increased by the projection operator on finitely many iterations k. The proof considers each partition separately to show (2.19). For states $(z^{\text{rep}}, w) \in \mathcal{S}_t^-$, we show by forward induction on m the existence of a finite index $\tilde{K}_t^m$ such that (2.19) holds for all iterations $k \geq \tilde{K}_t^m$. The proof utilizes stochastic sequences related to the noise terms and stochastic "bounding" sequences. For any state $(z^{\text{rep}}, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$ and a fixed m, by Lemma 6.4 of [43], we show the existence of a state-dependent random index $\hat{K}_t^m(z^{\text{rep}}, w)$ such that (2.19) holds for all $k \geq \hat{K}_t^m(z^{\text{rep}}, w)$.

See Appendix A.1.3 for the full details of the proof. □

Lemma 2.4.1 implies the convergence of the approximate slopes $\bar{v}^k$ to the true slopes v as long as the policy approximation converges correctly.

Theorem 2.4.1. *For* $\mathrm{x} \in \{\mathrm{rep}, \mathrm{dis}\}$*, the slope approximation* $\bar{v}_t^{\mathrm{x},k}(z^{\mathrm{x}}, w)$ *converges to the slope of the postdecision value function* $v_t^{\mathrm{x}}(z^{\mathrm{x}}, w)$ *almost surely for all* (z^{x}, w) *and* t*; the policy approximations* $\bar{\pi}_t^{\mathrm{rep},k}(r, w)$ *and* $\bar{\pi}_t^{\mathrm{dis},k}(z^{\mathrm{rep}}, w)$ *respectively converge to the optimal policies* $\pi_t^{\mathrm{rep}}(r, w)$ *and* $\pi_t^{\mathrm{dis}}(z^{\mathrm{rep}}, w)$ *almost surely for all* r*,* z^{rep}*,* w *and* t*.*

Sketch of Proof. The proof depends inductively on Lemma 2.4.1. Given its result for period t, we can then argue the convergence of policy approximations $\bar{\pi}_t^{\mathrm{rep},k}(r, w)$ and $\bar{\pi}_t^{\mathrm{dis},k}(z^{\mathrm{rep}}, w)$. This allows us to re-apply Lemma 2.4.1 on period $t-1$. The details are given in Appendix A.1.4. □

2.5 Numerical Experiments

In this section, we test the performance of our algorithm empirically and compare its convergence rate with other ADP algorithms on a common set of several benchmark problems with different state space sizes. Specifically, we compare with SPAR, a standard actor-critic method with a linear architecture, a policy gradient method with a linear architecture, and tabular Q-learning. We begin by giving a brief description of these algorithms.

- The multi-stage version of SPAR, introduced in [43], takes advantage of the concavity of the value function and uses the temporal difference to update slopes without a policy approximation. More specifically, in order to generate observations $\hat{V}_t^{\mathrm{rep},k}$ and $\hat{V}_t^{\mathrm{dis},k}$, instead of using (2.15) and (2.16), SPAR uses

$$\hat{V}_t^{\mathrm{rep},k}(z^{\mathrm{rep}}, w_t^k) = -c_{w_t^k} z^{\mathrm{rep}} + \max_{z^{\mathrm{dis}} \le z^{\mathrm{rep}}} \left\{ U^{\mu^*}_{w_t^k,0}\left(z^{\mathrm{rep}} - z^{\mathrm{dis}}, \xi_{t,0}^k\right) + \bar{V}_t^{\mathrm{dis},k-1}\left(z^{\mathrm{dis}}, w_t^k\right) \right\},$$

  and

$$\hat{V}_t^{\mathrm{dis},k}(z^{\mathrm{dis}}, w_t^k) = (c_{w_{t+1}} - h) z^{\mathrm{dis}} + \max_{z^{\mathrm{rep}} \ge z^{\mathrm{dis}}} \bar{V}_{t+1}^{\mathrm{rep},k-1}\left(z^{\mathrm{rep}}, w_{t+1}\right)$$

  respectively. Although the original specification of SPAR does not use an exploration policy, we implemented ϵ-greedy with exploration rate 0.1 for improved performance.

- We implement an actor-critic (AC) method [38] based on a linear approximation architecture for both the policy and value approximations. In both cases, the basis functions are chosen to be Gaussian radial basis functions (RBFs). The "critic" approximates the value function using a weighted sum of RBF basis functions. The "actor" is a stochastic policy with a parameter $h_t(r, w; z^{\text{rep}}, z^{\text{dis}})$ for each state-action pair $(r, w; z^{\text{rep}}, z^{\text{dis}})$, and is also approximated using a weighted sum of RBFs, which indicate the tendency of selecting action $(z^{\text{rep}}, z^{\text{dis}})$ in state (r, w). The associated stochastic policy is obtained through a softmax function, so that the probability of taking action $(z^{\text{rep}}, z^{\text{dis}})$ in state (r, w) is $\pi_t(z^{\text{rep}}, z^{\text{dis}} \,|\, r, w) = e^{h(r,w;z^{\text{rep}},z^{\text{dis}})} / \sum_{(z_1,z_2)} e^{h(r,w;z_1,z_2)}$. Detailed steps of the method are shown in Appendix A.2.
- Our policy gradient (PG) method [81, 82] updates the stochastic policy in each iteration. We adopt the Monte-Carlo policy gradient method where the policy approximation follows the same softmax policy as in the AC algorithm above. There is no value function and the policy parameters are updated using a sampled cumulative reward from t to T.
- The previous two algorithms use linear architectures for generalization. We also compare to the widely-used Q-learning (QL) algorithm [40], which is called *tabular* because each state-action pair is updated independently (structured actor-critic and SPAR lie in-between these two extremes as they generalize by enforcing structure). Q-learning aims to learn the state-action value function:

$$Q_t(r, w; z^{\text{rep}}, z^{\text{dis}}) = (c_w - h)r - c_w z^{\text{rep}} + \mathbf{E}_w\big[U^{\mu^*}_{w_t,0}(z^{\text{rep}} - z^{\text{dis}}, \Xi_{t,0}) + V_{t+1}(z^{\text{dis}}, W_{t+1})\big].$$

  Our implementation is a standard finite-horizon version of the algorithm that uses an ϵ-greedy exploration policy at a rate of 0.1.

Optimal benchmarks used to determine the effectiveness of the five algorithms were computed using standard backward dynamic programming (BDP). All computations in this chapter were performed using `Python` 3.5.

Table 1: Performance (% optimality) at iterations 500 and 1000.

		At iteration 500					At iteration 1000				
$R_{\max}$	$\lvert\mathcal{W}\rvert$	3	6	9	12	15	3	6	9	12	15
	AC	97.20	97.68	98.01	97.41	96.88	98.86	99.03	98.50	98.38	97.60
	PG	73.04	76.02	72.35	76.64	74.29	77.94	79.12	73.35	79.16	75.38
20	QL	30.02	33.86	28.36	27.85	35.53	32.60	35.91	31.75	31.20	37.63
	S-AC (ours)	**99.76**	**99.26**	**98.33**	**97.68**	**97.45**	**99.83**	**99.57**	**99.00**	**98.48**	**98.50**
	SPAR	97.82	95.11	95.10	94.69	92.36	96.95	97.55	93.80	94.33	95.87
	AC	97.21	96.40	95.75	95.17	94.91	97.65	97.13	96.40	96.31	95.27
	PG	69.97	72.24	76.48	73.36	78.19	76.07	74.15	76.91	81.04	78.30
30	QL	38.26	34.09	28.84	27.47	34.21	40.35	37.14	35.43	33.99	37.78
	S-AC (ours)	**99.58**	**99.36**	**98.53**	**97.70**	**97.61**	**99.83**	**99.67**	**99.18**	**98.67**	**98.60**
	SPAR	97.85	97.94	92.57	95.11	92.58	98.62	97.88	95.24	95.12	94.46
	AC	96.30	95.16	91.63	93.24	92.15	96.70	96.05	92.56	93.94	92.54
	PG	72.95	77.04	75.57	73.92	78.39	76.51	77.78	75.90	75.39	79.15
40	QL	39.65	35.40	26.71	24.70	32.36	42.20	40.57	35.20	33.44	37.63
	S-AC (ours)	**99.45**	**99.35**	**97.95**	**97.86**	**97.50**	**99.65**	**99.61**	**98.90**	**98.53**	**98.43**
	SPAR	97.46	96.08	93.50	93.79	93.74	96.79	96.62	95.33	93.81	92.07
	AC	90.96	90.56	86.47	88.00	88.02	91.65	91.76	87.18	89.03	89.73
	PG	72.06	70.67	66.57	69.34	76.81	73.95	74.75	67.71	71.71	78.05
50	QL	41.63	36.35	26.26	22.00	29.87	47.68	42.72	35.22	31.90	35.60
	S-AC (ours)	**99.42**	**99.15**	**97.49**	**97.47**	**97.09**	**99.52**	**99.46**	**98.18**	**98.25**	**98.01**
	SPAR	95.54	96.65	90.91	91.27	94.02	97.39	96.30	92.21	94.35	90.38
	AC	91.30	91.16	86.67	87.21	88.23	92.27	92.32	88.32	87.85	90.17
	PG	74.15	71.73	61.39	63.98	68.90	76.80	71.55	65.77	66.25	70.86
60	QL	42.03	33.88	22.51	19.95	27.38	47.13	41.37	33.10	31.44	33.40
	S-AC (ours)	**99.16**	**99.00**	**96.50**	**97.08**	**96.70**	**99.25**	**99.20**	**96.81**	**97.52**	**97.00**
	SPAR	96.40	95.55	91.52	93.63	90.73	95.89	95.51	94.18	92.64	91.51

2.5.1 Benchmark Instances and Parameters

We consider 10 PODs in these synthetic benchmark instances. Each POD has a randomly generated attribute ranging between 0 and 1 representing its priority, which is reflected in the utility function u. Let the stochastic utility function be $\tilde{u}\big(\min(y_i, D_i), \xi_i\big)$, with expectation $u_w(x_i, \xi_i) = \mathbf{E}_w\big[\tilde{u}\big(\min(\mu_i(x_i, \xi_i), D_i), \xi_i\big)\big]$, where μ_i and D_i are respectively the policy and the amount of demand in sub-period i. (In reality, the utility and demand might be revealed several periods later. For modeling purposes, we assume that they are revealed by the end of the current period in this section.) Let $\tilde{u}\big(z, \xi_i\big)$ be nondecreasing and discretely concave in z, then $\tilde{u}\big(\min(y_i, D_i), \xi_i\big)$ is $L^\natural$-concave in y_i based on Lemma 2 in [83], the structural properties are kept for this stochastic utility function. Specifically, we generate the stochastic utility function $\tilde{u}(z, \xi)$ by generating its unit utility function $\Delta\tilde{u}(z, \xi) = \tilde{u}(z, \xi) - \tilde{u}(z-1, \xi)$ as

Table 2: Performance (% optimality) after 5 and 10 seconds of CPU time.

		CPU time = 5s					CPU time = 10s				
$R_{\max}$	$\lvert\mathcal{W}\rvert$	3	6	9	12	15	3	6	9	12	15
20	AC	91.97	93.91	92.29	93.21	89.98	94.80	95.93	94.87	95.32	94.32
	PG	65.49	68.53	66.84	68.06	71.71	67.85	72.33	71.14	71.83	73.27
	QL	32.60	35.91	31.75	31.20	37.63	32.60	35.91	31.75	31.20	37.63
	S-AC (ours)	**99.79**	**99.52**	**98.71**	**98.20**	**98.03**	**99.83**	**99.57**	**99.00**	**98.48**	**98.50**
	SPAR	97.84	96.93	94.19	92.20	92.83	96.95	97.55	93.80	94.33	95.87
30	AC	89.78	89.61	88.64	88.31	88.74	93.20	91.90	92.62	91.26	91.77
	PG	68.84	64.37	71.91	67.15	74.40	67.55	68.01	73.35	65.14	74.44
	QL	40.35	37.14	35.43	33.99	37.78	40.35	37.14	35.43	33.99	37.78
	S-AC (ours)	**99.62**	**99.39**	**98.00**	**97.37**	**97.40**	**99.80**	**99.67**	**99.01**	**98.35**	**98.35**
	SPAR	95.97	97.42	94.97	94.85	94.96	97.53	97.88	92.86	94.90	95.37
40	AC	86.28	89.01	83.92	86.91	85.08	92.58	92.27	89.01	89.08	88.65
	PG	68.08	63.96	72.25	61.87	73.68	69.73	68.89	71.53	67.61	73.01
	QL	42.20	40.57	35.20	33.44	37.63	42.20	40.57	35.20	33.44	37.63
	S-AC (ours)	**99.43**	**98.94**	**96.85**	**96.58**	**95.63**	**99.58**	**99.41**	**98.32**	**97.94**	**97.57**
	SPAR	97.73	96.98	92.88	92.47	92.49	96.60	96.28	95.10	92.54	92.97
50	AC	86.79	81.48	79.76	81.90	80.34	88.10	88.05	82.45	84.92	83.29
	PG	67.47	68.04	63.74	60.86	69.41	66.57	65.26	65.34	65.01	71.29
	QL	47.60	42.72	35.22	31.34	35.60	47.68	42.72	35.22	31.90	35.60
	S-AC (ours)	**98.92**	**98.51**	**95.10**	**94.60**	**93.92**	**99.36**	**99.06**	**97.01**	**96.83**	**96.33**
	SPAR	96.85	96.62	93.17	88.63	91.35	95.02	96.27	94.17	92.61	92.27
60	AC	77.05	76.73	73.10	74.15	76.31	78.78	80.14	76.33	76.38	78.48
	PG	66.10	58.84	56.40	59.41	64.81	69.42	59.67	57.36	60.11	65.73
	QL	44.75	38.60	26.78	26.41	31.62	47.13	41.37	33.10	31.44	33.40
	S-AC (ours)	**98.65**	**98.02**	**93.17**	**93.20**	92.31	**99.05**	**98.75**	**95.41**	**96.16**	**95.11**
	SPAR	95.45	95.26	90.50	92.43	**92.41**	95.90	96.15	93.45	93.11	93.02

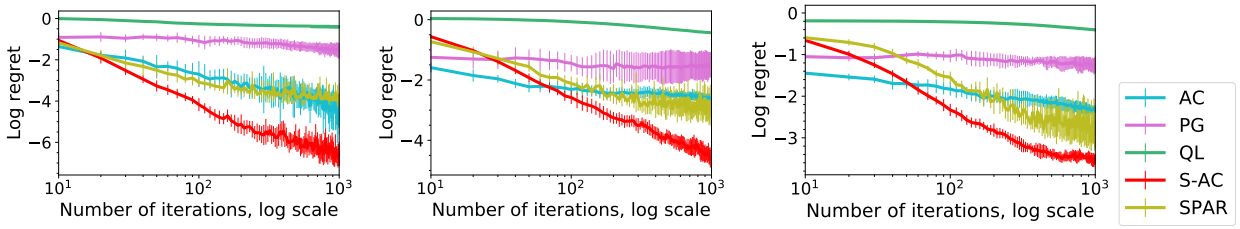
follows: $\Delta\tilde{u}(1,\xi) = 100\,(5\,\xi^3 + 1)$, $\Delta\tilde{u}(z,\xi) = \Delta\tilde{u}(z-1,\xi) - 10\,(5\,\xi^4)$. For each exogenous information realization w, we randomly generated 10 different patterns of the arriving POD sequences. A pattern of the arriving POD sequences was generated from randomly sampling ten elements from a pool which contains all the PODs and some empty elements. The number of the empty elements is dependent on w. For example, in the case of $|\mathcal{W}| = 3$, the numbers of the empty elements are 5, 10, and 15 for $w = 1$, 2, and 3 respectively. The utility of an empty element is 0.

Our interpretation of the stochastic process $\{W_t\}$ is a signal of the total demand for period t. An example for $\{W_t\}$ is the national trends of the particular public health situation, which may suggest higher demands in the region-of-interest. For benchmarking purposes, we use the model $W_{t+1} = \varphi_t W_t + \hat{W}_{t+1}$, where φ_t is deterministic and $\hat{W}_{t+1}$ is an independent noise

term that follows a mean zero discretized normal distribution with standard deviation σ_{t+1}. In this chapter, a continuously distributed random variable X is discretized to X_{disc} with $\mathbf{P}(X_{\text{disc}} = x) = \mathbf{P}(X \leq x) - \mathbf{P}(X \leq x - 1)$. Given a demand signal $W_t = w_t$, the realized demand $D_{t,i}$ is a discretized normal distribution with mean $d_i(w_t)$ and standard deviation $\tilde{\sigma}_t = 3$ for $i = 1, 2, \ldots, n$. All of the means above were generated randomly.

We created 25 benchmark problem instances by varying the sizes of the state, action, and outcome spaces (i.e., number of possible values of the exogenous information). Specifically, we consider problem instances with 21, 31, 41, 51, and 61 inventory levels and 3, 6, 9, 12, and 15 information states; these are the columns and rows shown in Tables 1 and 2. The sizes of the action spaces corresponding to inventory level sizes 21, 31, 41, 51, and 61 are respectively 231, 496, 861, 1326, and 1891. The time horizon for each instance is $T = 10$ and the cost parameters are $b = 0$, $h = 5$, $c_w \in [10, 50]$, $\mathbf{E}[c_w] = 30$.

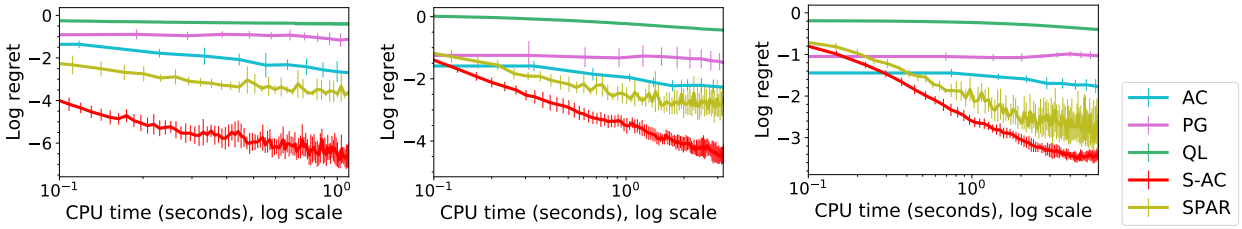
2.5.2 Optimality Gap of Approximate Policies



(a) $R_{\max} = 20, |\mathcal{W}| = 3$ (b) $R_{\max} = 40, |\mathcal{W}| = 9$ (c) $R_{\max} = 60, |\mathcal{W}| = 15$

Figure 4: Comparison of ADP algorithms with respect to iteration number.

To estimate the value $V_0^{\tilde{\pi}^k}$ of an approximate policy $\tilde{\pi}^k$, we averaged the value of initial states (r, w) drawn from a uniform distribution, where the value $V_0^{\tilde{\pi}^k}(r, w)$ is obtained from 100 Monte Carlo simulations following policy $\tilde{\pi}^k$. To evaluate the approximate policy learned from an ADP algorithm, we run 10 independent replications of the algorithm and average the performance of the learned approximate policy in each replication. Denote $\bar{V}_0^{\tilde{\pi}^k}$ the evaluation of the approximate policy learned from an algorithm. The percentage of optimality is the ratio of $\bar{V}_0^{\tilde{\pi}^k}$ to V_0, where the optimal value function V_0 is computed using BDP.



(a) $R_{\max} = 20, |\mathcal{W}| = 3$ (b) $R_{\max} = 40, |\mathcal{W}| = 9$ (c) $R_{\max} = 60, |\mathcal{W}| = 15$

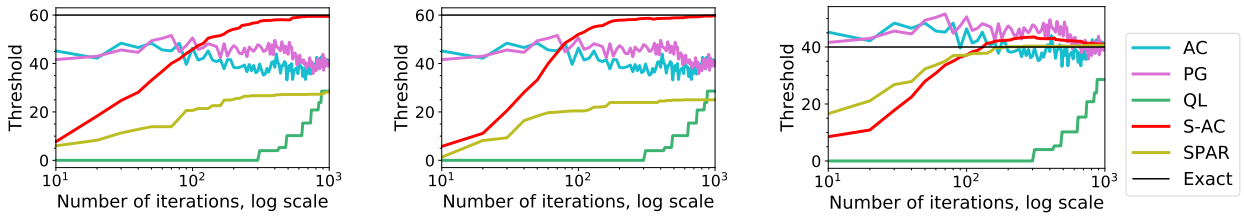
Figure 5: Comparison of ADP algorithms with respect to CPU time.

Tables 1 and 2 show the percentage of optimality of each algorithm at specific iterations and CPU times, across all problem instances. In almost all instances and comparison points, S-AC outperforms the baseline algorithms. AC is the most competitive baseline with respect to the number of iterations and SPAR is the most competitive when CPU time is of primary interest. Within the same number of iterations and CPU times, the performance of all the ADP algorithms becomes worse as the size of the problem increases; however, S-AC seems to be less sensitive than the others to problem size. Let us compare the percentage of optimality of the instance with $R_{\max} = 20$ and $|\mathcal{W}| = 3$ and the instance with $R_{\max} = 60$ and $|\mathcal{W}| = 15$ at iteration 1,000. The performance of AC, S-AC, and SPAR on the latter large instance is respectively 7.2, 2.8, and 6.5 percentage worse than the performance on the smaller instance. For the same instance at CPU time 10 seconds, the performance of the three algorithms on the larger instance is respectively 14.7, 4.7, and 4.9 percentage points worse than the performance on the smaller instance.

To further illustrate the performance of each ADP algorithm, we show the convergence curves of three instances with different sizes. Let us consider three problem instances: (1) $R_{\max} = 20$, $|\mathcal{W}| = 3$, (2) $R_{\max} = 40$, $|\mathcal{W}| = 9$, and (3) $R_{\max} = 60$, $|\mathcal{W}| = 15$. Figure 4 shows the rate of convergence of the ADP algorithms considered in this chapter as a function of the number of iterations, while Figure 5 shows the rate of convergence as a function of the computation time. We plot "log regret" (log of the suboptimality from 100%) to help improve the visualization.

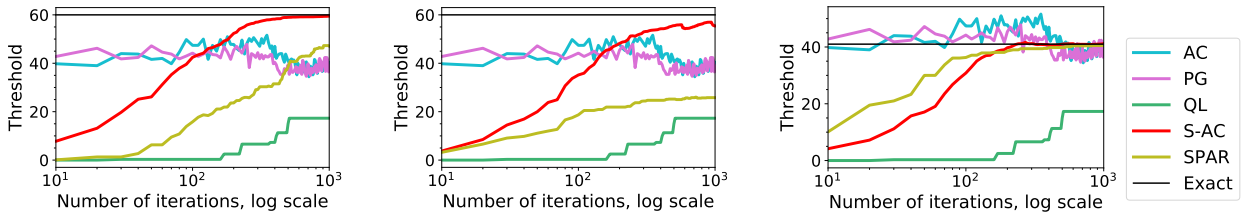
The policy approximations used in AC and PG are parameterized as stochastic policies initialized to take uniformly random actions in each state. This exploration helps to generate relatively high value in early iterations. AC and PG are very competitive with our S-AC algorithm when comparing performance with respect to the iteration count. However, this comes at a computational cost: although stochasticity encourages exploration, Figure 5 shows that each iteration is particularly time-consuming when compared to deterministic policies.

2.5.3 Convergence of Implied Basestock Thresholds



(a) $w_0 = 1$ (b) $w_0 = 4$ (c) $w_0 = 8$

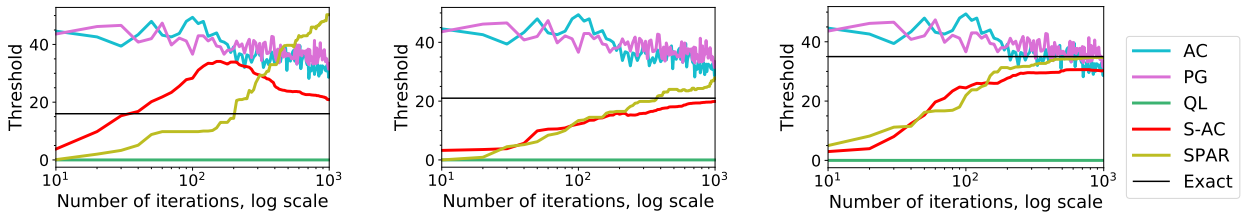
Figure 6: Convergence of replenish-up-to thresholds at $t = 0$ for the $R_{\max} = 60, |\mathcal{W}| = 9$ instance.



(a) $w_0 = 2$ (b) $w_0 = 6$ (c) $w_0 = 10$

Figure 7: Convergence of replenish-up-to thresholds at $t = 0$ for the $R_{\max} = 60, |\mathcal{W}| = 12$ instance.

Next, we are interested in examining how the implied replenish-up-to thresholds evolve as each algorithm progresses. The thresholds of AC and PG are selected as the actions with highest probabilities for state $r = 0$ and the thresholds of SPAR and QL correspond to the



(a) $w_0 = 2$ (b) $w_0 = 7$ (c) $w_0 = 12$

Figure 8: Convergence of replenish-up-to thresholds at $t = 0$ for the $R_{\max} = 60, |\mathcal{W}| = 15$ instance.

greedy policy with respect to the value function and state-action value function approximations. In this part, we take three problem instances as examples, whose storage capacities are all $R_{\max} = 60$, and exogenous information spaces are $|\mathcal{W}| = 9$, $|\mathcal{W}| = 12$ and $|\mathcal{W}| = 15$ respectively. Figures 6 to 8 show the convergence of approximate replenish-up-to threshold levels $\bar{l}^{\text{rep},k}$ as well as the optimal levels l^{rep} (denoted "Exact" in the plots) for three different exogenous information states w_0 at period $t = 0$ for the selected problem instances.

We see that the thresholds generated by S-AC quickly converge to the optimal ones in all instances. Due to the smoothing step of S-AC, the convergence is also observed to be relatively stable. On the other hand, the thresholds of AC, PG, QL, and SPAR tend to either have large gaps to the optimal thresholds or converge in a noisy manner. Stability of the basestock thresholds is particularly useful if S-AC is to be used in an online manner in practice, where drastic changes in the policy from one time period to the next (as observed in the competing algorithms) would be impractical. These results attest to the value of utilizing the structural properties of the policy and value function.

2.5.4 Sensitivity Analysis

In this section, we study the impact of model parameters. We take the instance with $R_{\max} = 50$ and $|\mathcal{W}| = 9$ in Section 2.5.1 as the base instance, and vary parameters in the model to evaluate the impact of each parameter. The results are summarized in Table 3. Each value in the table is an average of ten replications. For each replication, we take the policy

Table 3: Impact of parameters on ADP algorithms for the $R_{\max} = 50, |\mathcal{W}| = 9$ instance.

Parameter	Value	AC	PG	QL	S-AC	SPAR	Exact
Mean total demand	30, Normal	19,037	16,009	7,287	**20,313**	19,077	21,332
	30, Uniform	18,113	15,142	8,476	**20,865**	20,098	21,332
	50, Normal	28,422	23,237	10,318	**29,080**	28,278	29,387
	50, Uniform	28,023	23,112	10,286	**29,077**	28,150	29,387
Mean ordering cost	30	30,914	25,488	15,125	**33,532**	32,671	34,647
	50	18,037	14,009	7,287	**20,313**	19,077	20,689
	70	11,257	8,660	6,032	**11,866**	11,553	11,984
Holding cost	5	18,037	14,009	7,287	**20,313**	19,077	20,689
	20	18,402	15,064	7,189	**19,839**	19,285	20,131
	35	17,807	14,498	5,855	**19,381**	18,784	19,592
	50	17,150	15,011	4,582	**18,988**	18,418	19,203
	65	16,575	13,708	2,954	**18,597**	17,931	18,835

learned by the algorithm at iteration 1,000 and evaluate it by averaging 100 simulations. The first parameter we are interested in is the demand distribution. We consider two types of distribution, normal and uniform distributions. For each type of distribution, we consider two values of the average demand of all PODs in a period, 30 and 50. The table shows that the value is highly influenced by the expected demand, and that with the same expected demand, the type of distribution has relatively little impact on the performance. We are also interested in the impact of the costs in the model. The ordering cost has a much larger impact than the holding cost, and any increase in the ordering cost can significantly reduce the value of the policy. We also note that S-AC finds near-optimal policies in each of these cases.

2.6 Case Study: Naloxone for First Responders in Pennsylvania

Our case study is motivated by the need to distribute naloxone (a drug that can reverse overdoses within seconds to minutes) amidst the ongoing opioid overdose crisis, which is affecting communities across the state of Pennsylvania. Our case study makes use a time-series demand model for naloxone, fit using publicly available data from [84]. Our model

in this section contains a five-dimensional information state W_t, which makes the standard version of S-AC intractable. Instead, we leverage an aggregation-based version of S-AC, whose details are introduced in Appendix A.3. In essence, the method uses clusters of the exogenous information state (via k-means clustering) and learns a cluster-dependent policy. When implementing the policy, we use regression to interpolate between clusters. Our experimental results show that this simple extension of S-AC for the case of a continuous and multi-dimensional information state is surprisingly effective.

2.6.1 Description of Naloxone for First Responders in Pennsylvania

The rate of opioid overdose deaths has quadrupled since 1999 [85], with heroin deaths alone outpacing gun homicides in 2015 [86]. Moreover, in 2015, drug overdose deaths in U.S. exceeded the combined mortalities from car accidents and firearms [87, 88]. By August 2020, the number of deaths from synthetic opioids was 52% more than the previous year [89]. There is significant benefit for drug users, family members, community members, law enforcement officers, and medical professionals alike to have training and access to the overdose reversal drug naloxone for use in risky situations (see Pennsylvania's Act 139).

Table 4: Parameters used in the NFRP case study.

Parameter	Value	Meaning/Explanation
WTP/unit	\$31,000	Willingness to pay (WTP) for a unit of naloxone. Product of the next 2 entries.
WTP/QALY	\$50,000	WTP per quality-adjusted life-year (QALY) [90, 75].
QALY/unit	0.62	QALY adjustment factor for lives saved by naloxone. The average of utilities of "High-risk/low-risk prescription opioid use" and "Illicit opioid use" in [77].
Ordering cost	\$185.30	Approximate retail price of an auto-injector form of naloxone [91].
Treatment cost	\$2,976	The cost for EMS visit, EMS transport to hospital, and emergency department care [75].
$R_{\max}$	700	Capacity of the central storage.
h	\$10	Holding cost.
b	0	Disposal cost.

In this case study, we consider a somewhat simplified setting of a public health organization modeled after Naloxone for First Responders Program (NFRP), which distributes naloxone through a Centralized Coordination Entity (CCE). We use the top five counties in terms of overdose incidents responded to by emergency medical services (EMS) from publicly available data [84], Allegheny County, York County, Bucks County, Dauphin County, and Luzerne County (all of which have incident numbers over 1,000), as the five PODs (first responders) in our case study.

The parameters of our utility function are based on values found in [90], [75], [77], and [84]. Since the naloxone dispensed to first responders is used to reverse overdoses, we use willingness to pay (WTP) per unit of naloxone to measure the utility per demand satisfied. Specifically, let the WTP per unit of naloxone minus the treatment cost (EMS visit and related costs) be the unit utility Δu, similar to the approach taken by [75]. To reflect the different expected demand among counties, we adopt the following expected utility function in the case study: $u_w(y_i, \xi_i) = \Delta u \, \mathbf{E}_w\big[\min(y_i, D_i)\big]$, where D_i is the demand of POD i. The demand is computed as follows: based on data from [84], 1-9 doses of naloxone are administrated to reverse an overdose. The demand of POD i at period t equals to a sample of the doses of naloxone needed to reverse w^i incidents, where w^i is the i-th element of w. Further details (ordering cost, capacity of storage, and holding cost) are available in Table 4.

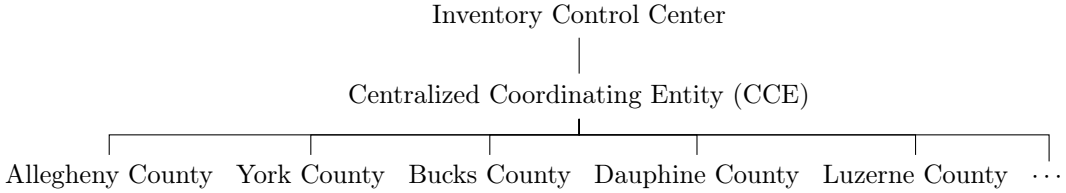
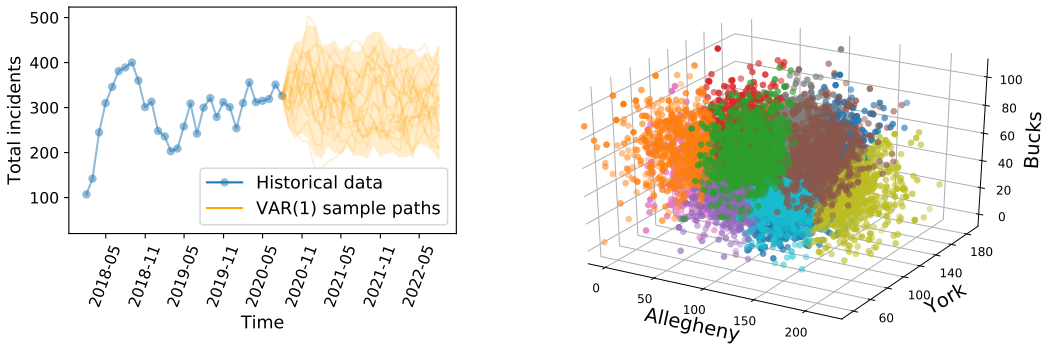


Figure 9: The hierarchical system structure used in the case study.

The system consists of an inventory control center, a dispensing coordinator, and multiple first responders as shown in Figure 9. Let the time horizon for the case study be $T = 12$ months. At each period t, the inventory control center replenishes the inventory of naloxone after observing the recent incident history, modeled as the county-level incident count of the last period (thus, $W_t \in \mathbb{R}^5$). The control center then decides the total amount of naloxone to



(a) Number of incidents and predictions.

(b) Three dimensions of k-means clustering.

Figure 10: Total overdose incidents of the five PODs and k-means visualization.

dispense in the current period. This naloxone is delivered to the CCE, who makes lower-level quantity-of-dispensing decisions based on the attribute of the arriving POD ξ_i, the current available naloxone in stock (the inventory level x_i), and the upper-level county-level incident count of the last period W_t. In the case study, the exogenous information is the incident history, which consists of the number of incidents from the five counties last month. We a vector autoregression (VAR) time-series model with a lag of 1.

Figure 10a shows the monthly number of overdose incidents in the five counties from January 1st, 2018 to July 31st, 2020, and 20 sample paths from the VAR(1) model for the next 24 months. The first planning period of the case study is July 2020. To generate the state aggregation, we sample 10,000 paths of the exogenous information, and use k-means clustering to cluster them into 12 clusters. Figure 10b shows the first three dimensions of the resulting clustering that is then used by S-AC.

2.6.2 Performance of the Algorithm

We denote the aggregate version of our algorithm `S-AC+DPR`, whose upper-level policies are learned by aggregate S-AC (see Appendix A.3). The learned cluster-dependent upper-level policies are then interpolated between clusters using Gaussian process regression. The lower-level policies are solved using a discretized DP we then interpolate using linear re-

gression (DPR). In this section, we first study the performance of S-AC+DPR compared with AC+DPR and a suite of heuristic strategies. Next, we illustrate the impact of the various approaches on the lower-level dispensing decisions to each POD, showing some stark differences between the methods. Finally, we show some sensitivity analysis of the cost parameters on the value of the learned policies and heuristics.

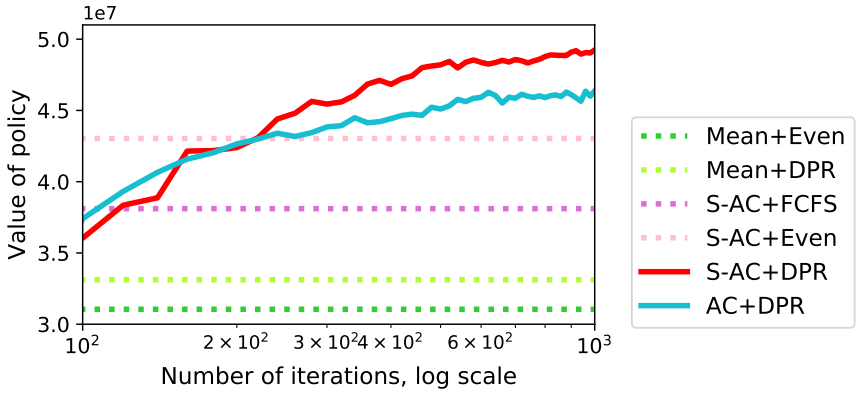


Figure 11: Convergence curve of S-AC and AC compared to performance of heuristics.

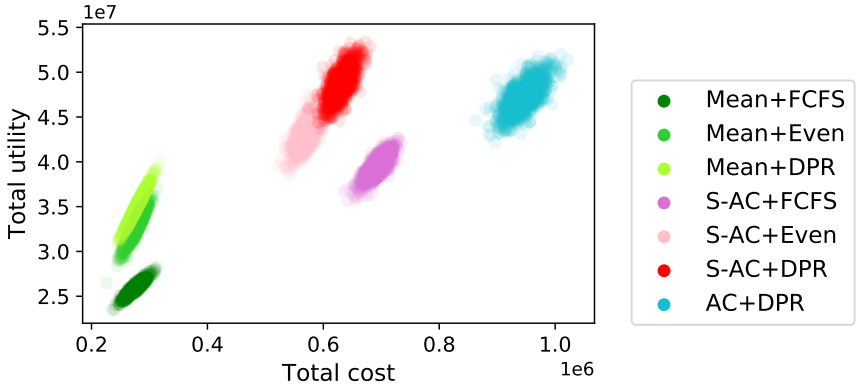


Figure 12: The relationship between total cost and total utility for each method.

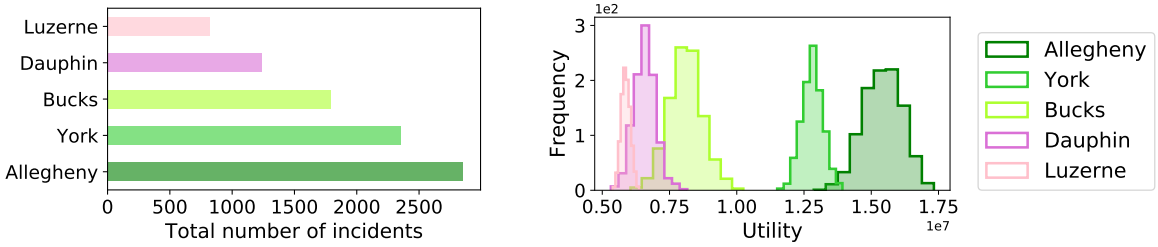
2.6.2.1 Convergence and Comparison with Heuristics We first describe the heuristic strategies to which we compare our new policy. We make a distinction between the upper-level and lower-level policies and consider two approaches for the upper-level and three

approaches for the lower-level, resulting in six combined strategies. On the upper-level, we either take the S-AC policies (`S-AC`) or always replenish-up-to the expected demand and dispense-down-to zero (`Mean`). The expected demand for a given exogenous information w equals to the sum of the elements of w times the average doses per reverse (1.517), which is computed by averaging the "dose count" in the dataset [84] (excluding the cases without applying naloxone). On the lower-level, the three strategies are: (1) take the policy trained using dynamic programming and interpolated to the continuous state space by linear regression (`DPR`), (2) evenly dispense naloxone to the five PODs (`Even`), and (3) follow the first-come-first-serve rule (`FCFS`), in which we dispense the expected demand of each POD upon its arrival until all the available resources are dispensed. We also apply `AC+DPR` as an alternative ADP method to which we can compare `S-AC+DPR`. We selected `AC` because it performs relatively well in Section 2.5 and is scalable to high-dimensional problems (unlike `QL` or `SPAR`).

Figure 11 shows the cumulative performance of the policies over a year, averaged over 100 simulations (the value of policy `Mean+FCFS` is smaller than 3e7 and is removed from the plot to better show the results). We see that compared with the upper-level heuristic `Mean`, applying `S-AC` on the upper-level improves the performance (i.e., compare `S-AC+DPR` with `Mean+DPR` and `S-AC+Even` with `Mean+Even`). This is due to the ability of the state-dependent basestocks to adapt to dynamic state information. On the lower-level, we see that `DPR` outperforms the heuristics `FCFS` and `Even` (i.e., compare `S-AC+DPR` with `S-AC+FCFS` and `S-AC+Even`, and `Mean+DPR` with `Mean+Even`) significantly. The reason is that the heuristics `FCFS` dispensing policy is unable to take advantage of the large initial gains in dispensing resources to all of the first responders, and the heuristic `Even` dispensing policy is unable to adjust the dispensing decision to a POD based on exogenous information.

Figure 12 shows the total cost vs. total utility for each method that we tested, which helps to illustrate the trade-offs associated with each. The total cost is mostly determined by the upper-level policy (i.e., the scatters of `Mean+FCFS`, `Mean+Even` and `Mean+DPR` are close on the x-axis, and the scatters of `S-AC+FCFS`, `S-AC+Even` and `S-AC+DPR` are close on the x-axis). The upper-level policy `AC` tends to always replenish the inventory up to a high level, which leads to the highest total cost. The heuristics `Mean` considers the exogenous information by always replenishing up to the expected demand and dispense all the inventory to the PODs;

this approach leads to the lowest total cost. The upper-level policy learned by `S-AC` is able to adapt to the exogenous information state and usually replenishes up to a level that is higher than the expected demand. It also sometimes retains a small portion of the inventory to the next period. With the same upper-level policy, although the total cost is similar, the total utility differs when applying different lower-level policy. This observation suggests that by applying a smarter lower-level policy `DPR`, it is possible to achieve more utility without spending much more cost. Overall, we see that our primary approach `S-AC+DPR` *attains the highest levels of utility while expending relatively moderate cost.*



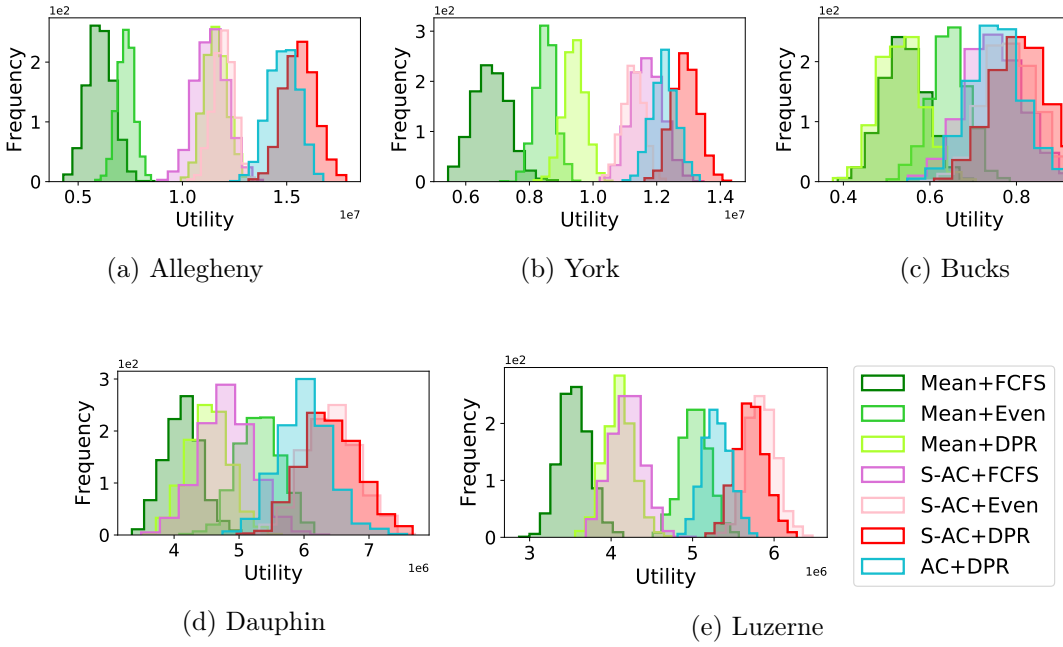
(a) County-level overdose incidents.

(b) Cumulative utilities achieved by `S-AC+DPR`.

Figure 13: Historical overdose incidents learned by `S-AC+DPR`.

2.6.2.2 Utilities of Different First Responders We now investigate the individual POD (or first responder) utilities achieved under each algorithm. Following the policy obtained after $1,000$ iterations of `S-AC+DPR`, we get the total utility of each POD during the entire planning horizon. Under our utility function definition and the parameters given in Table 4, the PODs with higher levels of overdose incidents are associated with a higher utilities than PODs fewer incidents. Thus, we expect that a good inventory and dispensing policy will learn to prioritize these high-utility PODs. Figure 13 shows the histograms of 1,000 simulations for the utilities of the five PODs alongside the historical county-level overdose incidents.

To investigate how each method prioritizes the different PODs, we show the utilities of each POD generated by each policy in Figure 14. `S-AC+DPR` leads to the highest utilities of the first three counties, while `S-AC+Even` leads to the highest utility of the last county. These



(a) Allegheny (b) York (c) Bucks

(d) Dauphin (e) Luzerne

Figure 14: Comparison of the cumulative utilities for each method.

two policies perform similarly for the fourth county. Moreover, the two ADP algorithms `S-AC+DPR` and `AC+DPR` are in the top three policies for all of the counties' utilities.

When both levels' policies are heuristics (i.e., `Mean+FCFS` and `Mean+Even`), the utility of all PODs are low, with `Mean+FCFS` leading to the lowest utilities in all cases. When only the upper-level is a heuristic (i.e., `Mean+DPR`), the utilities are still not particularly high; in fact, this method ranks in the bottom three policies for all the PODs except for Allegheny. When the upper-level is `S-AC` and the lower-level is a heuristic (i.e., `S-AC+FCFS` and `S-AC+Even`), the utilities of the top three counties are higher than the utilities by achieved using `Mean`. The policy `S-AC+Even` never falls in the bottom three policies; `S-AC+FCFS` performs reasonably but is part of the bottom three policies for Dauphin and Luzerne. In summary, both the upper-level and the lower-level policies play an important in this problem: a properly designed lower-level heuristic can achieve good utility values for some of the PODs; however, intelligent policies on both the upper and lower-levels is necessary to achieve the overall improvement.

Table 5: Simulated value of the policies on instances with different ordering costs (value in 10 million).

Ordering cost	Mean+FCFS	Mean+Even	Mean+DPR	S-AC+FCFS	S-AC+Even	S-AC+DPR	AC+DPR
185	2.07	3.10	3.31	3.81	4.30	**4.92**	4.64
925	1.94	2.96	3.39	3.70	4.10	**4.46**	4.34
1,850	1.75	2.77	3.04	3.12	3.56	**3.92**	3.76
3,700	1.39	2.41	2.00	2.06	2.62	**2.83**	2.56

2.6.2.3 Ordering Cost Sensitivity Analysis Table 5 shows the effect of the ordering cost on the performance (in terms of value achieved) of the various algorithms. The other costs (holding cost and disposal cost) exhibited very minor effects on the value and thus we omitted the results. Each value in the table is an average of twenty replications of the algorithm, and for each replication of the ADP algorithms, `S-AC+DPR` and `AC+DPR`, we take the policy learned by the algorithm at iteration 1,000 and evaluate it by averaging 100 simulations. The table shows that `S-AC+DPR` outperforms the other approaches in all settings. When the ordering cost increases to 5 times (increases from 185 to 925), the value of `S-AC+DPR` decreases 9.35%, and when it increases to 20 times (increases from 185 to 3,700), the value decreases 42.48%. These results indicate that the ordering cost of naloxone has a significant influence on the operations of a public health department.

2.6.3 Extensions

We showed how an aggregation-based version of S-AC along with k-means clustering can be used to handle the multi-dimensional continuous features used in the case study. There are also other possible extensions to S-AC that can make it more scalable to high-dimensional problems. For example, shape-constrained deep neural networks [92] [93] can handle both monotonicity and concavity via penalization of derivatives during training. In principle, our S-AC algorithm could be extended to use techniques like these, but the same core principles of S-AC would remain intact. We leave these investigations to future work.

2.7 Conclusions

In this chapter, we formulate a hierarchical MDP model for the sequential problem of optimizing inventory control and making dispensing decisions for a public health organization. We propose a novel, provably convergent actor-critic algorithm that utilizes problem structure in both the policy and value approximations (state-dependent basestock structure for the policy and concavity for the value functions). Although the algorithm was developed in the setting of our specific MDP, the general paradigm of a structured actor-critic algorithm is likely to be of broader methodological interest. Numerical experiments show that high-quality policies can be obtained in a small number of iterations and that the convergence of the policy is significantly less noisy when compared to competing algorithms. Lastly, we propose an aggregation-based version of our algorithm and provide a case study for the problem of dispensing naloxone to first responders.

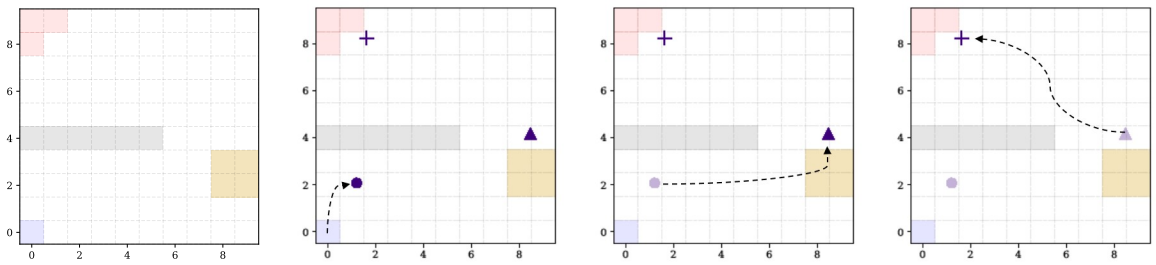
3.0 Subgoal-based Exploration via Bayesian Optimization

Reinforcement learning (RL) is becoming the standard for approaching control problems in environments whose dynamics – usually modeled by a Markov decision process (MDP) – are unknown and learned from data. In many applications, rewards are sparse and delayed, and since most RL algorithms rely, at least initially, on random exploration, this can cause an agent to require a large, often impractical number of interactions with the environment before obtaining any rewards. Simultaneously, in real-world settings, it is often the case that fast and cheap interactions with the environment are *not available*, making it nearly impossible to apply RL algorithms. To address the two issues of sparse rewards and expensive interactions, our goal in this chapter is to design methods for learning better exploration policies in a *cost-efficient* manner.

An illustrative example comes from the field of robotics: autonomous systems have long been used to explore unknown or dangerous terrains, including meteorite search in Antarctica [94], exploration of abandoned mines [95, 96], and navigation of terrains on Mars [97]. Offline policies are the norm in these situations, but it may be beneficial to introduce agents that execute an offline-learned exploration policy to guide the learning of an *online* policy that can better tailor to the details of the test environment. [97] describe the design of a rover for the Mars Pathfinder mission, where one of the main tasks is navigating the rover in a rocky terrain and reaching a goal. To train for the eventual mission, the engineers utilized an "indoor arena" that mimics the true environment. The need for cost-efficient training also arises in other settings where real robot interactions are used: automatic gait optimization [98, 99], safe robot navigation [100], and accurate object modeling using active touch strategies [101]. Existing approaches to exploration have largely ignored the need to be cost-efficient during training process and therefore are challenging to apply in the scenarios described here (see Section 3.1).

In our setup, an agent is given a fixed number of opportunities to train in environments randomly drawn from a distribution Ξ (henceforth, we refer to these as "training environments"), with the caveat that each interaction in the training environment *incurs a cost*.

After these opportunities are exhausted, the agent enters a random *test environment* $\xi \sim \Xi$ and executes an underlying RL algorithm to adapt to the particulars of ξ, while aided by the higher-level exploration strategy learned for Ξ. One can view this formulation as a meta-optimization problem with two levels: an upper-level problem to select the exploration strategy θ (for the distribution Ξ) and a lower-level RL problem that explores with the help of θ on an environment instance $\xi \sim \Xi$.



(a) Original (b) First subgoal (c) Second subgoal (d) Third subgoal

Figure 15: Example of a dynamic subgoal exploration strategy.

Note: The first, second, and third subgoals are denoted by the circle, triangle, and cross, respectively. The blue square is the starting location of the agent, the grey region is a wall, the yellow region is the location of the key, and the red region is the door (goal).

We propose optimizing over a class of *dynamic subgoal exploration strategies* in the upper-level optimization problem. To illustrate this concept, consider the sparse-reward environment shown in Figure 15a, where an agent is tasked with picking up a "key" in the yellow region, in order to exit the "door" in the red region. The grey region is a wall. An RL algorithm paired with a naive exploration strategy making use of random actions (such as ϵ-greedy) requires a prohibitively large number of random actions before finding a suitable path to the door though the key, while avoiding the wall. A dynamic subgoal strategy is an *ordered* set of subgoals (along with associated rewards leading to each subgoal, omitted here for illustrative clarity) that leads the agent on a trajectory where the underlying RL algorithm is likely to discover the optimal behavior. Figures 15b-15d together show an example of a dynamic subgoal exploration with three subgoals, which first leads the agent to the vicinity of the key and later towards the door. Note that the situation here in Figure

15 is simplified in that we are actually interested in finding dynamic subgoal strategies that work on average across a distribution of environments, rather than a single environment.

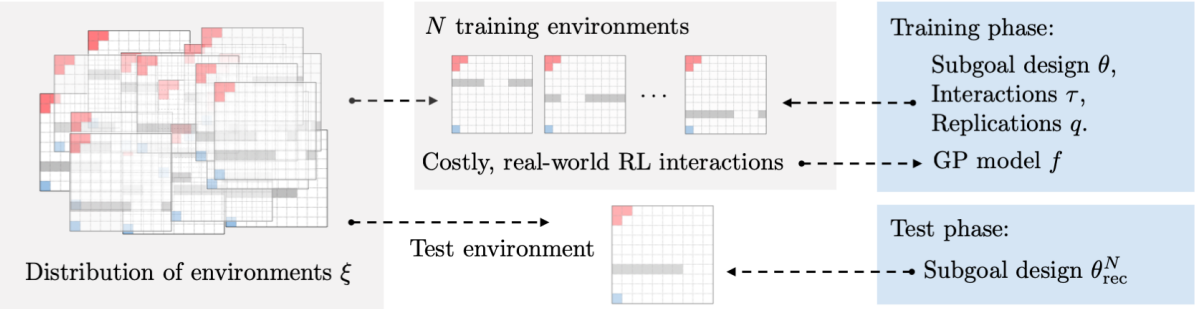


Figure 16: Outline of the BESD algorithm.

Note: During the training phase BESD optimizes an exploration strategy (represented as subgoals) on sampled training environments. It then utilizes the learned subgoal design as an exploration strategy in the test environment to train an effective policy within a limited number of interactions.

3.0.1 Our Contributions

Our contributions are as follows. We first propose a framework for *cost-efficient learning* of a dynamic subgoal exploration strategy for a distribution of environments; in other words, interactions with the environment are expensive *during training*, making most gradient-based approaches infeasible. We instead leverage the Bayesian optimization (BO) paradigm, a well-known class of sample-efficient optimization techniques [102, 103, 104, 105], and propose a new acquisition function as a core ingredient of our approach. The Gaussian process (GP) surrogate model used by the BO formulation has the ability to reason about the *learning curve* of the underlying RL algorithm, enabling us to introduce two additional levers in the BO learning process to improve cost-efficiency: (1) how long to run each episode of training, (2) the number of replications to run in each training environment. These levers allow us to intelligently trade-off running a longer trial versus more replications of shorter trials; the motivation is that, given $\tau_1 < \tau_2$, an accurate evaluation of a particular exploration strategy θ after τ_1 steps may be more informative than a noisy evaluation of θ after τ_2 steps, even though the same number of environment interactions are used in both

cases. The proposed algorithm, Bayesian exploratory subgoal design (`BESD`), is outlined in Figure 16. We also prove an asymptotic guarantee on the quality of the solution found by our approach, compared to the best possible subgoal-based exploration strategy within a given parameterized class.

3.1 Related Work

Our framework of cost-efficient learning of exploration strategies through BO appears to be distinct from existing formulations in its strong focus on expensive environmental interactions *during training*, made possible through the additional control levers of episode length and number of replications. Nevertheless, our work is related to a number of distinct areas of study: Bayesian optimization, exploration for RL, intrinsic reward and reward design in RL, multi-task RL, and transfer learning. Here, we attempt to give a tour through the various strands of relevance in each field.

3.1.1 Bayesian Optimization

Our approach follows the BO paradigm, a technique for optimizing black-box functions in a sample-efficient manner, in particular for tuning ML models and design of experiments [103, 102, 105, 104]. Our work also bears resemblance to methods for network architecture search and optimization with multiple information sources or fidelities [106, 107, 108, 109, 110] and in particular, the ability of our approach to select the length of an RL training episode builds upon [111] and [112], both of which propose acquisition functions that consider the ratio of "information gain" to cost of evaluation. Our approach also reasons about multiple replications in an environment, similar to the problem studied in [113] in the context of computer experiments. Our work fills a gap in the Bayesian optimization literature where the length of training and number of replications are *selected jointly* in a cost-aware setting, a natural and powerful idea that has not been considered in the literature. Our theoretical analysis builds upon techniques developed in [114] and

[111] but extend them in new directions, accounting for the ability to select the number of replications and providing a characterization of the asymptotic suboptimality due to using a discretized domain. (This is a common computational technique used when optimizing complex acquisition functions [115, 116, 111], but none of the previous works have addressed it in theoretical analyses.)

3.1.2 Exploration in Reinforcement Learning

Naive exploration strategies such as ϵ-greedy can lead to unreasonably large data requirements, making exploration a commonly studied topic in RL. Most existing work focus on proposing a fixed exploration strategy that is executed for a single underlying environment. For example, some previous related work employ approaches based on optimism [117, 118, 119, 120], while others use value-related methods [121, 122, 123, 124] to guide exploration. Our work departs from these existing studies in that we formulate the problem of exploration as a meta-optimization over a parameterized class of exploration strategies and aim to find a suitable strategy for a distribution of environments. A more closely related paper is [125], which extends the model-agnostic meta-learning (MAML) approach of [126] to the problem of exploration for a set of tasks in a way that is similar in spirit to our formulation. However, their gradient-based approach is not sample-efficient and costly environment interactions during training is not considered. In addition, [125] makes use of task-specific parameters during training, limiting their approach to a small set of environments. For a more comprehensive list of methods for exploration in RL, we refer the reader to the excellent survey of [127].

3.1.3 Options in Reinforcement Learning

The concept of options, which are temporally extended actions represented as a policy and a termination condition, is a way to improve the efficiency of RL through the use of previously acquired "skills" [128, 129]. These skills might be acquired with the help of a human, either fully user-specified (e.g., [130]) or obtained from expert demonstrations (e.g., [131] and [132]). Of particular relevance to our work is when options are automatically

discovered, a problem that is well-known to be challenging. One stream of work views option discovery to be (at least somewhat) detached from the RL reward maximization objective, using state visitation frequencies [133, 134, 135], clustering [136], novelty [137], local graph partitioning [138], or diversity objectives [139, 140], to name a few examples. Approaches that consider an integrated objective for option learning like ours [141, 142, 143, 144, 145] typically use large, neural network-based representations along with gradient-based (meta-) optimization and do not focus on cost-aware training. In contrast, our primary concern is the issue of expensive environment interactions during training and propose a novel BO algorithm to tackle this problem.

3.1.4 Intrinsic Reward and Reward Design

When a particular subgoal of our proposed dynamic subgoal exploration strategy is active, we "turn on" a set of artificial rewards that incentivize the agent to move toward that subgoal (these rewards are then removed after the agent moves on to the next subgoal). Hence, the literature on intrinsic reward and reward design in RL are also relevant. Intrinsic reward (also called *intrinsic motivation*) helps an agent learn increasingly complex behavior in a self-motivated way [146, 147, 148, 149, 150, 151, 152, 153, 154]. Several works from the *reward design* literature are most closely related to this chapter. [155] and [156] directly optimize the intrinsic reward parameters, via gradient ascent, to maximize the outcome of the learning process. Similarly, [157] use intrinsic rewards in policy gradient, and treat the parameters of policy as a function of the parameters of intrinsic rewards. Again, these methods differ from ours in that they do not consider the costliness of training and focus on finding intrinsic rewards for a single MDP.

3.1.5 Multi-task RL and Transfer Learning

Also related to our setting are methods that aim to train agents with the capability of solving (or adapting to) multiple sequential decision making tasks [158, 159, 160, 161, 162, 163, 126, 164, 165, 166, 167, 168]; such methods generally fall under the umbrella of *multi-task RL* or *transfer learning*. As before, many of these methods require the training of large

neural networks and are not designed for a cost-aware setting. Despite their stated purpose of being sample-efficient in adapting to new tasks, most multi-task RL or transfer learning approaches do not place a strong emphasis on cost-efficiency of training on existing tasks. This is an important distinction to our work. The two papers that are closest in spirit to our work are [158], where macro-actions are extracted from previous tasks, and [159], where shaped rewards are learned for a set of tasks. One drawback of [158] is that it assumes access to optimal policies for an initial set of MDPs. [159] directly uses previous value functions as shaped rewards (thereby requiring the agent to solve some tasks from scratch) and does not provide an avenue for cost-effective exploration.

3.2 Problem Formulation

In this section , we formulate the problem mathematically, by defining the original (sparse-reward) MDPs and how a dynamic subgoal exploration strategy induces an auxiliary, "subgoal-augmented" MDPs. We then describe the iterative training process.

3.2.1 Original MDPs $\mathcal{M}_\xi$ with Sparse Rewards

Consider a family of MDPs $\{\mathcal{M}_\xi = \langle \mathcal{S}, \mathcal{A}, T_\xi, R_\xi, \gamma \rangle\}$ parameterized by a random variable $\xi \sim \Xi$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, T_ξ is the transition matrix, $R_\xi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the extrinsic reward function and $\gamma \in [0, 1]$ is the discount factor. In Section 3.2.3, we describe how a dynamic subgoal exploration strategy supplements the extrinsic reward function with additional *intrinsic* rewards. Ξ is an arbitrary distribution, so our model handles the case where there is an infinite number of possible environments. The distribution Ξ is not assumed to be known. (Note that our approach also applies to the case of a *single environment* if the distribution contains only one environment.) In the sparse-reward setting, R_ξ is often only non-zero when the agent lands in a small number of "goal" states. We assume common state and action spaces across the distribution of MDPs (i.e., they are independent of ξ), while the reward and transition functions vary with ξ.

Given $\mathcal{S}$ and $\mathcal{A}$, a *policy* π is a mapping such that $\pi(\cdot \mid s)$ is a distribution over $\mathcal{A}$ for any state $s \in \mathcal{S}$. For any $\xi \sim \Xi$, define the *value function* of policy π at any state s as

$$V_\xi^\pi(s) = \mathbf{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_\xi(s_t, a_t, s_{t+1}) \,\middle|\, \pi, s\right], \tag{3.1}$$

where s is the initial state and $a_t \sim \pi(\cdot \mid s_t)$. For the MDP $\mathcal{M}_\xi$, its optimal value function and associated optimal policy are

$$V_\xi^*(s) = \sup_\pi V_\xi^\pi(s) \quad \text{and} \quad \pi_\xi^*(s) \in \arg\max_{a \in \mathcal{A}} \mathbf{E}\big[R_\xi(s, a, s') + \gamma V_\xi^*(s') \,\big|\, s, a\big].$$

When the extrinsic reward function R_ξ is sparse, it produces little to no learning signal for the agent. Under most RL algorithms, the agent essentially performs random exploration and does not start learning until the first time it wanders to the goal. The time it takes to find the goal under a random exploration strategy, such as ϵ-greedy, is often prohibitively long. For example, in a 20×20 gridworld with a sparse reward, the goal is not even reached for the first time by a standard Q-learning agent (let alone find an optimal policy) after 10 million interactions.

3.2.2 Dynamic Subgoal Exploration Strategies

Now, we formally define a *dynamic subgoal exploration strategy*, which uses a sequence of subgoals, along with a reward shaping function for each subgoal, to provide an artificial and intrinsic reward signal for the agent that, if properly designed, can direct the agent to explore useful parts of the state space.

Suppose there are K subgoals and let $\theta \in \Theta$ represent a subgoal parameterization. Let $\mathcal{G}_{\theta,j} \subseteq \mathcal{S}$ be a set of "target" states associated with the kth subgoal, for $k \in \{1, 2, \ldots, K\}$ (i.e., if the agent lands in some state in $\mathcal{G}_{\theta,k}$, then the kth subgoal is considered "completed"). In addition, we define an artificial reward function $g_{\theta,k}(s, s')$ that, when activated, provides a sequence of rewards that leads the agent toward subgoal k. Concretely, we use potential-based reward shaping from [147] to achieve this. Let $\Phi_{\theta,k}$ be a *potential function* over the full state space $\mathcal{S}$ such that target states in $\mathcal{G}_{\theta,k}$ have the highest potential. Then, let

$$g_{\theta,k}(s, s') = \gamma \Phi_{\theta,k}(s') - \Phi_{\theta,k}(s). \tag{3.2}$$

The definition of $g_{\theta,k}(s, s')$ in (3.2) can be interpreted as the difference in potential between states s' and s (with discount γ). This potential difference motivates the agent to move towards the target states (high potential) of kth subgoal. Thus, a parameterization of a set of K subgoals is fully described by

$$\left(\{\mathcal{G}_{\theta,k}\}_{k=1}^{K}, \{g_{\theta,j}\}_{k=1}^{K}\right),$$

the locations and associated reward shaping functions.

Example 3.2.1 (Key and Door Environment)**.** *Let us consider a distribution of maze MDPs with states* $\{(i,j)\}_{1\leq i,j\leq 10}$ *and a sparse reward in the upper left corner at* $(0,n)$*. In addition, suppose that the agent needs to pick up a key in order to receive the reward at* $(0,n)$*, but the location of the key is uncertain but likely to be in the* right half of the room*. The environment illustrated in Figure 15 can be considered to be one possible realization from this distribution of mazes. Now, let us consider a subgoal design with* $K = 3$ *subgoals. The simple parameterization* $\theta = (i_1, j_1, i_2, j_2, i_3, j_3)$*, with*

$$\mathcal{G}_{\theta,k} = \{(i_k, j_k)\} \quad \textit{and} \quad \Phi_{\theta,k}(s) = e^{-\|s-(i_k,j_k)\|^2}$$

specifies that for $k \in \{1,2,3\}$*, the kth subgoal is located at a single state* (i_k, j_k) *and the artificial reward potential is a Gaussian centered at* (i_k, j_k)*. Using Figure 15 as a visual reference, one can imagine that the subgoal design* $\theta = (1,2,8,4,2,8)$ *would be useful in guiding the agent toward the vicinity of the key on the right side of the room and then toward the vicinity of the goal. Once the agent is in the correct vicinity, the underlying RL algorithm can discover the precise locations of the key and goal in the particular environment realization more quickly.*

For the types of navigation tasks that we are concerned with in this chapter, the dimension of the subgoal parameterization θ need not scale with the dimension of the state s, which would pose a potential scalability issue. Instead, one general rule-of-thumb to keep in mind is that for a dynamic subgoal exploration strategy to be effective in navigation tasks, the dimension of θ only needs to scale with the number components of s that pertain to the *spatial positioning of the agent.* The next example provides an illustration.

Example 3.2.2 (Mountain Car Environment, with $\dim(\theta) < \dim(s)$)**.** *Consider the well-known Mountain Car problem, a continuous control task where an underpowered car, operating in a one-dimensional space, must make its way up a steep mountain [169, Example 10.1]. The state is two-dimensional, $s = (x, \dot{x})$, where $x \in [-1.2, 0.5]$ is the position of the agent while $\dot{x} \in [-0.07, 0.07]$ is its velocity. A possible subgoal design with $K = 2$ is $\theta = (i_1, i_2)$, with*

$$\mathcal{G}_{\theta,k} = \{(i_k, \dot{x}) \,|\, \dot{x} \in [-0.07, 0.07]\} \quad \textit{and} \quad \Phi_{\theta,k}(s) = e^{-(x-i_k)^2}$$

for each k. In other words, the agent reaches a subgoal target state if its position is i_k, for any value of its velocity. Also, the artificial reward only depends on the spatial position x rather than the full state $(x, \dot{x})$. In Section 3.4, we give numerical results for exactly this example.

One could imagine that the concept illustrated in Example 3.2.2 also applies to more complex robotics environments with a high-dimensional state, but where the components related to the spatial positioning is relatively small, meaning that the subgoal parameterization (and the resulting BO problem) is often of much lower dimension than that of the state itself.

3.2.3 Subgoal-Augmented MDPs $\mathcal{M}_{\xi,\theta}$

Now that we have described how a particular subgoal design is parameterized, the remaining question is how these are integrated in a useful way into the original, sparse-reward MDP described in Section 3.2.1. We propose the notion of a *subgoal-augmented*, auxiliary MDP, where the K subgoals are sequentially "activated." This way, we encode subgoal ordering into the exploration strategy, meaning that the agent only moves on to the next subgoal after finishing the current one. Without ordering, rewards from multiple subgoals can inhibit the agent's progress.

Let $\mathcal{M}_{\xi,\theta}$ denote an auxiliary, subgoal-augmented MDP based on an original MDP $\mathcal{M}_\xi$, except that it is includes rewards and transitions associated with the dynamic subgoal exploration strategy θ. We introduce an auxiliary state $i \in \mathcal{I} := \{0, 1, \ldots, K\}$, where i represents the number of subgoals reached by the agent so far. Initially, we have $i_0 = 0$. The state of

the $\mathcal{M}_{\xi,\theta}$ is $(s, i) \in \mathcal{S} \times \mathcal{I}$ and the transition for the auxiliary state is $i' = i + \mathbf{1}_{\{s' \in \mathcal{G}_{\theta,i+1}\}}$, where we take $\mathcal{G}_{\theta,K+1} = \varnothing$. This means the auxiliary state i is updated to $i + 1$ whenever s' reaches the next subgoal. Let the intrinsic reward of the agent be:

$$G_\theta(s_t, i_t, s_{t+1}) = \sum_{k=1}^{K} \mathbf{1}_{\{k=i_t\}} \cdot g_{\theta,k+1}(s_t, s_{t+1}),$$

where the indicator function encodes the logic that if i_t subgoals have been completed so far, then the current target is subgoal $i_t + 1$ and only the rewards leading to subgoal $j + 1$ should be active. The new reward function consists of both extrinsic and intrinsic rewards:

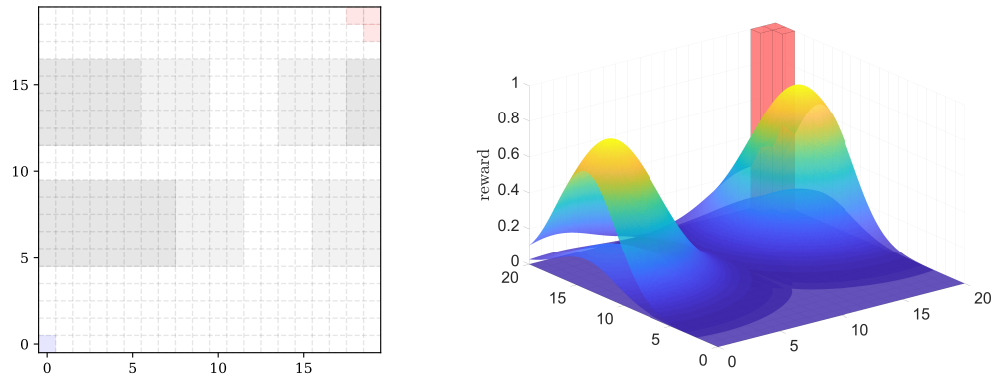
$$\hat{R}_{\xi,\theta}(s, i, a, s') = R_\theta(s, a, s') + G_\theta(s, i, s').$$

The value function for the new MDP $\mathcal{M}_{\xi,\theta}$ is written

$$\hat{V}^{\hat{\pi}}_{\xi,\theta}(s, i) = \mathbf{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \hat{R}_{\xi,\theta}(s_t, i_t, a_t, s_{t+1}) \,|\, \hat{\pi}, s, i\right], \tag{3.3}$$

where $\hat{\pi}(\cdot|s, i)$ is now a policy defined on the new state space $\mathcal{S} \times \mathcal{I}$.

Figure 17 gives an example when all the pieces are considered. Figure 17a shows the original MDP environment $\mathcal{M}_\xi$, where the dark gray cells are walls and the light gray represent uncertainty in the size of the "doors." Figure 17b shows the possible rewards the agent can encounter in the augmented MDP $\mathcal{M}_{\xi,\theta}$, for a random selection of subgoals θ. The sparse reward is represented by the red bar in the corner and the first subgoal is the one that is farther from the goal. Both subgoals are singletons and the potential functions are radial basis functions centered at the subgoal locations, similar to the parameterization described in Example 3.2.1. Note that this randomly selected set of subgoals θ is not a good exploration strategy for the environment in Figure 17a (as it leads the agent toward a wall), motivating the need for optimizing their locations, as we discuss in the next section.



(a) 20×20 Gridworld Environment

(b) Goal and Subgoal Rewards

Figure 17: An example that visualizes an environment and a random dynamic subgoal exploration strategy along with the rewards of the associated subgoal-augmented MDP.

3.2.4 Optimizing the Exploration Strategy

Selecting the best subgoal design θ depends on the agent's underlying learning algorithm, which could in principle be any RL algorithm that uses intermediate rewards for learning. However, for the time being, we do not place any restrictions on the RL algorithm and refer to it as `RL-ALGO`. In the numerical results of Section 3.4, our agent learns via Q-learning [170]. Let us use the notation $\texttt{RL-ALGO}[\tau, \mathcal{M}]$ to refer to the policy learned by `RL-ALGO` on MDP $\mathcal{M}$ after τ training interactions. We remind the reader that the subgoal-based exploration strategy is fixed *before* the test environment is revealed, so that the sequence of events in the test phase is as follows:

1. A subgoal design θ for exploration is selected.
2. The agent is placed in a new environment ξ.
3. The agent uses the subgoal-augmented MDP $\mathcal{M}_{\xi,\theta}$ and an RL algorithm with a budget of $\tau_{\max}$ interactions to learn a policy $\texttt{RL-ALGO}\big[\tau_{\max}, \mathcal{M}_{\xi,\theta}\big]$.
4. The agent's policy is evaluated in the original MDP with extrinsic reward function R_ξ.

Our goal is to find an exploration strategy $\theta \in \Theta$ such that a policy trained using θ behaves well in the original MDP situation in expectation:

$$\max_{\theta \in \Theta} \mathbf{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_\xi(s_t, a_t, s_{t+1}) \,\middle|\, \hat{\pi}_{\xi,\theta}^{\tau_{\max}}, s_0, i_0\right] \text{ where } \hat{\pi}_{\xi,\theta}^{\tau_{\max}} = \texttt{RL-ALGO}\big[\tau_{\max}, \mathcal{M}_{\xi,\theta}\big], \tag{3.4}$$

where (s_0, i_0) is the initial augmented state. Note that the dependence on the subgoal-augmented MDP $\mathcal{M}_{\xi,\theta}$ is through the policy learned from it, $\hat{\pi}_{\xi,\theta}^{\tau_{\max}}$. The expectation in (3.4) is taken over the random choice of a test environment ξ, the stochastic dynamics within $\mathcal{M}_\xi$, and the stochasticity of the learning algorithm itself. Moreover, it is convenient to explicitly define the following:

$$u(\theta, \tau) = \mathbf{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_\xi(s_t, a_t, s_{t+1}) \,\middle|\, \hat{\pi}_{\xi,\theta}^{\tau}, s_0, i_0\right] \text{ where } \hat{\pi}_{\xi,\theta}^{\tau} = \texttt{RL-ALGO}\big[\tau, \mathcal{M}_{\xi,\theta}\big].$$

Note that the objective function in (3.4) is $u(\theta, \tau_{\max})$, but the notation $u(\theta, \tau)$ will be useful in Section 3.3, where we discuss using fewer than $\tau_{\max}$ interactions to learn about $u(\theta, \tau_{\max})$ as a way of reducing cost.

3.2.5 Iterative Training and Additional Cost-Reduction Levers

In our setting, we observe the performance of exploration strategies and the resulting policies in a sequence of training environment realizations $\xi^1, \xi^2, \ldots, \xi^N$ drawn from the MDP distribution Ξ. By default, each complete evaluation of the objective function in (3.4) $u(\theta, \tau_{\max})$ for a fixed θ requires running `RL-ALGO` for $\tau_{\max}$ interactions. Since each interaction in the training environments is expensive (e.g., in robotics applications, this could involve time, labor, and equipment), we want to consider ways to reduce the number of training interactions. To do so, we propose two additional levers:

1. **Maximum episode length.** For each training environment ξ^n, we allow the specification of a maximum episode length τ^n chosen from a finite set $\mathcal{T}$. In the next section, we describe our probabilistic model of the RL training curve, which allows observations of short episodes to be informative about the final performance. This also can reduce the risk of spending too many interactions with an unpromising exploration strategy.

2. **Multiple replications.** We can reduce the variance of performance observations by averaging over the observed cumulative reward over q^n i.i.d. replications, for a total of $\tau^n q^n$ interactions in training environment ξ^{n+1}. We suppose that q^n is chosen from a finite set $\mathcal{Q}$. The idea here is that even with the same number of total interactions, a lower variance observation of a "preliminary" result could be more informative than a higher variance observation of the "full" result.

To summarize, three decisions are made at the beginning of each training opportunity n: (1) a choice of subgoal design θ^n, (2) the maximum episode length τ^n, and (3) the number q^n of replications of the training episode to use for this particular θ^n. For each of the q^n replications, we obtain a policy

$$\hat{\pi}^{\tau^n}_{\xi^{n+1},\theta^n} = \texttt{RL-ALGO}\big[\tau^n, \mathcal{M}_{\xi^{n+1},\theta^n}\big],$$

before observing a estimate of its performance. After the q^n training replications are complete, we compute the average performance over the q^n replications. Written more succinctly, our observation in episode n takes the form

$$y^{n+1}(\theta^n, \tau^n, q^n) = u(\theta^n, \tau^n) + \varepsilon^{n+1}_{\text{env}} + \varepsilon^{n+1}_{\text{rep}}(q^n),$$

where $\varepsilon^{n+1}_{\text{env}}$ represents the deviation from the $u(\theta^n, \tau^n)$ due to the random environment ξ^{n+1}, while the observation noise $\varepsilon^{n+1}_{\text{rep}}(q^n)$ represents the noise that can be reduced via multiple replications, i.e., the noise in $\hat{\pi}^{\tau^n}_{\xi^{n+1},\theta^n}$ due to a sample run of `RL-ALGO`. Thus, $\varepsilon^{n+1}_{\text{rep}}(q^n)$ depends on the number of replications q^n. Naturally, a larger number of replications implies a smaller observation noise. Note that the observations $\{y^n\}$ are i.i.d., since a new MDP is sampled in each iteration. The total training cost incurred is cumulative number of interactions: $\sum_{n=0}^{N-1} \tau^n q^n$.

After training opportunities $0, 1, \ldots, N-1$, we reach the *test phase* and commit to a final subgoal design θ^N_{rec}. This design is evaluated on the test MDP $\xi^{N+1} \sim \Xi$ with an agent that has a full budget of $\tau_{\max}$ interactions.

3.3 Bayesian Optimization for Cost-Efficient Exploration

The proposed BO approach for learning a dynamic subgoal exploration strategy consists of two components: a tailored probabilistic model and an acquisition function for selecting the next subgoal design, the maximum episode length, and the number of replications to run. Although shorter episodes and smaller number of replications are more cost-efficient, they also decrease the chance of reaching the goal and produce higher observation noise; the acquisition function must carefully trade off these downsides with the cost of interactions. We call this the *Bayesian Exploratory Subgoal Design* (`BESD`) acquisition function.

3.3.1 Surrogate Model

In order to enable the ability to dynamically select the maximum episode length of training, as described in Section 3.2.5, our approach uses a GP surrogate model over $u(\theta, \tau)$, rather than $u(\theta, \tau_{\max})$. In other words, our model is a function of both θ and τ rather than just θ, enabling it to capture the performance of a policy trained with subgoals θ, for a variety of episode lengths. Assume that $\Theta \subseteq \mathbb{R}^m$. We place a GP prior f on the latent function u with mean function $\mu : \Theta \times \mathcal{T} \to \mathbb{R}$ and covariance function $k : (\Theta \times \mathcal{T}) \times (\Theta \times \mathcal{T}) \to \mathbb{R}_+$. More precisely, we set μ to the mean of an initial set of samples and use a multidimensional product kernel, based on the kernel used in [112], to capture the structure of the RL learning curve:

$$k\big((\theta, \tau), (\theta', \tau')\big) = k_\theta(\theta, \theta')\, k_\tau(\tau, \tau'), \tag{3.5}$$

where the first kernel k_θ is the (5/2)-Matérn kernel and k_τ is a polynomial kernel $k_\tau(\tau, \tau') = \phi(\tau)^\intercal \Sigma_\phi \, \phi(\tau')$ with $\phi(\tau) = (1, \tau)^\intercal$ and hyperparameters Σ_ϕ. Note that the covariance under k is large only if the covariance is large under both k_θ and k_τ. We make the modeling assumption that $\varepsilon_{\text{env}}^{n+1}$ and $\varepsilon_{\text{rep}}^{n+1}(q^n)$ are independent, zero mean, and normally distributed with variances σ_{env}^2 and $\sigma_{\text{rep}}^2/q^n$, respectively. Although the assumption of normality is commonplace in BO for tractability of the posterior [105], other noise distributions can be used through an appropriate likelihood function (but this is often difficult to know a priori). This allows us to take advantage of standard GP machinery to analytically compute the posterior

on f conditioned on the history after n steps. This posterior is another GP, whose mean and kernel functions are denoted $\mu^n(\theta, \tau)$ and $k^n((\theta, \tau), (\theta', \tau'))$; the exact expressions can be found in, e.g., [171].

We remind the reader that the dimensionality of the GP surrogate model is $\dim(\Theta) + 1$, i.e., the dimension of the subgoal parameterization, along with an additional dimension for τ. As illustrated in Example 3.2.2, it will often be the case for navigation domains that the dimension of the subgoal parameterization is smaller than that of the state space of the underlying RL problem (due to the relatively small number of spatial components of the state). Therefore, dynamic subgoal exploration strategies can be tractably modeled and optimized for broad classes of navigation problems, even with vanilla GPs. Of course, when the need arises to optimize for high dimensional subgoal parameterizations, one may opt for scalable extensions of the model and optimization formulation (e.g., [172, 173, 174, 175] along with many more recent papers). We leave extensions in this direction to future work and focus on a more standard setting.

3.3.2 Acquisition Function

As described above, our framework proceeds in iterations, selecting one set of subgoals θ^n along with τ^n and q^n, to be evaluated in each training environment. We now propose the acquisition function for making these evaluation decisions. An overview of the BO setup is given in Algorithm 2.

Suppose the training budget is used up after training iterations $0, 1, \ldots, N-1$. Then, the optimal risk-neutral decision is to use subgoals on the test MDP ξ^{N+1} that have maximum expected performance under the posterior. The expected score of this choice is μ_*^n where

$$\mu_*^n := \max_\theta \mu^n(\theta, \tau_{\max}), \tag{3.6}$$

where $\mu^n(\theta, \tau_{\max}) = \mathbf{E}_n[f(\theta, \tau_{\max})]$. Here $\mathbf{E}_n$ is the conditional expectation with respect to the history after the first n observations: $(\theta^0, \tau^0, q^0, y^1, \ldots, \theta^{n-1}, \tau^{n-1}, q^{n-1}, y^n)$. Note that although we are allowed to use fewer than $\tau_{\max}$ interactions in training environments to reduce cost, the agent uses its full budget for the test MDP ξ^{N+1}.

Algorithm 2: Bayesian Exploratory Subgoal Design

Input: Set $n = 0$. Estimate hyperparameters of the GP prior f using initial samples.

Output: A subgoal recommendation θ^N_{rec} that maximizes $\mu^N(\theta, \tau_{\max})$.

for $n = 1, 2, \ldots, N$ **do**

  Compute next decision (θ^n, τ^n, q^n) according to the acquisition function (3.7).

  Train in environment ξ^{n+1} augmented with θ^n ($\mathcal{M}_{\xi^{n+1},\theta^n}$) using levers (τ^n, q^n).

  Observe $y^{n+1}(\theta^n, \tau^n)$ and update posterior on f.

end

We take the knowledge gradient, one-step lookahead approach [114, 176], i.e., we imagine for each training MDP that it is the last opportunity before the test MDP and act optimally. Full lookahead approaches require solving an intractable dynamic programming problem; however, we show that nonetheless, the one-step approach is asymptotically optimal in Theorem 3.3.1 and Theorem 3.3.2. If we evaluate (θ, τ, q), i.e., the subgoals θ for τ steps and q replications, then the expected gain in performance in the test MDP of the recommended exploration strategy after the evaluation, based on (3.6), with respect to the current best is

$$\nu^n(\theta, \tau, q) = \mathbf{E}_n\left[\mu_*^{n+1} \mid \theta^n = \theta, \tau^n = \tau, q^n = q\right] - \mu_*^n.$$

Therefore, the one-step optimal strategy is to choose the next subgoals θ^n, maximum episode length τ^n, and number of replications q^n so that ν^n is maximized.

However, this strategy would generally allocate a maximum number of steps $\tau_{\max}$ and replications $q_{\max}$ for the evaluation of the next subgoal design, as observing $\tau_{\max}$ during training is most informative of the test conditions, and repeating for $q_{\max}$ replications reduces the noise maximally. In other words, this strategy does not consider the cost of training. Hence, we propose an acquisition function that maximizes the *gain in performance per effort*, resulting in a policy that selects

$$(\theta^n, \tau^n, q^n) \in \arg\max_{\theta,\tau,q} \frac{\nu^n(\theta, \tau, q)}{q\tau}. \tag{3.7}$$

By construction BESD is Bayes optimal (per unit cost) for the last step, in expectation, as stated formally in the following proposition.

Proposition 3.3.1. *The acquisition function of (3.7) achieves an optimal expected information gain per unit cost for the case of $N = 1$.*

The optimization problem (3.7) is challenging when the domain Θ is continuous, so we take the approach of replacing it with a discrete domain $\bar{\Theta} \subseteq \Theta$ (for example, this could be selected by a Latin Hypercube design). This approach has been applied successfully in other knowledge gradient style acquisition functions [115, 116, 111]. Unlike previous work however, we provide a novel theoretical guarantee on the asymptotic suboptimality of a discretized optimization domain; see Theorem 3.3.2 in the next section.

3.3.3 Theoretical Analysis

We now provide our main theoretical results on the asymptotic optimality of BESD. Detailed proofs can be found in Appendix B.1. For convenience in this section, we suppose $\mu(\theta, \tau) = 0$ for all (θ, τ), and that the kernel $k(\cdot, \cdot)$ has continuous partial derivatives up to the fourth order. Recall that $\theta^N_{\text{rec}} \in \bar{\Theta}$ is the final recommendation made in iteration N:

$$\theta^N_{\text{rec}} \in \arg\max_{\theta \in \bar{\Theta}} \mu^N(\theta, \tau_{\max}).$$

Our first theorem is concerned with the finite, discretized optimization domain $\bar{\Theta}$.

Theorem 3.3.1. *The acquisition function described in (3.7) has the property of asymptotic optimality with respect to $\bar{\Theta}$, i.e.,*

$$\lim_{N \to \infty} f(\theta^N_{rec}, \tau_{max}) = \max_{\theta \in \bar{\Theta}} f(\theta, \tau_{max}),$$

almost surely. That is, the recommended design θ^N_{rec} becomes optimal as $N \to \infty$.

If the optimization domain $\bar{\Theta} = \Theta$, then Theorem 3.3.1 suffices. Unfortunately, for many applications, the subgoal parameterizations will naturally be continuous. Next, we provide an additive bound on the difference between the solution of `BESD` in $\bar{\Theta}$ and the unknown optimum in Θ, as the number of iterations N tends to infinity.

We use a probabilistic Lipschitz constant of a GP from [177] to quantify the performance with respect to the full, continuous subgoal parameter space. We make use of the fact that the derivative $df(\theta, \tau_{\max})/d\theta_i$ is another GP with covariance

$$k^{\partial i}(\theta, \theta') = \frac{\partial^2}{\partial \theta_i \partial \theta_i'} \, k\big((\theta, \tau_{\max}), (\theta', \tau_{\max})\big),$$

for all $i = 1, 2, \ldots, m$ [178, Section 9.4]. See also [179] and [180] for other uses of this property. For each $i = 1, 2, \ldots, m$, define the constant

$$L_\delta^i = k_{\max}^\partial \sqrt{2 \log\Big(\frac{2m}{\delta}\Big)} + 12\sqrt{6m} \max\left\{ k_{\max}^\partial, \sqrt{L_k^{\partial i} \max_{\theta, \theta' \in \Theta} \operatorname{dist}(\theta, \theta')} \right\}, \tag{3.8}$$

where $L_k^{\partial i}$ be a Lipschitz constant of the kernel $k^{\partial i}$ and $k_{\max}^\partial = \max_{\theta \in \Theta} \sqrt{k^{\partial i}(\theta, \theta)}$.

Theorem 3.3.2. *The acquisition function of (3.7) has bounded asymptotic suboptimality with respect to the original domain Θ in the sense that with probability at least $1 - \delta$, it holds that*

$$\lim_{N \to \infty} f(\theta_{\text{rec}}^N, \tau_{\max}) \geq \max_{\theta \in \Theta} f(\theta, \tau_{\max}) - d \cdot \|L_\delta\|$$

where $d = \max_{\theta \in \Theta} \min_{\theta' \in \bar{\Theta}} \operatorname{dist}(\theta, \theta')$ is a measure on the "coarseness" of the discretization and L_δ is the vector $(L_\delta^1, L_\delta^2, \ldots, L_\delta^m)$, with each L_δ^i defined as in (3.8).

3.4 Numerical Experiments

We now provide numerical experiments to demonstrate the cost-effectiveness of the `BESD` framework. `BESD` was implemented in `Python` 2.7 using the `MOE` package [181] and will be open-sourced upon acceptance of the manuscript.

In the experiments that follow, we use the proposed `BESD` approach to optimize dynamic subgoal exploration strategies consisting of two or three subgoals. `BESD` is given a few choices

for the episode length τ and number of replications q (values reported for each benchmark below). Each replication of the `BESD` is given an initial set of 10 observations for each value of τ (these initial observations incur interaction costs just like future observations). The potential function at state s with the jth subgoal activated is $\Phi_j(s) = w_1 \exp[-0.5(s - j)^2/w_2]$, where the "height" is $w_1 = 0.2$ and "width" is $w_2 = 10$.

3.4.1 Baseline Algorithms

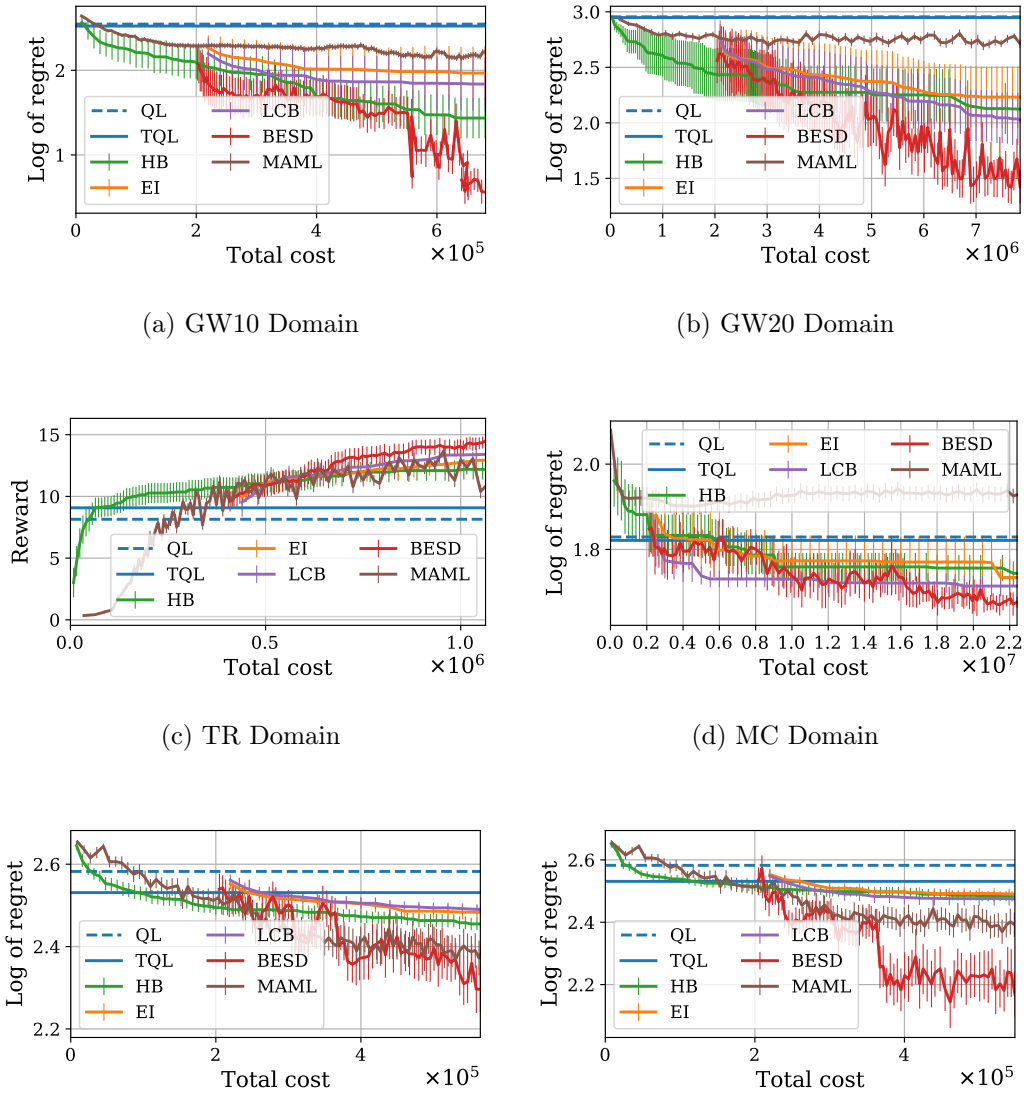
Given the somewhat unique positioning of the `BESD` framework, it is important for us to compare against from several streams of literature. Due to our strong focus on cost-efficiency, non-gradient-based approaches are from the BO literature are particularly relevant. Two of the most common approaches are *expected improvement* [182, 183] and *lower/upper confidence bound*, often called "GP-UCB" when used with a Gaussian process model [184, 185]. "Lower" when minimizing the objective and "upper" when maximizing. Expected improvement (`EI`) allocates one sample in each round, selecting a point that maximizes the expected improvement beyond currently sampled points:

$$\texttt{EI}(\theta) = \mathbf{E}_n\big[\big(\min\{y^1, \ldots, y^n\} - y^{n+1}(\theta, \tau_{\max})\big)^+\big].$$

In each iteration, we evaluate the `EI` selection using $\tau_{\max}$ iterations. Lower confidence bound (`LCB`) controls the exploration-exploitation trade-off using a "bonus term" proportional to the standard deviation at each point:

$$\texttt{LCB}(\theta) = \mu^n(\theta, \tau_{\max}) - \kappa\sqrt{k^n((\theta, \tau_{\max}), (\theta, \tau_{\max}))}.$$

The parameter κ is set to 2. Both `EI` and `LCB` are implemented in `Python` 2.7 using the `GPyOpt` package [186].



(a) GW10 Domain

(b) GW20 Domain

(c) TR Domain

(d) MC Domain

(e) KEY2 Domain

(f) KEY3 Domain

Figure 18: Performance as a function of the total training costs.

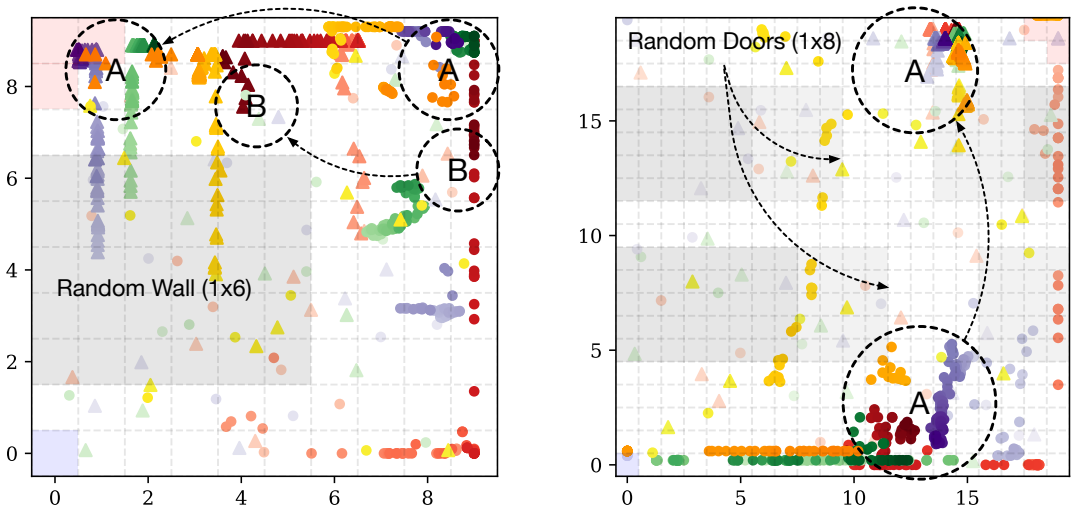
Note: The curves are averaged over 50 replications of the meta-optimization problem and the error bars indicate ± 2 standard errors of the mean. For each replication, to assess the performance at a particular point in the process, we take its latest recommendation and test it by averaging its performance on a random sample of 200 test MDPs (i.e., ξ^N) The x-axis is the cumulative cost including the initial sampling cost. The y-axis is typically the log regret, where regret is defined as the number of additional steps needed to reach the goal when compared to the optimal policy. The exception is in (c), where the y-axis is discounted reward (since in TR, the performance is measured by both reward and steps). Note that the curves associated with the BO methods, `BESD`, `LCB`, `EI`, start later due to the use of a set of initial points for initializing the GP model.

We also compare against two "default RL" baselines, that do not incorporate an aspect of tuning the exploration strategy. The first baseline is the Q-learning algorithm (`QL`) [40] with no subgoals or reward shaping: that is, we directly run `QL` on environment ξ^N for $\tau_{\max}$ interactions. The second one is a heuristic based on the approximate Q-values learned by `QL`, which we call "transfer" Q-learning (`TQL`): for the test instance, we initialize the Q-values using the previously stored results from a randomly chosen training environment. This heuristic is inspired by the idea of *policy reuse* proposed in [161] for transferring learned strategies to new tasks.

An alternative to applying BO or bandit algorithms to hyperparameter optimization is the idea of *adaptive configuration evaluation*, which focuses on improving the throughput of configuration evaluation by quickly eliminating ones that are not promising. From this line of thinking, the Hyperband algorithm (`HB`) of [110] stands out as a popular and representative approach. It treats hyperparameter optimization as a pure-exploration infinite-armed bandit problem; it uses sophisticated techniques for adaptive resource allocation and early-stopping to concentrate its learning efforts on promising designs. Setting $\eta = 3$ (the default value) and $R = 81$, `HB` consists of $\lfloor \log_\eta R \rfloor$ rounds. The first round starts with R samples of subgoal designs θ from a Latin hypercube sample. Following `HB`'s motivation of early-stopping unpromising designs, each θ is evaluated for $\tau_{\min}$ steps. The best $1/\eta$-fraction designs are kept for the next round. In round i, Hyperband samples R/η^{i-1} subgoal designs to evaluate for $\tau_{\min}\,\eta^{i-1}$ steps.

Finally, we chose a representative algorithm from the multi-task RL literature, the well-known *model-agnostic meta-learning* algorithm (`MAML`) proposed by [126]. `MAML` consists of two optimization loops. The outer loop provides an initialization to the inner loop, and the inner loop solves new tasks with a small number of examples. As with `QL` and `TQL`, `MAML` does not make use of subgoal exploration and uses neural network representations as in the original paper [126]. It utilizes stochastic gradient descent in both loops to optimize the parameters. `MAML` is implemented in `Python` 3.6 using [187]. To keep consistent with other baselines, the batch size of the outer loop is 1.

3.4.2 Windy Gridworlds with Walls



(a) GW10 Domain

(b) GW20 Domain

Figure 19: Recommendation paths for GW10 and GW20.

Note: The blue and red shaded regions denote the starting points and goals, respectively. Dark and light gray regions possible locations of walls and doors, respectively. Each plot displays four realizations of the "recommendation paths" of `BESD`. Each color corresponds to one sample realization, and the color becomes darker as n increases, with the lightest points being the initial samples. The circles and triangles represent the first and second subgoals, respectively, of the exploration strategy. The 'A' and 'B' labels point out two example sets of subgoals displaying notable behaviors.

The first set of environments (GW10) is a distribution over 10×10 gridworlds, where the goal is to reach the upper left square that is shaded red in Figure 19a to collect a reward of one. The agent starts from the lower-left grid square shaded in blue and may in each step choose an action from the action space consisting of the four compass directions. Each gridworld is partitioned by a wall into two rooms. The wall, randomly located in one of the middle five rows in the gridworld, has a door located on four grid squares on its right. The agent will stay in the current location when it hits the wall.

There is a small amount of "wind" or noise in the transition: the agent moves in a random direction with a probability that is itself uniformly distributed between 0 and 0.02 (thus, a

particular environment instance drawn from the distribution has a random wall location and wind probability).

We use $\mathcal{T} = \{200, 600, 1000\}$ for the possible values of τ and $\mathcal{Q} = \{5, 20\}$ for the possible values of q. We parameterize the exploration strategy using two subgoals, whose locations are optimized. Subgoal locations are limited to the continuous subset of $\mathbb{R}^2$ which contains the grid, i.e., $\Theta = ([0, 10] \times [0, 10])^2$ for GW10. Figure 18a shows the performance of the recommendations by `BESD` as a function of total expended cost compared to the baselines. We will discuss the baseline comparisons in more detail in Section 3.4.7.

3.4.2.1 Recommendation Paths for GW10 In order to visualize the qualitative behavior of `BESD`, we show in Figure 19a the evolution of the recommended subgoals over time (iterations), a concept that we refer to as a *recommendation path*. The plot displays four recommendation path realizations of `BESD` using distinct colors. Within each color, the lightest points are the initial samples while the darker points represent recommendations for larger n. Also within each color, the circles represent the first subgoal of the exploration strategy, while the triangles represent the second subgoal. We point out two types of exploration behaviors discovered by `BESD` in Figure 19a:

- Behavior 'A': The pairs of regions labeled 'A' are the final recommendations of the orange, green, and purple sample paths. The strategy leads the agent toward the upper right corner, in order to bypass the wall, and then after that, directly towards the goal.
- Behavior 'B': The final recommendation of the red sample path is labeled by 'B.' Note that in behavior 'A', a direct path to the first subgoal (upper right corner) is blocked by the random wall for some realizations of the environment. Behavior 'B' might be interpreted as a slight remedy of this situation by targeting a lower region of the right edge, creating a more direct path around the wall.

3.4.3 Larger, Three-Room Windy Gridworlds

The second domain (GW20) is a distribution of larger 20×20 gridworlds with three rooms separated by two walls. As shown in Figure 19b, the walls are randomly located in

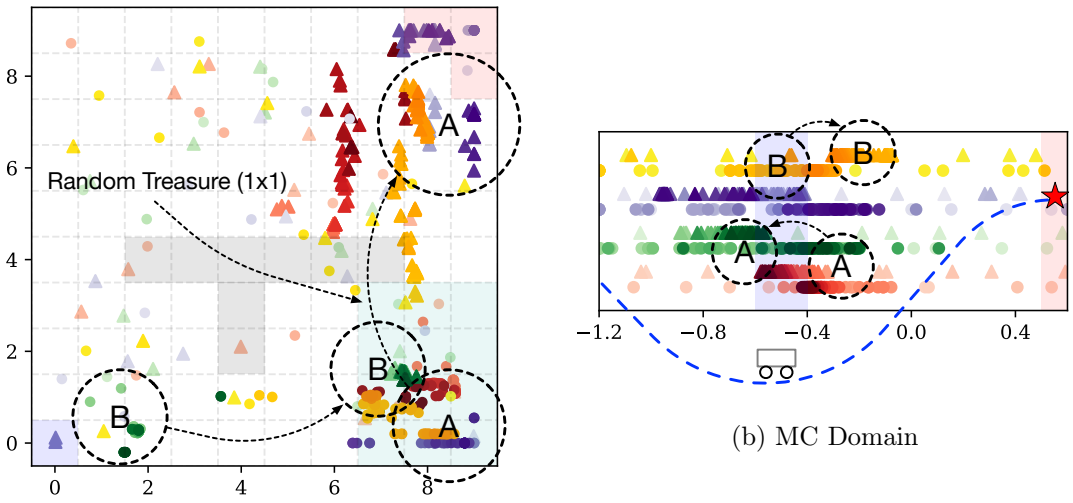
the middle rows (dark gray). A door of size 8 is randomly located somewhere within the wall, shaded in light gray. The starting location is the blue square in the lower left and the goal is displayed in red in the upper right. As in GW10, we optimize the locations of a two-subgoal exploration strategy, with $\Theta = ([0, 20] \times [0, 20])^2$. The noise due to wind is the same as in GW10. In this experiment, we consider the case of only allowing `BESD` to select the maximum episode length from $\mathcal{T} = \{4000, 7000, 10000\}$, while keeping $q = 20$ fixed. The performance comparison with the baseline algorithms is shown in Figure 18b.

3.4.3.1 Recommendation Paths for GW20 Recommendation paths are shown in Figure 19b. Unlike the case of GW10, all four of the realizations converge to roughly the same exploration strategy, labeled by 'A.' Focusing on the lighter red and orange circles, we can notice a trend of the first subgoal initially being placed (naively) near the goal, but as learning progresses, they move downward toward the entrance of the first door. The second subgoal converges toward the exit of the second door, moving the agent near the goal.

Regarding the placement of the first subgoal near the goal and inducing a direct path, it is worth pointing out this strategy might work for *some* environments (i.e., those where the first door is at its leftmost position and the second door is at its rightmost position). However, `BESD` learns that in order to perform well *across the distribution* of environments, the strategy of first moving rightward is better.

3.4.4 Treasure-in-Room

The third domain (TR) is a distribution of 10×10 gridworlds with a "treasure" hidden in a small room; see Figure 20a. The light green area shows the possible positions of the treasure. The agent gets a reward of 10 upon entering the square with treasure, and a reward of 10 upon reaching the goal. The cumulative reward, however, is zero if the agent does not find the goal within the interaction budget. The discount factor is set to $\gamma = 0.98$ to encourage policies that collect the reward earlier. We set $\mathcal{T} = \{400, 1200, 2000\}$ and $\mathcal{Q} = \{5, 20\}$. See Figure 18c for the comparison to baselines.



(a) TR Domain

(b) MC Domain

Figure 20: Recommendation paths for TR and MC.

Note: The first panel, Figure 20a, largely follows the same design as Figures 19a and 19b, except that the green squares represent possible location of the treasure. In the second panel, Figure 20b, since the location of the mountain-car is one-dimensional, we visualize the four recommendation paths by spacing them vertically to avoid crowding. The initial location of the car is colored in blue, while the goal is in red, corresponding to the overlay of the mountain.

3.4.4.1 Recommendation Paths for TR The recommendation paths for TR are in Figure 20a. We observe that two strategies were discovered by `BESD` across these four realizations:

- Behavior 'A': This appears to be the ideal behavior and was discovered in the orange, purple, and red sample paths: first lead the agent to the treasure and then toward the goal through the upper right. It is also notable that the first subgoal is located at the *bottom* of the room, meaning that wherever the treasure turns out to be, the agent can pick it up without backtracking.
- Behavior 'B': The green sample path's final recommendation coincides with the (apparently suboptimal) exploration strategy denoted by 'B' simply leads the agent to the treasure, but does not provide any guidance toward the goal. We highlight that this is an

instance where BESD's learning is not yet complete, evidenced by the fact that behavior 'B' is often recommended in *earlier iterations of the orange sample path.* In that case however, BESD eventually discovers behavior 'A' in later iterations.

3.4.5 The Mountain Car Problem (MC)

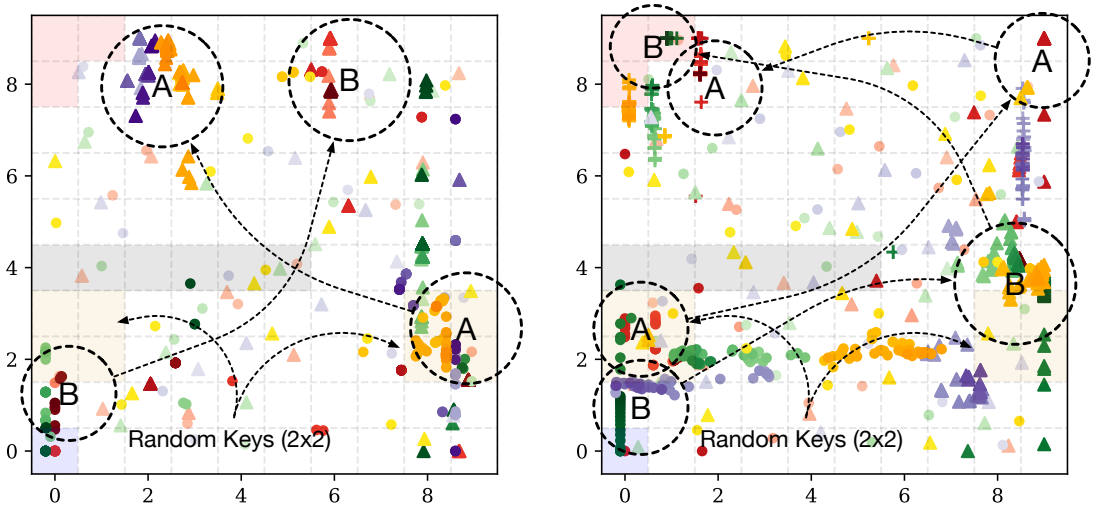
The mountain car (MC) domain, as we introduced in Example 3.2.2, is a commonly used RL benchmark environment that tests an agent's ability to explore, as it is required to go in the opposite direction of the goal in order to reach the top of the mountain; see, e.g., [169, Example 10.1]. For this experiment, we created a distribution of environments Ξ by randomizing the starting location of the agent, which is chosen uniformly from $[-0.6, -0.4]$. Here, we set $\mathcal{T} = \{4000, 7000, 10000\}$ and $\mathcal{Q} = \{10, 50\}$. Figure 18d compares the performance of BESD to baseline approaches.

3.4.5.1 Recommendation Paths for MC The subgoal-pairs discovered by BESD are shown in Figure 20b; they tend to be on opposite sides of the agent's starting location, thereby creating back-and-forth movement needed to generate momentum and move up the mountain. It is worth noting that the symmetric behaviors of going from left to right (Behavior 'B' in Figure 20b, for the orange sample path) and going from right to left (Behavior 'A', exhibited by the green, red, and purple sample paths) can both be found in the results of BESD.

3.4.6 Key-Door with Highly Varying Key Locations (KEY2 and KEY3)

In our last experiment, we test for the situation where the distribution of environments Ξ contains environments that might vary dramatically from one another. We also consider how the exploration behavior changes when we add an additional subgoal to the strategy.

In domains KEY2 (with two subgoals) and KEY3 (with three subgoals), we consider a 10×10 gridworld with one wall, where a "key" needs to be picked up before opening a closed door at the upper-right corner of the grid. The location of the key, however, is highly varying and is either near the left wall or the right wall. The environment is visualized in



(a) KEY2 Domain

(b) KEY3 Domain

Figure 21: Recommendation paths for KEY2 and KEY3.

Note: The blue and red shaded regions denote the starting points and goals, respectively. Dark and light gray regions possible locations of walls and doors, respectively. Each plot displays four realizations of the "recommendation paths" of `BESD`. Each color corresponds to one sample realization, and the color becomes darker as n increases, with the lightest points being the initial samples. The circles, triangles, and crosses represent the first, second, and third subgoals, respectively. The 'A' and 'B' labels point out two example sets of subgoals displaying notable behaviors.

Figures 21a and 21b. We set $\mathcal{T} = \{400, 700, 1000\}$ and $\mathcal{Q} = \{5, 20\}$. Figures 18e and 18f gives the baseline comparison.

3.4.6.1 Recommendation Paths for KEY2/KEY3 It is important that the agent moves in the *vicinity of both keys* in order for it to perform well across the distribution of environments. We now discuss how this is achieved by the two- and three-subgoal exploration strategies, using the annotations in Figures 21a and 21b.

- Behavior 'A' in KEY2 (Figure 21a): In the first exploration behavior discovered by `BESD`, the agent is first directed to the right-most key location and then towards the door. This is behavior is reasonable in the sense that the agent's initial location is near the left-

most key location; hence, the naive exploration (e.g., ϵ-greedy) "built-in" to `RL-ALGO` would likely find the key (if it is there) without additional subgoal rewards.

- Behavior 'B' in KEY2 (Figure 21a): The second exploration behavior that we highlight takes a similar approach. This strategy incentivizes the agent to first check the left-most key location (going upwards from the initial location). Interestingly, the second subgoal is neither the other key location nor the goal: instead, the agent is directed toward the upper edge of the environment, slightly right of center. Upon examination, one might conclude that this path *compromises* between the second key location and the goal. On its way from the first to second subgoal, the agent enters the vicinity of the second key location and also ends up not far from the goal. In other words, the exploration strategy puts the agent in a position such that `RL-ALGO`'s naive exploration is more likely to be successful.
- Behavior 'A' in KEY3 (Figure 21b): With an additional subgoal to work with, `BESD` is able to find more flexible exploration strategies. For behavior 'A', we see that the first subgoal is near the left-most key location, the second subgoal indirectly leads the agent toward the vicinity of the right-most key location, and the third subgoal is at the goal. The placement of the second subgoal is reminiscent of behavior 'B' of KEY2, but this time, a third subgoal allows `BESD` to directly lead the agent towards the goal
- Behavior 'B' in KEY3 (Figure 21b): This strategy is more intuitive (indeed, more replications converge to behavior 'B' than behavior 'A') and leads the agent to check each of the possible key locations (the closer one first) and then sends the agent directly toward the goal.

3.4.7 Takeaways from Baseline Comparisons in Figure 18

We now offer some observations and takeaways from the performance plots of Figures 18a-18f, where `BESD` is compared to a variety of baseline approaches.

1. **Comparison to MAML.** `BESD` significantly outperforms `MAML` in all domains except KEY2, where performance is similar. We also see that in some domains (e.g., GW10, GW20, MC), `MAML` is unable to make much progress at all within the interaction budgets

that we considered. This is not surprising as MAML relies on an abundance of data for gradient-based updates *during training* (despite the fact that it is designed for sample-efficient adaptation in the test environment). In addition, we note that since MAML's default hyperparameters worked even more poorly – we tuned the learning rate and batch sizes to improve performance. Note that BESD's "hyperparameters" (subgoal parameterization) are relatively more intuitive, especially given some domain knowledge. Importantly, there are no learning rates.

2. **Comparison to Hyperband.** HB is reasonably competitive against BESD on two of the easier domains, GW10 and TR. In particular, we notice that HB tends to have good performance early on (as it is able to use early stopping to quickly eliminate inferior subgoal strategies). However, as the interaction budget grows, we see that in most domains, BESD is eventually able to make better use of its evaluations, likely explained by BESD's use of a tailored surrogate model.
3. **Comparison to other BO methods.** The popular BO methods EI and LCB tend to perform similarly to each other in all domains. Compared to BESD, however, they are less cost-efficient. Since all three approaches make use of underlying GP surrogate models, but EI and LCB are constrained in always using $q_{\max}\tau_{\max}$ interactions, this is evidence that being able to reduce the episode lengths and the number of replications is valuable.
4. **Impact of more subgoals.** Lastly, we point out that Figures 18e and 18f show that although a two-subgoal exploration strategy achieves better results than the baselines, a three-subgoal strategy performs even better. This demonstrates the benefit of expanding the dimension of the parameterization in certain environments. Choosing the number of subgoals to use in a particular set of environments is not an exact science; in general, a higher dimensional subgoal parameterization makes the BO meta-optimization problem more challenging and each acquisition function optimization is also more time-consuming. We recommend the following guidelines: (1) Consider the total interaction budget across all training iterations. A rule-of-thumb is that a d-dimensional subgoal parameterization should have $2d - 1$ random initial points. The interaction cost of the initial points should be less than $1/3$ of the total budget in order to give BESD adequate time to make progress (if the cost of initial points is too high, then one might want to reduce d). (2)

Optimizing the acquisition function becomes more time consuming as d increases, so d should be small enough such that (3.7) can be computed in one's allotted per-iteration time budget for acquisition function optimization.

3.4.8 How Much Does a Dynamic Subgoal Exploration Strategy Help RL?

In Section 3.4, Figures 19, 20, and 21 gave visual intuition about the types of exploration behaviors that were discovered by `BESD`. In this section, we show how the final dynamic subgoal strategy θ^N_{rec} recommended by `BESD` helps throughout the course of RL. Let $\pi^\tau_\xi = \texttt{RL-ALGO}\big[\tau, \mathcal{M}_\xi\big]$ be the policy learned using `RL-ALGO` on the original, sparse-reward environment (i.e., no subgoal exploration strategy). For a given problem domain, we define the agent's performance ratio after τ interactions to be:

$$\text{performance ratio}(\tau) = u(\theta^N_{\text{rec}}, \tau) \;/\; \mathbf{E}\big[V^{\pi^\tau_\xi}(s_0)\big].$$

In other words, this is the ratio of the performance of the policy learned by `RL-ALGO` when using the dynamic subgoal exploration strategy θ^N_{rec} in the subgoal-augmented MDP to the performance of the policy learned by `RL-ALGO` in the original environment. On GW10, GW20, MC, KEY2, and KEY3, a smaller performance ratio indicates a more effectiveness of the exploration strategy. Since for TR we measure performance using rewards instead of costs, a larger performance ratio is desired. Table 6 displays the performance ratios as a function of the number of interactions used in the test environment. We can see that an optimized exploration strategy corresponds to dramatic improvements, ranging from roughly 3x in the worst cases (MC, KEY2, and KEY3) to nearly 20x in the best cases (GW10, GW20, and TR).

3.5 Conclusion and Future Work

The problem of finding exploration strategies for a distribution of environments with a strong focus on *cost-awareness during training* has not been adequately studied in the literature. This can be a deterrent to applying RL in real-world settings where interactions with

Table 6: Performance ratios as a function of interactions in the test environment.

τ	GW10	GW20	TR	MC	KEY2	KEY3
m	0.458	0.779	0.436	0.980	1.456	1.025
$2m$	0.218	0.492	2.823	1.048	0.736	0.940
$3m$	0.086	0.234	2.823	0.949	1.277	0.698
$4m$	0.080	0.224	0.917	0.896	0.704	0.788
$5m$	0.070	0.108	6.723	0.987	1.355	0.531
$6m$	0.086	0.088	8.939	0.878	0.856	0.503
$7m$	0.080	0.068	9.908	1.077	0.920	0.623
$8m$	0.087	0.075	10.216	0.877	0.883	0.532
$9m$	0.069	0.059	23.2936	0.512	0.232	0.566
$10m$	0.069	0.058	18.011	0.354	0.332	0.361

Note: GW10, GW20 TR, MC, KEY2, and KEY3 are evaluated every $m = 100, 1000, 200, 1000, 500, 500$ steps respectively.

the environment are limited and expensive (and where cheap simulators are not available). This chapter proposes a solution based on Bayesian optimization; in a cost-aware manner, our approach finds subgoals with an intrinsic shaped reward that aids the agent in scenarios with sparse and delayed rewards, thereby reducing the number of interactions needed to obtain a good solution. An experimental evaluation demonstrates that `BESD` achieves considerably better solutions than a comprehensive field of baseline methods on a variety of benchmark problems. Moreover, an examination of its "recommendation paths" shows that `BESD` discovers solutions that induce interesting exploration strategies. There are several exciting avenues for extensions of this chapter:

- **Richer BO formulations.** Extensions to the BO formulation could be made in various ways. For example, one interesting direction is to allow the acquisition function to determine the *number of subgoals* as an additional lever. Based on a few informal observations, such a formulation is likely only interesting in settings where *more subgoals incur additional experimentation cost.* We ran a small number of informal experiments where we allowed BO to select the number of subgoals, but found that `BESD` almost

immediately gravitates to the largest number of subgoals (as subgoals come at no cost). Since in the applications that we have in mind, subgoal cost was not a primary concern, we did not pursue this direction as it did not bring any particularly strong insights for the standard case. Alternatively, the acquisition function itself could be extended with additional features, such as encouraging successive subgoal evaluations to be nearby previous ones (i.e., to reduce setup cost) or the ability to reason about (known) symmetries in the domain. Such advanced features might be enabled by dynamic programming formulations *of the BO problem itself*, which can be tackled using multi-step lookahead BO [188, 189, 190, 191]. Other possiblities include the ability to handle expensive-to-evaluate constraints [192, 193, 194] or total cost budgets [195, 196].

- **Case study in an application domain.** Our experiments gave proof-of-concept results on benchmarks where the RL training itself did not use prohibitive amounts of computation, in order for us to stay within a reasonable computational budget. This is because statistically distinguishable results for baseline algorithms require many replications of the meta-optimization problem (i.e., the BO routines), each of which require many iterations of RL training. One immediate area of future work is to "productionize" the dynamic subgoal exploration strategies in a real-world application involving a navigation task.
- **The task-aware setting.** Finally, our problem formulation does not include "labels" for environments, as we were motivated by the case where the randomness of the test environments is due to the decision maker's uncertainty in its parameters. The situation often studied in the multi-task RL setting, however, often comes with task identifiers, where the agent knows that it is operating in particular task. An extension to this setting might be useful for certain applications, where exploration strategies that are good for one task (e.g., biking through an environment) are also useful for other tasks (e.g., walking through the same environment).

4.0 Frozen-State Approximate Value Iteration for Fast-Slow Markov Decision Processes

We consider sequential decision problems, modeled as Markov decision processes (MDPs), that are endowed with a new type of "fast-slow" structure: a *fast-slow* MDP has a state that can be divided into two parts, a *slow state* and a *fast state*. At each time step, the transition of the slow state results in a change that is relatively small compared to that of the fast state. An alternative view from the perspective of the reward function (rather than the transition function) is that the reward is less sensitive to changes in the slow state. Our models allow for the slow state process to be either (1) fully exogenous to the system, where actions taken do not affect its dynamics, (One interpretation of the exogenous slow state setting is that of a standard MDP whose the rewards and transitions are modulated by an external process (the slow state) similar to the formulation given in [197].) or (2) endogenous to the system, where a separability assumption holds on the action space. The latter will be presented as an extension to the fully exogenous case, which we focus on initially to build intuition and theoretical foundations. Fast-slow structure is common in important real-world problems where sequential decisions need to be made at high frequencies, yet information that varies at a slower timescale also influences the optimal policy.

1. **Service allocation in multi-class queues.** The first example is a dynamic service allocation problem for a multi-class queue [198, 199], with the addition of stochastic holding costs (i.e., the cost of leaving items in the queue) that vary slowly and can be viewed as the slow state [200]. One prominent motivation is the case of energy-aware job scheduling in data centers, where variations of electricity prices over time can influence the holding cost [201, 202, 203]. Another motivation is the case of content moderation queues for online platforms, where the costs of delayed review of harmful content can depend on a variety of factors, including content popularity and the state of current events [200, 204].
2. **Energy demand response.** We can also apply the fast-slow framework in sequential decision problems from the realm of demand response in the electricity market. Specifi-

cally, we consider the problem faced an energy aggregator who observes a day-ahead price and then simultaneously bids a reduction quantity into the demand response market and sets the compensation for demand reduction from consumers [205, 206, 207, 208, 209]. Essentially, the aggregator hopes to generate profit from the difference between the contracted price for delivery of demand reduction to the market and the price that offers customers for that reduction. However, the aggregator has to consider the demand elasticity of its customers, along with the stochasticity of day-ahead prices and real-time prices (which determine the "penalty" for mistmatch between the promised and realized quantities of demand reduction). Since real-time prices are much more volatile compared to the day-ahead prices, it is reasonable to view day-ahead prices as the slow state.

3. **Multi-product joint procurement and pricing.** A third example is the multi-product joint procurement and pricing problem with price-dependent demand [210, 211, 212]. In some situations, the demand can differ quite dramatically between products (i.e., the demand for some of the products is high, and the demand for the other products is relatively low). For low-demand products, inventory review can occur less frequently to simplify the decision making process.

Attempts to optimally solve a model that incorporates the full state space along with the true decision-making frequency often encounter computational issues, due to the challenge of solving an MDP with a large state space over a large number of periods. Therefore, both practitioners and academic researchers may elect to design simplified decision models that *ignore* the effect of the slow state on components of their problems. In other words, these states might be "left out" of the state variable by, e.g., fixing them to constant values. Although such an approach results in policies that can be obtained in a computationally tractable manner, we see in Section 4.6 that they can incur significant regret compared to the optimal policy.

4.0.1 Main Contributions

In this paper, we propose somewhat of a *compromise* between the full MDP and ignoring slow states, by designing a framework around periodically "freezing" and "releasing"

slow states, and re-using policies that are computed based on a frozen slow-state model. Specifically, we make the following contributions:

1. We first consider a fast-slow MDP with exogenous slow state and provide an (exact) reformulation into an MDP with hierarchical structure. The "upper level" is a slow-timescale infinite horizon MDP and the "lower level" is a fast-timescale finite horizon MDP with T periods. One period of the upper-level problem is composed of a complete lower-level problem. We propose a *frozen-state approximation* to the reformulated MDP, where the slow state is frozen in the lower-level problem, while each period in the upper-level problem "releases" the slow state. Computational benefits arise in several ways: (1) re-use of the lower-level policy (which is computed once) when applying value iteration in the upper level, (2) frozen states simplify the dynamics of the lower-level MDP (dramatically fewer successor states), and (3) the lower-level MDP thus becomes separable into independent MDPs, opening the door to speedups via parallel computation. Solving the frozen-state approximation gives a policy that switches between the one action from upper-level policy and $T - 1$ actions from the lower-level policy. We give a theoretical analysis that upper bounds the expected regret from applying this policy compared to the optimal policy.
2. We then discuss an additional step of approximation that further reduces computational requirements called the *nominal-state approximation*, which takes advantage of a factored reward function assumption and approximates the lower-level MDP using a fixed set of "nominal" slow states. The consequence is that instead of solving the lower-level MDP for all slow states, this approximation allows us to solve it only for the set of nominal slow states, which are then used to approximate the lower-level value for other slow states. We also provide an upper bound on the expected regret of the policy obtained from the nominal state approximation.
3. We then extend our model to consider a fast-slow MDP with an *endogenous* slow state, under an action space separability assumption, where we suppose that each action can be broken up into two parts: one part that affects the slow state's transition dynamics and another part that affects the fast state's transition dynamics. In this version of the model, the action for the slow state is taken in the upper-level policy, while the lower-level

policy focuses on the action for the fast state. In the theoretical analysis, we account for an extra error related to the actions for the slow state.

4. Next, we show how the fast-slow framework can also be exploited in an approximate dynamic programming (ADP) setting [37, 39]. Specifically, we design an approximate value iteration (AVI) algorithm that mimics the nominal state approximation, in that we perform AVI in both the lower and upper levels. However, to allow for generalization across the state space, we make use of a feature-based linear approximation that combines estimated values of a set of pre-selected states to form approximations of the value function at other states, based on the technique introduced in [213]. We provide an analysis of the expected regret for the policy that is greedy with respect to the fixed point of the upper-level AVI.
5. Lastly, we perform a systematic empirical study on four problem settings (machine maintenance, dynamic service allocation, energy demand response, and multi-product joint procurement and pricing). We show that with either exogenous or endogenous slow states, the proposed frozen-state approximation algorithms, especially the nominal state approximation, converge faster than standard (approximate) value iteration, a baseline that ignores slow states, Q-learning, and DQN (deep Q-networks). We also give qualitative evidence that policies generated by the frozen-state approach have structural features resembling those of the optimal policy.

4.1 Related Work

In this section, we provide a brief review of related literature. First, there exists a stream of literature focused on sequential decision making problems with *exact* hierarchical, multi-timescale structure. [214] study multi-timescale MDPs, which are composed of M different decisions that are made on M different discrete timescales. The authors consider the impact of upper-level states and actions on the transition of the lower levels, an idea is also present in our fast-slow framework. Multi-timescale MDPs have often been applied in supply chain problems, including production planning in semiconductor fabrication [215, 216], hydropower

portfolio management [217], and strategic network growth for reverse supply chains [218]. [209] transfer the finite-horizon MMDP into a linear programming problem, exploit the threshold structure of the optimal solution, and propose a row-generation-based algorithm to solve the problem. [219] consider "piecewise stationary" MDPs, where the transition and reward functions are "renewed" every $N+1$ periods, motivated by problems where routine decisions are periodically interrupted by higher-level decisions. For the case of large renewal periods, they propose a policy called the "initially stationary policy" which uses a fixed decision rule for some number of initial periods in each renewal cycle. Our fast-slow model focuses on a novel fast-slow structure present in many MDPs and does not assume any natural/exact hierarchical structure. Instead, we focus on how a particular type of (frozen-state) hierarchical structure can be used as an *approximation* to the true MDP (and derive error bounds). However, we note that many MDPs with natural two-timescale structure can also fit into our framework, and in that sense, our model can be roughly viewed as more general.

Our proposed frozen-state algorithms are also related to literature on *hierarchical reinforcement learning*, which are methods that artificially decompose a complex problem into smaller sub-problems [220]. Approaches include the options framework [128], the hierarchies of abstract machines (HAMs) approach [221], and MAXQ value function decomposition [222].

The options framework is the most closely related to this paper. A Markov option (macro-action, or temporally extended action) is composed of a policy, a termination condition, and an initiation set [128, 223]. One of the biggest challenges is to automatically construct options that can effectively speed up reinforcement learning. A large portion among this research is based on *subgoals*, states that might be beneficial to reach [224, 134, 225, 226]. The subgoals are identified by utilizing the learned model of the environment [227, 136, 137, 138], or through trajectories without learning a model [134, 133]. The options (and subgoals) framework is largely motivated by robotics and navigation-related tasks, while we are particularly interested in solving problems that arise in operations research and operations management domains. The problems that we study do not decompose naturally into "subgoals" — leading us to identify and focus on the fast-slow structure, which does indeed arise naturally.

In addition, our proposed methods avoid the challenge of constructing the set of options by using a fixed length $T-1$ for the lower-level MDP (and we are able to provide guidance on selecting the value of T that introduces an acceptable amount of error). The idea of freezing states to reduce computational cost is also unique to our approach.

4.2 Fast-Slow MDPs with Exogenous Slow States

In this section, we introduce the *base model*, the original MDP to be solved and formally introduce the notion of a *fast-slow* MDP with exogenous slow states. We then provide a hierarchical reformulation of the base model using fixed-horizon policies, and show the equivalence (in optimal value) between the two models.

4.2.1 Base Model

Consider a discrete-time MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$, where $\mathcal{S}$ is the finite state space, $\mathcal{A}$ is the finite action space, $\mathcal{W}$ is the space of possible realizations of an exogenous, independent and identically distributed (i.i.d.) noise process $\{w_t\}$, $f : \mathcal{S} \times \mathcal{A} \times \mathcal{W} \to \mathcal{S}$ is the transition function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function with bound $r_{\max}$, and $\gamma \in [0, 1)$ is the discount factor for future rewards [228]. The objective is

$$U^*(s) = \max_{\{\nu_t\}} \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t \, r\big(s_t, \nu_t(s_t)\big) \,\middle|\, s_0 = s\right], \tag{4.1}$$

where states transition according to $s_{t+1} = f(s_t, a_t, w_{t+1})$ and we optimize over sequences of policies $\nu_t : \mathcal{S} \to \mathcal{A}$, which are deterministic mappings from states to actions. The expectation is taken over exogenous noise process $\{w_t\}_{t=1}^{\infty}$.

Assumption 4.2.1 (Separability and the Fast-Slow Property)**.** *Suppose the following hold:*

(i) The state space $\mathcal{S}$ is separable and can be written as $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$. We call $\mathcal{X}$ the "slow state space" and $\mathcal{Y}$ the "fast state space."

(ii) *Let $s_t = (x_t, y_t) \in \mathcal{S}$, where $x_t \in \mathcal{X}$ is the slow state and $y_t \in \mathcal{Y}$ the fast state, $a_t \in \mathcal{A}$, and $w_{t+1} \in \mathcal{W}$. The transition dynamics $s_{t+1} = f(s_t, a_t, w_{t+1}) \in \mathcal{S}$ is separable in the following sense:*

$$x_{t+1} = f_{\mathcal{X}}(x_t, w_{t+1}) \in \mathcal{X} \quad \textit{and} \quad y_{t+1} = f_{\mathcal{Y}}(x_t, y_t, a_t, w_{t+1}) \in \mathcal{Y},$$

for some $f_{\mathcal{X}} : \mathcal{X} \times \mathcal{W} \to \mathcal{X}$ and $f_{\mathcal{Y}} : \mathcal{S} \times \mathcal{A} \times \mathcal{W} \to \mathcal{Y}$.

(iii) *For any state $(x, y) \in \mathcal{S}$, action $a \in \mathcal{A}$, and exogenous noise $w \in \mathcal{W}$, suppose the one-step transitions of x and y satisfy:*

$$\left\|y - f_{\mathcal{Y}}(x, y, a, w)\right\|_2 \leq d_y \quad \textit{and} \quad \left\|x - f_{\mathcal{X}}(x, w)\right\|_2 \leq \alpha d_y,$$

for some $d_y < \infty$ and $\alpha \in [0, 1]$.

Note that from Assumption 4.2.1(ii), the slow state transition is exogenous in that it does not depend on the action a_t, nor does it depend on the fast state y_t (the transition of y_t, however, is allowed to depend on all available information, including x_t). We relax the assumption of exogenous slow states in Section 4.4, but it is instructive to begin with this case.

We assume throughout that $\mathcal{S}$, $\mathcal{A}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{S} \times \mathcal{A}$ are equipped with the Euclidean metric, which is naturally the case for many applications. However, as long as the relevant spaces are metric spaces, the framework continues to hold. We choose Euclidean metrics as they are natural for our applications. For a given reward function r on $\mathcal{S} \times \mathcal{A}$, define

$$L_r = \max_{(s,a) \neq (s',a')} \frac{|r(s, a) - r(s', a')|}{\|(s, a) - (s', a')\|_2}, \tag{4.2}$$

which is the maximum growth rate of the reward function and serves as a Lipschitz constant for r. Similarly, we define

$$L_f = \max_{(s,a) \neq (s',a'), w} \frac{\|f(s, a, w) - f(s', a', w)\|_2}{\|(s, a) - (s', a')\|_2}, \tag{4.3}$$

which serves as a Lipschitz constant for the transition function f.

Definition 4.2.1 (Fast-Slow MDP)**.** *An MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$ is called a (α, d_y, L_r, L_f)-fast-slow MDP if Assumption 4.2.1 is satisfied, the reward function r satisfies (4.2), and the transition function f satisifes (4.3). Throughout the paper, we will often denote a fast-slow MDP with the expanded notation $\langle \mathcal{X} \times \mathcal{Y}, \mathcal{A}, \mathcal{W}, f_{\mathcal{X}}, f_{\mathcal{Y}}, r, \gamma \rangle$.*

Given any state $s = (x, y)$, noise w, and policy ν, we use the notation $f^{\nu}(s, w) = f(s, \nu(s), w)$, $f^{\nu}_{\mathcal{Y}}(x, y, w) = f_{\mathcal{Y}}\big(x, y, \nu(x, y), w\big)$, and $r(x, y, \nu) = r(x, y, \nu(x, y))$ throughout the paper. The value of a stationary policy (It is well-known that there exists an optimal policy to (4.1) that is both stationary and deterministic. See [228].) ν at state (x, y) is the expected cumulative reward starting from state (x, y) following policy ν, i.e.,

$$U^{\nu}(x, y) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r\big(x_t, y_t, \nu\big) \,\middle|\, (x_0, y_0) = (x, y)\right] = r\big(x, y, \nu\big) + \gamma\, \mathbf{E}\big[U^{\nu}(x', y')\big], \tag{4.4}$$

where $x' = f_{\mathcal{X}}(x, w)$, $x_{t+1} = f_{\mathcal{X}}(x_t, w_t)$, $y = f^{\nu}_{\mathcal{Y}}(x, y, w)$, $y_{t+1} = f^{\nu}_{\mathcal{Y}}(x_t, y_t, w_t)$ for all t. The optimal value function at state $U^*(x, y)$, as defined in (4.1), satisfies the Bellman equation, i.e.,

$$U^*(x, y) = \max_a \; r(x, y, a) + \gamma\, \mathbf{E}\big[U^*(x', y')\big]. \tag{4.5}$$

A policy that is greedy with respect to the optimal value function, i.e.,

$$\nu^*(x, y) = \arg\max_a \; r(x, y, a) + \gamma\, \mathbf{E}\big[U^*(x', y')\big].$$

is an optimal policy, and the optimal value U^* and the value of the optimal policy U^{ν^*} are the same.

4.2.2 Hierarchical Reformulation using Fixed-Horizon Policies

In this section, we derive an exact hierarchical reformulation with the original timescale broken up into groups of T periods each. The reformulation holds for a general MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$, but the concepts that we introduce in this section will serve as the basis for developing our frozen-state computational approach for fast-slow MDPs.

Denote $(\mu, \boldsymbol{\pi})$ a *T-horizon policy*, which is a sequence of T policies $(\mu, \pi_1, \ldots, \pi_{T-1})$, $\mu : \mathcal{S} \to \mathcal{A}$, $\pi_t : \mathcal{S} \to \mathcal{A}$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{T-1})$. Following $(\mu, \boldsymbol{\pi})$ means that we take the

first action according to μ and then next $T-1$ actions according to $\boldsymbol{\pi}$. Given any state s_0, the T-period reward function (of the base model) associated with $(\mu, \boldsymbol{\pi})$ is written as:

$$R(s_0, \mu(s_0), \boldsymbol{\pi}) = r(s_0, \mu) + \sum_{t=1}^{T-1} \gamma^t \, r(s_t, \pi_t), \tag{4.6}$$

where $s_1 = f^{\mu}(s_0, w_1)$ and $s_{t+1} = f^{\pi_t}(s_t, w_{t+1})$ for $t > 0$.

A *T-periodic policy* $(\mu, \boldsymbol{\pi})$ refers to the infinite sequence that repeatedly applies the T-horizon policy $(\mu, \boldsymbol{\pi})$, i.e., $(\mu, \boldsymbol{\pi}, \mu, \boldsymbol{\pi}, \ldots)$. Note that the T-periodic policy $(\mu, \boldsymbol{\pi})$ can be implemented in the infinite horizon problem defined in (4.1). The value of the T-periodic policy $(\mu, \boldsymbol{\pi})$ at state s_0 is

$$\bar{U}^{\mu}(s_0, \boldsymbol{\pi}) = \mathbf{E}\left[\sum_{k=0}^{\infty} \gamma^{kT} R(s_k, \mu(s_k), \boldsymbol{\pi}) \,\middle|\, s_0 = s\right] = \mathbf{E}\big[R(s_0, \mu(s_0), \boldsymbol{\pi}) + \gamma^T \, \bar{U}^{\mu}(s_T, \boldsymbol{\pi})\big], \tag{4.7}$$

where, again, $s_1 = f^{\mu}(s_0, w_1)$ and $s_{t+1} = f^{\pi_t}(s_t, w_{t+1})$ for $t > 0$ within each cycle of T periods. The optimal value function satisfies the following Bellman equation:

$$\bar{U}^*(s_0) = \max_{(\mu, \boldsymbol{\pi})} \mathbf{E}\big[R(s_0, \mu(s_0), \boldsymbol{\pi}) + \gamma^T \bar{U}^*(s_T)\big], \tag{4.8}$$

where the "action" now involves selecting the $\boldsymbol{\pi}$ as well. Denote $(\mu^*, \boldsymbol{\pi}^*)$ an optimal T-periodic policy, which solves (4.8). In Proposition 4.2.1, we prove that the base model (4.5) and the hierarchical reformulation (4.8) are equivalent in a certain sense.

Proposition 4.2.1. *Given an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$, the following hold:*

(i) The optimal value of the base model (4.5) *is equal to the optimal value of the hierarchical reformulation* (4.8)*, i.e., $U^* = \bar{U}^*$.*

(ii) An optimal stationary policy ν^* with respect to the base model (4.5) *is also an optimal policy for the hierarchical reformulation* (4.8)*, i.e., $\bar{U}^* = \bar{U}^{\nu^*}$.*

The proof is in Appendix C.1.2. Part (i) of Proposition 4.2.1 is most relevant to our situation in the sense that the optimal T-periodic policy $(\mu^*, \boldsymbol{\pi}^*)$ is no better than the stationary optimal policy ν^*. Therefore, solving the hierarchical reformulation (4.8) allows us to achieve the same value as the ν^*, the optimal policy to the original base model (4.5).

4.3 The Frozen-State Approximation

We now propose our frozen-state approximation, where slow states are frozen for T periods at a time. This allows us to construct an auxiliary problem that proceeds at a timescale that is a factor of T *slower* than the MDP of the base model (equivalently, the discount factor becomes γ^T instead of γ), naturally leading to ADP algorithms (see Sections 4.3.3 and 4.3.7) with computational benefits. The number of periods T to freeze the state is a parameter to the approach.

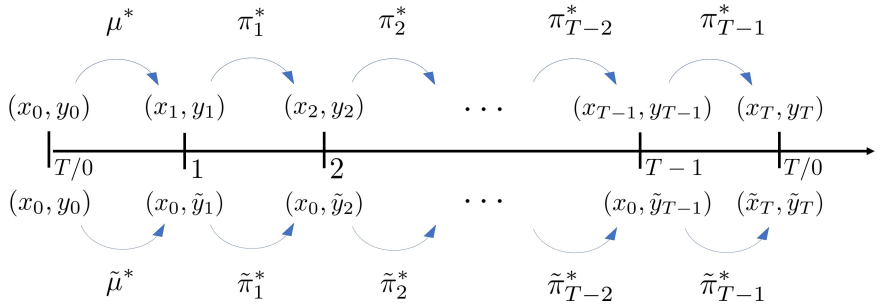


Figure 22: Illustration of the base model versus the frozen-state approximation

Consider a fast-slow MDP $\langle \mathcal{X} \times \mathcal{Y}, \mathcal{A}, \mathcal{W}, f_{\mathcal{X}}, f_{\mathcal{Y}}, r, \gamma \rangle$. The frozen-state process proceeds in rounds of length T and within each round, we make decisions on periods $0, 1, \ldots T-1$. The slow state x remains frozen from period 0 to period $T-1$, but during the transition from $T-1$ to T, the slow state x is updated. Period T is also labeled period 0 for the next round. See the illustration of this process in Figure 22.

Remark 4.3.1. *It is important to note that the freezing of states only occurs "within the algorithm" as a step toward more efficient computation of policies. Our resulting policies are then implemented in the underlying base model MDP, which proceeds naturally according to its true dynamics. Our theoretical and empirical results always attempt to answer the question: how well does a policy that is computed by pretending certain states are frozen perform in the true model?*

4.3.1 The Lower-Level MDP (Frozen Slow States)

We view the problem from period 1 to period T as the "lower level" of the frozen-state approximation. This corresponds to the periods relevant to $\boldsymbol{\pi}$ from $(\mu, \boldsymbol{\pi})$ in the hierarchical reformulation (4.8), whose structure the frozen-state approximation mimics. To form the lower-level problem of the frozen-state approximation, we consider this $T-1$ period problem in isolation:

$$J_1^*(x,y) = \max_{\tilde{\boldsymbol{\pi}}} \mathbf{E}\left[\sum_{t=1}^{T-1} \gamma^{t-1} r(x_0, y_t, \tilde{\pi}_t) \,\middle|\, (x_0, y_0) = (x,y)\right], \tag{4.9}$$

where x remains frozen, $y_{t+1} = f_{\mathcal{Y}}(x, y_t, w_{t+1})$, and $\tilde{\boldsymbol{\pi}} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_{T-1})$. The problem (4.9) can be solved using finite-horizon dynamic programming: accordingly, let the terminal $J_T^* \equiv 0$ and for $t = 1, 2, \ldots, T-1$, let

$$J_t^*(x,y) = \max_a r(x,y,a) + \gamma \mathbf{E}\big[J_{t+1}^*(x,y')\big], \tag{4.10}$$

where $y' = f_{\mathcal{Y}}(x,y,a,w)$. Also, let $\tilde{\boldsymbol{\pi}}^* = (\tilde{\pi}_1^*, \ldots, \tilde{\pi}_{T-1}^*)$ be the finite-horizon policy that is greedy with respect to J_t^*:

$$\tilde{\pi}_t^*(x,y) = \arg\max_a r(x,y,a) + \gamma \mathbf{E}\big[J_{t+1}^*(x,y')\big].$$

It may not immediately be clear why freezing slow states is desired. There are two main computational benefits to solving (4.10) instead of an analogous version of (4.10) *without* freezing x:

- In algorithms like value iteration [228], each update requires computing expectations over successor states. In most practical implementations of MDP solvers, the transition probability matrix is stored. In the case where $|\mathcal{W}| >> |\mathcal{S}|$,(This is often the case for tabular settings, because multiple random outcomes of w can lead to the same state s.) the number of successor states impacts the number of operations for each VI iteration. When x is frozen, the number of successor states is much smaller since we only have successor fast states: in other words, we only need to compute $\mathbf{E}\big[J_{t+1}^*(x,y')\big]$ instead of $\mathbf{E}\big[J_{t+1}^*(x',y')\big]$. Even in the case that the expectation is approximated via sampling, the former requires sampling from a lower-dimensional successor state distribution.

- Second, (4.10) can effectively be viewed as $|\mathcal{X}|$ independent MDPs, one for each $x \in \mathcal{X}$, allowing for the possibility of computing the policy with additional parallelism. In the nominal-state approximation discussed Section 4.3.7, we analyze the error of an approach that solves only a small number out of the $|\mathcal{X}|$ independent MDPs.

4.3.2 The Upper-Level MDP (True State Dynamics)

Let us now consider the upper-level problem of the frozen-state approximation, which is an infinite horizon problem with groups of T periods aggregated. Denote the stationary upper-level policy by $\mu : \mathcal{S} \to \mathcal{A}$, which is the policy that we are attempting to optimize in the upper-level problem. The upper-level problem takes two "inputs" related to the lower-level problem: (1) J_1, an approximation of the optimal lower-level value J_1^*, (2) $\boldsymbol{\pi}$, a lower-level finite-horizon policy. Fixing these inputs, the value at state $s_0 = (x_0, y_0)$ by executing policy μ is

$$V^{\mu}(s_0, J_1, \boldsymbol{\pi}) = \mathbf{E}\big[\tilde{R}(s_0, \mu(s_0), J_1) + \gamma^T V^{\mu}(s_T, J_1, \boldsymbol{\pi})\big]. \tag{4.11}$$

where s_T is the state reached according to the true system dynamics by following $(\mu, \boldsymbol{\pi})$, starting from s_0 and

$$\tilde{R}(s_0, a, J_1) = r(s_0, a) + \gamma\, J_1\big(f_{\mathcal{X}}(x_0, w), f_{\mathcal{Y}}(x_0, y_0, a, w)\big)$$

is a one-step approximation to the T-period reward function R, defined in (4.6). The optimal value (for this approximation) at state s_0 can be written as

$$V^*(s_0, J_1, \boldsymbol{\pi}) = \max_a \mathbf{E}\big[\tilde{R}(s_0, a, J_1) + \gamma^T V^*(s_T, J_1, \boldsymbol{\pi})\big]. \tag{4.12}$$

Recall that the optimal lower-level policy (that solves the frozen-state model) is denoted $\tilde{\boldsymbol{\pi}}^*$ and its optimal value is J_1^*. Let $\tilde{\mu}^*$ be the optimal upper-level policy corresponding to these inputs, i.e., the policy greedy with respect to $V^*(s_0, J_1^*, \tilde{\boldsymbol{\pi}}^*)$.

Thus, $(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*)$ is the resulting T-periodic policy from the overall frozen-state hierarchical approximation; we refer to it as the *T-periodic frozen-state policy.*

Algorithm 3: Frozen-State Value Iteration (FSVI)

Input: Initial values $J_T(\cdot,\cdot) = 0$, $V_0(\cdot,\cdot,\cdot) = 0$; terminal condition Δ.

Output: Optimal T-periodic policy $(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*)$.

for $t = T-1, T-2, \ldots, 1$ **do**
  for *each slow state* $x \in \mathcal{X}$ **do**
    for *each fast state* $y \in \mathcal{Y}$ **do**
      $J_t^*(x,y) = \max_a \mathbf{E}[r(x,y,a) + \gamma J_{t+1}^*(x, f_{\mathcal{Y}}(x,y,a,w))]$.
      $\tilde{\pi}_t^*(x,y) = \arg\max_a \mathbf{E}[r(x,y,a) + \gamma J_{t+1}^*(x, f_{\mathcal{Y}}(x,y,a,w))]$.
    end
  end
end
while $\|V_k - V_{k-1}\|_\infty > \Delta$ **do**
  for $s_0 = (x_0, y_0)$ *in the state space* $\mathcal{X} \times \mathcal{Y}$ **do**
    $V_k(x_0, y_0, J_1^*, \tilde{\boldsymbol{\pi}}^*) = \max_a \mathbf{E}\big[\tilde{R}(s_0, a, J_1^*) + \gamma^T V_{k-1}(x_T, y_T, J_1^*, \tilde{\boldsymbol{\pi}}^*)\big]$.
  end
end
for $s_0 = (x_0, y_0)$ *in the state space* $\mathcal{X} \times \mathcal{Y}$ **do**
  $\tilde{\mu}^*(x_0, y_0) = \arg\max_a \mathbf{E}\big[\tilde{R}(s_0, a, J_1^*) + \gamma^T V_k(x_T, y_T, J_1^*, \tilde{\boldsymbol{\pi}}^*)\big]$.
end

4.3.3 Frozen-State Value Iteration

The full *frozen-state value iteration* (FSVI) algorithm is given in Algorithm 3. The idea is to first solve the lower-level MDP with frozen states and then feed the resulting policy into the upper-level problem. We then apply value iteration (VI) in the upper level, which is a problem with discount factor γ^T. In this section, we will show an upper bound for the on the regret from applying applying the T-periodic policy $(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*)$ instead of the optimal policy ν^* in the base model.

Definition 4.3.1 (Regret of the Frozen-State Policy)**.** *Consider a fast-slow MDP with initial state s_0 and optimal policy ν^*. The regret of the T-periodic frozen-state policy $(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*)$ is defined as:*

$$\mathcal{R}(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*, T) = U^{\mu^*}(s_0) - \bar{U}^{\tilde{\mu}^*}(s_0, \tilde{\boldsymbol{\pi}}^*) = \bar{U}^*(s_0) - \bar{U}^{\tilde{\mu}^*}(s_0, \tilde{\boldsymbol{\pi}}^*),$$

where the second equality uses the value equivalence between the base model and its hierarchical reformulation.

Remark 4.3.2. *As a follow-up comment to Remark 4.3.1, notice that $V^*(s_0, J_1^*, \tilde{\boldsymbol{\pi}}^*)$ does not enter the regret definition as this is the optimal value of the frozen-state approximation, not the value of the policy (generated by the frozen-state approximation) when evaluated in the base model.*

4.3.4 Exact and Frozen-State (Lower-Level) Bellman Operators

Next, we introduce the two Bellman operators, one for the base model and another one for lower level of the frozen-state approximation. Denote by H the Bellman operator of the base model; for any state (x, y) and value function U, define

$$(HU)(x, y) = \max_a \, r(x, y, a) + \gamma \, \mathbf{E}\big[U(f_{\mathcal{X}}(x, w), f_{\mathcal{Y}}(x, y, a, w))\big].$$

Recall that $(\mu^*, \boldsymbol{\pi}^*)$ is an optimal T-periodic policy of the base model's hierarchical reformulation (4.8). Suppose $\boldsymbol{\pi}^*$ is available. Then, the Bellman equation of the base model reformulation can be written as

$$U^*(x_0, y_0) = \bar{U}^*(x_0, y_0)$$

$$
\begin{aligned}
&= \max_a \mathbf{E}\big[R(x_0, y_0, a, \boldsymbol{\pi}^*) + \gamma^T \bar{U}^*(x_T, y_T)\big] \\
&= \max_a \mathbf{E}\left[r(x_0, y_0, a) + \sum_{t=1}^{T-1} \gamma^t\, r(x_t, y_t, \pi_t^*) + \gamma^T\, U^*(x_T, y_T)\right] \\
&= \max_a \mathbf{E}\Big[r(x_0, y_0, a) + \gamma\,\big(H^{T-1}U^*\big)(x_1, y_1)\Big]. \qquad (4.13)
\end{aligned}
$$

Therefore, the expected T-horizon reward can be written as

$$
\mathbf{E}\big[R(x_0, y_0, a, \boldsymbol{\pi}^*)\big] = \mathbf{E}\Big[r(x_0, y_0, a) + \gamma\,\big(H^{T-1}U^*\big)(x_1, y_1) - \gamma^T\, U^*(x_T, y_T)\Big]. \qquad (4.14)
$$

Moving on to the frozen-state approximation, we denote by $\tilde{H}$ the Bellman operator of the lower-level problem, which is on the same timescale as the base model (hence, the discount factor is γ), but the transition of the slow-state x is frozen. For any state (x, y) and lower-level value function J_{t+1}, (We include time indexing on the value function to emphasize that this Bellman operator is used in a finite-horizon (i.e., non-stationary) setting, but the definition of $\tilde{H}$ itself does not depend on t.) define:

$$
\big(\tilde{H} J_{t+1}\big)(x, y) = \max_a\, r(x, y, a) + \gamma\, \mathbf{E}\big[J_{t+1}(x, f_{\mathcal{Y}}(x, y, a, w))\big].
$$

Given the optimal value J_1^* of the lower level (4.10), the T-horizon reward of the upper level (4.12) can be written as

$$
\begin{aligned}
\mathbf{E}\big[\tilde{R}(x_0, y_0, a, J_1^*)\big] &= r(x_0, y_0, a) + \gamma\, \mathbf{E}\big[J_1^*(x_1, y_1)\big] \\
&= r(x_0, y_0, a) + \gamma\big(\tilde{H}^{T-1} J_T^*\big)(x_1, y_1), \\
&= r(x_0, y_0, a) + \gamma\big(\tilde{H}^{T-1}\, \mathbf{0}\big)(x_1, y_1), \qquad (4.15)
\end{aligned}
$$

where $\mathbf{0}$ is the all-zero value function. The difference between (4.14) and (4.15) can be interpreted as follows: in the former, we follow a lower-level policy that is *aware* of a terminal value U^* (but exclude value when defining the T-horizon reward), while in the latter, we follow a lower-level policy that sees zero terminal reward at the end of the $T-1$ periods.

4.3.5 Analyzing the Regret of Frozen-State Policy

In this section, we derive a bound on $\mathcal{R}(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*, T)$, the regret of applying T-periodic policy $(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*)$ to the base model.

We notice that the difference between the base model and the frozen-state approximation exists in their transition functions and reward functions. For two MDPs, MDP_1 and MDP_2, with different transition functions and reward functions, Lemma 4.3.1 bounds the difference between their optimal value functions, and Lemma 4.3.2 bounds the error of apply the optimal policy of MDP_2 to MDP_1.

Lemma 4.3.1 (Optimal Value Bound of Different MDPs)**.** *Consider two MDPs who differ in their transition and reward functions $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_1, r_1, \gamma \rangle$ and $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_2, r_2, \gamma \rangle$. Let U_1^* and U_2^* be their respective optimal value functions. Suppose that*

(a) $|r_1(s,a) - r_2(s,a)| \le \epsilon_r$ *for all* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$*;*

(b) $\|f_1(s,a,w) - f_2(s,a,w)\|_2 \le d$ *for all* $s \in \mathcal{S}$*,* $a \in \mathcal{A}$ *and* $w \in \mathcal{W}$*; and*

(c) there exists $L_1 > 0$ *such that* $|U_1^*(s) - U_1^*(s')| \le L_1 \|s - s'\|_2$ *for all* $s, s' \in \mathcal{S}$*.*

Then, the difference in optimal values of the two MDPs can be bounded as follows:

$$\left|U_1^*(s) - U_2^*(s)\right| \le \epsilon_U = \frac{\epsilon_r + \gamma L_1 d}{1 - \gamma}$$

for all $s \in \mathcal{S}$*.*

The proof is in Appendix C.1.3.

Lemma 4.3.2 (Simulation Lemma Variant)**.** *Consider two MDPs who differ in their transition and reward functions $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_1, r_1, \gamma \rangle$ and $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_2, r_2, \gamma \rangle$. Let U_1^* and U_2^* be their respective optimal value functions, and let π_2^* be an optimal policy for the second MDP. Suppose that*

(a) $|r_1(s,a) - r_2(s,a)| \le \epsilon_r$ *for all* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$*;*

(b) $\|f_1(s,a,w) - f_2(s,a,w)\|_2 \le d$ *for all* $s \in \mathcal{S}$*,* $a \in \mathcal{A}$ *and* $w \in \mathcal{W}$*;*

(c) there exists $L_1 > 0$ *such that* $|U_1^*(s) - U_1^*(s')| \le L_1 \|s - s'\|_2$ *for any* $s, s' \in \mathcal{S}$*; and*

(d) $|U_1^*(s) - U_2^*(s)| \le \epsilon_U$ *for all* $s \in \mathcal{S}$*.*

Then, the value of π_2^* when implemented in the first MDP has regret bounded by:

$$U_1^*(s) - U_1^{\pi_2^*}(s) \leq \frac{2\epsilon_r + 2\gamma\epsilon_U + \gamma L_1 d}{1-\gamma}.$$

for all $s \in \mathcal{S}$.

The proof is in Appendix C.1.4.

Proposition 4.3.1. *Consider a (α, d_y, L_r, L_f)-fast-slow MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$. If $\gamma L_f < 1$, then the optimal value U^* of the base model (4.5) satisfies:*

$$\left|U^*(x,y) - U^*(\tilde{x},\tilde{y})\right| \leq \frac{L_r}{1-\gamma L_f}\big(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2\big). \tag{4.16}$$

for any states $(x,y) \in \mathcal{S}$ and $(\tilde{x},\tilde{y}) \in \mathcal{S}$.

The proof is in Appendix C.1.5.

Proposition 4.3.2. *Consider a (α, d_y, L_r, L_f)-fast-slow MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$ with $\gamma L_f < 1$. Let $\boldsymbol{\pi}^*$ be the optimal lower-level policy for the base model reformulation (4.8) and J_1^* be the optimal (first-stage) value of the lower-level problem in the frozen-state approximation (4.10). For any state $s_0 = (x_0, y_0)$ and action a, the approximation error between the T-horizon reward of hierarchical reformulation and the frozen-state approximation, i.e., the discrepancy between (4.14) and (4.15), can be bounded as:*

$$\begin{aligned}
&\left|\mathbf{E}[R(s_0, a, \boldsymbol{\pi}^*)] - \mathbf{E}[\tilde{R}(s_0, a, J_1^*)]\right| \\
&\leq \epsilon_r(\gamma, \alpha, d_y, L_r, L_f, T) \\
&= (\alpha+2)\, L_r\, d_y \left[\frac{\gamma^2}{(1-\gamma)^2} - A(\gamma,\alpha,L_f)\,\gamma^T + B(\gamma,\alpha,L_f)\,T\gamma^T\right],
\end{aligned}$$

where

$$A(\gamma,\alpha,L_f) = \frac{\gamma^2}{(1-\gamma)^2} + \frac{1}{1-\gamma L_f} \quad \textit{and} \quad B(\gamma,\alpha,L_f) = \frac{2}{1-\gamma L_f} - \frac{\gamma}{1-\gamma}.$$

The proof is in Appendix C.2. Proposition 4.3.2 shows that the distance between the two reward functions is dependent on the problem and the choice of T. We discuss the bound on T for a given error level in Section 4.3.6.

The expected regret $\mathcal{R}(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*, T)$ of applying a suboptimal policy learned from the frozen-state hierarchical approximation to the base model is bounded in Theorem 4.3.1. The proof is based on Lemmas 4.3.1 and 4.3.2, and Propositions 4.3.1 and 4.3.2.

Theorem 4.3.1. *Consider a (α, d_y, L_r, L_f)-fast-slow MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$ with $\gamma L_f < 1$. The regret of applying $(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}})$ in the base model is bounded by*

$$\mathcal{R}(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*, T) = \frac{1}{(1-\gamma^T)^2}\Big(2\epsilon_r(\gamma, \alpha, d_y, L_r, L_f, T) + \frac{L_r}{1-\gamma L_f} d(\alpha, d_y, T)\gamma^T(1+\gamma^T)\Big),$$

where $d(\alpha, d_y, T) = 2(\alpha+1)d_y(T-1)$, and

$$\epsilon_r(\gamma, \alpha, d_y, L_r, L_f, T) = (\alpha+2)L_r d_y\Big(\frac{\gamma^2}{(1-\gamma)^2} - A(\gamma, \alpha, L_f)\gamma^T + B(\gamma, \alpha, L_f)T\gamma^T.$$

Proof. Let $\text{MDP}_1 = \langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_1, r_1, \gamma^T \rangle$ and $\text{MDP}_2 = \langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_2, r_2, \gamma^T \rangle$ be the base model reformulation and the frozen-state hierarchical approximation respectively. Denote $(\mu^*, \boldsymbol{\pi}^*)$ and $(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*)$ the optimal T-horizon policies of MDP_1 and MDP_2 respectively. The reward functions r_1 and r_2 are defined as $r_1(s,a) = \mathbf{E}[R(s,a,\boldsymbol{\pi}^*)]$ and $r_2(s,a) = \mathbf{E}[\tilde{R}(s,a,J_1^*)]$. Proposition 4.3.2 provides a bound that $|r_1(s,a) - r_2(s,a)| \le \epsilon_r(\gamma, \alpha, d_y, L_r, L_f, T)$.

Given noise sequence $\boldsymbol{w} = (w_0, w_1, \ldots, w_{T-1})$, $s_T = (x_T, y_T) = f_1(s, a, \boldsymbol{w})$ is the state starting from s, and taking action a, then following policy $\boldsymbol{\pi}^*$ for the next $T-1$ steps. As for MDP_2, $\tilde{s}_T = (\tilde{x}_T, \tilde{y}_T) = f_2(s, a, \boldsymbol{w})$ is the state starting from s, and taking action a, then following policy $\tilde{\boldsymbol{\pi}}^*$ for the next $T-1$ steps. According to Lemma C.1.1, $\|x_T - \tilde{x}_T\|_2 \le 2(T-1)\alpha d_y$, and $\|y_T - \tilde{y}_T\|_2 \le 2(T-1)d_y$. Therefore,

$$\begin{aligned}
\|f_1(s,a,\boldsymbol{w}) - f_2(s,a,\boldsymbol{w})\|_2 &\le d(\alpha, d_y, T) \\
&= \max_{s,a,\boldsymbol{w}}(\|x_T - \tilde{x}_T\|_2 + \|y_T - \tilde{y}_T\|_2) \\
&\le 2(\alpha+1)d_y(T-1),
\end{aligned}$$

and the bound $\big|U_1^*(s) - U_2^*(s)\big|$ in Lemma 4.3.1 becomes

$$\big|U_1^*(s) - U_2^*(s)\big| \le \epsilon_U(\gamma, \alpha, d_y, L_r, L_f, T) = \frac{\epsilon_r(\gamma, \alpha, d_y, L_r, L_f, T) + \frac{L_r}{1-\gamma L_f} d(\alpha, d_y, T)\gamma^T}{1-\gamma^T}.$$

With all the above information, the regret bound in Lemma 4.3.2 becomes

$$\begin{aligned}\mathcal{R}(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*, T) &= \max_{x,y} \bar{U}^*(x,y) - \bar{U}^{\tilde{\mu}^*}(x,y,\tilde{\boldsymbol{\pi}}^*) \\ &\leq \frac{2\epsilon_r(\gamma,\alpha,d_y,L_r,L_f,T) + 2\gamma^T\epsilon_U(\gamma,\alpha,d_y,L_r,L_f,T) + \frac{L_r}{1-\gamma L_f} d(\alpha,d_y,T)\gamma^T}{1-\gamma^T} \\ &= \frac{1}{(1-\gamma^T)^2}\Big(2\epsilon_r(\gamma,\alpha,d_y,L_r,L_f,T) + \frac{L_r}{1-\gamma L_f} d(\alpha,d_y,T)\gamma^T(1+\gamma^T)\Big).\end{aligned}$$

□

4.3.6 Discussion of the Choice of T

In this section, we focus on the exact VIs for the base model and the frozen-state hierarchical approximation, and provide a bound on the value of T for a desired error level. Denote U_k and V_k the value functions in iteration k of VI for the base model and the upper level of the frozen-state hierarchical approximation respectively, denote ν^k and $\tilde{\mu}^k$ the greedy policies in iteration k of the two models respectively. For the base model, Proposition 4.3.3 gives the required number of iterations to achieve a desired error for exact VI.

Proposition 4.3.3. *The base model requires at least*

$$K_{\text{base}}(\xi) = \frac{1}{\log(\gamma)} \log\Big(\frac{(1-\gamma)^2\xi}{4r_{\max}}\Big) - 1$$

iterations to achieve an error of ξ for exact VI, i.e., $\mathcal{R}(K_{\text{base}}(\xi)) = \|U^{\nu_{K_{\text{base}}(\xi)}} - U^*\|_\infty \leq \xi$.

Let us next consider the frozen-state hierarchical approximation. The lower-level value functions are exactly solved by value iteration. For the upper-level problem, denote F and F^μ the Bellman operators of the upper-level problem of the approximation (4.12), i.e.,

$$(FV)(x_0,y_0,J_1,\boldsymbol{\pi}) = \max_a \mathbf{E}\big[\tilde{R}(x_0,y_0,a,J_1) + \gamma^T V(x_T,y_T,J_1,\boldsymbol{\pi})\big],$$

and

$$(F^\mu V)(x_0,y_0,J_1,\boldsymbol{\pi}) = \mathbf{E}\big[\tilde{R}(x_0,y_0,\mu(x_0,y_0),J_1) + \gamma^T V^\mu(x_T,y_T,J_1,\boldsymbol{\pi})\big].$$

Proposition 4.3.4 gives the the required number of iterations to achieve a desired error for exact VI.

Proposition 4.3.4. *The frozen-state hierarchical approximation requires*

$$K_{\text{frozen}}(\xi, T) = \frac{1}{T\log(\gamma)} \log\Big(\frac{\xi(1-\gamma)(1-\gamma^T)}{4r_{\max}}\Big) - 1$$

iterations to achieve an error of ξ for exact VI, i.e.,

$$\mathcal{R}(K_{\text{frozen}}(\xi, T)) = \|V^{\tilde{\mu}_{K_{\text{frozen}}(\xi,T)}}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) - V^*(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*)\|_\infty \leq \xi.$$

Propositions 4.3.3 and 4.3.4 show the required number of iterations to achieve desired error level ξ for exact VI for the base model (4.5) and the upper level of the frozen-state approximation (4.12) respectively. Let us now discuss the regret of applying the policy learned from frozen-state VI to the base model in Corollary 4.3.1. The regret is composed of two parts, one is an error caused by freezing slow state x for every T periods, the other is the VI error discussed in Proposition 4.3.4.

Corollary 4.3.1. *If T satisfies $\epsilon_U(\gamma, \alpha, d_y, L_r, L_f, T) \leq \xi_2$, and $k \geq K_{\text{frozen}}(\xi_1, T)$, then the error of applying T-periodic policy $(\tilde{\mu}_k, \tilde{\boldsymbol{\pi}}^*)$ to the base model is bounded by*

$$\|\tilde{U}^{\tilde{\mu}_k}(\cdot,\cdot,\tilde{\boldsymbol{\pi}}^*) - U^*\|_\infty \leq \xi_1 + 2\xi_2,$$

where $\epsilon_U(\gamma, \alpha, d_y, L_r, L_f, T) = \frac{1}{1-\gamma^T}\big(\epsilon_r(\gamma, \alpha, d_y, L_r, L_f, T) + \frac{L_r}{1-\gamma L_f} d(\alpha, d_y, T)\gamma^T\big)$.

The proofs for this section are in Appendix C.2.1.

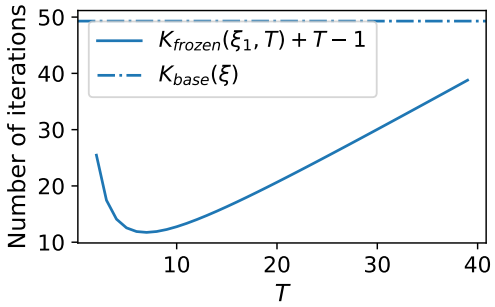
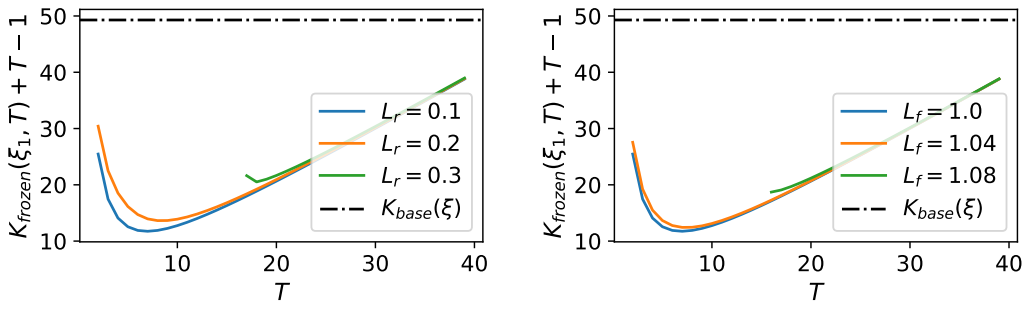
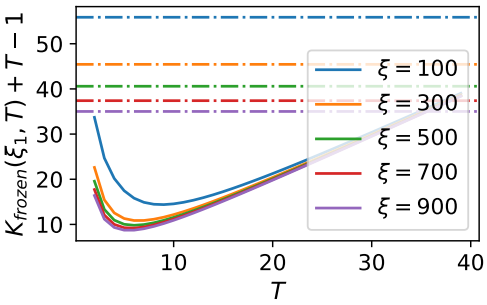


Figure 23: The choice of T



(a) L_r (b) L_f



(c) ξ

Figure 24: Sensitivity analysis for the choice of T

We next show the benefits of adopting frozen-state model and discuss the choice of T numerically. Consider a fast-slow MDP with parameters $\gamma = 0.9$, $r_{\max} = 100$, $\alpha = 0.2$, $d_y = 1$ $L_r = 0.1$, $L_f = 1$. We consider an error level $\xi = 200$, which is 80% of the optimal value, and let $\mathcal{R}(K_{\text{base}}(\xi)) \leq \xi$ and $\|\tilde{U}^{\tilde{\mu}_k}(\cdot, \cdot, \tilde{\boldsymbol{\pi}}^*) - U^*\|_\infty \leq \xi$ respectively. The latter one leads to $\mathcal{R}(K_{\text{frozen}}(\xi, T)) \leq \xi_1 = \xi - 2\epsilon_U(\gamma, \alpha, d_y, L_r, L_f, T)$. According to Proposition 4.3.4, the minimum number of iterations $K_{\text{frozen}}(\xi, T)$ to achieve error ξ of the frozen-state model is impacted by the value of T. Figure 23 shows $K_{\text{frozen}}(\xi_1, T) + T - 1$, the number of iterations for error level ξ_1 plus the number of periods of the lower level, as a function of T given the parameters provided above. The value of $K_{\text{frozen}}(\xi_1, T) + T - 1$ first decreases then increases as T increases, and the minimum value is taken at $T = 7$. The dotted line denotes the value

of $K_{\text{base}}(\xi)$. In the range of T shown in the plot, the number of iterations of the frozen-state model (including the computation for the lower level) is smaller than the number of iterations of the base model to achieve the same error ξ.

Figure 24 shows the value of $K_{\text{frozen}}(\xi_1, T)+T-1$ as a function of T for different Lipschitz constants L_r, L_f and ξ. As L_r increases, $K_{\text{frozen}}(\xi_1, T)+T-1$ increases, while the increment diminishes as T increases. When L_r is large ($L_r = 0.3$), the error from freezing slow state is high that $\xi_1+2\xi_2 > \xi$ for small T ($T < 15$). The impact of L_f is similar, $K_{\text{frozen}}(\xi_1, T)+T-1$ increases as L_f increases. Note that the value of γL_f must be smaller than 1. Otherwise, the analysis in this paper will follow another path discussed in Appendix C.1.5.1, and the discussion of the choice of T will also be different. We focus on the case with $\gamma L_f < 1$ in this paper. As for ξ, it impact both $K_{\text{base}}(\xi)$ (the dotted lines) and $K_{\text{frozen}}(\xi_1, T)+T-1$. A small error level ξ requires more number of iterations of the base model and the frozen-state model. The error level ξ has higher impact in on $K_{\text{base}}(\xi)$ than $K_{\text{frozen}}(\xi_1, T) + T - 1$. Moreover, when ξ is large and T is large, $K_{\text{frozen}}(\xi_1, T) + T - 1$, is larger than $K_{\text{base}}(\xi)$.

4.3.7 Nominal-State Approximation

Assumption 4.3.1 (Nearly Factored Reward)**.** *The reward function is "nearly factored," in the sense that* $|g(x) + h(y, a) - r(x, y, a)| \leq \zeta$ *for any* $x \in \mathcal{X}$*,* $y \in \mathcal{Y}$*, and* $a \in \mathcal{A}$*.*

Property 1. *The functions* $g : \mathcal{X} \to \mathbb{R}$ *and* $h : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ *are Lipschitz w.r.t. the state and action variables, i.e., there exists* $L'_r > 0$ *such that for all* $x, x' \in \mathcal{X}$*,* $y, y' \in \mathcal{Y}$ *and* $a, a' \in \mathcal{A}$*,*

$$|g(x) - g(x')| \leq L'_r \|x - x'\|_2, \tag{4.17}$$

and

$$|h(y, a) - h(y', a')| \leq L'_r \|(y, a) - (y', a')\|_2. \tag{4.18}$$

This terminology comes from the notion *factored* MDPs, a commonly-studied type of weakly-connected structure that notably assumes an additive reward function [229, 230]. Note that although the analysis in this paper is based on the $|g(x)+h(y, a)-r(x, y, a)| \leq \zeta$, it is easy to extend the analysis to other types of separable rewards: for example, $|g(x)h(y, a)-r(x, y, a)| \leq \zeta$.

Algorithm 4: Value Iteration for the Nominal State Approximation

Input: The nominal slow state x^*; initial values $\bar{J}_T(x^*, \cdot) = 0$, $\bar{V}_0(\cdot, \cdot, \cdot, \cdot) = 0$; terminal condition Δ.

Output: Optimal T-periodic policy $(\bar{\mu}^*, \bar{\boldsymbol{\pi}}^*)$.

for $t = T-1, T-2, \ldots, 1$ **do**
  for *y in the fast state space* $\mathcal{Y}$ **do**
    $\bar{J}_t(x^*, y) = \max_a \mathbf{E}[g(x^*) + h(y, a) + \gamma \bar{J}_{t+1}(x^*, f(x^*, y, a, w))]$.
    $\bar{\pi}_t^*(x, y) = \arg\max_a \mathbf{E}[g(x^*) + h(y, a) + \gamma \bar{J}_{t+1}(x^*, f(x^*, y, a, w))]$, $\forall x$.
  end
end
while $\|\bar{V}_k - \bar{V}_{k-1}\|_\infty > \Delta$ **do**
  for $s_0 = (x_0, y_0)$ *in the state space* $\mathcal{X} \times \mathcal{Y}$ **do**
    $\bar{V}_k(x_0, y_0, \bar{J}_1, \bar{\boldsymbol{\pi}}^*) = \max_a \mathbf{E}\big[\bar{R}(s_0, a, \bar{J}_1) + \gamma^T \bar{V}_{k-1}(x_T, y_T, \bar{J}_1, \bar{\boldsymbol{\pi}}^*)\big]$.
  end
end
for $s_0 = (x_0, y_0)$ *in the state space* $\mathcal{X} \times \mathcal{Y}$ **do**
  $\bar{\mu}^*(x_0, y_0) = \arg\max_a \mathbf{E}\big[\bar{R}(s_0, a, \bar{J}_1) + \gamma^T \bar{V}_k(x_T, y_T, \bar{J}_1, \bar{\boldsymbol{\pi}}^*)\big]$.
end

To further efficiently solve the lower level of the frozen-state hierarchical approximation, we make use of a nominal slow state x^*. In this section, we discuss the case with a single nominal slow state x^* for simplicity. It is easy to be extended to multiple nominal slow states.

Let us first introduce the lower-level MDP with factored reward function $g(x) + h(y, a)$ (Assumption 4.3.1). The value function can be written as, for $t = 1, 2, \ldots, T-1$,

$$\bar{J}_t(x, y) = \max_{a \in \mathcal{A}} \mathbf{E}\big[g(x) + h(y, a) + \gamma \bar{J}_{t+1}(x, f_{\mathcal{Y}}(x, y, a, w))\big], \tag{4.19}$$

and $\bar{J}_T(\cdot, \cdot) = 0$. Denote $\bar{J}_t^*$ the optimal value, $\bar{\boldsymbol{\pi}}^*$ the optimal $(T-1)$-period policy. Let us solve (4.19) only for the nominal slow state x^*. The value of other slow state $x \in \mathcal{X}$ can be approximated by $\bar{J}_t(x^*, y_t)$ as follows,

$$\bar{J}_t(x, y) = \sum_{i=0}^{T-t-1} \gamma^i (g(x) - g(x^*)) + \max_{a \in \mathcal{A}} \mathbf{E}\big[g(x^*) + h(y, a) + \gamma \bar{J}_{t+1}(x^*, f_{\mathcal{Y}}(x^*, y, a, w))\big] \tag{4.20}$$

$$= \sum_{i=0}^{T-t-1} \gamma^i (g(x) - g(x^*)) + \bar{J}_t(x^*, y). \tag{4.21}$$

According to Assumption 4.3.1, using $g(x) + h(y, a)$ as the reward function in the lower level instead of $r(x, y, a)$ incurs an error. The value approximation (4.21) also incurs an error. Lemma 4.3.3 bounds the error in the lower level.

Lemma 4.3.3. *The error in the lower level value function introduced by using nominal state approximation to the frozen-state hierarchical approximation is*

$$\begin{aligned} |\bar{J}_t(x, y) - J_t(\tilde{x}, \tilde{y})| \leq & \sum_{i=0}^{T-t-1} \gamma^i \zeta + L_r \sum_{i=0}^{T-t-1} (\gamma L_f)^i (\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2) \\ & + \gamma L_r L_f \Big(\sum_{i=0}^{T-t-2} L_f^i \sum_{j=i}^{T-t-2} \gamma^j \Big) \|x - x^*\|_2. \end{aligned}$$

The proof is in Appendix C.2.2. Denote $\bar{H}$ and $\bar{H}^{\boldsymbol{\pi}}$ the Bellman operators of lower level of the nominal state approximation, i.e.,

$$(\bar{H}\bar{J}_t)(x,y) = \sum_{i=0}^{T-t-1} \gamma^i (g(x) - g(x^*)) + \max_{a\in\mathcal{A}} \mathbf{E}\big[g(x^*) + h(y,a) + \gamma \bar{J}_{t+1}(x^*, f_{\mathcal{Y}}(x^*,y,a,w))\big],$$

and

$$(\bar{H}^{\boldsymbol{\pi}}\bar{J}_t)(x,y) = \sum_{i=0}^{T-t-1} \gamma^i (g(x) - g(x^*)) + \mathbf{E}\big[g(x^*) + h(y,\pi_t(x^*,y)) + \gamma \bar{J}_{t+1}(x^*, f^{\pi_t}_{\mathcal{Y}}(x^*,y,w))\big].$$

Let us now consider the upper-level problem. Given the lower-level policies $\boldsymbol{\pi}$, the value at state $s_0 = (x_0, y_0)$ by executing policy μ is

$$\bar{V}^{\mu}(x_0, y_0, \bar{J}_1, \boldsymbol{\pi}) = \mathbf{E}\big[\bar{R}(s_0, \mu(s_0), \bar{J}_1) + \gamma^T \bar{V}^{\mu}(x_T, y_T, \bar{J}_1, \boldsymbol{\pi})\big]. \tag{4.22}$$

where

$$\begin{aligned}\mathbf{E}[\bar{R}(s_0,a,\bar{J}_1)] &= \mathbf{E}\big[r(x_0,y_0,a) + \gamma \bar{J}_1(x_1,y_1)\big] \\ &= \mathbf{E}\big[r(x_0,y_0,\mu) + (g(x_1) - g(x^*)) \sum_{i=1}^{T-1} \gamma^i + \gamma \bar{J}_1(x^*,y_1)\big],\end{aligned}$$

and $x_1 = f_{\mathcal{X}}(x_0, w_0)$, $y_1 = f^{\mu}_{\mathcal{Y}}(x_0, y_0, w_0)$. The value at state $s_0 = (x_0, y_0)$ can be written as

$$\bar{V}(x_0, y_0, \bar{J}_1, \boldsymbol{\pi}) = \max_a \mathbf{E}\big[\bar{R}(s_0, a, \bar{J}_1) + \gamma^T \bar{V}(x_T, y_T, \bar{J}_1, \boldsymbol{\pi})\big]. \tag{4.23}$$

Given the optimal lower-level policy $\bar{\boldsymbol{\pi}}^*$, denote $\bar{V}^*(x_0, y_0, \bar{J}_1, \bar{\boldsymbol{\pi}}^*)$ the optimal upper-level value, and $\bar{\mu}^*$ the corresponding optimal policy. Denote $(\bar{\mu}^*, \bar{\boldsymbol{\pi}}^*)$ the optimal T-periodic policy for the nominal state approximation. The algorithm is introduced in Algorithm 4.

Next, we discuss the regret in the value of applying T-periodic policy $(\bar{\mu}^*, \bar{\boldsymbol{\pi}}^*)$ to the base model reformulation, i.e.,

$$\mathcal{R}(\tilde{\mu}^*, \tilde{\boldsymbol{\pi}}^*, T) = \|\bar{U}^*(\cdot,\cdot) - \bar{U}^{\bar{\mu}^*}(\cdot,\cdot,\bar{\boldsymbol{\pi}}^*)\|_{\infty}. \tag{4.24}$$

The analysis is similar to the analysis in Section 4.3.3. We focus on the extra error introduced by leveraging the nominal slow state in the lower level in this section. The extra error is bounded in Proposition 4.3.5.

Proposition 4.3.5. *Leveraging the nominal slow state leads to an error in the lower level,*

$$\begin{aligned}\left|\mathbf{E}[\tilde{R}(s_0,a,J_1^*)]-\mathbf{E}[\bar{R}(s_0,a,\bar{J}_1^*)]\right| &\le \epsilon_r'(\gamma,L_r,L_f,T)\\ &= \sum_{i=1}^{T-1}\gamma^i\zeta+\gamma^2L_rL_f\Big(\sum_{i=0}^{T-3}L_f^i\sum_{j=i}^{T-3}\gamma^j\Big)\max_x\|x-x^*\|_2.\end{aligned}$$

The proof is in Section C.2.3. Combining Lemma 4.3.3 and Proposition 4.3.5 with the analysis of Theorem 4.3.1, we show the expected regret $\mathcal{R}(\bar{\mu}^*,\bar{\boldsymbol{\pi}}^*,T)$ in Theorem 4.3.2.

Theorem 4.3.2. *Consider a (α,d_y,L_r,L_f)-fast-slow MDP. The expected regret is bounded by*

$$\begin{aligned}&\mathcal{R}(\bar{\mu}^*,\bar{\boldsymbol{\pi}}^*,T)\\ &=\|\bar{U}^*(\cdot,\cdot)-\bar{U}^{\bar{\mu}^*}(\cdot,\cdot,\bar{\boldsymbol{\pi}}^*)\|_\infty\\ &\le\frac{1}{(1-\gamma^T)^2}\Big(2(\epsilon_r(\gamma,\alpha,d_y,L_r,L_f,T)+\epsilon_r'(\gamma,L_r,L_f,T))+(1+\gamma^T)\frac{L_r}{1-\gamma L_f}d(\alpha,d_y,T)\gamma^T\Big),\end{aligned}$$

where $\epsilon_r(\gamma,\alpha,d_y,L_r,L_f,T)$ is the error in Proposition 4.3.2, $\epsilon_r'(\gamma,L_r,L_f,T)$ is the error in Proposition 4.3.5, and $d(\alpha,d_y,T)=2(\alpha+1)d_y(T-1)$.

Proof. The proof is similar to the proof of Theorem 4.3.1. Consider three MDPs, let $\text{MDP}_1=\langle\mathcal{S},\mathcal{A},\mathcal{W},f_1,r_1,\gamma^T\rangle$ be the base model reformulation, $\text{MDP}_2=\langle\mathcal{S},\mathcal{A},\mathcal{W},f_2,r_2,\gamma^T\rangle$ be the frozen-state hierarchical approximation, and $\text{MDP}_3=\langle\mathcal{S},\mathcal{A},\mathcal{W},f_3,r_3,\gamma^T\rangle$ be the nominal state approximation. Their optimal policies are $(\mu^*,\boldsymbol{\pi}^*)$, $(\tilde{\mu}^*,\tilde{\boldsymbol{\pi}}^*)$ and $(\bar{\mu}^*,\bar{\boldsymbol{\pi}}^*)$. The reward functions r_1, r_2 and r_3 are defined as $r_1(s,a)=\mathbf{E}[R(s,a,\boldsymbol{\pi}^*)]$, $r_2(s,a)=\mathbf{E}[\tilde{R}(s,a,J_1^*)]$ and $r_3(s,a)=\mathbf{E}[\bar{R}(s,a,\bar{J}_1^*)]$. According to Propositions 4.3.2 and 4.3.5, we have

$$\begin{aligned}|r_1(s,a)-r_3(s,a)| &\le |r_1(s,a)-r_2(s,a)|+|r_2(s,a)-r_3(s,a)|\\ &\le \epsilon_r(\gamma,\alpha,d_y,L_r,L_f,T)+\epsilon_r'(\gamma,L_r,L_f,T),\end{aligned}$$

where $\epsilon_r(\gamma,\alpha,d_y,L_r,L_f,T)$ is the error in Proposition 4.3.2, $\epsilon_r'(\gamma,L_r,L_f,T)$ is the error in Proposition 4.3.5.

Given noise sequence $\boldsymbol{w}=(w_0,w_1,\ldots,w_{T-1})$, initial state s and initial action a, denote $s_T=(x_T,y_T)=f_1(s,a,\boldsymbol{w})$, $\tilde{s}_T=(\tilde{x}_T,\tilde{y}_T)=f_2(s,a,\boldsymbol{w})$, and $\bar{s}_T=(\bar{x}_T,\bar{y}_T)=f_3(s,a,\boldsymbol{w})$ the final states of the three models after taking action a and then following policies $\boldsymbol{\pi}^*$, $\tilde{\boldsymbol{\pi}}^*$,

and $\bar{\boldsymbol{\pi}}^*$ for the next $T-1$ steps respectively. According to the analysis in Theorem 4.3.1, $\|f_1(s,a,\boldsymbol{w}) - f_3(s,a,\boldsymbol{w})\|_2 \leq d(\alpha, d_y, T) = 2(\alpha+1)d_y(T-1)$.

Therefore, the regret bound is

$$
\begin{aligned}
&\mathcal{R}(\bar{\mu}^*, \bar{\boldsymbol{\pi}}^*, T) \\
&= \max_{x,y} \bar{U}^*(x,y) - \bar{U}^{\bar{\mu}^*}(x,y,\bar{\boldsymbol{\pi}}^*) \\
&\leq \frac{1}{(1-\gamma^T)^2}\Big(2\big(\epsilon_r(\gamma,\alpha,d_y,L_r,L_f,T) + \epsilon_r'(\gamma,L_r,L_f,T)\big) + (1+\gamma^T)\frac{L_r}{1-\gamma L_f} d(\alpha,d_y,T)\gamma^T\Big).
\end{aligned}
$$

□

4.4 The Case of Endogenous Slow States

This section considers the MDP with endogenous slow states, i.e., $\langle \mathcal{X}\times\mathcal{Y}, \mathcal{A}_\mathcal{X}\times\mathcal{A}_\mathcal{Y}, \mathcal{W}, f_\mathcal{X}, f_\mathcal{Y}, r, \gamma\rangle$, where the transition functions are $f_\mathcal{X}: \mathcal{X}\times\mathcal{A}_\mathcal{X}\times\mathcal{W}\to\mathcal{X}$ and $f_\mathcal{Y}: \mathcal{S}\times\mathcal{A}_\mathcal{Y}\times\mathcal{W}\to\mathcal{Y}$. In this section, in violation of the notations in Sections 4.2 and 4.3, we use the *bar* notations for the value function, the policy, and the Bellman operators of the base model and the hierarchical approximation model. Assume that the slow state and the fast state are nearly separable in the reward function.

Assumption 4.4.1. *(Factored reward function). The reward function is factored over* $\mathcal{X}\times\mathcal{Y}\times\mathcal{A}$ *such that* $|g(x,a_x) + h(y,a_y) - r(x,y,a_x,a_y)| \leq \zeta$.

An example is the inventory management with two products, where the demand for product 1 is much lower than the demand for product 2. Since demand of product 1 is low, its inventory level changes slowly. Checking the inventory and making the procurement decision every period might be inefficient. Therefore, it is beneficial for the decision maker to manage the inventory of product 1 every several periods, while manage the inventory of product 2 every period. Details are in Section 4.6.4.

In the hierarchical approximation, the lower level is independent from the actions corresponding to the slow states $\mathcal{A}_\mathcal{X}$. The action a_x is taken every T periods at the upper level.

For the nominal state approximation, the value function of the lower level problem can be written as, for $t = 1, 2, \ldots, T-1$,

$$\bar{J}_t(x, a_x, y) = \max_{a_y \in \mathcal{A}_\mathcal{Y}} \mathbf{E}\big[g(x, a_x) + h(y, a_y) + \gamma \bar{J}_{t+1}(x, a_x, f_\mathcal{Y}(x, y, a_y, w))\big], \tag{4.25}$$

and $\bar{J}_T(\cdot, \cdot, \cdot) = 0$. Denote $\bar{J}_t^*$ the optimal value, $\bar{\boldsymbol{\pi}}^* : \mathcal{S} \times \mathcal{A}_\mathcal{X} \to \mathcal{A}_\mathcal{Y}$ the optimal $T-1$-period policy. Let us solve (4.25) only for the nominal slow state-action pair (x^*, a_x^*). The value of other slow state-action pairs $(x, a_x) \in \mathcal{X} \times \mathcal{A}_\mathcal{X}$ can be approximated by $\bar{J}_t(x^*, a_x^*, y_t)$:

$$\bar{J}_t(x, a_x, y) = \sum_{i=0}^{T-t-1} \gamma^i (g(x, a_x) - g(x^*, a_x^*)) \tag{4.26}$$

$$+ \max_{a_y \in \mathcal{A}_\mathcal{Y}} \mathbf{E}\big[g(x^*, a_x^*) + h(y, a_y) + \gamma \bar{J}_{t+1}(x^*, a_x^*, f_\mathcal{Y}(x^*, y, a_y, w))\big] \tag{4.27}$$

$$= \sum_{i=0}^{T-t-1} \gamma^i (g(x, a_x) - g(x^*, a_x^*)) + \bar{J}_t(x^*, a_x^*, y). \tag{4.28}$$

Denote $\bar{H}$ and $\bar{H}^{\boldsymbol{\pi}}$ the Bellman operators of lower level of the nominal state approximation, i.e.,

$$(\bar{H}\bar{J}_t)(x, a_x, y) = \sum_{i=0}^{T-t-1} \gamma^i (g(x, a_x) - g(x^*, a_x^*)) + \max_{a_y \in \mathcal{A}_\mathcal{Y}} \mathbf{E}\big[g(x^*, a_x^*)) + h(y, a_y) + \gamma \bar{J}_{t+1}(x^*, a_x^*, f_\mathcal{Y}(x^*, y, a_y, w))\big],$$

and

$$(\bar{H}^{\boldsymbol{\pi}}\bar{J}_t)(x, a_x, y) = \sum_{i=0}^{T-t-1} \gamma^i (g(x, a_x) - g(x^*, a_x^*)) + \mathbf{E}\big[g(x^*, a_x^*)) + h(y, \pi_t(x^*, y)) + \gamma \bar{J}_{t+1}(x^*, a_x^*, f_\mathcal{Y}^{\pi_t}(x^*, y, w))\big].$$

Given the lower-level policies $\boldsymbol{\pi}$, the value at state $s_0 = (x_0, y_0)$ by executing policy $\mu : \mathcal{S} \to \mathcal{A}$ is

$$\bar{V}^\mu(x_0, y_0, \bar{J}_1, \boldsymbol{\pi}) = \mathbf{E}\big[\bar{R}(s_0, \mu(s_0), \bar{J}_1) + \gamma^T \bar{V}^\mu(x_T, y_T, \bar{J}_1, \boldsymbol{\pi})\big]. \tag{4.29}$$

where

$$\mathbf{E}[\bar{R}(s_0, a_x, a_y, \bar{J}_1)] = \mathbf{E}\big[r(x_0, y_0, a_x, a_y)) + \gamma \bar{J}_1(x_1, a_x, y_1)\big]$$

$$= \mathbf{E}\big[r(x_0, y_0, \mu) + (g(x_1, a_x) - g(x^*, a_x^*)) \sum_{i=1}^{T-1} \gamma^i + \gamma \bar{J}_1(x^*, a_x^*, y_1)\big],$$

and $x_1 = f_{\mathcal{X}}^{\mu}(x_0, w_0)$, $y_1 = f_{\mathcal{Y}}^{\mu}(x_0, y_0, w_0)$. The value at state $s_0 = (x_0, y_0)$ can be written as

$$\bar{V}(x_0, y_0, \bar{J}_1, \boldsymbol{\pi}) = \max_{a \in \mathcal{A}} \mathbf{E}\big[\bar{R}(s_0, a, \bar{J}_1) + \gamma^T \bar{V}(x_T, y_T, \bar{J}_1, \boldsymbol{\pi})\big]. \tag{4.30}$$

Given the optimal lower-level policy $\bar{\boldsymbol{\pi}}^*$, denote $\bar{V}^*(x_0, y_0, \bar{J}_1, \bar{\boldsymbol{\pi}}^*)$ the optimal upper-level value, and $\bar{\mu}^*$ the corresponding optimal policy. Denote $(\bar{\mu}^*, \bar{\boldsymbol{\pi}}^*)$ the optimal T-periodic policy for the nominal state approximation.

Theorem 4.4.1. *Consider a (α, d_y, L_r, L_f)-fast-slow MDP. The expected regret is bounded by*

$$\begin{aligned}\mathcal{R}(\bar{\mu}^*, \bar{\boldsymbol{\pi}}^*, T) &= \|\bar{U}^*(\cdot, \cdot) - \bar{U}^{\bar{\mu}^*}(\cdot, \cdot, \bar{\boldsymbol{\pi}}^*)\|_\infty \\ &\leq \frac{1}{(1-\gamma^T)^2}\Big(2\bar{\epsilon}_r(\gamma, \alpha, d_y, L_r, L_f, T, \zeta) + (1+\gamma^T)\frac{L_r}{1-\gamma L_f} d(\alpha, d_y, T)\gamma^T\Big),\end{aligned}$$

where

$$\begin{aligned}\bar{\epsilon}_r(\gamma, \alpha, d_y, L_r, L_f, T, \zeta) = d_y(\alpha+2)\big(L_r \sum_{i=1}^{t} i\gamma^i + \frac{L_r}{1-\gamma L_f} t\gamma^t\big) + \sum_{i=0}^{T-t-1} \gamma^i \zeta \\ + \gamma L_r L_f \Big(\sum_{i=0}^{T-t-2} L_f^i \sum_{j=i}^{T-t-2} \gamma^j\Big) \max_{(x, a_x)} \|(x, a_x) - (x^*, a_x^*)\|_2\end{aligned}$$

and $d(\alpha, d_y, T) = 2(\alpha+1)d_y(T-1)$.

The proof is in Appendix C.2.4. In the next section, we will focus on the exogenous slow state model and discuss solving the nominal state approximation problem approximately.

4.5 Approximate Value Iteration for Nominal State Approximation

This section introduces parameterized approximation to VI for the nominal state approximation, and proves the converges of the algorithm.

4.5.1 The Algorithm

In this section, we use linear architecture to approximate the value functions for the lower and upper level problems of the nominal state approximation. Formally, the value functions are approximated by the following form,

$$\hat{J}_t(s, \boldsymbol{\omega}_t) = \boldsymbol{\phi}^\intercal(s)\boldsymbol{\omega}_t, \quad \forall t,$$

$$\hat{V}(s, \hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}) = \boldsymbol{\phi}^\intercal(s)\boldsymbol{\nu},$$

where $\hat{J}_1$ and $\hat{\boldsymbol{\pi}}$ are a given lower level value and policy, $\boldsymbol{\phi}(s) = \big(\phi_1(s), \phi_2(s), \ldots, \phi_M(s)\big)^\intercal \in \mathbb{R}^M$ is the feature vector associated with state s, $\boldsymbol{\omega}_t$ and $\boldsymbol{\nu}$ are the parameter vectors of the lower level and upper level respectively.

Let $\tilde{\mathcal{S}} = \{s_1, s_2, \ldots, s_M\}$ be the pre-selected states for AVI. For the lower level, the parameter $\boldsymbol{\omega}_t$ is estimated by first evaluating $\bar{H}\hat{J}_{t+1}(s, \boldsymbol{\omega}_{t+1})$ at the pre-selected states, and then computing $\boldsymbol{\omega}_t$ so that $\hat{J}_t(s, \boldsymbol{\omega}_t) = \bar{H}\hat{J}_{t+1}(s, \boldsymbol{\omega}_{t+1})$ for $s \in \tilde{\mathcal{S}}$. For the upper level, the parameter $\boldsymbol{\nu}_k$ is improved to $\boldsymbol{\nu}_{k+1}$ in iteration $k+1$ by first evaluating $F\hat{V}(s, \hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}_k)$ at the pre-selected states, then computing $\boldsymbol{\nu}_{k+1}$ so that $\hat{V}(s, \hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}_{k+1}) = F\hat{V}(s, \hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}_k)$ for $s \in \tilde{\mathcal{S}}$. Assume the feature vectors satisfy Assumption 4.5.1.

Assumption 4.5.1. *Let $\tilde{\mathcal{S}} = \{s_1, s_2, \ldots, s_M\}$ be the pre-selected states.*

1. *The vectors $\boldsymbol{\phi}(s_1), \boldsymbol{\phi}(s_2), \ldots, \boldsymbol{\phi}(s_M)$ are linearly independent.*
2. *There exists a value $\gamma' \in [\gamma^T, 1)$ such that for any state $s \in \mathcal{S}$, there exist $\theta_1(s), \theta_2(s), \ldots, \theta_M(s) \in \mathbb{R}$ with*

$$\sum_{m=1^M} |\theta_m(s)| \leq 1,$$

and

$$\boldsymbol{\phi}(s) = \frac{\gamma'}{\gamma^T} \sum_{m=1}^{M} \theta_m(s)\boldsymbol{\phi}(s_m).$$

With Assumption 4.5.1, we introduce the matrices for AVI. Let i be the ith state in the state space $\mathcal{S}$. Recall that s_m is the mth state in the pre-selected state set $\tilde{\mathcal{S}} \subset \mathcal{S}$. Let $\Phi \in \mathbb{R}^{N \times M}$ be a matrix with the ith row equal to $\boldsymbol{\phi}^\intercal(i)$. Let $L \in \mathbb{R}^{M \times M}$ be a matrix with the mth row equal to $\boldsymbol{\phi}^\intercal(s_m)$. The matrix L has a unique matrix inverse $L^{-1} \in \mathbb{R}^{M \times M}$ Since its rows are linearly independent. Define $\Phi^\dagger \in \mathbb{R}^{M \times N}$ as follows: for $s_m = i \in \tilde{\mathcal{S}}$, the

ith column equals to the mth column of L^{-1}; the other entries are zero. Without loss of generality, assume that $s_1 = 1, s_2 = 2, \ldots, s_K = K$, we have

$$\Phi^\dagger \Phi = [L^{-1}\ 0] \begin{bmatrix} L \\ G \end{bmatrix} = L^{-1}L = I,$$

where $I \in \mathbb{R}^{M \times M}$ is the identity matrix and G is the remaining rows of Φ. Therefore, $\Phi^\dagger$ is a left inverse of Φ. For the lower level, the approximate value can be written as

$$\hat{J}_t(\boldsymbol{\omega}_t) = \Phi \boldsymbol{\omega}_t,$$

and the parameter vector $\boldsymbol{\omega}_t$ is updated as follows

$$\boldsymbol{\omega}_t = \bar{H}'(\boldsymbol{\omega}_{t-1}),$$

where $\bar{H}' = \Phi^\dagger \circ \bar{H} \circ \Phi$. Define the linear architecture approximation error as

$$\epsilon_{\mathrm{L}} = \max_t \min_{\boldsymbol{\omega}} \|\bar{J}_t^* - \hat{J}_t(\boldsymbol{\omega}_t)\|_\infty.$$

The AVI for the upper level can be written as

$$\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}) = \Phi \boldsymbol{\nu},$$

and the parameter vector $\boldsymbol{\nu}$ is updated as follows

$$\boldsymbol{\nu}_{k+1} = F'(\boldsymbol{\nu}_k),$$

where $F' = \Phi^\dagger \circ F \circ \Phi$. Define error ϵ_{U} as

$$\epsilon_{\mathrm{U}}(\hat{J}_1, \hat{\boldsymbol{\pi}}) = \min_{\boldsymbol{\nu}} \|\bar{V}^*(\cdot, \hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\cdot, \hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu})\|_\infty.$$

Lemma 4.5.1. *For any vectors J and J',*

$$\|\Phi\Phi^\dagger(J) - \Phi\Phi^\dagger(J')\|_\infty \leq \frac{\gamma'}{\gamma}\|J - J'\|_\infty < \frac{\gamma'}{\gamma^T}\|J - J'\|_\infty.$$

The proof of Lemma 4.5.1 is in Appendix C.2.5.1.

Algorithm 5: Approximate Value Iteration for the Nominal State Approximation

Input: The nominal slow state x^*; pre-selected states $\tilde{\mathcal{S}} = \{s_1, s_2, \ldots, s_M\}$; initial weights $\boldsymbol{\omega}_T^{x^*}$, $\boldsymbol{\nu}_0$.

Output: Weights $\boldsymbol{\omega}_t$, $\boldsymbol{\nu}_K$.

for $t = T-1, T-2, \ldots, 1$ **do**
  for $s = (x^*, y) \in \tilde{\mathcal{S}}$ **do**
    Observe value $J_t(x^*, y) = \max_{a \in \mathcal{A}} \mathbf{E}\big[g(x^*) + h(y, a) + \gamma \boldsymbol{\phi}(s')^\intercal \boldsymbol{\omega}_{t+1}^{x^*}\big]$.
  end
  Update $\boldsymbol{\omega}_t^{x^*}$: $\boldsymbol{\omega}_t^{x^*} = \Phi^\dagger J_t$, where J_t is as follows: for $s_m = i \in \tilde{\mathcal{S}}$, the ith entry equals to $J_t(s_m)$; the other entries are zero.
  for $s = (x, y) \in \mathcal{S}$ **do**
    The policy $\hat{\pi}_t(x, y) = \arg\max_a \mathbf{E}[g(x^*) + h(y, a) + \gamma \boldsymbol{\phi}(s')^\intercal \boldsymbol{\omega}_{t+1}^{x^*}]$.
  end
end
for $k = 1, 2, \ldots, K$ **do**
  for $s \in \tilde{\mathcal{S}}$ **do**
    Observe value: $V(s, J_1, \hat{\boldsymbol{\pi}}) = \max_a \mathbf{E}\big(\bar{R}(s, a, \bar{J}_1) + \gamma^T \phi(s')^\intercal \boldsymbol{\nu}_{k-1}\big)$.
    Update $\boldsymbol{\nu}_k$: $\boldsymbol{\nu}_k = \Phi^\dagger V$, where V is as follows: for $s_m = i \in \tilde{\mathcal{S}}$, the ith entry equals to $V(s_m, J_1, \hat{\boldsymbol{\pi}})$; the other entries are zero.
  end
end

4.5.2 Convergence of the Lower Level

Lemma 4.5.2. *If* $(\boldsymbol{\omega}_1^*, \boldsymbol{\omega}_2^*, \ldots, \boldsymbol{\omega}_T^*)$ *is the optimal solution of* $\boldsymbol{\omega}_t = \bar{H}'\boldsymbol{\omega}_{t+1}$*, then*

$$\|\bar{J}_t^* - \hat{J}_t(\boldsymbol{\omega}_t^*)\|_\infty \leq \epsilon_{\mathrm{L}}\big(1 + \frac{\gamma+1}{\gamma}\sum_{i=1}^{T-t}(\gamma')^i\big) < \epsilon_{\mathrm{L}}\big(1 + \frac{\gamma+1}{\gamma}(T-t)\big).$$

The proof of Lemma 4.5.2 is in Appendix C.2.5.2.

Lemma 4.5.3. *If* $\|\bar{J}_t^* - \hat{J}_t(\boldsymbol{\omega}_t^*)\|_\infty \leq \epsilon_t^{\mathrm{bias}} \leq \epsilon^{\mathrm{bias}}$ *for all* t*, then*

$$\|\bar{J}_t^* - \bar{J}_t^{\hat{\boldsymbol{\pi}}^*}\|_\infty \leq 2\gamma \sum_{\tau=t+1}^{T-1} \epsilon_\tau^{\mathrm{bias}} \leq 2\gamma\epsilon^{\mathrm{bias}}(T-t-1),$$

where $\hat{\boldsymbol{\pi}}^*$ *is the greedy policy w.r.t.* $\hat{J}_t(\boldsymbol{\omega}_t^*)$*.*

The proof of Lemma 4.5.3 is in Appendix C.2.5.3.

Theorem 4.5.1. *Let* $\hat{\boldsymbol{\pi}}^*$ *be the greedy policy w.r.t.* $\hat{J}_t(\boldsymbol{\omega}_t^*)$*. The error bound* $\|\bar{J}_1^* - \bar{J}_1^{\hat{\boldsymbol{\pi}}^*}\|_\infty$ *is*

$$\|\bar{J}_1^* - \bar{J}_1^{\hat{\boldsymbol{\pi}}^*}\|_\infty \leq 2\gamma\epsilon_{\mathrm{L}}(T-2)\big(1 + \frac{\gamma+1}{\gamma}(T-1)\big).$$

4.5.3 Convergence of the Upper Level

Lemma 4.5.4. *The mapping* $F' = \Phi^\dagger \circ F \circ \Phi$ *is a contraction with coefficient* γ'*, w.r.t. a norm* $\|\cdot\|$ *on* $\mathbb{R}^M$ *defined by* $\|\boldsymbol{\nu}\| = \|\Phi\boldsymbol{\nu}\|_\infty$*, i.e.,*

$$\|F'(\boldsymbol{\nu}) - F'(\boldsymbol{\nu}')\| \leq \gamma'\|\boldsymbol{\nu} - \boldsymbol{\nu}'\|.$$

The proof of Lemma 4.5.4 is in Appendix C.2.5.4.

Lemma 4.5.5. *If* $\boldsymbol{\nu}^*$ *is the fixed point of* F'*, i.e.,* $\boldsymbol{\nu}^* = F'\boldsymbol{\nu}^*$*, then*

$$\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \leq \frac{\gamma^T + \gamma'}{\gamma^T(1-\gamma')}\epsilon_{\mathrm{U}}(\hat{J}_1, \hat{\boldsymbol{\pi}}).$$

The proof of Lemma 4.5.5 is in Appendix C.2.5.5.

Lemma 4.5.6. *If* $\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \leq \epsilon^{\mathrm{bias}}$*, then*

$$\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \leq \frac{2\gamma^T}{1-\gamma^T}\epsilon^{\mathrm{bias}},$$

where $\hat{\mu}^*$ *is the greedy policy w.r.t.* $\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)$*.*

Lemma 4.5.7. *The optimal value of the lower level of nominal state approximation $\bar{J}_1^*$ is Lipschitz w.r.t. the state, i.e.*

$$\begin{aligned}\|\bar{J}_1^*(x,y)-\bar{J}_1^*(\bar{x},\bar{y})\|_\infty &\le L_r'\sum_{i=0}^{T-2}\gamma^i(\|x-\tilde{x}\|_2+\|y-\tilde{y}\|_2)+2d_yL_r'\sum_{i=1}^{T-2}i\gamma^i\\ &\le \frac{1-\gamma^T}{1-\gamma}L_r'(\|x-\tilde{x}\|_2+\|y-\tilde{y}\|_2)+\Delta_J,\end{aligned}$$

where $\Delta_J=\frac{2d_yL_r'}{1-\gamma}\big(\frac{\gamma}{1-\gamma}-\frac{\gamma}{1-\gamma}\gamma^T-T\gamma^T\big)$.

The proof of Lemma 4.5.7 is a mimic of the analysis in Lemmas C.1.2, C.1.3 and C.1.4.

Theorem 4.5.2. *Let $\hat{\mu}^*$ be the greedy policy w.r.t. $\hat{V}(\hat{J}_1,\hat{\boldsymbol{\pi}},\boldsymbol{\nu}^*)$. The error bound is*

$$\begin{aligned}&\|\bar{V}^*(\bar{J}_1^*,\boldsymbol{\pi}^*)-\bar{V}^{\hat{\mu}^*}(\hat{J}_1,\hat{\boldsymbol{\pi}})\|_\infty\\ &\le 2\gamma^T\Big(G(\gamma,\alpha)L_r'd_y+\frac{1+\alpha}{1-\gamma}L_r'd_yT+H(\gamma,\alpha)L_r'd_y\gamma^T-\frac{2+\alpha}{1-\gamma}L_r'd_yT\gamma^T\Big)\\ &\quad+\epsilon_{\mathrm{L}}-\frac{\gamma+1}{\gamma}+\frac{\gamma+1}{\gamma}\epsilon_{\mathrm{L}}T+\frac{2(\gamma^T+\gamma')}{(1-\gamma^T)(1-\gamma')}\epsilon_{\mathrm{U}}(\hat{J}_1,\hat{\boldsymbol{\pi}}).\end{aligned}$$

Proof. Let $\hat{s}=\arg\max_s|\bar{V}^*(s,\bar{J}_1^*,\bar{\boldsymbol{\pi}}^*)-\bar{V}^*(s,\hat{J}_1\hat{\boldsymbol{\pi}})|$. We have

$$\|\bar{V}^*(\bar{J}_1^*,\bar{\boldsymbol{\pi}}^*)-\bar{V}^*(\hat{J}_1,\hat{\boldsymbol{\pi}})\|_\infty\le|V^*(\hat{s},\bar{J}_1^*,\bar{\boldsymbol{\pi}}^*)-V^*(\hat{s},\hat{J}_1,\hat{\boldsymbol{\pi}})|$$

By Lemmas 4.5.5 and 4.5.6, we have

$$\|\bar{V}^*(\hat{J}_1,\hat{\boldsymbol{\pi}})-\bar{V}^{\hat{\mu}^*}(\hat{J}_1,\hat{\boldsymbol{\pi}})\|_\infty\le\frac{2(\gamma^T+\gamma')}{(1-\gamma^T)(1-\gamma')}\epsilon_{\mathrm{U}}(\hat{J}_1,\hat{\boldsymbol{\pi}}).$$

We next consider the distance between $\bar{V}^*(\bar{J}_1^*,\boldsymbol{\pi}^*)$ and $\bar{V}^*(\hat{J}_1,\hat{\boldsymbol{\pi}})$. They are the optimal values of two MDPs with different reward functions and transition functions. According to (4.23), the distance between the two reward functions is $|\bar{J}_1^*(x,y)-\hat{J}_1(x,y)|$, which is bounded by $\epsilon_{\mathrm{L}}\big(1+\frac{\gamma+1}{\gamma}(T-1)\big)$ according to Lemma 4.5.2. The distance between the two transition functions is bounded by $d=2(\alpha+1)d_y(T-1)$ according to the analysis in Theorem 4.3.1. Therefore, according to Lemmas 4.3.1 and 4.5.7, $d=2(\alpha+1)d_y(T-1)$, $\Delta_J=\frac{2d_yL_r'}{1-\gamma}\big(\frac{\gamma}{1-\gamma}-\frac{\gamma}{1-\gamma}\gamma^T-T\gamma^T\big)$, and

$$\begin{aligned}&\|\bar{V}^*(\bar{J}_1^*,\boldsymbol{\pi}^*)-\bar{V}^*(\hat{J}_1,\hat{\boldsymbol{\pi}})\|_\infty\\ &\le\frac{1}{1-\gamma^T}\big(\epsilon_{\mathrm{L}}\big(1+\frac{\gamma+1}{\gamma}(T-1)\big)+\gamma^T(\frac{1-\gamma^T}{1-\gamma}L_r'd+\Delta_J)\big)\end{aligned}$$

$$
\begin{aligned}
&= \epsilon_{\mathrm{L}} - \frac{\gamma+1}{\gamma} + \frac{\gamma+1}{\gamma}\epsilon_{\mathrm{L}} T \\
&\quad + 2\gamma^T \Big(G(\gamma,\alpha) L_r' d_y + \frac{1+\alpha}{1-\gamma} L_r' d_y T + H(\gamma,\alpha) L_r' d_y \gamma^T - \frac{2+\alpha}{1-\gamma} L_r' d_y T \gamma^T \Big),
\end{aligned}
$$

where $G(\gamma,\alpha) = \frac{\gamma-(1-\gamma)(1+\alpha)}{(1-\gamma)^2}$, $H(\gamma,\alpha) = \frac{(1-\gamma)(1+\alpha)-\gamma}{(1-\gamma)^2}$. Therefore,

$$
\begin{aligned}
&\|\bar{V}^*(\bar{J}_1^*, \boldsymbol{\pi}^*) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&\le \|\bar{V}^*(\bar{J}_1^*, \boldsymbol{\pi}^*) - \bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty + \|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&\le 2\gamma^T \Big(G(\gamma,\alpha) L_r' d_y + \frac{1+\alpha}{1-\gamma} L_r' d_y T + H(\gamma,\alpha) L_r' d_y \gamma^T - \frac{2+\alpha}{1-\gamma} L_r' d_y T \gamma^T \Big) \\
&\quad + \epsilon_{\mathrm{L}} - \frac{\gamma+1}{\gamma} + \frac{\gamma+1}{\gamma}\epsilon_{\mathrm{L}} T + \frac{2(\gamma^T+\gamma')}{(1-\gamma^T)(1-\gamma')}\epsilon_{\mathrm{U}}(\hat{J}_1, \hat{\boldsymbol{\pi}}).
\end{aligned}
$$

□

4.6 Numerical Experiment

In this section, we apply our algorithms to a few MDP problems, including a machine maintenance problem, a dynamic serve allocation for a multi-class queuing model, a demand response problem, and a multi-product joint procurement and pricing problem. Specifically, we apply VI to the first two problems, and AVI to the last two problem. For AVI, we use Gaussian radial basis function. We respectively call the frozen-state algorithm and nominal state algorithm "FSVI" and "Nominal FSVI" when VI is used, and "FSAVI" and "Nominal FSAVI" when AVI is used. In the procurement and pricing problem, the length of the state-frozen periods is $T = 5$, in other problems, the length is $T = 10$. In the policy evaluation, we sample the cumulative reward over $10\,T$ periods of all the initial states. Each evaluation is an average over 10 runs.

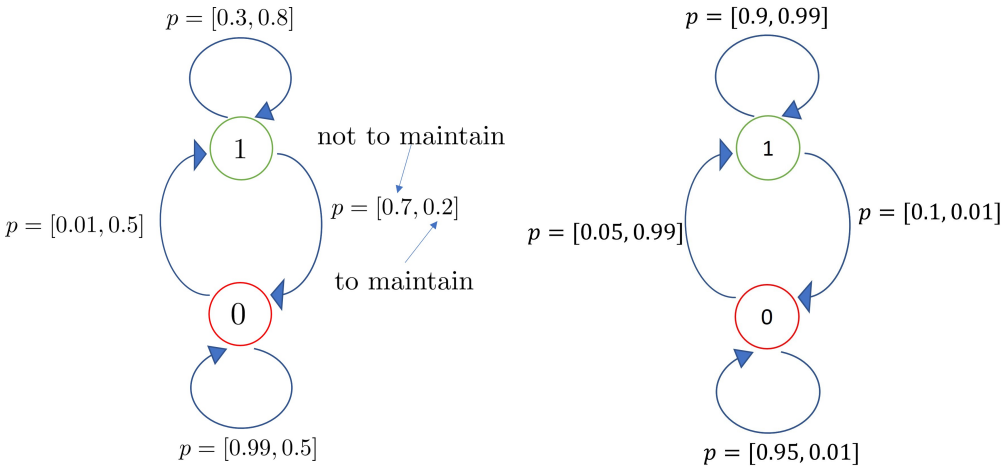
To illustrate the convergence and the computation efficiency of our algorithms, we compare against a few baseline algorithms. The first baseline is the base model (4.5), denoted as "base VI" when using VI and "base AVI" when using AVI. Another popular baseline is the Q-learning algorithm (QL) [40, 170] and Deep Q network (DQN) [231, 232].

We also propose Slow-agnostic model that ignores the slow state, i.e., the value function is

$$U(y) = \max_{a \in \mathcal{A}} \mathbf{E}\big[r(\cdot, y, a) + \gamma U(y')\big], \tag{4.31}$$

where the expectation is over the slow state x and the exogenous noise w.

4.6.1 Machine Maintenance



(a) The system condition is "bad"

(b) The system condition is "good"

Figure 25: Transition matrices in different system conditions

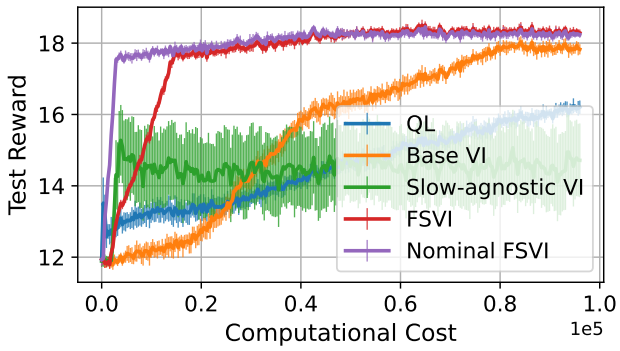


Figure 26: Performance of VI for the maintenance problem

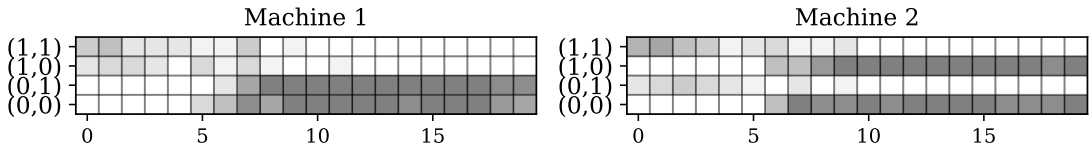
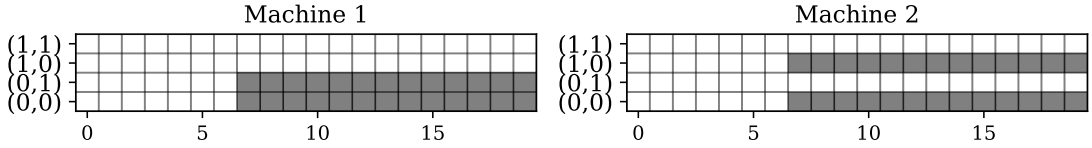
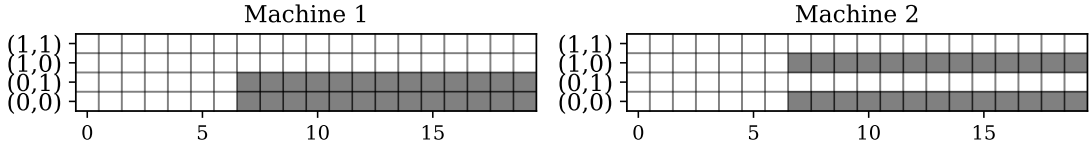


Figure 27: Policy for the maintenance problem: Base VI



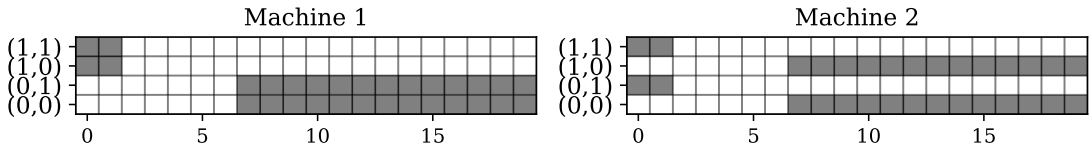
(a) Upper level



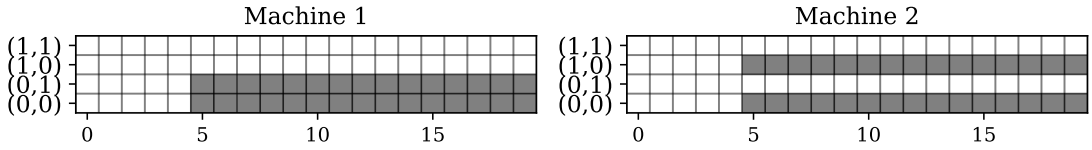
(b) Lower level $t = 5$

Figure 28: Policy for the maintenance problem: FSVI

Consider a machine maintenance problem [233, 234, 235] with $q = 2$ machines, each machine i has two states: operating ($y_i = 1$) and not operating ($y_i = 0$). At the end of each period, the operator decides which machines to maintain. The state of machine i in the next period is influenced by three factors, its current state y_i, whether it is maintained a_i, and the condition of the system x. The machines have higher probability to stop operating when the system condition is "bad" than when the system condition is "good", as shown in Figure 25. In general, if the machine is operating and is maintained at the end of the period, and the system is in good condition, it will have higher probability to be operating in the next period.



(a) Upper level



(b) Lower level $t = 5$

Figure 29: Policy for the maintenance problem: Nominal FSVI

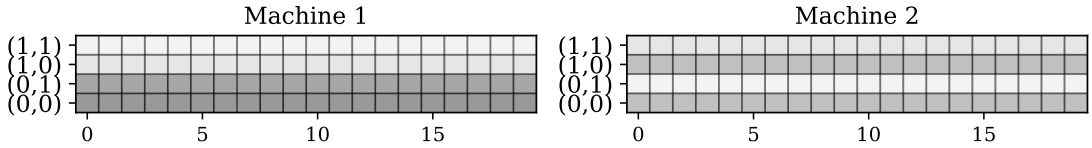


Figure 30: Policy for the maintenance problem: Slow-agnostic VI

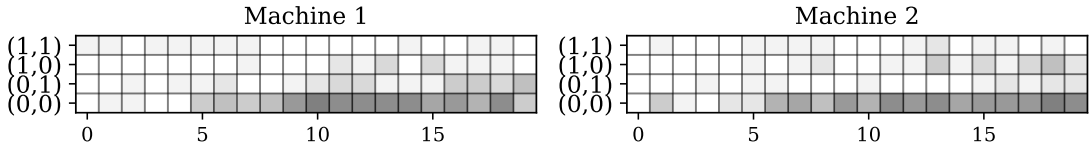


Figure 31: Policy for the maintenance problem: QL

The system condition x is the slow state, while the operating status of the machines $y_t = (y_{t,1}, y_{t,2})$ is the fast state. We consider 25 values of x ($x \in \{0, 1, \ldots, 24\}$). Each x transits to $x + 2$, $x + 1$, $x - 1$ and $x - 2$ with probabilities 0.05, 0.15, 0.15, and 0.05

respectively. At the end of each period, the operator makes a decision on the machines to be maintained at the end of the period, $a_t = (a_{t,1}, a_{t,2})$. The immediate reward function is $r(x_t, y_t, a_t) = 2\sum_{i=1}^{q} y_{t,i} - \sum_i a_{t,i}$.

Figure 26 shows the performance of the algorithms as a function of the computational cost. Each evaluation is the cumulative reward over 100 periods. Each point in the plot is an average over 30 runs. The plot shows that except for Slow-agnostic VI, all other algorithms converge. What's more, the proposed algorithms, especially Nominal FSVI, converge much faster than the baseline algorithms.

To understand the difference in the performance of the algorithms, Figures 27 to 29 show the policies at the last iteration of the algorithms. In each of the figures, the x-axis represent the exogenous information, the y-axis represent the status (y_1, y_2) of the two machines. The left plot shows whether to maintenance machine 1, the right plot shows whether to maintenance machine 2. A grey square means maintaining the machine, a white square means the opposite. The shade of the grey color represents the frequency of taking the maintenance action over 10 runs. We have the following observations.

1. Among the policies of the five algorithms, Base VI, FSVI and Nominal FSVI are similar to each other in that: *(i)* the policies to a machine are stationary and the same when the machine failed, that is to maintain it; and *(ii)* when the machine did not fail and the system status is bad (the x-axis value is small), the policies of FSVI and Nominal FSVI are the same, that is to maintain when $x < 5$ no matter the machine failed or not; while the policies from the 10 runs of Base VI are unstable, but there is a higher probability to maintain as the system status gets worse.
2. The policy of Slow-agnostic VI is not influenced by the system status x, and shade reflects that the policies are not stationary. The policies from the 10 runs of QL are unstable and the influence of x and y is not clear.

4.6.2 Dynamic Service Allocation for a Multi-class Queuing Model

In the service allocation problem [199], we consider a single server and two classes of customers. the arrival rates and service rates for the two classes are $\mu_1 = \mu_2 = 0.2$ and

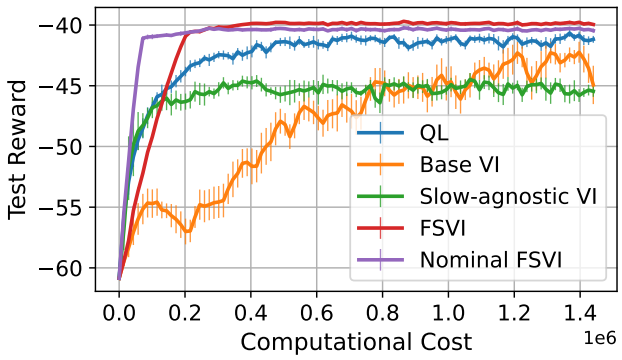


Figure 32: Performance of VI for the queuing problem

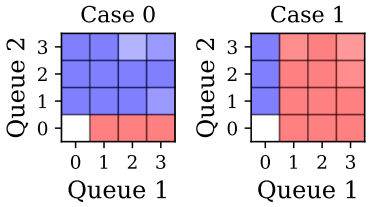
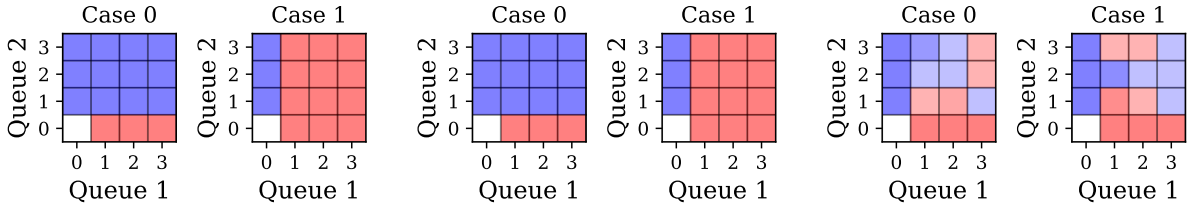


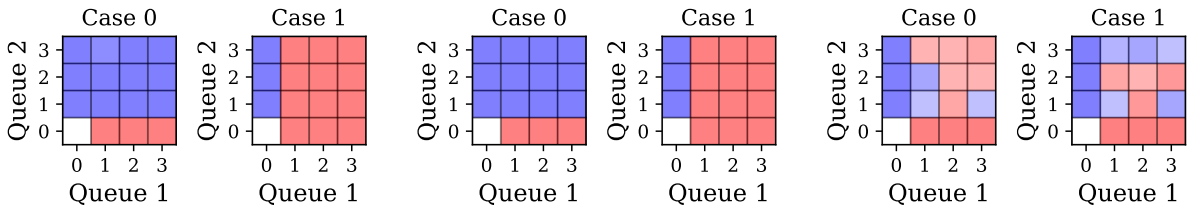
Figure 33: Policy for the queuing problem: Base VI



(a) Upper level (b) Lower level $t = 8$ (c) Lower level $t = 9$

Figure 34: Policy for the queuing problem: FSVI

$\lambda_1 = \lambda_2 = 0.3$. The problem is converted to a discrete-time model through "uniformization" [228]. The capacities of the queues are $Q_1 = 3$ and $Q_2 = 3$. Denote $y_{t,1}$ and $y_{t,2}$ the number of customers in the queues, $z_t \in \{0, 1, 2\}$ the class of customer that is currently served (0



(a) Upper level (b) Lower level $t = 8$ (c) Lower level $t = 9$

Figure 35: Policy for the queuing problem: Nominal FSVI

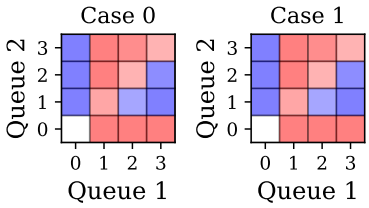


Figure 36: Policy for the queuing problem: Slow-agnostic VI

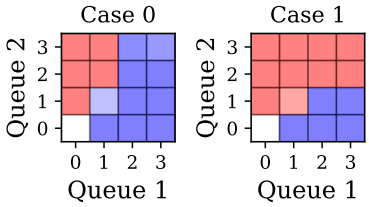


Figure 37: Policy for the queuing problem: QL

means no customer being served). The transition of the length of the queue j is a function of the current number of customers $y_{t,j}$ and the customer that is currently served z_t.

1. With probability $\mu_{t,j}$, a class j customer arrives, $y_{t+1,j} = \min(y_{t,j} + 1, Q_j)$.
2. With probability λ_{z_t}, the server completed serving the current customer, $y_{t+1,j} = y_{t,z_t} - 1$.
3. With probability $1 - \lambda_i - \sum_j \mu_{t,j}$, no event happens, $y_{t+1,j} = y_{t,j}$.

When the server completed serving the current customer, the decision maker decides the class of customer to serve, denoted as $a_t \in \{0, 1, 2\}$, where 0 represents when there is no decision to make. Customers waiting in the queues incurs a cost. The unit cost of queue j is stochastic, denoted as $x_{t,j}$. Consider 6 values of the unit cost $x_{t,j}$, which is uniformly distributed in $[0.01, 0.2]$. The next unit cost $x_{t+1,j}$ follows a truncated discrete normal distribution centered at $x_{t,j}$ with standard deviation 0.01. The immediate reward function is $r(x_{t,1}, x_{t,2}, y_{t,1}, y_{t_2}) = -\sum_{j=1}^{2} x_{t,j} y_{t,j}$.

Figure 32 shows the performance of the algorithms. Nominal FSVI converges the fastest. FSVI and QL also converge fast. Base VI, although converges slowly, improves as the computational cost increases. On the other hand, the performance of Slow-agnostic VI is good at the beginning, but does not improve as the computational cost increases.

Figures 33 to 35 show the policies at the last iteration of the algorithms. We show policies for two different unit cost: case 1, the unit cost of queue 1 is lower than the unit cost of queue 2; and case 2, the unit cost of queue 1 is higher than the unit cost of queue 2. In each plot, the x- and y-axis represents the length of the first and the second queues respectively. The red color denotes serving customers in queue 1, and the blue color denotes serving customers in queue 2. The shade of the colors represents the frequency of taking the maintenance action over 10 runs. When the class to be served at a state is different in different runs, we show the color of the class that is the majority in all the runs. We have the following observations:

1. The policies of Base VI, the upper levels and the first 8 periods of the lower levels of FSVI and Nominal FSVI are similar. That is to always serve customers in the queue with a higher unit cost as long as the queue is nonempty. As for the last period of the lower levels of FSVI and Nominal FSVI, when one of the queues is empty, the policies are to serve customers in the nonempty queue; when both queues are nonempty, the policies are stochastic. This is because no matter serving queue 1 or queue 2, the lengths of the queues are not shortened, which means the immediate rewards are the same. Considering this is the last period of the two lower-level problems, the decision does not impact the following periods.

2. The policy of Slow-agnostic VI is not impacted by the slow state $(x_{t,1}, x_{t,2})$. Given any slow state, the policy is to serve the nonempty queue when the other queue is empty, and to serve the shorter queue when both queues are nonempty.
3. As for the policy of QL, the server serves the queue with higher unit cost when both queues have more than 1 customers. When there is 0 or 1 customer in the higher cost queue and more than 1 customer in the lower cost queue, the server serves the longer queue (the lower cost queue).

4.6.3 Energy Demand Response

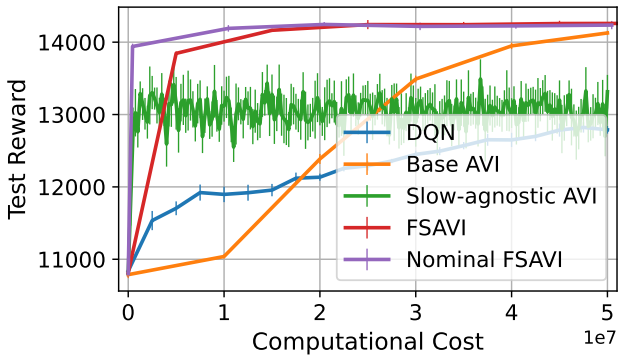
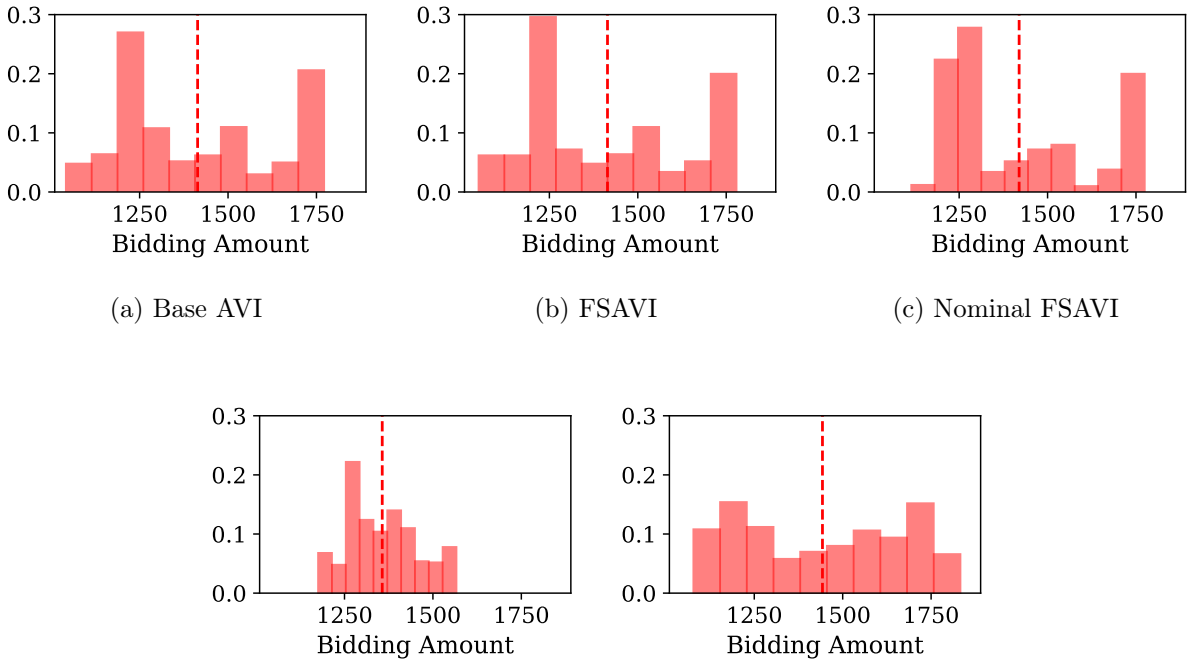


Figure 38: AVI for the demand response problem

In the demand response problem in the energy market [208], the energy consumption reduction is dependent on the unit compensation. The state is composed of the DA price x_t, which follows discretized Ornstein Uhlenbeck process $x_{t+1} - x_t = c_1(c_2 - x_t) + \epsilon_t$, and the real-time (RT) shortage price and the RT overage price, which are represented as fractions of the the DA price, i.e., the shortage price is $p_t^- = x_t y_t^-$, the overage price is $p_t^+ = x_t y_t^+$. Assume (y_t^-, y_t^+) satisfies that $y_t^+ < 1 < y_t^-$. At the beginning of each period t, the aggregator commits an amount of a_t to a forward contract for energy in the DA market at the DA price. The aggregator must deliver the amount of a_t energy in the RT market. If the delivered energy falls short of the forward contract, the aggregator must purchase the shortfall at the RT shortage price p_t^-. If the delivered energy exceeds the forward contract, the aggregator has to sell the excess amount at the RT overage price p_t^+.



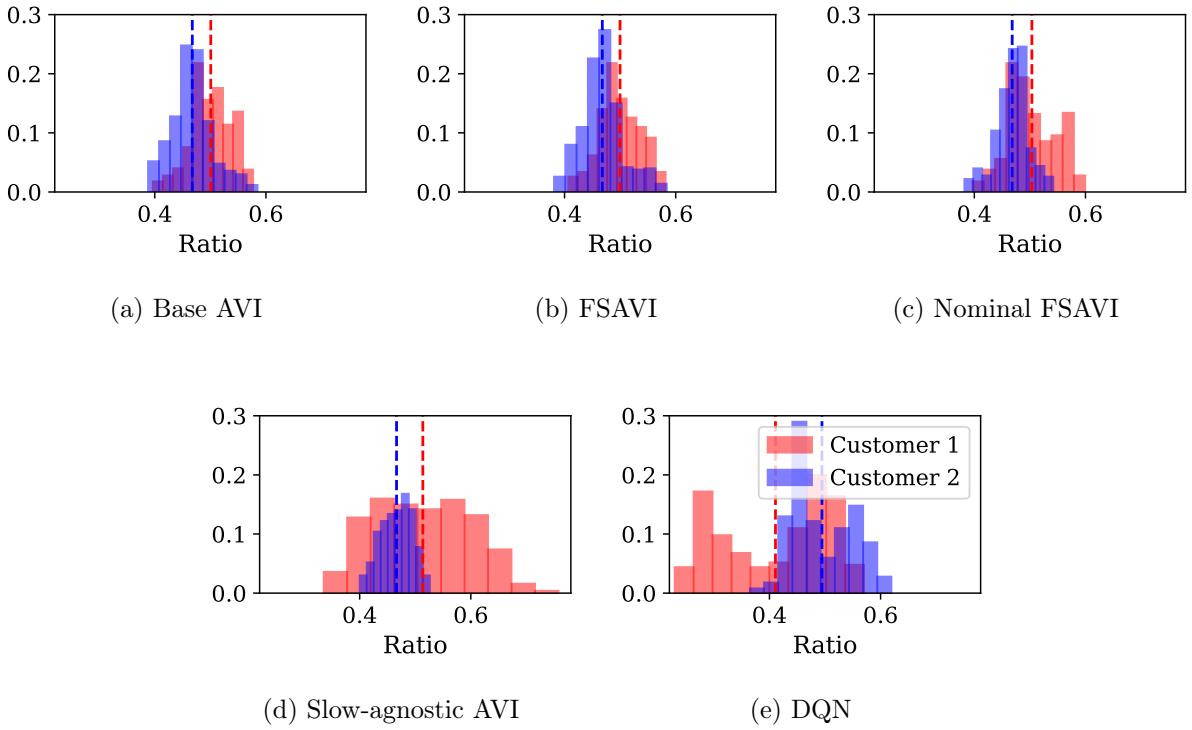
(a) Base AVI (b) FSAVI (c) Nominal FSAVI

(d) Slow-agnostic AVI (e) DQN

Figure 39: The bidding amount of the algorithms

To meet the forward contract, the aggregator must elicit a reduction in demand from its customers. We consider two customers in this section, i.e., $m \in \{1, 2\}$. The reduction in demand is call demand response, which is a function of the compensation provided by the aggregator. Represent the compensation as fractions of the the DA price, i.e., $q_{t,m} = x_t \alpha_{t,m}$, where $\alpha_{t,m} < 1$. The demand response is modeled as $d_m(x_t, \alpha_{t,m}) = b_{m,1} + b_{m,2} x_t \alpha_{t,m} + \sigma_{t,m}$, where $\sigma_{t,m}$ is the noise. Let $b_{1,2} > b_{2,2}$, $b_{1,1} < b_{2,1}$, and let the maximum expected demand response of customer 1 be bigger than that of customer. The immediate reward function is

$$r(x_t, y_t^+, y_t^-, a_t, \boldsymbol{\alpha}_t) = x_t a_t + \mathbf{E}\big[-\sum_m x_t \alpha_{t,m} d_m(x_t, \alpha_{t,m}) + x_t y_t^+ \big(\sum_m d_m(x_t, \alpha_{t,m}) - a_t\big)^+ - x_t y_t^- \big(a_t - \sum_m d_m(x_t, \alpha_{t,m})\big)^+\big],$$



(a) Base AVI (b) FSAVI (c) Nominal FSAVI

(d) Slow-agnostic AVI (e) DQN

Figure 40: The proportion of the bidding amount satisfied by each customer

where $\boldsymbol{\alpha}_t = (\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,m})$. The reward function can be approximated as $h(x_t)\, g(x_t^*, y_t^+, y_t^-, a_t, \boldsymbol{\alpha}_t)$, where

$$g(x_t^*, y_t^+, y_t^-, a_t, \boldsymbol{\alpha}_t) = a_t + \mathbf{E}\big[-\sum_m \alpha_{t,m} d_m(x_t^*, \alpha_{t,m}) + y_t^+ \big(\sum_m d_m(x_t^*, \alpha_{t,m}) - a_t\big)^+ - y_t^- \big(a_t - \sum_m d_m(x_t^*, \alpha_{t,m})\big)^+\big].$$

Figure 38 shows the performance of the algorithms as a function of the computational cost. The proposed algorithms, especially Nominal FSAVI, converge fast. The performance of Base AVI and DQN improves slowly. The performance of Slow-agnostic AVI is good comparing to other algorithms at the beginning, but oscillates at a small value and does not converge.

To understand the policies of the algorithms, Figure 39 shows the histograms of the cumulative bidding amount over 100 periods starting from 1000 random states. Figures 40 shows histograms of the proportions of the bidding amount that is satisfied by the two customers. We have the following observations:

1. The bidding amount of Base AVI, FSAVI and Nominal FSAVI are similar. Their mean values are all about 1400. The mean bidding amount of Slow-agnostic AVI is smaller than 1400, and the amount of DQN is higher than 1400.
2. The shapes of the histograms of Base AVI, FSAVI and Nominal FSAVI are similar. The average proportion of customer 1 is a higher than that of customer 2, the difference is small. The histograms of customer 1 of Slow-agnostic AVI and DQN are wider than the other three models. DQN is the only algorithm that the average proportion of customer 1 is smaller than that of customer 2.

4.6.4 Multi-product Joint Procurement and Pricing

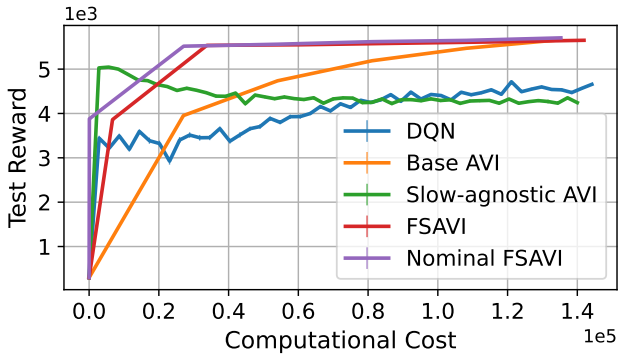
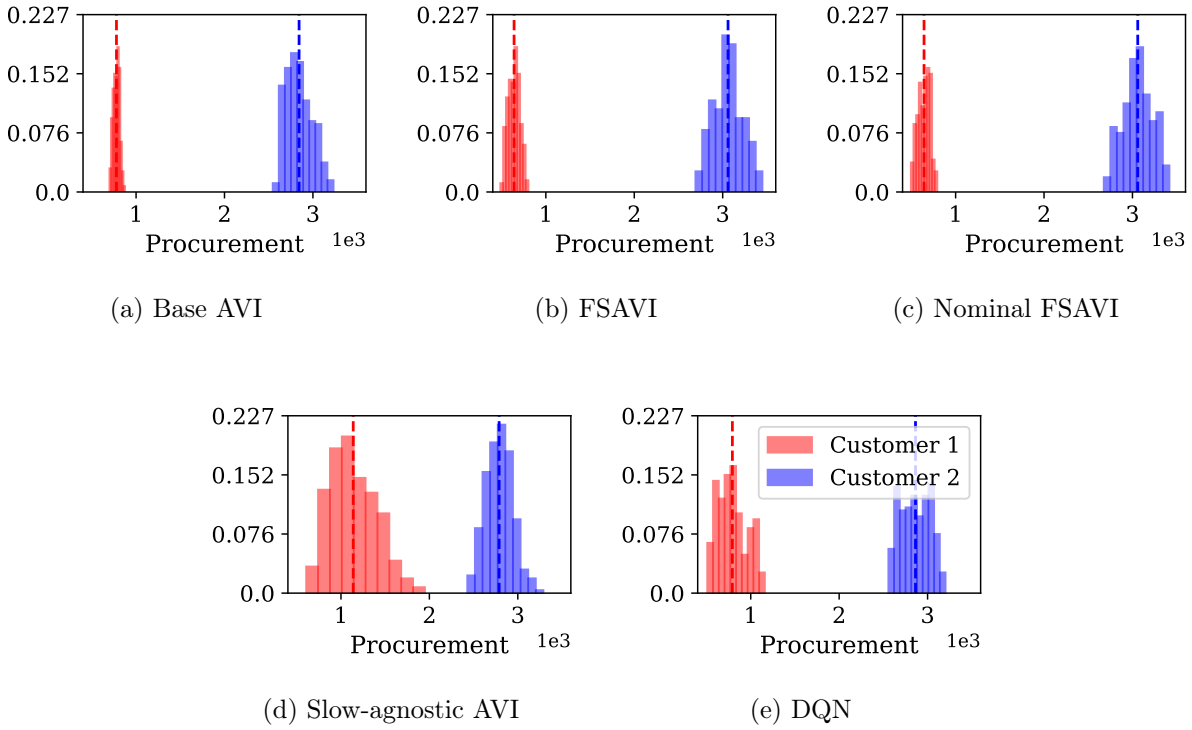


Figure 41: AVI for the joint procurement and pricing problem

Finally, we apply the algorithms to an MDP with endogenous slow state. We consider the multi-product joint procurement and pricing problem with price-dependent demand [210, 211, 212] with two products. Assume that the demand for product 1 is high, and the demand for product 2 is relatively low. The state is the procurement costs and the inventory levels at the beginning of period t, denoted by $(c_{t,1}, y_{t,1}, c_{t,2}, y_{t,2})$, where $c_{t,1}, c_{t,2} \in$



(a) Base AVI (b) FSAVI (c) Nominal FSAVI

(d) Slow-agnostic AVI (e) DQN

Figure 42: The procurement quantities of the algorithms

$\{1, 1.05, \ldots, 2\}$, $y_{t,1}, y_{t,2} \in \{0, 10, \ldots, 100\}$. The cost $(c_{t,1}, c_{t,2})$ forms the exogenous Markov process: $(c_{t+1,1}, c_{t+2,2}) = f_C(c_{t,1}, c_{t,2}, w_t)$. Denote the stochastic demand $d_i(p_{t,i}) = \alpha_i - \beta_i p_{t,i}$, where $\alpha_1 = 40$, $\beta_1 = 6.7$, $\alpha_2 = 160$, and $\beta_2 = 26.7$. The transition function of the inventory level is $y_{t+1,i} = y_{t,i} + a_{t,i} - D_{t,i}$, where $D_{t,i}$ is a realization of the demand $d_i(p_{t,i})$. Given the state, the decision maker decides the procurement quantities and selling prices $\boldsymbol{a}_t = (a_{t,1}, p_{t,1}, a_{t,2}, p_{t,2})$, where $a_{t,1} \in \{0, 20, \ldots, 100\}$, $p_{t,1} \in \{3, 4.5\}$. The immediate reward function is

$$r(\boldsymbol{c}_t, \boldsymbol{y}_t, \boldsymbol{a}_t) = \mathbf{E}_t\Big[\sum_{i=1,2} p_{t,i} d_i(p_{t,i}) - c_{t,i} a_{t,i} \\ - h_i^+ (y_{t,i} + a_{t,i} - d_i(p_{t,i}))^+ - h_i^- (d_i(p_{t,i}) - y_{t,i} - a_{t,i})^+\Big],$$

where $h^+ = 0$ is the inventory cost, $h^- = 3$ is the lost-sales cost. Consider the large difference in the expected demand of the two products, we treat $(c_{t,1}, y_{t,1})$ as the slow state. That is,

the decision maker checks and updates the inventory level of product 1 every T periods, and makes the procurement and pricing decisions every T periods. During the lower level, the procurement quantity of product 1 is 0 (no procurement), and its price is frozen.

Figure 41 shows the performance of the algorithms as a function of the computational cost. FSAVI and Nominal FSAVI converge fast. The performance of Base AVI improves slowly. The performance of Slow-agnostic AVI never improves as the computational cost increases.

To illustrate the policies of the algorithms, Figure 42 shows the histograms of the total procurement quantities over 50 periods starting from 1000 random states. We have the following observations:

1. The procurement quantities of product 1 is smaller than the procurement quantities of product 2 in every algorithm. This coincides with the difference in the expected demand of the two products.

2. The average procurement quantity of product 1 in Slow-agnostic AVI is 1139, much higher than the values of DQN (792), Base AVI (778), FSAVI (640) and Nominal FSAVI (643). In the latter four algorithms, the distributions of the procurement quantity of product 1 are different, Base AVI has the smallest variance, DQN has the largest variance. That means starting from different states, the cumulative procurement decisions are more stable in the Base AVI than the other algorithms.

3. The average procurement quantities of product 2 in the algorithms are similar, which are 2789 in Slow-agnostic AVI, 2864 in DQN, 2844 in Base AVI, 3061 in FSAVI, and 3060 in Nominal FSAVI. Hierarchical algorithms leads to higher procurement quantity than the base model. This is a result of freezing the slow states, i.e., the cost and inventory level of product 1, which influences the transition function of the cost of product 2, which in turn impacts the pricing decision, the demand and the procurement decision of product 2.

4.6.5 Discussion

We offer some important observations and takeaways from figures of the VI (or AVI) performance and the policies of the algorithms for all the studied problems.

1. The proposed models converge faster than the baseline algorithms, and the nominal state model converges even faster than the frozen-state model. The fast convergence comes from two parts: one is freezing the slow state in the lower level, which reduce the computational cost of estimating the expected value of next state; the other is utilizing the nominal slow states in the lower level, which reduce the number of lower level MDPs.
2. The policies of the proposed hierarchical approximation algorithms have similar performance to the base model.
3. It is risky to ignore the slow state, it leads to convergence to a worse value.

4.7 Conclusions

In this chapter, we study the MDPs with fast-slow structure, in which the slow state is either exogenous or endogenous with a separability assumption on the action space. We respectively propose hierarchical value iteration algorithms based on the idea of freezing the slow states, solving a set of finite-horizon MDPs, and applying value iteration to an auxiliary MDP that transitions on a slower timescale. We show the errors caused by freezing slow states and adopting nominal slow states in the approximation. We then discuss the choice of T, the length of the state-frozen periods, to achieve the desired error level. We propose VI and AVI for the hierarchical approximation problems. The proposed algorithms and the baseline algorithms are applied to four problems to show the performance and the difference in the learned policies.

5.0 Conclusions and Future Work

In this thesis, we exploit structural properties to efficiently solve the MDPs. We first study the problems with concave value function and basestock policy. The inspiration is the PODs operated by first responders or first receivers for distributing critical medical supplies during emergency situations. We develop a hierarchical, finite-horizon MDP model, where the upper-level MDP is an inventory model that controls the lower-level dispensing problem. The MDP features basestock-like structure in a discrete state setting and discretely-concave value functions. Based on the properties, we propose a new actor-critic algorithm that exploits the structural properties of the MDP. In the algorithm, both policy and value function approximations are tracked and the structural properties are utilized to improve the empirical convergence rate. We also apply an aggregation-based version of the algorithm to a case study for the problem of dispensing naloxone. We showed how an aggregation-based version of S-AC along with k-means clustering can be used to handle the multi-dimensional continuous features used in the case study. There are also other possible extensions to S-AC that can make it more scalable to high-dimensional problems. For example, shape-constrained deep neural networks [92] [93] can handle both monotonicity and concavity via penalization of derivatives during training. In principle, our S-AC algorithm could be extended to use techniques like these, but the same core principles of S-AC would remain intact. We leave these investigations to future work.

Considering the challenge of exploration in unknown environments, as interactions with the environment are usually expensive or limited, we secondly study the problem of exploration in RL. We focus on problems with sparse and delayed rewards, and that has a distribution of possible environments (or tasks) that are related through common state and action spaces. The agent faces an unknown task in the future and is given prior opportunities to "practice" on related tasks where the interactions are still expensive. We propose a cost-aware Bayesian optimization approach that efficiently searches over a class of dynamic subgoal-based exploration strategies. The algorithm adjusts a variety of levers — the locations of the subgoals, the length of each episode, and the number of replications per trial

— in order to overcome the challenges of sparse rewards, expensive interactions, and noise. Our experimental evaluation demonstrates that, when averaged across problem domains, the proposed algorithm outperforms the meta-learning algorithm MAML by 19%, the hyperparameter tuning method Hyperband by 23%, BO techniques EI and LCB by 24% and 22%, respectively. We also provide a theoretical foundation and prove that the method asymptotically identifies a near-optimal subgoal design from the search space. Future work includes scalable extensions of the model and optimization formulation for high dimensional subgoal parameterizations, and apply the dynamic subgoal exploration strategies in a real-world application involving a navigation task.

Finally, we consider infinite horizon MDPs with fast-slow structure, meaning that certain parts of the state space move fast (and are more influential) while other parts of the state space transition move slowly (and are less influential). We propose the idea of freezing the slow states, and develop frozen-state algorithm and nominal state algorithm which solve a set of finite-horizon MDPs and apply value iteration to an auxiliary MDP that transitions on a slower timescale. Theoretically, we bound the loss caused by freezing the slow state and leveraging the nominal slow states respectively. We provide a bound over T given desired error level. Empirically, we compare the proposed algorithms and a few benchmarks. We show that when the slow state is exogenous, the policies from the proposed algorithms are similar to the base model value iteration. We also show that with either exogenous or endogenous slow state, the proposed algorithms converge faster than base model value iteration.

Appendix A

A.1 Proofs for Chapter 2

In this section, we give the proofs of results from the main paper: Proposition 2.3.2, Lemma 2.4.1, and Theorem 2.4.1.

A.1.1 Proof of Proposition 2.3.1

We prove the $L^\natural$-concavity of the Q-value function of the lower-level problem by backward induction. Note that if this is true, then the discrete concavity of $U_{w,i}(x,\xi)$ in x follows by Lemma 2 of [83]. Let $J_{w,i}(x,\xi,y)$ be the Q-value for a given state-action pair (x,ξ,y) at period i:

$$J_{w,i}(x,\xi,y) = u_w\big(y,\xi\big) + \mathbf{E}_w\big[U_{w,i+1}(X_{i+1},\Xi_{i+1})\big].$$

The base case is $J_{w,n+1}(x,\xi,y) = b\,x$, which is $L^\natural$-concave in (x,y). The induction hypothesis is that $J_{w,i+1}(x,\xi,y)$ is $L^\natural$-concave in (x,y).

We analyze $J_{w,i}(x,\xi,y)$ by breaking it up into two terms. The first term is $L^\natural$-concave in y according to Assumption 2.3.1. The second term $U_{w,i+1}(X_{i+1},\Xi_{i+1})$ is $L^\natural$-concave in X_{i+1} according to Lemma 2 of [83]. Since $X_{i+1} = x - y$, $U_{w,i+1}(X_{i+1},\Xi_{i+1})$ is $L^\natural$-concave in (x,y) by Lemma 2 in [66]. This concludes the proof.

A.1.2 Proof of Proposition 2.3.2

First, we prove part 1. Let us define the *state-action value function* (or the Q-value). The terminal value is defined as $Q_T^{\mathrm{rep}}(r,w;z^{\mathrm{rep}}) = -b\,r$. For $t < T$, replenish-up-to decision $z^{\mathrm{rep}} \in \bar{\mathcal{Z}}(r)$, and dispense-down-to decision $z^{\mathrm{dis}} \in \underline{\mathcal{Z}}(z^{\mathrm{rep}})$,

$$Q_t^{\mathrm{rep}}(r,w;z^{\mathrm{rep}}) = (c_w - h)\,r - c_w z^{\mathrm{rep}} + V_t^{\mathrm{dis}}(z^{\mathrm{rep}},w), \tag{A.1}$$

$$Q_t^{\text{dis}}(z^{\text{rep}}, w; z^{\text{dis}}) = \mathbf{E}_w\big[U_{w,0}\big(z^{\text{rep}} - z^{\text{dis}}, \Xi_{t,0}\big) + V_{t+1}^{\text{rep}}(R_{t+1}, W_{t+1})\big], \tag{A.2}$$

where $R_{t+1} = z^{\text{dis}}$. We now prove the $L^\natural$-concavity of Q-value by backward induction. Note that if this is true, then the $L^\natural$-concavity of $\tilde{V}_t^{\text{rep}}$ and $\tilde{V}_t^{\text{dis}}$ follows. The base case is $Q_T^{\text{rep}}(r, w; z^{\text{rep}}) = -b\, r$, which is $L^\natural$-concave in (r, z^{rep}), and the induction hypothesis is the same property for $Q_{t+1}^{\text{rep}}(r, w; z^{\text{rep}})$.

We first analyze (A.2) by breaking it up into two terms. The first term $\mathbf{E}_w\big[U_{w,0}\big(z^{\text{rep}} - z^{\text{dis}}, \Xi_{t,0}\big)\big]$ is discretely concave in $(z^{\text{rep}}, z^{\text{dis}})$ according to Proposition 2.3.1 and Lemma 2 in [66]. In the second term, $V_{t+1}^{\text{rep}}(r, w) = \max_{z^{\text{rep}} \in \bar{\mathcal{Z}}(r)} Q_{t+1}^{\text{rep}}(r, w; z^{\text{rep}})$. Lemma 2 of [83] shows that $V_{t+1}^{\text{rep}}(r, w)$ is $L^\natural$ concave in r. Since $R_{t+1} = z^{\text{dis}}$, the term $V_{t+1}^{\text{rep}}(R_{t+1}, W_{t+1})$ is $L^\natural$-concave in z^{dis}. $L^\natural$-concavity is preserved under expectations, so $Q_t^{\text{dis}}(z^{\text{rep}}, w; z^{\text{dis}})$ is $L^\natural$-concave in $(z^{\text{rep}}, z^{\text{dis}})$.

Next, we analyze (A.1) by breaking it up into two terms. The first term $(c_w - h)\, r - c_w z^{\text{rep}}$ is clearly $L^\natural$-concave in (r, z^{rep}). The second term is

$$V_t^{\text{dis}}(z^{\text{rep}}, w) = \max_{z^{\text{dis}} \in \underline{\mathcal{Z}}(z^{\text{rep}})} Q_t^{\text{dis}}(z^{\text{rep}}, w; z^{\text{dis}}).$$

Lemma 2 of [83] shows that $V_t^{\text{dis}}(z^{\text{rep}}, w)$ is $L^\natural$ concave in z^{rep}. This concludes Part 1.

A.1.3 Proof of Lemma 2.4.1

Let us show part (1), the convergence of $\bar{v}_t^{\text{rep},k}(z^{\text{rep}}, w)$. The convergence of $\bar{v}_t^{\text{dis},k}(z^{\text{dis}}, w)$ in part (2) of the lemma is similar. Since the demand $D_{t,i}$ is bounded by $D_{\max}$, there exists a $v_{\max}^{\text{rep}} > 0$ such that $|v_t^{\text{rep}}(z^{\text{rep}}, w)| \le v_{\max}^{\text{rep}}$ for all t, z^{rep}, and w. We first construct two deterministic sequences $\{G^m\}$ and $\{I^m\}$ such that $G^0 = v^{\text{rep}} + v_{\max}^{\text{rep}}$ and $I^0 = v^{\text{rep}} - v_{\max}^{\text{rep}}$ with

$$G^{m+1} = \frac{G^m + v^{\text{rep}}}{2} \quad \text{and} \quad I^{m+1} = \frac{I^m + v^{\text{rep}}}{2}. \tag{A.3}$$

It is easy to show that

$$G^m \to v^{\text{rep}} \quad \text{and} \quad I^m \to v^{\text{rep}}. \tag{A.4}$$

Our goal in this proof is to show that for any m and sufficiently large k,

$$I_t^m(z^{\text{rep}}, w) \le \bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \le G_t^m(z^{\text{rep}}, w). \tag{A.5}$$

If (A.5) is true, then we can conclude the result of Lemma 2.4.1 by (A.4).

We now introduce a random set of states $\mathcal{S}_t^-$ that are increased by the projection operator (16) on finitely many iterations k. Formally, let

$$\mathcal{S}_t^- = \big\{(z^{\text{rep}}, w) \in \mathcal{S} : \tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) < \bar{v}_t^{\text{rep},k}(z^{\text{rep}}, w) \text{ finitely often}\big\}.$$

Let $\bar{K}$ be the random variable that describes the iteration number after which states in $\mathcal{S}_t^-$ are no longer increased by the projection step; i.e., for all $(z^{\text{rep}}, w) \in \mathcal{S}_t^-$, it holds that $\tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) \geq \bar{v}_t^{\text{rep},k}(z^{\text{rep}}, w)$ for all $k \geq \bar{K}$. We break apart (A.5) into two separate inequalities; this proof will focus on showing that for a fixed m, there exists a finite random index $\hat{K}_t^m$ such that for all $k \geq \hat{K}_t^m$,

$$\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq G_t^m(z^{\text{rep}}, w). \tag{A.6}$$

The state space $\mathcal{S}$ can be partitioned into two parts: (1) states $(z^{\text{rep}}, w) \in \mathcal{S}_t^-$ and (2) states $(z^{\text{rep}}, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$. The proof of (A.6) will consider each partition separately. We now define some noise terms and stochastic sequences. Recall from (2.15) and (2.17) that $\hat{v}_t^{\text{rep},k} = \hat{V}_t^{\text{rep},k}(z_t^{\text{rep},k}, w_t^k) - \hat{V}_t^{\text{rep},k}(z_t^{\text{rep},k} - 1, w_t^k)$, where

$$\begin{aligned}\hat{V}_t^{\text{rep},k}(z^{\text{rep}}, w_t^k) = &- c_{w_t^k} z^{\text{rep}} + U_{w_t^k,0}^{\mu^*}\big(z^{\text{rep}} - \tilde{\pi}_t^{\text{dis},k-1}(z^{\text{rep}}, w_t^k), \check{\xi}_{t,0}^k\big) \\ &+ f_t^{\text{dis}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), z^{\text{rep}}\big),\end{aligned}$$

By our assumption that $\bar{l}_\tau^{\text{rep},k}(w) \to l_\tau^{\text{rep}}(w)$ almost surely for $\tau \geq t+1$, and $\bar{l}_\tau^{\text{dis},k}(w) \to l_\tau^{\text{dis}}(w)$ almost surely for $\tau \geq t$, and the fact that $f_t^{\text{dis}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), z^{\text{rep}}\big)$ depends only on the replenish-up-to thresholds for periods $t+1$ onward and the dispense-down-to thresholds for periods t onward, it follows that the simulated value of $\tilde{\pi}^{\text{rep},k-1}$ and $\tilde{\pi}^{\text{dis},k-1}$ becomes unbiased asymptotically:

$$\mathbf{E}_w\big[f_t^{\text{dis}}\big(\tilde{\pi}^{\text{rep},k-1}, \tilde{\pi}^{\text{dis},k-1}; \mathbf{Z}_t^k(w), z^{\text{rep}}\big)\big] \to \tilde{V}_t^{\text{dis}}\big(\pi_t^{\text{dis},*}(z^{\text{rep}}, w), w\big) \quad \text{a.s.} \tag{A.7}$$

We define the noise term $\epsilon_t^k(z_t^{\text{rep},k}, w_t^k)$ such that

$$\epsilon_t^k(z_t^{\text{rep},k}, w_t^k) = \mathbf{E}\big[\hat{v}_t^{\text{rep},k}\big] - v_t^{\text{rep}}(z_t^{\text{rep},k}, w_t^k). \tag{A.8}$$

Note that we can conclude from (A.7) that $\epsilon_t^k(z_t^{\text{rep},k}, w_t^k) \to 0$ almost surely. We define another noise term $\varepsilon_t^k(z_t^{\text{rep},k}, w_t^k)$ such that $\varepsilon_t^k(z_t^{\text{rep},k}, w_t^k) = \hat{v}_t^{\text{rep},k} - \mathbf{E}\big[\hat{v}_t^{\text{rep},k}\big]$. Thus, we can see that

$$\hat{v}_t^{\text{rep},k} = v_t^{\text{rep}}(z_t^{\text{rep},k}, w_t^k) + \epsilon_t^k(z_t^{\text{rep},k}, w_t^k) + \varepsilon_t^k(z_t^{\text{rep},k}, w_t^k) \tag{A.9}$$

Next, we need to define some stochastic sequences related to these noise terms. Let $\{\bar{s}_t^k\}$ be defined such that for $k < \bar{K}$, $\bar{s}_t^k(z^{\text{rep}}, w) = 0$, and for $k \geq \bar{K}$,

$$\bar{s}_t^k(z^{\text{rep}}, w) = \big(1 - \alpha_t^k(z^{\text{rep}}, w)\big)\, \bar{s}_t^{k-1}(z^{\text{rep}}, w) + \alpha_t^k(z^{\text{rep}}, w)\big[\epsilon_t^k(z_t^{\text{rep},k}, w_t^k) + \varepsilon_t^k(z_t^{\text{rep},k}, w_t^k)\big]. \tag{A.10}$$

This sequence averages both of the noise terms. Since ϵ_t^k is unbiased and ε_t^k converges to zero, we can apply Theorem 2.4 of [80], a standard stochastic approximation convergence result, to conclude that $\bar{s}_t^k(z^{\text{rep}}, w) \to 0$ almost surely. We then define a stochastic bounding sequence $\{\bar{g}_t\}$ such that for $k < \bar{K}$, $\bar{g}_t^k(z^{\text{rep}}, w) = G_t^k(z^{\text{rep}}, w)$ and for $k \geq \bar{K}$,

$$\bar{g}_t^k(z^{\text{rep}}, w) = \big(1 - \alpha_t^k(z^{\text{rep}}, w)\big)\, \bar{g}_t^{k-1}(z^{\text{rep}}, w) + \alpha_t^k(z^{\text{rep}}, w)\, v_t^{\text{rep}}(z^{\text{rep}}, w). \tag{A.11}$$

As in [43], we provide an ω-wise argument, meaning that we consider a fixed $\omega \in \Omega$ (although the dependence of random variables on ω is omitted for notational simplicity). Here, we show the existence of a finite index $\tilde{K}_t^m$ such that for all states $(z^{\text{rep}}, w) \in \mathcal{S}_t^-$, it holds that for all iterations $k \geq \tilde{K}_t^m$, $\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq G_t^m(z^{\text{rep}}, w)$. The proof is a forward induction on m where the base case is $m = 0$. The base case can be easily proved by applying the definition of G^0 (note that we can select $\tilde{K}_t^m \geq \bar{K}$. The induction hypothesis is that there exists an integer $\tilde{K}_t^m \geq \bar{K}$ such that for all $k \geq \tilde{K}_t^m$, the inequality (A.6) is true. The next step is $m + 1$: we must show the existence of an integer $\tilde{K}_t^{m+1} \geq \bar{K}$ such that for all states $(z^{\text{rep}}, w) \in \mathcal{S}_t^-$, it holds that

$$\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq G_t^{m+1}(z^{\text{rep}}, w) \tag{A.12}$$

for all iterations $k \geq \tilde{K}_t^{m+1}$. We require the following lemma.

Lemma A.1.1. *The inequality*

$$\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq \bar{g}_t^{k-1}(z^{\text{rep}}, w) + \bar{s}_t^{k-1}(z^{\text{rep}}, w) \tag{A.13}$$

holds almost everywhere on $\{k \geq \tilde{K}_t^m,\ (z^{\text{rep}}, w) \in \mathcal{S}_t^-\}$.

Proof. When $k = \tilde{K}_t^m$, the relationship (A.13) can be shown using the definitions of $\bar{g}_t^{k-1}(z^{\text{rep}}, w)$ and $\bar{s}_t^{k-1}(z^{\text{rep}}, w)$, along with the induction hypothesis (A.6). We now induct on k. Suppose that (A.13) is true for a given $k \geq \tilde{K}_t^m$. The inductive step is to show $\bar{v}_t^{\text{rep},k}(z^{\text{rep}}, w) \leq \bar{g}_t^k(z^{\text{rep}}, w) + \bar{s}_t^k(z^{\text{rep}}, w)$. To simplify notation, let $\check{\alpha}_t^k$, $\check{v}_t^k$, $\check{s}_t^k$, and $\check{g}_t^k$ respectively denote $\alpha_t^k(z^{\text{rep}}, w)$, $\bar{v}_t^{\text{rep},k}(z^{\text{rep}}, w)$, $\bar{s}_t^k(z^{\text{rep}}, w)$ and $\bar{g}_t^k(z^{\text{rep}}, w)$. For state $(z^{\text{rep}}, w) = (z_t^{\text{rep},k}, w_t^k)$, we have

$$\begin{aligned}
\check{v}_t^k &= \tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) = (1 - \check{\alpha}_t^k)\check{v}_t^{k-1} + \check{\alpha}_t^k \hat{v}_t^{\text{rep},k} \\
&\leq (1 - \check{\alpha}_t^k)\big(\check{g}_t^{k-1} + \check{s}_t^{k-1}\big) + \check{\alpha}_t^k \hat{v}_t^{\text{rep},k} - \check{\alpha}_t^k v_t^{\text{rep}}(z_t^{\text{rep},k}, w_t^k) + \check{\alpha}_t^k v_t^{\text{rep}}(z_t^{\text{rep},k}, w_t^k) \\
&= (1 - \check{\alpha}_t^k)\big(\check{g}_t^{k-1} + \check{s}_t^{k-1}\big) + \check{\alpha}_t^k \big[\epsilon_t^k(z_t^{\text{rep},k}, w_t^k) + \varepsilon_t^k(z_t^{\text{rep},k}, w_t^k)\big] + \check{\alpha}_t^k v_t^{\text{rep}}(z_t^{\text{rep},k}, w_t^k) \\
&= (1 - \check{\alpha}_t^k)\check{g}_t^{k-1} + \check{s}_t^k + \check{\alpha}_t^k v_t^{\text{rep}}(z_t^{\text{rep},k}, w_t^k) \\
&= \check{g}_t^k + \check{s}_t^k.
\end{aligned}$$

The first equality is due to the fact that $(z^{\text{rep}}, w) = (z_t^{\text{rep},k}, w_t^k)$, which is unaltered by the projection operator (16). The second inequality follows from the induction hypothesis (A.13). The last three steps follow by (A.9), (A.10) and (A.11) respectively.

For $(z^{\text{rep}}, w) \neq (z_t^{\text{rep},k}, w_t^k)$, which are the states that are not updated by a direct observation of the sample slope at iteration k, period t, the stepsize $\check{\alpha}_t^k = 0$. Then, we have

$$\check{s}_t^k = \check{s}_t^{k-1} \quad \text{and} \quad \check{g}_t^k = \check{g}_t^{k-1}.$$

Therefore, from the definition of set $\mathcal{S}_t^-$, the fact that $\tilde{K}_t^m \geq \bar{K}$, and the induction hypothesis, we have

$$\check{v}_t^k \leq \tilde{v}_t^{\text{rep},k}(z^{\text{rep}}, w) = \check{v}_t^{k-1} \leq \check{g}_t^{k-1} + \check{s}_t^{k-1} = \check{g}_t^k + \check{s}_t^k,$$

which concludes the proof of (A.13). □

Since $G^m \geq G^{m+1} \geq v^{\text{rep}}$ for all m, when $G_t^m(z^{\text{rep}}, w) = v_t^{\text{rep}}(z^{\text{rep}}, w) = G_t^{m+1}(z^{\text{rep}}, w)$, the inequality $\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq G_t^m(z^{\text{rep}}, w)$ implies that $\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq G_t^{m+1}(z^{\text{rep}}, w)$. Thus, the only remaining states to consider are the ones where $G_t^m(z^{\text{rep}}, w) > v_t^{\text{rep}}(z^{\text{rep}}, w)$. Let δ^m be the minimum of the quantity $[G_t^k(z^{\text{rep}}, w) - v_t^{\text{rep}}(z^{\text{rep}}, w)]/4$ over states $(z^{\text{rep}}, w) \in$

$\mathcal{S}_t^-$ with $G_t^m(z^{\text{rep}}, w) > v_t^{\text{rep}}(z^{\text{rep}}, w)$. Define an integer $K^G \geq \tilde{K}_t^m$ such that for all states $(z^{\text{rep}}, w) \in \mathcal{S}_t^-$,

$$\textstyle\prod_{k=\tilde{K}_t^m}^{K^G-1} \big(1 - \alpha_t^k(z^{\text{rep}}, w)\big) \leq 1/4 \quad \text{and} \quad \bar{s}_t^k(z^{\text{rep}}, w) \leq \delta^m.$$

for every iteration $k \geq K^G$. We can find such a K^G because the stepsize conditions of Assumption 2.4.1 imply that

$$\textstyle\prod_{k=\tilde{K}_t^m}^{\infty} \big(1 - \alpha_t^k(z^{\text{rep}}, w)\big) = 0,$$

and because $\bar{s}_t^k(z^{\text{rep}}, w)$ converges to zero.

Now we are ready to show (A.12). The definition of the sequence $\{\bar{g}_t^k\}$ implies that $\bar{g}_t^k(z^{\text{rep}}, w)$ is a convex combination of $G_t^k(z^{\text{rep}}, w)$ and $v_t^{\text{rep}}(z^{\text{rep}}, w)$, of the form

$$\bar{g}_t^k(z^{\text{rep}}, w) = \hat{\alpha}_t^k(z^{\text{rep}}, w)\, G_t^k(z^{\text{rep}}, w) + \big(1 - \hat{\alpha}_t^k(z^{\text{rep}}, w)\big)\, v_t^{\text{rep}}(z^{\text{rep}}, w),$$

where $\hat{\alpha}_t^k(z^{\text{rep}}, w) = \prod_{k=\tilde{K}_t^m}^{K-1} \big(1 - \alpha_t^k(z^{\text{rep}}, w)\big) \leq 1/4$ for $k \geq K^G$. Because $G^m \geq v^{\text{rep}}$ for any m, it follows that

$$\begin{aligned}
\bar{g}_t^k(z^{\text{rep}}, w) &\leq \tfrac{1}{4}\, G_t^k(z^{\text{rep}}, w) + \tfrac{3}{4}\, v_t^{\text{rep}}(z^{\text{rep}}, w) \\
&= \tfrac{1}{2}\, G_t^k(z^{\text{rep}}, w) + \tfrac{1}{2}\, v_t^{\text{rep}}(z^{\text{rep}}, w) - \tfrac{1}{4}\, \big(G_t^k(z^{\text{rep}}, w) - v_t^{\text{rep}}(z^{\text{rep}}, w)\big) \\
&\leq G_t^{k+1}(z^{\text{rep}}, w) - \delta^m,
\end{aligned}$$

where the second inequality follows from (A.3) and the definition of δ^m. Recall that we are concentrating on the case where $G_t^m(z^{\text{rep}}, w) > v_t^{\text{rep}}(z^{\text{rep}}, w)$, so δ^m is well-defined and positive. This inequality, together with Lemma A.1.1 and $\bar{s}_t^k(z^{\text{rep}}, w) \leq \delta^m$, imply that for all $k \geq K^G$,

$$\bar{g}_t^k(z^{\text{rep}}, w) \leq G_t^{k+1}(z^{\text{rep}}, w) - \delta^m + \bar{s}_t^k(z^{\text{rep}}, w) \leq G_t^{k+1}(z^{\text{rep}}, w) - \delta^m + \delta^m \leq G_t^{k+1}(z^{\text{rep}}, w).$$

We conclude Part (1) of the proof by letting $\tilde{K}_t^{m+1} = K^G$.

We now focus on the states $(z^{\text{rep}}, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$ that are increased infinitely often. For a fixed m and state $(z^{\text{rep}}, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$, we wish to prove the existence of a random index $\hat{K}_t^m(z^{\text{rep}}, w)$ such that for all $k \geq \hat{K}_t^m(z^{\text{rep}}, w)$, it holds that $\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq G_t^m(z^{\text{rep}}, w)$. Note that $\hat{K}_t^m(z^{\text{rep}}, w)$ differs from $\tilde{K}_t^m$ in that it depends on a specific $(z^{\text{rep}}, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$ (while we $\tilde{K}_t^m$ is chosen uniformly for all states in $\mathcal{S}_t^-$). The crux of the proof depends on the following lemma.

Lemma A.1.2. *Fix $m \geq 0$ and consider a state $(z^{\text{rep}} - 1, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$ and suppose that there exists a random index $\hat{K}_t^m(z^{\text{rep}}, w)$ such that the required condition $\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \leq G_t^m(z^{\text{rep}}, w)$ is true, then there exists another random index $\hat{K}_t^m(z^{\text{rep}} - 1, w)$ such that*

$$\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}} - 1, w) \leq G_t^m(z^{\text{rep}} - 1, w)$$

for all iterations $k \geq \hat{K}_t^m(z^{\text{rep}} - 1, w)$.

Proof. See the proof of Lemma 6.4 of [43]. The only modification that needs to be made is to redefine the Bellman operator 'H' from [43] so that it maps to the optimal value function slopes v for any argument (we no longer interpret H as a Bellman operator as our algorithm is not based on value iteration). □

Consider some $m \geq 0$ and a state $(z^{\text{rep}}, w) \in \mathcal{S} \setminus \mathcal{S}_t^-$. Now, let state $(z_{\min}^{\text{rep}}, w)$ where $z_{\min}^{\text{rep}}$ is the minimum replenish-up-to postdecision resource level such that $z_{\min}^{\text{rep}} > z^{\text{rep}}$ and $(z_{\min}^{\text{rep}}, w) \in \mathcal{S}_t^-$. We note that such a state certainly exists because $(R_{\max}, w) \in \mathcal{S}_t^-$. The state $(z_{\min}^{\text{rep}}, w)$ satisfies the condition of Lemma A.1.2 with $\hat{K}_t^m(z_{\min}^{\text{rep}}, w) = K_t^m$, so we may conclude that there is an index $\hat{K}_t^m(z_{\min}^{\text{rep}} - 1, w)$ associated with state $(z_{\min}^{\text{rep}} - 1, w)$ such that for all $k \geq \hat{K}_t^m(z_{\min}^{\text{rep}} - 1, w)$, the required condition $\bar{v}_t^{\text{rep},k-1}(z_{\min}^{\text{rep}} - 1, w) \leq G_t^m(z_{\min}^{\text{rep}} - 1, w)$ holds. This process can be repeated until we reach the state of interest (z^{rep}, w), which provides the required $\hat{K}_t^m(z^{\text{rep}}, w)$. Finally, if we choose an iteration large enough, i.e.,

$$K_t^m = \max\{\tilde{K}_t^m, \max_{(z^{\text{rep}},w) \in \mathcal{S} \setminus \mathcal{S}_t^-} \hat{K}_t^m(z^{\text{rep}}, w)\},$$

then (A.6) is true for all $k \geq \hat{K}_t^m$ and states $(z^{\text{rep}}, w) \in \mathcal{S}$. A symmetric proof can be given to verify that the other half of the inequality (A.5), $\bar{v}_t^{\text{rep},k-1}(z^{\text{rep}}, w) \geq I_t^m(z^{\text{rep}}, w)$, holds for sufficiently large k, which completes the proof.

A.1.4 Proof of Theorem 2.4.1

The proof of Theorem 2.4.1 is a backward induction over time periods t. For the replenish-up-to value function and threshold, the base case is $t = T$, where the convergence of $\bar{v}_T^{\text{rep},k}(z^{\text{rep}}, w)$ and $\bar{l}_T^{\text{rep},k}(w)$ to their optimal counterparts (both equal to zero) are trivial by assumption (see Section 2.4.3). The induction hypothesis is that $\bar{l}_\tau^{\text{rep},k}(w) \to l_\tau^{\text{rep}}(w)$ almost surely for $\tau \geq t+1$, and $\bar{\pi}_\tau^{\text{dis},k}(z^{\text{rep}}, w) \to \pi_\tau^{\text{dis}}(z^{\text{rep}}, w)$ almost surely for $\tau \geq t$. Now, consider period t. The almost sure convergence of $\bar{v}_t^{\text{rep},k}(z^{\text{rep}}, w)$ to $v_t^{\text{rep}}(z^{\text{rep}}, w)$ follows by Lemma 2.4.1. Therefore, by Assumption 2.4.2, we can conclude that

$$\hat{l}_t^{\text{rep},k} = \arg\max_{z^{\text{rep}} \in \mathcal{Z}(0)} \sum_{j=0}^{z^{\text{rep}}} \bar{v}_t^{\text{rep},k}(j, w_t^k) \to l_t^{\text{rep}}(w) \quad \text{a.s.}$$

Combining this with the update formula for $\bar{l}_t^{\text{rep},k}(w)$, the stepsize properties of Assumption 2.4.1, and Theorem 2.4 of [80], we see that $\bar{l}_t^{\text{rep},k}(w)$ converges to $l_t^{\text{rep}}(w)$ almost surely.

For the dispense-down-to value function and policy, the proof is similar. We only need to notice that the dispense-down-to decision is made after the replenish-up-to decision, and the induction hypothesis for it is that $\bar{l}_\tau^{\text{rep},k}(w) \to l_\tau^{\text{rep}}(w)$ and $\bar{\pi}_\tau^{\text{dis},k}(z^{\text{rep}}, w) \to \pi_\tau^{\text{dis}}(z^{\text{rep}}, w)$ almost surely for $\tau \geq t+1$.

A.2 Actor-Critic Method

The actor-critic method is shown in Algorithm 6.

Algorithm 6: Actor-Critic Method

Input: RBFs $\boldsymbol{\psi}(r, w)$ for the state value, and $\boldsymbol{\phi}(r, w; z^{\text{rep}}, z^{\text{dis}})$ for the policy.
Initial parameter estimate $\boldsymbol{\eta}^0$ and $\boldsymbol{\theta}^0$.
Stepsize rules $\tilde{\alpha}_t^k$ and $\tilde{\beta}_t^k$ for all t, k.

Output: Parameters $\boldsymbol{\eta}^k$ and $\boldsymbol{\theta}^k$.

for $k = 1, 2, \ldots, K$ **do**
  Sample an initial state s_0^k.
  for $t = 0, 1, \ldots, T-1$ **do**
    Observe w_t^k and $\xi_{t,1}^k$.
    Take action $\big(z_t^{k,\text{rep}}, z_t^{k,\text{dis}}\big) \sim \pi_t^{k-1}(z^{\text{rep}}, z^{\text{dis}} | r_t^k, w_t^k; \boldsymbol{\theta}^{k-1})$, observe the next state (r_{t+1}^k, w_{t+1}^k) and the immediate reward $C_t = (c_{w_t^k} - h)\, r - c_{w_t^k} z_t^{k,\text{rep}} + U_{w_t^k,0}\big(z_t^{k,\text{rep}} - z_t^{k,\text{dis}}, \xi_{t,0}^k\big)$.
    Calculate the temperal difference $\delta_t \leftarrow C_t + \boldsymbol{\psi}(r_{t+1}^k, w_{t+1}^k)^T \boldsymbol{\eta}_{t+1}^k - \boldsymbol{\psi}(r_t^k, w_t^k)^T \boldsymbol{\eta}_t^k$.
    Critic update: $\boldsymbol{\eta}_t^k = \boldsymbol{\eta}_t^{k-1} + \alpha_t^k(r, w)\delta_t \boldsymbol{\psi}(r_t^k, w_t^k)$, where $\alpha_t^k(r, w) = \tilde{\alpha}_t^k \mathbb{1}\{(r, w) = (r_t^k, w_t^k)\}$.
    Actor update: $\boldsymbol{\theta}_t^k = \boldsymbol{\theta}_t^{k-1} + \beta_t^k(r, w; z^{\text{rep}}, z^{\text{dis}})\delta_t \Delta_{\boldsymbol{\theta}_t^{k-1}} \ln \dot{\pi}_t^{k-1}(z^{\text{rep}}, z^{\text{dis}} | r_t^k, w_t^k; \boldsymbol{\theta}^{k-1})$, where $\beta_t^k(r, w; z^{\text{rep}}, z^{\text{dis}}) = \tilde{\beta}_t^k \mathbb{1}\{(r, w; z^{\text{rep}}, z^{\text{dis}}) = (r_t^k, w_t^k; z_t^{k,\text{rep}}, z_t^{k,\text{dis}})\}$.
  end
end

A.3 A Practical, Aggregation-based Version of S-AC

To deal with potentially continuous information states $W_t \in \mathcal{W}$, we now introduce a practical version of our algorithm that utilizes aggregation in the information state. The essential idea is that the structural results from Section 2.3 continue to hold when we perform aggregation, so the S-AC idea can be applied almost directly. We partition the exogenous information space $\mathcal{W}$ into J sets, i.e., let

$$\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2 \cup \ldots \cup \mathcal{W}_J \quad \text{with} \quad \mathcal{W}_i \cap \mathcal{W}_j = \emptyset \quad \text{if} \quad i \neq j.$$

Note that we do not aggregate in the inventory state and only do so in the information state. Each partition $\mathcal{W}_j$ contains a representative state, denoted $\dot{w}_j \in \mathcal{W}_j$, similar to what is done in [213]. We also assign a distribution over each partition and we suppose that the distribution is described with a density function $p^j(w)$, with $w \in \mathcal{W}_j$. This allows us to map the original MDP to an aggregate version by integrating with respect to this distribution (which should be thought of as a design choice). For the remainder of the paper, we assume that $p^j(\cdot)$ is a uniform density function, but remark that the algorithm can easily accommodate other aggregation distributions by including a likelihood ratio factor.

We use "dot" notation to denote variables related to state aggregation. For example, $\dot{W}_t$ denotes the aggregate exogenous information at period t. Further, let $\dot{V}_t^{\text{rep}}(r, \dot{w}_j)$ and $\dot{V}_t^{\text{dis}}(z^{\text{rep}}, \dot{w}_j)$ respectively denote the *optimal aggregate value functions* for the replenish-up-to decision and the dispense-down-to decision, let $\dot{\tilde{V}}_t^{\text{rep}}(z, \dot{w}_j)$ and $\dot{\tilde{V}}_t^{\text{dis}}(z^{\text{rep}}, \dot{w}_j)$ respectively denote their corresponding *aggregate postdecision value function*, let $\dot{\tilde{\pi}}^{\text{rep}}$ and $\dot{\tilde{\pi}}^{\text{dis}}$ be the rounded policies under state aggregation. The terminal aggregate value function is $\dot{V}_T^{\text{rep}}(r, \dot{w}_j) = -b\, r$ and for $t < T$, we have

$$\dot{V}_t^{\text{rep}}(r, \dot{w}_j) = \max_{z^{\text{rep}} \in \bar{\mathcal{Z}}(r)} \int_{w \in \mathcal{W}_j} p^j(w) \left\{ (c_w - h)r - c_w z^{\text{rep}} + \dot{V}_t^{\text{dis}}(z^{\text{rep}}, \dot{w}_j) \right\} dw,$$

$$\dot{V}_t^{\text{dis}}(z^{\text{rep}}, \dot{w}_j) = \max_{z^{\text{dis}} \in \underline{\mathcal{Z}}(z^{\text{rep}})} \int_{w \in \mathcal{W}_j} p^j(w) \left\{ \mathbf{E}_w \left[U_{w,0}^{\bar{\mu}}(z^{\text{rep}} - z^{\text{dis}}, \Xi_{t,0}) + \dot{V}_{t+1}^{\text{rep}}\left(z^{\text{dis}}, \dot{W}_{t+1}\right) \right] \right\} dw,$$

where the transition to $\dot{W}_{t+1}$ satisfies $\dot{W}_{t+1} = \sum_{j=1}^{k} \dot{W}_j \mathbb{1}\{W_{t+1} \in \dot{W}_j\}$, and $\bar{\mu}$ is the approximate policy for the lower-level. For the lower-level dispensing problem, similar the the

discrete state space version, we solve the optimal policy μ^* for each aggregate state. Then the policy is extrapolated to the continuous state space by linear regression. Similar to the definition of postdecision value functions (2.6) and (2.7), define

$$\dot{\tilde{V}}_t^{\text{rep}}(z, \dot{w}_j) = \int_{w \in \mathcal{W}_j} p^j(w) \left\{ -c_w z^{\text{rep}} + \dot{V}_t^{\text{dis}}(z^{\text{rep}}, \dot{w}_j) \right\} dw,$$

$$\dot{\tilde{V}}_t^{\text{dis}}(z^{\text{rep}}, \dot{w}_j) = \int_{w \in \mathcal{W}_j} p^j(w) \, \mathbf{E}_w \left[\dot{V}_{t+1}^{\text{rep}}\left(z^{\text{dis}}, \dot{W}_{t+1}\right) \right] dw.$$

The optimal replenish-up-to and dispense-down-to policies under state aggregation can be written as

$$\dot{\pi}_t^{\text{rep},*}(r, \dot{w}_j) \in \arg\max_{z^{\text{rep}} \in \bar{\mathcal{Z}}(r)} \dot{\tilde{V}}_t^{\text{rep}}(z, \dot{w}_j),$$

$$\dot{\pi}_t^{\text{dis},*}(z^{\text{rep}}, \dot{w}_j) \in \arg\max_{z^{\text{dis}} \in \underline{\mathcal{Z}}(z^{\text{rep}})} \dot{\tilde{V}}_t^{\text{dis}}(z^{\text{dis}}, \dot{w}_j),$$

The postdecision Bellman equation under state aggregation is $\dot{\tilde{V}}_{T-1}^{\text{dis}}(z^{\text{dis}}, \dot{w}_j) = -b\, z^{\text{dis}}$, and for any $t < T-1$,

$$\begin{aligned}\dot{\tilde{V}}_t^{\text{rep}}(z^{\text{rep}}, \dot{w}_j) = \int_{w \in \mathcal{W}_j} p^j(w) \Big\{ -c_w z^{\text{rep}} + \mathbf{E}_w \left[U_{w,0}^{\bar{\mu}}\left(z^{\text{rep}} - \dot{\pi}_t^{\text{dis},*}(z^{\text{rep}}, \dot{w}_j), \Xi_{t,0}\right) \right] \\ + \dot{\tilde{V}}_t^{\text{dis}}\left(\dot{\pi}_t^{\text{dis},*}(z^{\text{rep}}, \dot{w}_j), \dot{w}_j\right) \Big\} dw,\end{aligned}$$

$$\dot{\tilde{V}}_t^{\text{dis}}(z^{\text{dis}}, \dot{w}_j) = \int_{w \in \mathcal{W}_j} p^j(w) \left\{ \mathbf{E}_w \left[(c_{\dot{W}_{t+1}} - h) z^{\text{dis}} + \dot{\tilde{V}}_{t+1}^{\text{rep}}\left(\dot{\pi}_t^{\text{rep},*}(z^{\text{dis}}, \dot{W}_{t+1}), \dot{W}_{t+1}\right) \right] \right\} dw.$$

The properties of the aggregate problem are stated in Proposition A.3.1. The result follows from the proof of Proposition 2.3.2 and the fact that $L^\natural$-concavity is preserved under expectations.

Proposition A.3.1. *Suppose Assumption 2.3.1 is satisfied. Then, the structural properties in Proposition 2.3.2 hold for the aggregate postdecision value functions* $\dot{\tilde{V}}_t^{\text{rep}}(z^{\text{rep}}, \dot{w}_j)$ *and* $\dot{\tilde{V}}_t^{\text{dis}}(z^{\text{dis}}, \dot{w}_j)$ *as well as the thresholds* $\dot{l}_t^{\text{rep}}(\dot{w}_j)$ *and* $\dot{l}_t^{\text{dis}}(\dot{w}_j)$.

Proposition A.3.1 is the theoretical basis of the algorithm for the aggregate problem. At each iteration and each period in the algorithm, we sample/observe the true exogenous information process as in Algorithm 1, while using the corresponding aggregate exogenous information states to update values and thresholds. The details are in Appendix A.3.1.

A.3.1 Algorithm for the Aggregate Problem

We define some other notations. At iteration k and period t, we use the same notations as in Section 2.4 to represent the observation of the exogenous information and the attribute, which are w_t^k and $\xi_{t,1}^k$ respectively. The corresponding information partition and the aggregate exogenous information are $\mathcal{W}_t^k$ and $\dot{w}_t^k$ respectively. For the process $\mathbf{Z}_t^k(w) = \big\{(\check{w}_\tau^k, \check{\xi}_{\tau,1}^k) : \tau = t, \ldots, T-1\big\}$, denote $\check{\mathcal{W}}_t^k$ and $\dot{\check{w}}_t^k$ the corresponding information partition and the aggregate exogenous information at period τ respectively, and we have $\check{w}_t^k \in \check{\mathcal{W}}_t^k$. Let $\dot{f}_t^{\text{rep}}\big(\dot{\bar{\pi}}^{\text{rep},k-1}, \dot{\bar{\pi}}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), r_t\big)$ be the Monte Carlo estimates of the replenish-up-to postdecision value starting in period t under the current aggregate policy approximations and an initial state (r_t, w_t):

$$\begin{aligned}\dot{f}_t^{\text{rep}}\big(\dot{\bar{\pi}}^{\text{rep},k-1}, \dot{\bar{\pi}}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), r_t\big) = \textstyle\sum_{\tau=t}^{T-2}\big[-c_{\check{w}_\tau^k}\dot{\tilde{z}}_\tau^{\text{rep}} + U^{\bar{\mu}}_{\check{w}_\tau^k}\big(\dot{\tilde{z}}_\tau^{\text{rep}} - \dot{\tilde{z}}_\tau^{\text{dis}}, \check{\xi}_{\tau,0}^k\big) + (c_{\check{w}_{\tau+1}^k} - h)\dot{\tilde{z}}_\tau^{\text{dis}}\big] \\ - c_{\check{w}_{T-1}^k}\dot{\tilde{z}}_{T-1}^{\text{rep}} + U^{\bar{\mu}}_{\check{w}_{T-1}^k,0}\big(\dot{\tilde{z}}_{T-1}^{\text{rep}} - \dot{\tilde{z}}_{T-1}^{\text{dis}}, \check{\xi}_{T-1,0}^k\big) - b\,\dot{\tilde{z}}_{T-1}^{\text{dis}},\end{aligned}$$

where for all $\tau \geq t$, the aggregate policies are

$$\dot{\tilde{z}}_\tau^{\text{rep}} = \dot{\tilde{\pi}}_\tau^{\text{rep},k-1}(r_\tau, \dot{\check{w}}_\tau^k), \quad \dot{\tilde{z}}_\tau^{\text{dis}} = \dot{\tilde{\pi}}_\tau^{\text{dis},k-1}\big(\tilde{\pi}_\tau^{\text{rep},k-1}(r_\tau, \dot{\check{w}}_\tau^k), \dot{\check{w}}_\tau^k\big).$$

Let $\dot{f}_t^{\text{dis}}\big(\dot{\bar{\pi}}^{\text{rep},k-1}, \dot{\bar{\pi}}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), z_t^{\text{rep}}\big)$ be the Monte Carlo estimates of the dispense-down-to postdecision value starting in period t under the current aggregate policy approximations and an initial state (z_t^{rep}, w_t):

$$\begin{aligned}&\dot{f}_t^{\text{dis}}\big(\dot{\bar{\pi}}^{\text{rep},k-1}, \dot{\bar{\pi}}^{\text{dis},k-1}; \mathbf{Z}_t^k(w_t), z_t^{\text{rep}}\big) \\ &\quad = \textstyle\sum_{\tau=t}^{T-2}\big[(c_{\check{w}_{\tau+1}^k} - h)\dot{\tilde{z}}_\tau^{\text{dis}} - c_{\check{w}_{\tau+1}^k}\dot{\tilde{z}}_{\tau+1}^{\text{rep}} + U^{\bar{\mu}}_{\check{w}_{\tau+1}^k,0}\big(\tilde{z}_{\tau+1}^{\text{rep}} - \dot{\tilde{z}}_{\tau+1}^{\text{dis}}, \check{\xi}_{\tau+1,0}^k\big)\big] - b\,\dot{\tilde{z}}_{T-1}^{\text{dis}},\end{aligned}$$

where $\dot{\tilde{z}}_t^{\text{dis}} = \dot{\tilde{\pi}}_t^{\text{dis},k-1}\big(z_t^{\text{rep}}, \dot{\check{w}}_\tau^k\big)$, and for all $\tau \geq t+1$,

$$\dot{\tilde{z}}_\tau^{\text{rep}} = \dot{\tilde{\pi}}_\tau^{\text{rep},k-1}(r_\tau, \dot{\check{w}}_\tau^k), \quad \dot{\tilde{z}}_\tau^{\text{dis}} = \dot{\tilde{\pi}}_\tau^{\text{dis},k-1}\big(\dot{\tilde{z}}_\tau^{\text{rep}}, \dot{\check{w}}_\tau^k\big).$$

At each period t, to compute the approximate slopes, we use $\dot{f}_t^{\text{dis}}$ to observe values $\dot{\hat{V}}_t^{\text{rep},k}(z_t^{\text{rep},k}, \dot{w}_t^k)$ and $\dot{\hat{V}}_t^{\text{rep},k}(z_t^{\text{rep},k} - 1, \dot{w}_t^k)$, and $\dot{f}_{t+1}^{\text{rep}}$ to observe values $\dot{\hat{V}}_t^{\text{dis},k}(z_t^{\text{dis},k}, \dot{w}_t^k)$ and $\dot{\hat{V}}_t^{\text{dis},k}(z_t^{\text{dis},k} - 1, \dot{w}_t^k)$, where $\dot{f}_t^{\text{dis}}$ and $\dot{f}_{t+1}^{\text{rep}}$ are implied by the current aggregate policies $\dot{\bar{\pi}}^{\text{rep},k-1}$

and $\dot{\bar{\pi}}^{\text{dis},k-1}$; specifically, for $z^{\text{rep}}, z^{\text{dis}} \geq 0$, the observations $\dot{\hat{V}}_t^{\text{rep},k}(z^{\text{rep}}, \dot{w}_t^k)$ and $\dot{\hat{V}}_t^{\text{dis},k}(z^{\text{dis}}, \dot{w}_t^k)$ are

$$\begin{aligned}\dot{\hat{V}}_t^{\text{rep},k}(z^{\text{rep}}, \dot{w}_t^k) = &- c_{\dot{w}_t^k} z^{\text{rep}} + U^{\bar{\mu}}_{\check{w}_t^k,0}\big(z^{\text{rep}} - \dot{\bar{\pi}}_t^{\text{dis},k-1}(z^{\text{rep}}, \dot{w}_t^k), \check{\xi}_{t,0}^k\big) \\ &+ \dot{f}_t^{\text{dis}}\big(\dot{\bar{\pi}}^{\text{rep},k-1}, \dot{\bar{\pi}}^{\text{dis},k-1}; \mathbf{Z}_t^k(\dot{w}_t), z^{\text{rep}}\big),\end{aligned}$$

and

$$\hat{V}_t^{\text{dis},k}(z^{\text{dis}}, \dot{w}_t^k) = (c_{w_{t+1}} - h) z^{\text{dis}} + \dot{f}_{t+1}^{\text{rep}}\big(\dot{\bar{\pi}}^{\text{rep},k-1}, \dot{\bar{\pi}}^{\text{dis},k-1}; \mathbf{Z}_{t+1}^k(w_{t+1}), z^{\text{dis}}\big),$$

where w_{t+1} is a realization from the distribution $W_{t+1} \,|\, W_t = \dot{w}_t^k$. The approximate slopes $\dot{\hat{v}}_t^{\text{rep},k}$ and $\dot{\hat{v}}_t^{\text{dis},k}$ are given by:

$$\dot{\hat{v}}_t^{\text{rep},k} = \dot{\hat{V}}_t^{\text{rep},k}(z_t^{\text{rep},k}, \dot{w}_t^k) - \dot{\hat{V}}_t^{\text{rep},k}(z_t^{\text{rep},k} - 1, \dot{w}_t^k), \tag{A.14}$$

$$\dot{\hat{v}}_t^{\text{dis},k} = \dot{\hat{V}}_t^{\text{dis},k}(z_t^{\text{dis},k}, \dot{w}_t^k) - \dot{\hat{V}}_t^{\text{dis},k}(z_t^{\text{dis},k} - 1, \dot{w}_t^k), \tag{A.15}$$

where we define $\dot{\hat{V}}_t^{\text{rep},k}(-1, \dot{w}_t^k) = \dot{\hat{V}}_t^{\text{dis},k}(-1, \dot{w}_t^k) \equiv 0$. Under the assumption that $p^j(\cdot)$ is a uniform density function for all j, an algorithm for the aggregate problem is given in Algorithm 7.

Algorithm 7: Aggregate Structured Actor-Critic Method

Input: Lower-level approximate policy $\bar{\mu}$ (learned from backward dynamic programming in the aggregate state space and extrapolated to continuous state space by linear regression). Initial policy estimates $\dot{\bar{l}}^{\text{rep},0}$ and $\dot{\bar{\pi}}^{\text{dis},0}$, and value estimates $\dot{\bar{v}}^{\text{rep},0}$ and $\dot{\bar{v}}^{\text{dis},0}$ (nonincreasing in z^{rep} and z^{dis} respectively). Stepsize rules $\tilde{\alpha}_t^k$ and $\tilde{\beta}_t^k$ for all t, k.

Output: Approximations $\{\dot{\bar{l}}^{\text{rep},k}\}$, $\{\dot{\bar{\pi}}^{\text{dis},k}\}$, $\{\dot{\bar{v}}^{\text{rep},k}\}$, and $\{\dot{\bar{v}}^{\text{dis},k}\}$.

for $k = 1, 2, \ldots$ **do**

  Sample initial states $z_0^{\text{rep},k}$ and $z_0^{\text{dis},k}$.

  for $t = 0, 1, \ldots, T-1$ **do**

    Observe w_t^k and $\xi_{t,1}^k$, then observe $\dot{\hat{v}}_t^{\text{rep},k}$ and $\dot{\hat{v}}_t^{\text{dis},k}$ according to (A.14) and (A.15) respectively.

    Perform SA step:

    $\dot{\tilde{v}}_t^{\text{rep},k}(z^{\text{rep}}, \dot{w}) = \big(1 - \alpha_t^k(z^{\text{rep}}, \dot{w})\big)\, \dot{\bar{v}}_t^{\text{rep},k-1}(z^{\text{rep}}, \dot{w}) + \alpha_t^k(z^{\text{rep}}, \dot{w})\, \dot{\hat{v}}_t^{\text{rep},k}$,

    $\dot{\tilde{v}}_t^{\text{dis},k}(z^{\text{dis}}, \dot{w}) = \big(1 - \alpha_t^k(z^{\text{dis}}, \dot{w})\big)\, \dot{\bar{v}}_t^{\text{dis},k-1}(z^{\text{dis}}, \dot{w}) + \alpha_t^k(z^{\text{dis}}, \dot{w})\, \dot{\hat{v}}_t^{\text{dis},k}$.

    Perform the concavity projection operation (16):

    $\dot{\bar{v}}_t^{\text{rep},k} = \Pi_{z_t^{\text{rep},k}, \dot{w}_t^k}(\dot{\tilde{v}}_t^{\text{rep},k}), \quad \dot{\bar{v}}_t^{\text{dis},k} = \Pi_{z_t^{\text{dis},k}, \dot{w}_t^k}(\dot{\tilde{v}}_t^{\text{dis},k})$.

    Observe and update the replenish-up-to threshold:

    $\dot{\hat{l}}_t^{\text{rep},k} = \arg\max_{z^{\text{rep}} \in \bar{\mathcal{Z}}(0)} \sum_{j=0}^{z^{\text{rep}}} \dot{\bar{v}}_t^{\text{rep},k}\big(j, \dot{w}_t^k\big)$,

    $\dot{\bar{l}}_t^{\text{rep},k}(\dot{w}) = \big(1 - \beta_t^k(\dot{w})\big)\, \dot{\bar{l}}_t^{\text{rep},k-1}(\dot{w}) + \beta_t^k(\dot{w})\, \dot{\hat{l}}_t^{\text{rep},k}$.

    Observe and update the dispense-down-to policy:

    for $z_t^{\text{rep}} = 0, 1, \ldots, R_{\max}$ **do**

      $\dot{\hat{\pi}}_t^{\text{dis}} = \arg\max_{z^{\text{dis}} \in \underline{\mathcal{Z}}(z_t^{\text{rep}})} U_{w_t^k, 0}^{\bar{\mu}}\big(z_t^{\text{rep}} - z^{\text{dis}}, \xi_{t,0}^k\big) + \sum_{j=0}^{z^{\text{dis}}} \dot{\bar{v}}_t^{\text{dis},k}\big(j, \dot{w}_t^k\big)$,

      $\dot{\bar{\pi}}_t^{\text{dis},k}(z^{\text{rep}}, \dot{w}) = \big(1 - \alpha^k(z^{\text{rep}}, \dot{w})\big)\, \dot{\bar{\pi}}_t^{\text{dis},k-1}(z^{\text{rep}}, \dot{w}) + \alpha^k(z^{\text{rep}}, \dot{w})\, \dot{\hat{\pi}}_t^{\text{dis}}$.

    end

    If $t < T-1$, take $z_{t+1}^{\text{rep},k}$ and $z_{t+1}^{\text{dis},k}$ according to the ϵ-greedy exploration policy.

  end

end

Appendix B

B.1 Proofs for Chapter 3

B.1.1 Proof of Theorem 3.3.1

The proof is based on theoretical results of [111]. Our result, however, includes the ability to select the number of replications q. Denote $\lambda(\theta, \tau, q) = \sigma^2_{\text{env}} + \sigma^2_{\text{rep}}/q$. Also, let $\mathscr{F}^n$ denote the σ-algebra generated by the history H^n. The expectation $\mathbf{E}_n := \mathbf{E}[\,\cdot\,|\mathscr{F}^n]$ is taken with respect to $\mathscr{F}^n$. Recall that μ^n and k^n are the mean and covariance matrix of the time n belief on f. Define the quantities

$$Z^{n+1} = \frac{y^{n+1}(\theta,\tau) - \mu^n(\theta,\tau)}{\sqrt{\text{Var}\big[y^{n+1}(\theta,\tau) - \mu^n(\theta,\tau)\,|\,\mathscr{F}^n\big]}},$$

and

$$\tilde{\sigma}^n_q\big((\theta',\tau'),(\theta,\tau)\big) = \frac{k^n\big((\theta',\tau'),(\theta,\tau)\big)}{\sqrt{\lambda(\theta,\tau,q) + k^n\big((\theta,\tau),(\theta,\tau)\big)}}.$$

Observe that Z^{n+1} is standard normal (conditional on $\mathscr{F}^n$). We have the following recursive updating equation for μ^{n+1}:

$$\mu^{n+1}(\theta,\tau) = \mu^n(\theta,\tau) + \tilde{\sigma}^n_{q^{n+1}}\big((\theta,\tau),(\theta^{n+1},\tau^{n+1})\big)\, Z^{n+1}, \tag{B.1}$$

and another recursive formula k^{n+1}:

$$\begin{aligned} k^{n+1}\big((\theta',\tau'),(\theta,\tau)\big) &= k^n\big((\theta',\tau'),(\theta,\tau)\big) \\ &\quad - \tilde{\sigma}^n_{q^{n+1}}\big((\theta',\tau'),(\theta^{n+1},\tau^{n+1})\big)\,\big[\tilde{\sigma}^n_{q^{n+1}}\big((\theta,\tau),(\theta^{n+1},\tau^{n+1})\big)\big]^\top. \end{aligned} \tag{B.2}$$

These updating equations are based on the Sherman-Woodbury identity; see [176] for a full derivation. The objective of the acquisition function is thus:

$$\frac{\nu^n(\theta,\tau,q)}{q\tau} = \frac{1}{q\tau}\,\mathbf{E}_n\big[(\mu^{n+1}_* - \mu^n_*)\,|\,(\theta^n,\tau^n,q^n) = (\theta,\tau,q)\big]$$

$$= \frac{1}{q\tau}\,\mathbf{E}_n\Big[\max_{\theta'}\big\{\mu^n(\theta',\tau_{\max}) + \tilde{\sigma}^n_q\big((\theta',\tau_{\max}),(\theta,\tau)\big)\,Z^{n+1}\big\} \\ - \max_{\theta'}\mu^n(\theta',\tau_{\max})\;\Big|\;(\theta^n,\tau^n,q^n) = (\theta,\tau,q)\Big]. \tag{B.3}$$

We also define the quantity

$$V^n(\theta,\tau,\theta',\tau') = \mathbf{E}_n[f(\theta,\tau)\cdot f(\theta',\tau')] = k^n\big((\theta,\tau),(\theta',\tau')\big) + \mu^n(\theta,\tau)\cdot\mu^n(\theta',\tau'). \tag{B.4}$$

Next, we restate a useful technical lemma from [111].

Lemma B.1.1 (Restatement of Lemma 1 of [111])**.** *Let $\tau,\tau' \in \mathcal{T}$ and $\theta,\theta' \in \Theta$. The limits of the series $\{\mu^n(\theta,\tau)\}_n$ and $\{V^n(\theta,\tau,\theta',\tau')\}_n$ exist. Denote them by $\mu^\infty(\theta,\tau)$ and $V^\infty(\theta,\tau,\theta',\tau')$ respectively. We have*

$$\lim_{n\to\infty}\mu^n(\theta,\tau) = \mu^\infty(\theta,\tau), \tag{B.5}$$

$$\lim_{n\to\infty}V^n(\theta,\tau,\theta',\tau') = V^\infty(\theta,\tau,\theta',\tau') \tag{B.6}$$

almost surely. If (θ',τ') is sampled infinitely often, then

$$\lim_{n\to\infty}V^n(\theta,\tau,\theta',\tau') = \mu^\infty(\theta,\tau)\cdot\mu^\infty(\theta',\tau').$$

Fix a sample path ω, which corresponds to a particular path of measurements and observations

$$\{(\theta^n,\tau^n,q^n,y^{n+1}(\theta^n,\tau^n,q^n))\}_n.$$

By the finiteness of $\bar{\Theta}$, $\mathcal{T}$, and $\mathcal{Q}$, there must exist a configuration (θ',τ',q') that is visited infinitely often on sample path ω. The following lemma states the asymptotic behavior of $\nu^n(\theta',\tau',q')/(q'\tau')$ for $n\to\infty$ as a function of $\mu^n(\cdot,\cdot)$ and $\tilde{\sigma}^n_{\cdot}\big((\cdot,\cdot),(\cdot,\cdot)\big)$.

Lemma B.1.2. *Consider the sample path ω and (θ',τ',q') described above. Then, on that sample path ω, it holds that*

$$\lim_{n\to\infty}\tilde{\sigma}^n_{q'}\big((\theta'',\tau_{max}),(\theta',\tau')\big) = 0$$

for every $\theta'' \in \Theta$. Also, the acquisition value tends to zero: $\lim_{n\to\infty}\nu^n(\theta',\tau',q')/(q'\tau') = 0$

Proof. It follows from Lemma B.1.1 that

$$k^n\big((\theta,\tau),(\theta',\tau')\big) = \mathbf{E}_n[f(\theta,\tau)\cdot f(\theta',\tau')] - \mu^n(\theta,\tau)\cdot\mu^n(\theta',\tau') \xrightarrow{n\to\infty} 0$$

for any $\theta\in\Theta$, $\tau\in\mathcal{T}$. Then for all $\theta''\in\bar{\Theta}$, we have

$$\lim_{n\to\infty}\tilde{\sigma}^n_{q'}\big((\theta'',\tau_{\max}),(\theta',\tau')\big) = \lim_{n\to\infty}\frac{k^n\big((\theta'',\tau_{\max}),(\theta',\tau')\big)}{\sqrt{\lambda\big(\theta',\tau',q'\big)+k^n\big((\theta',\tau'),(\theta',\tau')\big)}} = 0.$$

Note that we made use of the fact that the observation noise $\lambda(\theta',\tau',q')>0$ for any q'. From the proof of Lemma 1 of [111], it is shown that for any $\theta''\in\bar{\Theta}$,

$$\big\{\mu^n(\theta'',\tau_{\max})\big\}_n \quad\text{and}\quad \big\{\tilde{\sigma}^n_{q'}\big((\theta'',\tau_{\max}),(\theta',\tau')\big)\big\}_n$$

are uniformly integrable (u.i.) families of random variables that converge almost surely to their respective limits $\mu^\infty(\theta'',\tau_{\max})$ and $\tilde{\sigma}^\infty_{q'}\big((\theta'',\tau_{\max}),(\theta',\tau')\big)=0$. Note that the family of random variables $\big\{\tilde{\sigma}^n_{q'}\big((\theta'',\tau_{\max}),(\theta',\tau',)\big)\,Z^{n+1}\big\}_n$ is also uniformly integrable since Z^{n+1} is independent of $\tilde{\sigma}^n_{q'}\big((\theta'',\tau_{\max}),(\theta',\tau')\big)$. Let Z be a standard normal random variable (independent from all other quantities). It holds that

$$\begin{aligned}
&\lim_{n\to\infty}\frac{\nu^n(\theta',\tau',q')}{q'\tau'}\\
&\quad= \frac{1}{q'\tau'}\Big[\int_{-\infty}^{+\infty}\phi(Z)\max_{\theta''\in\bar{\Theta}}\Big\{\mu^\infty(\theta'',\tau_{\max})+\tilde{\sigma}^\infty_{q'}\big((\theta'',\tau_{\max}),(\theta',\tau')\big)\,Z\Big\}\,dZ \\
&\qquad\qquad -\max_{\theta''\in\bar{\Theta}}\mu^\infty(\theta'',\tau_{\max})\Big] \qquad\qquad (B.7)\\
&\quad= 0.
\end{aligned}$$

The first equality is due to (B.3) and the fact that the operations of summing and taking maximum over a finite set of uniform integrable random variables maintains uniform integrability. □

From (3.7), we know that in each iteration n, the configuration (θ^n, τ^n, q^n) is selected from according to $\arg\max_{\theta,\tau,q} \nu^n(\theta,\tau,q)/(q\tau)$. Now, for the sake of contradiction, suppose that there exists some configuration $(\breve{\theta}, \breve{\tau}, \breve{q})$ such that $\lim_{n\to\infty} \nu^n(\breve{\theta}, \breve{\tau}, \breve{q})/(\breve{q}\breve{\tau}) > 0$. This immediately leads to a contradiction, since then it cannot be the case that (θ', τ', q') is visited infinitely often.

Since the sample path ω was arbitrary, we conclude that

$$\lim_{n\to\infty} \nu^n(\theta,\tau,q)/(q\tau) = 0 \quad \text{a.s.} \tag{B.8}$$

for all $\theta \in \bar{\Theta}$, $\tau \in \mathcal{T}$, and $q \in \mathcal{Q}$.

Lemma B.1.3. *Given that (B.8) holds, we have that*

$$\arg\max_{\theta\in\bar{\Theta}} \mu^\infty(\theta, \tau_{max}) = \arg\max_{\theta\in\bar{\Theta}} f(\theta, \tau_{max})$$

almost surely.

Proof. We can conclude from (B.4) and Lemma B.1.1 that

$$\lim_{n\to\infty} k_n\big((\theta,\tau_{\max}),(\theta,\tau_{\max})\big) = k^\infty\big((\theta,\tau_{\max}),(\theta,\tau_{\max})\big) \quad \text{a.s.}$$

for all $\theta \in \bar{\Theta}$. In the case that the posterior variance $k^\infty((\theta,\tau_{\max}),(\theta,\tau_{\max})) = 0$ for all $\theta \in \bar{\Theta}$, then the maximizer is known perfectly and we are done.

If not, then we define $\hat{\Theta} = \big\{\theta \in \bar{\Theta} \,|\, k^\infty((\theta,\tau_{\max}),(\theta,\tau_{\max})) > 0\big\}$ and consider some $\hat{\theta} \in \hat{\Theta}$ where the posterior variance is positive. Fix any $\hat{q} \in \mathcal{Q}$. We now argue that

$$\tilde{\sigma}^\infty_{\hat{q}}\big((\hat{\theta},\tau_{\max}),(\hat{\theta},\tau_{\max})\big) = \tilde{\sigma}^\infty_{\hat{q}}\big((\theta'',\tau_{\max}),(\hat{\theta},\tau_{\max})\big) \tag{B.9}$$

for all $\theta'' \in \bar{\Theta}$. Suppose, for the sake of contradiction, that there exist some $\theta_1, \theta_2 \in \bar{\Theta}$ with

$$\tilde{\sigma}^\infty_{\hat{q}}\big((\theta_1,\tau_{\max}),(\hat{\theta},\tau_{\max})\big) \neq \tilde{\sigma}^\infty_{\hat{q}}\big((\theta_2,\tau_{\max}),(\hat{\theta},\tau_{\max})\big). \tag{B.10}$$

Recall (B.7) and note that it can be rewritten as

$$\lim_{n\to\infty} \frac{\nu^n(\theta',\tau',q')}{q'\tau'} = \frac{1}{q'\tau'}\Big[\mathbb{E}\big[h(Z)\big] - \max_{\theta''\in\bar{\Theta}} \mu^\infty(\theta'',\tau_{\max})\Big], \tag{B.11}$$

where $h(z) = \max_{\theta'' \in \bar{\Theta}} \Big\{ \mu^{\infty}(\theta'', \tau_{\max}) + \tilde{\sigma}^{\infty}_{q'}\big((\theta'', \tau_{\max}), (\theta', \tau')\big)\, z \Big\}$. Since $\bar{\Theta}$ is finite and each function within the maximization in h is affine in z, the $h(z)$ is convex[1] and piecewise linear. Since h is convex, there is an affine function l such that

$$l(0) = h(0), \quad l(z) \leq h(z) \;\text{ for all }\; z \in \mathbb{R}.$$

The assumption we made in (B.10), which effectively says that the h is created by taking maximum over affine functions of *differing slopes*, implies h cannot itself be affine (and indeed, must consist of various "pieces"). Therefore, there exists an interval $\mathcal{I}$, either of the form (z_0, ∞) or $(-\infty, z_0)$, such that $l(z) < h(z)$ for $z \in \mathcal{I}$. It follows that $\mathbb{E}[l(Z)] < \mathbb{E}[h(Z)]$. By th linearity of l, we have

$$\mathbb{E}[l(Z)] = l(\mathbb{E}[Z]) = l(0) = h(0) = \max_{\theta'' \in \bar{\Theta}} \mu^{\infty}(\theta'', \tau_{\max}) < \mathbb{E}\big[h(Z)\big].$$

This implies that (B.11) is strictly positive, contradicting (B.8). We thus conclude that (B.9) holds, which is equivalent to

$$\frac{k^{\infty}\big((\theta'', \tau_{\max}), (\hat{\theta}, \tau_{\max})\big)}{\sqrt{\lambda(\hat{\theta}, \tau_{\max}, \hat{q}) + k^{\infty}\big((\hat{\theta}, \tau_{\max}), (\hat{\theta}, \tau_{\max})\big)}} = \frac{k^{\infty}\big((\theta''', \tau_{\max}), (\hat{\theta}, \tau_{\max})\big)}{\sqrt{\lambda(\hat{\theta}, \tau_{\max}, \hat{q}) + k^{\infty}\big((\hat{\theta}, \tau_{\max}), (\hat{\theta}, \tau_{\max})\big)}},$$

for all $\theta'', \theta''' \in \bar{\Theta}$. Moreover, since $\hat{\theta}$ was chosen from $\hat{\Theta}$, we know that

$$\lambda(\hat{\theta}, \tau_{\max}, \hat{q}) + k^{\infty}\big((\hat{\theta}, \tau_{\max}), (\hat{\theta}, \tau_{\max})\big) > 0,$$

and hence $k^{\infty}\big((\theta''', \tau_{\max}), (\hat{\theta}, \tau_{\max})\big) = k^{\infty}\big((\theta'', \tau_{\max}), (\hat{\theta}, \tau_{\max})\big)$ for all $\theta'', \theta''' \in \bar{\Theta}$.

This means the covariance matrix of $\{f(\theta, \tau_{\max}) \,|\, \theta \in \bar{\Theta}\}$ is proportional to the all-ones matrix, and that draws from $f(\theta, \tau_{\max}) - \mu^{(\infty)}(\theta, \tau_{\max})$ are *constant* across $\theta \in \bar{\Theta}$. Therefore, $\arg\max_{\theta \in \bar{\Theta}} \mu^{(\infty)}(\theta, \tau_{\max}) = \arg\max_{\theta \in \bar{\Theta}} f(\theta, \tau_{\max})$ and the statement of the theorem holds. □

[1]Pointwise maximum of convex functions is convex.

B.1.2 Proof of Theorem 3.3.2

In Theorem 3.3.2, we establish an additive bound on the loss of the solution obtained by `BESD`, $f(\bar{\theta}, \tau_{\max})$, with respect to the unknown optimum $f(\theta^{\mathrm{OPT}}, \tau_{\max})$, as the number of iterations $N \to \infty$. Recall that we suppose $\mu(\theta, \tau) = 0$ for all θ, τ, and that the kernel $k(\cdot, \cdot)$ has continuous partial derivatives up to the fourth order. According to Theorem 3.2 of [177], for any $\delta \in (0, 1]$, with probability at least $1 - \delta$, the quantity

$$\|L_\delta\| = \left\|(L_\delta^1, L_\delta^2, \cdots, L_\delta^m)\right\|$$

is a Lipschitz constant of f on Θ , i.e., it holds that

$$|f(\theta, \tau_{\max}) - f(\theta', \tau_{\max})| \leq \|L_\delta\| \cdot \mathrm{dist}(\theta, \theta'),$$

where $\theta, \theta' \in \Theta$. By the definition of d, there exists a $\bar{\theta} \in \bar{\Theta}$ such that $\mathrm{dist}(\bar{\theta}, \theta^{\mathrm{OPT}}) \leq d$. Therefore, it follows that the suboptimality due to optimizing in $\bar{\Theta}$ is bounded by

$$f(\theta^{\mathrm{OPT}}, \tau_{\max}) - f(\bar{\theta}, \tau_{\max}) \leq \|L_\delta\| \cdot d. \tag{B.12}$$

Theorem 3.3.1 completes the proof of Theorem 3.3.2 since (B.12) holds with probability $1 - \delta$.

Appendix C

C.1 Proofs for Chapter 4

C.1.1 Additional Lemmas

Lemma C.1.1. *Consider a (α, d_y, L_r, L_f)-fast-slow MDP. For any states (x_0, y_0) and $(\tilde{x}_0, \tilde{y}_0)$, let (x_t, y_t) and $(\tilde{x}_t, \tilde{y}_t)$ be the states reached after t transitions under a policy $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_{t-1})$, i.e., $(x_t, y_t) = f^{\boldsymbol{\pi}}(x_{t-1}, y_{t-1}, w_t)$ and $(\tilde{x}_t, \tilde{y}_t) = f^{\boldsymbol{\pi}}(\tilde{x}_{t-1}, \tilde{y}_{t-1}, w_t)$. Then, for any policy $\boldsymbol{\pi}$, we have*

(i) $\|x_t - \tilde{x}_0\|_2 \leq t\alpha d_y + \|x_0 - \tilde{x}_0\|_2$,

(ii) $\|x_t - \tilde{x}_t\|_2 \leq 2t\alpha d_y + \|x_0 - \tilde{x}_0\|_2$,

(iii) $\|y_t - \tilde{y}_t\|_2 \leq 2td_y + \|y_0 - \tilde{y}_0\|_2$.

Proof. Lemma C.1.1 is a consequence of Assumption 4.2.1. □

C.1.2 Proof of Proposition 4.2.1

We consider an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$ and note that U^* is the unique optimal solution of the base model (4.5), and there exists a stationary optimal policy $\nu^*(x, y) = \arg\max U^*(x, y)$ that attains this optimal value [236, Proposition 4.3]. Fix a state $s_0 \in \mathcal{S}$ and for $t > 0$ and a sequence of policies $\pi_0, \ldots, \pi_{t-1}$, define the notation:

$$s_1(\pi_0) = f^{\pi_0}(s_0, w_1) \quad \text{and} \quad s_{t'+1}(\pi_0, \ldots, \pi_{t'}) = f^{\pi_{t'}}(s_{t'}(\pi_0, \ldots, \pi_{t'-1}), w_{t'+1})$$

for $t' \geq 1$. Therefore, we have

$$U^*(s_0) = \max_{\pi_0} r(s_0, \pi_0) + \gamma \, \mathbf{E}\big[U^*(s_1(\pi_0))\big] = r(s_0, \nu^*) + \gamma \, \mathbf{E}\big[U^*(s_1(\nu^*))\big]. \tag{C.1}$$

By expanding the $U^*(s_1(\pi_0))$ and $U^*(s_1(\nu^*))$ terms in (C.1), we have the following:

$$\begin{aligned} U^*(s_0) &= \max_{\pi_0,\pi_1} \mathbf{E}\Big[r(s_0,\pi_0) + \gamma\, r(s_1(\pi_0),\pi_1) + \gamma^2\, U^*(s_2(\pi_0,\pi_1))\Big] \\ &= \mathbf{E}\Big[r(s_0,\nu) + \gamma\, r(s_1(\nu^*),\nu^*) + \gamma^2\, U^*(s_2(\nu^*,\nu^*))\Big]. \end{aligned}$$

Let $\boldsymbol{\pi} = (\pi_0, \pi_1, \ldots, \pi_{T-1})$. Repeating the expansion, we obtain:

$$U^*(s_0) = \max_{\boldsymbol{\pi}} \mathbf{E}\left[\sum_{t=0}^{T-1} \gamma^t\, r\big(s_t(\pi_0,\ldots,\pi_{t-1}),\pi_t\big) + \gamma^T U^*\big(s_T(\pi_0,\ldots,\pi_{T-1})\big)\right] \tag{C.2}$$

$$= \mathbf{E}\left[\sum_{t=0}^{T-1} \gamma^t\, r\big(s_t(\nu^*,\ldots,\nu^*),\nu^*\big) + \gamma^T\, U^*\big(s_T(\nu^*,\ldots,\nu^*)\big)\right]. \tag{C.3}$$

Observe that (C.2) is in same form as the Bellman equation (4.8) for the hierarchical reformulation (with T-horizon reward function R and value function $\bar{U}$), which has a unique optimal solution $\bar{U}^*$. Therefore $U^*(s_0) = \bar{U}^*(s_0)$ and (i) is proved when we recall that s_0 was chosen arbitrarily. Part (ii) follows because by (C.3), it is clear that $(\nu^*,\ldots,\nu^*)$ solves (C.2) and hence also (4.8).

C.1.3 Proof of Lemma 4.3.1

Let $\hat{s} = \arg\max_{s\in\mathcal{S}} \big|U_1^*(s) - U_2^*(s)\big|$. We have $\big|U_1^*(s) - U_2^*(s)\big| \le \big|U_1^*(\hat{s}) - U_2^*(\hat{s})\big|$. Let us show the bound of $\big|U_1^*(\hat{s}) - U_2^*(\hat{s})\big|$.

$$\begin{aligned} &\big|U_1^*(\hat{s}) - U_2^*(\hat{s})\big| \\ &= \big|\max_{a\in\mathcal{A}}\big(r_1(\hat{s},a) + \gamma\mathbf{E}[U_1^*(f_1(\hat{s},a,w))]\big) - \max_{b\in\mathcal{A}}\big(r_2(\hat{s},b) + \gamma\mathbf{E}[U_2^*(f_2(\hat{s},b,w))]\big)\big| \\ &\le \max_{a\in\mathcal{A}}\big|r_1(\hat{s},a) + \gamma\mathbf{E}[U_1^*(f_1(\hat{s},a,w))] - r_2(\hat{s},a) - \gamma\mathbf{E}[U_2^*(f_2(\hat{s},a,w))]\big| \\ &\le \max_{a\in\mathcal{A}}\big|r_1(\hat{s},a) - r_2(\hat{s},a)\big| + \max_{a\in\mathcal{A}}\gamma\big|\mathbf{E}[U_1^*(f_1(\hat{s},a,w))] - \mathbf{E}[U_2^*(f_2(\hat{s},a,w))]\big| \\ &\le \epsilon_r + \max_{a\in\mathcal{A}}\gamma\big|\mathbf{E}[U_1^*(f_1(\hat{s},a,w))] - \mathbf{E}[U_1^*(f_2(\hat{s},a,w))]\big| \\ &\quad + \max_{a\in\mathcal{A}}\gamma\big|\mathbf{E}[U_1^*(f_2(\hat{s},a,w))] - \mathbf{E}[U_2^*(f_2(\hat{s},a,w))]\big| \\ &\le \epsilon_r + \max_{a\in\mathcal{A},w\in\mathcal{W}}\gamma L_1\|f_1(\hat{s},a,w) - f_2(\hat{s},a,w)\| + \max_{a\in\mathcal{A}}\gamma\big|U_1^*(\hat{s}) - U_2^*(\hat{s})\big| \\ &\le \epsilon_r + \gamma L_1 d + \gamma\big|U_1^*(\hat{s}) - U_2^*(\hat{s})\big|. \end{aligned}$$

Therefore,

$$\big|U_1^*(\hat{s}) - U_2^*(\hat{s})\big| \le \frac{\epsilon_r + \gamma L_1 d}{1-\gamma}.$$

C.1.4 Proof of Lemma 4.3.2

The proof follows the technique of Corollary 1 in [237]. Let π_1^* be an optimal policy for MDP_1. Since π_2^* is the optimal policy for MDP_2, for any $s \in \mathcal{S}$,

$$r_2(s, \pi_1^*(s)) + \gamma \mathbf{E}[U_2^*(f_2(s, \pi_1^*(s), w))] \leq r_2(s, \pi_2^*(s)) + \gamma \mathbf{E}[U_2^*(f_2(s, \pi_2^*(s), w))].$$

According to the assumptions of the lemma, for any s and a, $U_1^*(s) - \epsilon_U \leq U_2^*(s) \leq U_1^*(s) + \epsilon_U$, and $r_1(s,a) - \epsilon_r \leq r_2(s,a) \leq r_1(s,a) + \epsilon_r$. Therefore,

$$\begin{aligned} r_1(s, \pi_1^*(s)) - \epsilon_r + \gamma(\mathbf{E}[U_1^*(f_2(s, \pi_1^*(s), w))] - \epsilon_U) \\ \leq r_1(s, \pi_2^*(s)) + \epsilon_r + \gamma(\mathbf{E}[U_1^*(f_2(s, \pi_2^*(s), w))] + \epsilon_U), \end{aligned}$$

which can be transformed into

$$\begin{aligned} r_1(s, \pi_1^*(s)) - r_1(s, \pi_2^*(s)) \leq 2\epsilon_r + 2\gamma\epsilon_U + \gamma\big(\mathbf{E}[U_1^*(f_2(s, \pi_2^*(s), w))] \\ - \mathbf{E}[U_1^*(f_2(s, \pi_1^*(s), w))]\big). \end{aligned} \tag{C.4}$$

Let state $\hat{s}$ be the state that achieves the maximum loss, i.e., $L_{\pi_2^*}(\hat{s}) \geq L_{\pi_2^*}(s)$ for all $s \in \mathcal{S}$. The maximum loss is

$$\begin{aligned} L_{\pi_2^*}(\hat{s}) &= U_1^*(\hat{s}) - U_1^{\pi_2^*}(\hat{s}) \\ &= r_1(\hat{s}, \pi_1^*(\hat{s})) - r_1(\hat{s}, \pi_2^*(\hat{s})) + \gamma\big(\mathbf{E}[U_1^*(f_1(\hat{s}, \pi_1^*(\hat{s}), w))] - \mathbf{E}[U_1^{\pi_2^*}(f_1(\hat{s}, \pi_2^*(\hat{s}), w))]\big). \end{aligned}$$

Substituting from (C.4) gives

$$\begin{aligned} L_{\pi_2^*}(\hat{s}) &\leq 2\epsilon_r + 2\gamma\epsilon_U + \gamma\big(\mathbf{E}[U_1^*(f_2(\hat{s}, \pi_2^*(\hat{s}), w))] - \mathbf{E}[U_1^*(f_2(\hat{s}, \pi_1^*(\hat{s}), w))] \\ &\quad + \mathbf{E}[U_1^*(f_1(\hat{s}, \pi_1^*(\hat{s}), w))] - \mathbf{E}[U_1^{\pi_2^*}(f_1(\hat{s}, \pi_2^*(\hat{s}), w))]\big) \\ &= 2\epsilon_r + 2\gamma\epsilon_U + \gamma\big(\mathbf{E}[U_1^*(f_2(\hat{s}, \pi_2^*(\hat{s}), w))] - \mathbf{E}[U_1^{\pi_2^*}(f_1(\hat{s}, \pi_2^*(\hat{s}), w))]\big) \\ &\quad + \gamma\big(\mathbf{E}[U_1^*(f_1(\hat{s}, \pi_1^*(\hat{s}), w))] - \mathbf{E}[U_1^*(f_2(\hat{s}, \pi_1^*(\hat{s}), w))]\big) \\ &\leq 2\epsilon_r + 2\gamma\epsilon_U + \gamma L_{\pi_2^*}(\hat{s}) + \gamma L_1 \max_{w \in \mathcal{W}} \|f_1(\hat{s}, \pi_1^*(\hat{s}), w) - f_2(\hat{s}, \pi_1^*(\hat{s}), w)\|_2 \\ &\leq 2\epsilon_r + 2\gamma\epsilon_U + \gamma L_{\pi_2^*}(\hat{s}) + \gamma L_1 d. \end{aligned}$$

Therefore,

$$L_{\pi_2^*}(\hat{s}) \leq \frac{2\epsilon_r + 2\gamma\epsilon_U + \gamma L_1 d}{1 - \gamma}.$$

C.1.5 Proof of Proposition 4.3.1

Lemma C.1.2. *Consider the base model $\langle \mathcal{X} \times \mathcal{Y}, \mathcal{A}, \mathcal{W}, f_{\mathcal{X}}, f_{\mathcal{Y}}, r, \gamma \rangle$ and let $U : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a value function such that there exists L_U, for any states (x, y) and $(\tilde{x}, \tilde{y})$,*

$$|U(x, y) - U(\tilde{x}, \tilde{y})| \leq L_U \|(x, y) - (\tilde{x}, \tilde{y})\|_2. \tag{C.5}$$

Define

$$Q(x, y, a) = r(x, y, a) + \gamma \mathbf{E}[U(f_{\mathcal{X}}(x, w), f_{\mathcal{Y}}(x, y, a, w))].$$

Then for any state-action pairs (x, y, a) and $(\tilde{x}, \tilde{y}, \tilde{a})$, Q satisfies

$$\big|Q(x, y, a) - Q(\tilde{x}, \tilde{y}, \tilde{a})\big| \leq (L_r + \gamma L_U L_f)(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a - \tilde{a}\|_2).$$

Proof. For any state-action pairs (x, y, a) and $(\tilde{x}, \tilde{y}, \tilde{a})$,

$$\begin{aligned}
&\big|Q(x, y, a) - Q(\tilde{x}, \tilde{y}, \tilde{a})\big| \\
&= \big|r(x, y, a) + \gamma \mathbf{E}[U(x', y') - r(\tilde{x}, \tilde{y}, \tilde{a}) - \gamma \mathbf{E}[U(\tilde{x}', \tilde{y}')\big| \\
&\leq \big|r(x, y, a) - r(\tilde{x}, \tilde{y}, \tilde{a})\big| + \gamma \big|\mathbf{E}[U(x', y') - \mathbf{E}[U(\tilde{x}', \tilde{y}')\big| \\
&\leq L_r(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a - \tilde{a}\|_2) + \gamma L_U \max_{x', y', \tilde{x}', \tilde{y}'} \|(x', y') - (\tilde{x}', \tilde{y}')\|_2 \qquad \text{(C.6)} \\
&\leq L_r(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a - \tilde{a}\|_2) + \gamma L_U L_f \|(x, y) - (\tilde{x}, \tilde{y})\|_2 \qquad \text{(C.7)} \\
&\leq (L_r + \gamma L_U L_f)(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a - \tilde{a}\|_2),
\end{aligned}$$

where (C.6) is from the lemma assumption, (C.7) is from (4.3). □

Lemma C.1.3. *For the base model $\langle \mathcal{X} \times \mathcal{Y}, \mathcal{A}, \mathcal{W}, f_{\mathcal{X}}, f_{\mathcal{Y}}, r, \gamma \rangle$, let $Q : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ be a Q-value function that there exists L_Q such that for any state-action pairs (x, y, a) and $(\tilde{x}, \tilde{y}, \tilde{a})$,*

$$\big|Q(x, y, a) - Q(\tilde{x}, \tilde{y}, \tilde{a})\big| \leq L_Q(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a - \tilde{a}\|_2).$$

Define

$$U(x, y) = \max_a Q(x, y, a).$$

Then for any states (x, y) and $(\tilde{x}, \tilde{y})$, U satisfy

$$\big|U(x, y) - U(\tilde{x}, \tilde{y})\big| \leq L_Q(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2).$$

Proof. Consider states (x, y) and $(\tilde{x}, \tilde{y})$,

$$\begin{aligned}\big|U(x,y) - U(\tilde{x},\tilde{y})\big| &= \big|\max_a Q(x,y,a) - \max_{\tilde{a}} Q(\tilde{x},\tilde{y},\tilde{a})\big| \\ &\leq \max_a \big|Q(x,y,a) - Q(\tilde{x},\tilde{y},a)\big| \\ &\leq L_Q(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2).\end{aligned}$$

□

Lemma C.1.4. *Consider the base model $\langle \mathcal{X} \times \mathcal{Y}, \mathcal{A}, \mathcal{W}, f_{\mathcal{X}}, f_{\mathcal{Y}}, r, \gamma \rangle$. Start with $U_0 = 0$ and recursively define Q_{k+1} and U_{k+1} as follows:*

$$Q_{k+1}(x,y,a) = r(x,y,a) + \gamma \mathbf{E}[U_k(f_{\mathcal{X}}(x,w), f_{\mathcal{Y}}(x,y,a,w))],$$

and

$$U_{k+1}(x,y) = \max_a Q_{k+1}(x,y,a).$$

Then U_k is Lipschitz continuous and its Lipschitz constant L_{U_k} satisfies

$$L_{U_k} = L_r + \gamma L_f L_{U_{k-1}}.$$

Proof. The proof is an induction. For $k = 1$, $|Q_1(x,y,a) - Q_1(\tilde{x},\tilde{y},\tilde{a})| = |r(x,y,a) - r(\tilde{x},\tilde{y},\tilde{a})| \leq L_r(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2 + \|a-\tilde{a}\|_2)$ by Property 4.2. Then, $|U_1(x,y) - U_1(\tilde{x},\tilde{y})| \leq L_r(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2)$ by Lemma C.1.3.

Now, assume that L_{U_k} satisfy

$$L_{U_k} = L_r + \gamma L_f L_{U_{k-1}}.$$

Then, by Lemma C.1.2, Q_{k+1} is $(L_r + \gamma L_f L_{U_k})$-Lipschitz continuous. By Lemma C.1.3, U_{k+1} is $(L_r + \gamma L_f L_{U_k})$-Lipschitz continuous. □

According to Proposition 7.3.1 of [238], the value U_k in Lemma C.1.4 converges to the optimal value U^*. Let $k \to \infty$, we have that for any states (x, y) and $(\tilde{x}, \tilde{y})$,

$$|U^*(x,y) - U^*(\tilde{x},\tilde{y})| \leq \frac{L_r}{1-\gamma L_f}(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2).$$

C.1.5.1 The Case that $\gamma L_f \geq 1$ Next, we consider the case that $\gamma L_f \geq 1$, optimal value at different states $|U^*(x,y) - U^*(\tilde{x},\tilde{y})|$ cannot be bounded by the technique above. Instead, we use Proposition C.1.1.

Proposition C.1.1. *The optimal value U^* of the base model (4.5) satisfies that, for any states (x,y) and $(\tilde{x},\tilde{y})$,*

$$|U^*(x,y) - U^*(\tilde{x},\tilde{y})| \leq \frac{1}{1-\gamma} L_r(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty) + \Delta_U, \tag{C.8}$$

where $\Delta_U = \frac{2}{(1-\gamma)^2}(\alpha+1)d_y L_r$.

To prove Proposition C.1.1, we need the following lemmas.

Lemma C.1.5. *Consider the base model $\langle \mathcal{X}\times\mathcal{Y}, \mathcal{A}, \mathcal{W}, f_\mathcal{X}, f_\mathcal{Y}, r, \gamma\rangle$ and let $U : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ be a value function such that there exist $L_U, \Delta > 0$ such that for any states (x,y) and $(\tilde{x},\tilde{y})$,*

$$|U(x,y) - U(\tilde{x},\tilde{y})| \leq L_U(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty) + \Delta. \tag{C.9}$$

Define

$$Q(x,y,a) = r(x,y,a) + \gamma \mathbf{E}[U(f_\mathcal{X}(x,w), f_\mathcal{Y}(x,y,a,w))].$$

Then for any state-action pairs (x,y,a) and $(\tilde{x},\tilde{y},\tilde{a})$, Q satisfies

$$\begin{aligned}
&\big|Q(x,y,a) - Q(\tilde{x},\tilde{y},\tilde{a})\big| \\
&\quad \leq (L_r + \gamma L_U)(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty + \|a-\tilde{a}\|_\infty) + 2(\alpha+1)d_y\gamma L_U + \gamma\Delta.
\end{aligned}$$

Proof. For any state-action pairs (x,y,a) and $(\tilde{x},\tilde{y},\tilde{a})$,

$$\begin{aligned}
&\big|Q(x,y,a) - Q(\tilde{x},\tilde{y},\tilde{a})\big| \\
&= \big|r(x,y,a) + \gamma\mathbf{E}[U(x',y') - r(\tilde{x},\tilde{y},\tilde{a}) - \gamma\mathbf{E}[U(\tilde{x}',\tilde{y}')\big| \\
&\leq \big|r(x,y,a) - r(\tilde{x},\tilde{y},\tilde{a})\big| + \gamma\big|\mathbf{E}[U(x',y') - \mathbf{E}[U(\tilde{x}',\tilde{y}')\big| \\
&\leq L_r(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty + \|a-\tilde{a}\|_\infty) \\
&\quad + \gamma L_U \max_{x',\tilde{x}',y',\tilde{y}'} \big(\|x'-\tilde{x}'\|_\infty + \|y'-\tilde{y}'\|_\infty\big) + \gamma\Delta \qquad \text{(C.10)} \\
&\leq L_r(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty + \|a-\tilde{a}\|_\infty) \\
&\quad + \gamma L_U\big(2(\alpha+1)d_y + \|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty\big) + \gamma\Delta \qquad \text{(C.11)}
\end{aligned}$$

$$\leq (L_r + \gamma L_U)(\|x - \tilde{x}\|_\infty + \|y - \tilde{y}\|_\infty + \|a - \tilde{a}\|_\infty) + 2(\alpha + 1)d_y \gamma L_U + \gamma\Delta,$$

where (C.10) is from Property 4.2, (C.11) is from Lemma C.1.1. □

Lemma C.1.6. *For the base model $\langle \mathcal{X} \times \mathcal{Y}, \mathcal{A}, \mathcal{W}, f_\mathcal{X}, f_\mathcal{Y}, r, \gamma \rangle$, let $Q : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ be a Q-value function that there exist $L_Q, \Delta > 0$ such that for any state-action pairs (x, y, a) and $(\tilde{x}, \tilde{y}, \tilde{a})$,*

$$\big|Q(x, y, a) - Q(\tilde{x}, \tilde{y}, \tilde{a})\big| \leq L_Q(\|x - \tilde{x}\|_\infty + \|y - \tilde{y}\|_\infty + \|a - \tilde{a}\|_\infty) + \Delta.$$

Define

$$U(x, y) = \max_a Q(x, y, a).$$

Then for any states (x, y) and $(\tilde{x}, \tilde{y})$, U satisfy

$$\big|U(x, y) - U(\tilde{x}, \tilde{y})\big| \leq L_Q(\|x - \tilde{x}\|_\infty + \|y - \tilde{y}\|_\infty) + \Delta.$$

Proof. Consider states (x, y) and $(\tilde{x}, \tilde{y})$,

$$\begin{aligned}
\big|U(x, y) - U(\tilde{x}, \tilde{y})\big| &= \big|\max_a Q(x, y, a) - \max_{\tilde{a}} Q(\tilde{x}, \tilde{y}, \tilde{a})\big| \\
&\leq \max_a \big|Q(x, y, a) - Q(\tilde{x}, \tilde{y}, a)\big| \\
&\leq L_Q(\|x - \tilde{x}\|_\infty + \|y - \tilde{y}\|_\infty) + \Delta.
\end{aligned}$$

□

Lemma C.1.7. *Consider the base model $\langle \mathcal{X} \times \mathcal{Y}, \mathcal{A}, \mathcal{W}, f_\mathcal{X}, f_\mathcal{Y}, r, \gamma \rangle$. Start with $U_0 = 0$ and recursively define Q_{k+1} and U_{k+1} as follows:*

$$Q_{k+1}(x, y, a) = r(x, y, a) + \gamma \mathbf{E}[U_k(f_\mathcal{X}(x, w), f_\mathcal{Y}(x, y, a, w))],$$

and

$$U_{k+1}(x, y) = \max_a Q_{k+1}(x, y, a).$$

Then for any states (x, y) and $(\tilde{x}, \tilde{y})$,

$$|U_k(x, y) - U_k(\tilde{x}, \tilde{y})| \leq L_r \sum_{i=0}^{k-1} \gamma^i (\|x - \tilde{x}\|_\infty + \|y - \tilde{y}\|_\infty) + 2(\alpha + 1)d_y L_r \sum_{i=1}^{k-1} i\gamma^i.$$

Proof. The proof is an induction. For $k = 1$, $|Q_1(x,y,a) - Q_1(\tilde{x},\tilde{y},\tilde{a})| = |r(x,y,a) - r(\tilde{x},\tilde{y},\tilde{a})| \leq L_r(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty + \|a-\tilde{a}\|_\infty)$ by Property 4.2. Then, $|U_1(x,y) - U_1(\tilde{x},\tilde{y})| \leq L_r(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty)$ by Lemma C.1.6.

Now, assume that U_k satisfy

$$|U_k(x,y) - U_k(\tilde{x},\tilde{y})| \leq L_r \sum_{i=0}^{k-1} \gamma^i (\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty) + 2(\alpha+1) d_y L_r \sum_{i=1}^{k-1} i\gamma^i.$$

Then, by Lemma C.1.5,

$$\begin{aligned}
&|Q_{k+1}(x,y,a) - Q_{k+1}(\tilde{x},\tilde{y},\tilde{a})| \\
&\leq (L_r + \gamma L_r \sum_{i=0}^{k-1} \gamma^i)(\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty + \|a-\tilde{a}\|_\infty) \\
&\quad + 2(\alpha+1) d_y \gamma L_r \sum_{i=0}^{k-1} \gamma^i + \gamma 2(\alpha+1) d_y L_r \sum_{i=1}^{k-1} i\gamma^i \\
&= L_r \sum_{i=0}^{k} \gamma^i (\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty + \|a-\tilde{a}\|_\infty) + 2(\alpha+1) d_y L_r (\sum_{i=1}^{k} \gamma^i + \sum_{i=1}^{k-1} i\gamma^{i+1}) \\
&= L_r \sum_{i=0}^{k} \gamma^i (\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty + \|a-\tilde{a}\|_\infty) + 2(\alpha+1) d_y L_r \sum_{i=1}^{k} i\gamma^i.
\end{aligned}$$

By Lemma C.1.6,

$$|U_{k+1}(x,y) - U_{k+1}(\tilde{x},\tilde{y})| \leq L_r \sum_{i=0}^{k} \gamma^i (\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty) + 2(\alpha+1) d_y L_r \sum_{i=1}^{k} i\gamma^i.$$

□

According to Proposition 7.3.1 of [238], the value U_k in Lemma C.1.7 converges to the optimal value U^*. Let $k \to \infty$, we have that for any states (x,y) and $(\tilde{x},\tilde{y})$,

$$|U^*(x,y) - U^*(\tilde{x},\tilde{y})| \leq \frac{1}{1-\gamma} L_r (\|x-\tilde{x}\|_\infty + \|y-\tilde{y}\|_\infty) + \frac{2}{(1-\gamma)^2}(\alpha+1) d_y L_r.$$

C.2 Proof of Proposition 4.3.2

C.2.0.1 Additional Lemmas The following two lemmas are about properties of Bellman operators H and $\tilde{H}$.

Lemma C.2.1. *For any state (x, y) and any value functions V and $\tilde{V} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we have*

$$|(H^t V)(x, y) - (H^t \tilde{V})(x, y)| \leq \max_{x_t, y_t} \gamma^t \big|V(x_t, y_t) - \tilde{V}(x_t, y_t)\big|,$$

and

$$|(\tilde{H}^t V)(x, y) - (\tilde{H}^t \tilde{V})(x, y)| \leq \max_{y_t} \gamma^t \big|V(x, y_t) - \tilde{V}(x, y_t)\big|,$$

where (x_t, y_t) is the state reached after t transitions under a policy $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_{t-1})$.

Lemma C.2.2. *Suppose there exists $L_V > 0$ that for any states (x, y) and $(\tilde{x}, \tilde{y})$, any value functions V and $\tilde{V} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $|V(x, y) - V(\tilde{x}, \tilde{y})| \leq L_V \|(x, y) - (\tilde{x}, \tilde{y})\|_2$, then*

$$\begin{aligned}
&|(H^t V)(x, y) - (\tilde{H}^t V)(\tilde{x}, \tilde{y})| \\
&\leq (\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2)\big(L_r \sum_{i=0}^{t-1} \gamma^i + L_V \gamma^t\big) + d_y(\alpha + 2)\big(L_r \sum_{i=1}^{t} i\gamma^i + L_V t \gamma^t\big).
\end{aligned}$$

Proof.

$$\begin{aligned}
&|(H^t V)(x, y) - (\tilde{H}^t V)(\tilde{x}, \tilde{y})| \\
&= \big|\max_a r(x, y, a) + \gamma \mathbf{E}[(H^{t-1} V)(x', y')] - \max_b \big(r(\tilde{x}, \tilde{y}, b) + \gamma \mathbf{E}[(\tilde{H}^{t-1} V)(\tilde{x}, \tilde{y}')]\big)\big| \\
&\leq \max_{a_0 \in \mathcal{A}, x_1, y_1, \tilde{y}_1} \big|r(x, y, a_0) + \gamma (H^{t-1} V)(x_1, y_1) - r(\tilde{x}, \tilde{y}, a_0) - \gamma (\tilde{H}^{t-1} V)(\tilde{x}, \tilde{y}_1)\big| \\
&\leq \max_{a_0 \in \mathcal{A}} \big|r(x, y, a_0) - r(\tilde{x}, \tilde{y}, a_0)\big| + \gamma \max_{x_1, y_1, \tilde{y}_1} \big|(H^{t-1} V)(x_1, y_1) - (\tilde{H}^{t-1} V)(\tilde{x}, \tilde{y}_1)\big| \\
&\leq L_r(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2) + \gamma \max_{x_1, y_1, \tilde{y}_1} \big|(H^{t-1} V)(x_1, y_1) - (\tilde{H}^{t-1} V)(\tilde{x}, \tilde{y}_1)\big| \\
&\leq \ldots \\
&\leq L_r(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2) + \sum_{i=1}^{t-1} \max_{x_i, y_i, \tilde{y}_i} \gamma^i L_r(\|x_i - \tilde{x}\|_2 + \|y_i - \tilde{y}_i\|_2) \\
&\quad + \max_{x_t, y_t, \tilde{y}_t} \gamma^t \big|V(x_t, y_t) - V(\tilde{x}, \tilde{y}_t)\big|
\end{aligned}$$

$$
\begin{aligned}
&\leq L_r(\|x-\tilde{x}\|_2+\|y-\tilde{y}\|_2)+\sum_{i=1}^{t-1}\max_{x_i,y_i,\tilde{y}_i}\gamma^i L_r(\|x_i-\tilde{x}\|_2+\|y_i-\tilde{y}_i\|_2)\\
&\quad+\max_{x_t,y_t,\tilde{y}_t}\gamma^t L_V(\|x_t-\tilde{x}\|_2+\|y_t-\tilde{y}_t\|_2)\\
&\leq L_r(\|x-\tilde{x}\|_2+\|y-\tilde{y}\|_2)+\sum_{i=1}^{t-1}\gamma^i L_r(i\alpha d_y+\|x-\tilde{x}\|_2+2id_y+\|y-\tilde{y}\|_2)\\
&\quad+\gamma^t L_V(t\alpha d_y+\|x-\tilde{x}\|_2+2td_y+\|y-\tilde{y}\|_2) \qquad \text{(C.12)}\\
&=(\|x-\tilde{x}\|_2+\|y-\tilde{y}\|_2)\big(L_r\sum_{i=0}^{t-1}\gamma^i+L_V\gamma^t\big)+d_y(\alpha+2)\big(L_r\sum_{i=1}^{t}i\gamma^i+L_V t\gamma^t\big),
\end{aligned}
$$

where (C.12) is from Lemma C.1.1. □

C.2.0.2 Proof of Proposition 4.3.2 The difference between the two reward functions can be expanded as follows.

$$
\begin{aligned}
&\big|\mathbf{E}[R(s_0,a,\boldsymbol{\pi}^*)]-\mathbf{E}[\tilde{R}(s_0,a,J_1^*)]\big|\\
&=\big|r(x_0,y_0,a)+\gamma\mathbf{E}[(H^{T-1}U^*)(x_1,y_1)]-\gamma^T\mathbf{E}[U^*(x_T,y_T)]\\
&\quad-r(x_0,y_0,a)-\gamma\mathbf{E}[(\tilde{H}^{T-1}0)(x_1,y_1)]\big|\\
&=\gamma\big|\mathbf{E}[(H^{T-1}U^*)(x_1,y_1)]-\mathbf{E}[(\tilde{H}^{T-1}0)(x_1,y_1)]-\gamma^{T-1}\mathbf{E}[U^*(x_T,y_T)]\big|\\
&\leq\gamma\big|\mathbf{E}[(H^{T-1}U^*)(x_1,y_1)]-\mathbf{E}[(\tilde{H}^{T-1}U^*)(x_1,y_1)]\big|\\
&\quad+\gamma\big|\mathbf{E}[(\tilde{H}^{T-1}U^*)(x_1,y_1)]-\mathbf{E}[(\tilde{H}^{T-1}0)(x_1,y_1)]-\gamma^{T-1}\mathbf{E}[U^*(x_T,y_T)]\big|\\
&\leq d_y(\alpha+2)\big(L_r\sum_{i=1}^{T-1}i\gamma^{i+1}+\frac{L_r}{1-\gamma L_f}(T-1)\gamma^T\big)\\
&\quad+\max_{x_T,y_T,y_T'}\gamma^T\big|U^*(x,y_T')-\mathbf{E}[U^*(x_T,y_T)]\big| \qquad \text{(C.13)}\\
&\leq d_y(\alpha+2)\big(L_r\sum_{i=1}^{T-1}i\gamma^{i+1}+\frac{L_r}{1-\gamma L_f}(T-1)\gamma^T\big)\\
&\quad+\max_{x_T,y_T,y_T'}\frac{L_r\gamma^T}{1-\gamma L_f}(\|x-x_T\|_2+\|y_T-y_T'\|_2) \qquad \text{(C.14)}\\
&\leq d_y(\alpha+2)\big(L_r\sum_{i=1}^{T-1}i\gamma^{i+1}+\frac{L_r}{1-\gamma L_f}(T-1)\gamma^T\big)+\frac{L_r}{1-\gamma L_f}(\alpha+2)d_yT\gamma^T \qquad \text{(C.15)}\\
&=d_y(\alpha+2)L_r\sum_{i=1}^{T-1}i\gamma^{i+1}+\frac{L_r}{1-\gamma L_f}(\alpha+2)d_y(T-1)\gamma^T+\frac{L_r}{1-\gamma L_f}(\alpha+2)d_yT\gamma^T
\end{aligned}
$$

$$= \frac{1}{(1-\gamma)^2}(\alpha+2)L_r d_y\Big(\gamma^2(1-\gamma^T) - (1-\gamma)\gamma T\gamma^T\Big) + \frac{1}{1-\gamma L_f}(\alpha+2)L_r d_y\big(2T\gamma^T - \gamma^T\big)$$
$$= \frac{1}{(1-\gamma)^2}(\alpha+2)L_r d_y\Big(\gamma^2 - \gamma^2\gamma^T - (1-\gamma)\gamma T\gamma^T\Big) + \frac{1}{1-\gamma L_f}(\alpha+2)L_r d_y\big(2T\gamma^T - \gamma^T\big)$$
$$= (\alpha+2)L_r d_y\Big(\frac{\gamma^2}{(1-\gamma)^2} - \big(\frac{\gamma^2}{(1-\gamma)^2} + \frac{1}{1-\gamma L_f}\big)\gamma^T + \big(\frac{2}{1-\gamma L_f} - \frac{\gamma}{1-\gamma}\big)T\gamma^T\Big),$$

where (C.13) is from Lemmas C.2.1 and C.2.2, (C.14) comes from Proposition 4.3.1, (C.15) comes from Lemma C.1.1.

C.2.1 Proof for Section 4.3.6

C.2.1.1 Additional Lemmas

Lemma C.2.3. *For any vectors J and J', the Bellman operators $\tilde{H}$ and $\tilde{H}^{\boldsymbol{\pi}}$ satisfy*

$$\|\tilde{H}J - \tilde{H}J'\|_\infty \le \gamma\|J - J'\|_\infty, \;\; \|\tilde{H}^{\boldsymbol{\pi}}J - \tilde{H}^{\boldsymbol{\pi}}J'\|_\infty \le \gamma\|J - J'\|_\infty.$$

Lemma C.2.4. *For any vectors V and V', the Bellman operators F and F^μ satisfy*

$$\|FV - FV'\|_\infty \le \gamma^T\|V - V'\|_\infty, \;\; \|F^\mu V - F^\mu V'\|_\infty \le \gamma^T\|V - V'\|_\infty.$$

Lemmas C.2.3 and C.2.4 are from Lemma 2.5 of [37].

Lemma C.2.5. *Let $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma\rangle$ be an MDP. Let U^* be its optimal value function, and let U_k be the value function in iteration k of VI. If the immediate reward $R(s,a)$ is bounded within $[-r_{\max}, r_{\max}]$, then*

$$\|U_k - U^*\|_\infty \le \frac{2r_{\max}\gamma^k}{1-\gamma}. \tag{C.16}$$

Proof. Since $R(s,a)$ is bounded within $[-r_{\max}, r_{\max}]$, the value $U(s)$ is bounded within $[-\frac{r_{\max}}{1-\gamma}, \frac{r_{\max}}{1-\gamma}]$. In the worst case, the value function is initialize as

$$\|U_0 - U^*\|_\infty = \frac{2r_{\max}}{1-\gamma}.$$

Let H be the Bellman operator such that

$$(HU)(s) = \max_{a\in\mathcal{A}} \sum_{s'} P(s'|s,a)\big[R(s,a) + \gamma U(s')\big].$$

For each iteration k of VI, we have $U_k = HU_{k-1}$. Therefore,

$$\|U_k - U^*\|_\infty = \|HU_{k-1} - HU^*\|_\infty \leq \gamma\|U_{k-1} - U^*\|_\infty \leq \ldots \leq \gamma^k\|U_0 - U^*\|_\infty \leq \frac{2r_{\max}\gamma^k}{1-\gamma}.$$

□

Lemma C.2.6. *Let $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma\rangle$ be an MDP. Let ν_k be the greedy policy w.r.t. $R+\gamma\mathbf{E}[U_k]$. If $\|U_k - U^*\|_\infty \leq \varepsilon$, then*

$$\|U^{\nu_k} - U^*\|_\infty \leq \frac{2\gamma}{1-\gamma}\varepsilon. \tag{C.17}$$

Proof.

$$\begin{aligned}
\|U^{\nu_k} - U^*\|_\infty &= \|H^{\nu_k}U^{\nu_k} - U^*\|_\infty \\
&\leq \|H^{\nu_k}U^{\nu_k} - H^{\nu_k}U_{K_{\text{base}}(\xi)}\|_\infty + \|H^{\nu_k}U_{K_{\text{base}}(\xi)} - U^*\|_\infty \\
&\leq \gamma\|U^{\nu_k} - U_{K_{\text{base}}(\xi)}\|_\infty + \|HU_{K_{\text{base}}(\xi)} - HU^*\|_\infty \\
&\leq \gamma\|U^{\nu_k} - U^* + U^* - U_{K_{\text{base}}(\xi)}\|_\infty + \gamma\|U_{K_{\text{base}}(\xi)} - U^*\|_\infty \\
&\leq \gamma\|U^{\nu_k} - U^*\|_\infty + 2\gamma\|U_{K_{\text{base}}(\xi)} - U^*\|_\infty.
\end{aligned}$$

Therefore, $\|U^{\nu_k} - U^*\|_\infty \leq \frac{2\gamma}{1-\gamma}\|U_{K_{\text{base}}(\xi)} - U^*\|_\infty \leq \frac{2\gamma}{1-\gamma}\varepsilon$. □

Proposition C.2.1. *Consider a (α, d_y, L_r, L_f)-fast-slow MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma\rangle$. Denote ν a policy. If $\gamma L_f < 1$, then the value U^ν of the base model (4.5) satisfies:*

$$\left|U^\nu(x,y) - U^\nu(\tilde{x},\tilde{y})\right| \leq \frac{L_r}{1-\gamma L_f}\big(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2\big). \tag{C.18}$$

for any states $(x,y) \in \mathcal{S}$ and $(\tilde{x},\tilde{y}) \in \mathcal{S}$.

Proof. The proof follows the same technique in Appendix C.1.5 by replacing $U(x,y)$ and $Q(x,y,a)$ with $U^\nu(x,y)$ and $Q^\nu(x,y,a)$ respectively. □

Lemma C.2.7. *Consider two MDPs who differ in their transition and reward functions $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_1, r_1, \gamma\rangle$ and $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_2, r_2, \gamma\rangle$. Let U_1^* and U_2^* be their respective optimal value functions. Suppose that*

(a) $|r_1(s,a) - r_2(s,a)| \leq \epsilon_r$ *for all* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$;

(b) $\|f_1(s,a,w) - f_2(s,a,w)\|_2 \leq d$ *for all* $s \in \mathcal{S}$, $a \in \mathcal{A}$ *and* $w \in \mathcal{W}$; *and*

(c) *there exists* $L_1 > 0$ *such that* $|U_1^\nu(s) - U_1^\nu(s')| \le L_1\|s - s'\|_2$ *for all states* $s, s' \in \mathcal{S}$ *and policy* ν.

Then, the difference in optimal values of the two MDPs can be bounded as follows:

$$\left|U_1^\nu(s) - U_2^\nu(s)\right| \le \epsilon_U = \frac{\epsilon_r + \gamma L_1 d}{1-\gamma}$$

for all $s \in \mathcal{S}$.

Proof. Let $\hat{s} = \arg\max_{s\in\mathcal{S}}\left|U_1^\nu(s) - U_2^\nu(s)\right|$. We have $\left|U_1^\nu(s) - U_2^\nu(s)\right| \le \left|U_1^\nu(\hat{s}) - U_2^\nu(\hat{s})\right|$. Let us show the bound of $\left|U_1^\nu(\hat{s}) - U_2^\nu(\hat{s})\right|$.

$$\begin{aligned}
&\left|U_1^\nu(\hat{s}) - U_2^\nu(\hat{s})\right| \\
&= \left|\left(r_1(\hat{s},\nu) + \gamma\mathbf{E}[U_1^\nu(f_1^\nu(\hat{s},w))]\right) - \left(r_2(\hat{s},\nu) + \gamma\mathbf{E}[U_2^\nu(f_2^\nu(\hat{s},w))]\right)\right| \\
&\le \max_{a\in\mathcal{A}}\left|r_1(\hat{s},a) + \gamma\mathbf{E}[U_1^\nu(f_1(\hat{s},a,w))] - r_2(\hat{s},a) - \gamma\mathbf{E}[U_2^\nu(f_2(\hat{s},a,w))]\right| \\
&\le \max_{a\in\mathcal{A}}\left|r_1(\hat{s},a) - r_2(\hat{s},a)\right| + \max_{a\in\mathcal{A}}\gamma\left|\mathbf{E}[U_1^\nu(f_1(\hat{s},a,w))] - \mathbf{E}[U_2^\nu(f_2(\hat{s},a,w))]\right| \\
&\le \epsilon_r + \max_{a\in\mathcal{A}}\gamma\left|\mathbf{E}[U_1^\nu(f_1(\hat{s},a,w))] - \mathbf{E}[U_1^\nu(f_2(\hat{s},a,w))]\right| \\
&\quad + \max_{a\in\mathcal{A}}\gamma\left|\mathbf{E}[U_1^\nu(f_2(\hat{s},a,w))] - \mathbf{E}[U_2^\nu(f_2(\hat{s},a,w))]\right| \\
&\le \epsilon_r + \max_{a\in\mathcal{A},w\in\mathcal{W}}\gamma L_1\|f_1(\hat{s},a,w) - f_2(\hat{s},a,w)\| + \max_{a\in\mathcal{A}}\gamma\left|U_1^\nu(\hat{s}) - U_2^\nu(\hat{s})\right| \\
&\le \epsilon_r + \gamma L_1 d + \gamma\left|U_1^\nu(\hat{s}) - U_2^\nu(\hat{s})\right|.
\end{aligned}$$

Therefore,

$$\left|U_1^\nu(\hat{s}) - U_2^\nu(\hat{s})\right| \le \frac{\epsilon_r + \gamma L_1 d}{1-\gamma}.$$

□

C.2.1.2 Proof of Propositions 4.3.3 and 4.3.4 For the base model $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f_{\mathcal{X}}, f_{\mathcal{Y}}, r, \gamma \rangle$, the reward function r is bounded by $-r_{\max}$ and $r_{\max}$. According to Lemmas C.2.5 and C.2.6,

$$\|U^{\nu_k} - U^*\|_\infty \leq \frac{2\gamma}{1-\gamma} \frac{2r_{\max}\gamma^k}{1-\gamma} = \frac{4r_{\max}\gamma^{k+1}}{(1-\gamma)^2}.$$

Let $\|U^{\nu_k} - U^*\|_\infty \leq \xi$, solving inequality $\xi \leq \frac{4r_{\max}\gamma^{k+1}}{(1-\gamma)^2}$ gives a bound on the number of iteration k:

$$k \geq K_{\text{base}}(\xi) = \frac{1}{\log(\gamma)} \log\Big(\frac{(1-\gamma)^2\xi}{4r_{\max}}\Big) - 1.$$

For the upper level MDP of the hierarchical approximation with nominal slow state, the immediate reward function R is bounded within $[-\frac{1-\gamma^T}{1-\gamma}r_{\max}, \frac{1-\gamma^T}{1-\gamma}r_{\max}]$, and the discount factor is γ^T. According to Lemmas C.2.5 and C.2.6,

$$\|\bar{V}^{\mu_k}(\cdot,\cdot,\bar{J}_1^*,\bar{\boldsymbol{\pi}}^*) - \bar{V}^*(\cdot,\cdot,\bar{J}_1^*,\bar{\boldsymbol{\pi}}^*)\|_\infty \leq \frac{2\gamma^T}{1-\gamma^T} \frac{2\big(\frac{1-\gamma^T}{1-\gamma}r_{\max}\big)\big(\gamma^{kT}\big)}{1-\gamma^T} = \frac{4r_{\max}\gamma^{T(k+1)}}{(1-\gamma)(1-\gamma^T)}.$$

Let $\|\bar{V}^{\mu_k}(\cdot,\cdot,\bar{J}_1^*,\bar{\boldsymbol{\pi}}^*) - \bar{V}^*(\cdot,\cdot,\bar{J}_1^*,\bar{\boldsymbol{\pi}}^*)\|_\infty \leq \xi$, then the number of iterations is bounded by

$$k \geq K_{\text{frozen}}(\xi, T) = \frac{1}{T\log(\gamma)} \log\Big(\frac{\xi(1-\gamma)(1-\gamma^T)}{4r_{\max}}\Big) - 1.$$

C.2.1.3 Proof of Corollary 4.3.1 This corollary is a result of Propositions 4.3.4.

$$\begin{aligned}
&\|\bar{U}^{\tilde{\mu}_k}(\cdot,\cdot,\tilde{\boldsymbol{\pi}}^*) - U^*\|_\infty \\
&= \|\bar{U}^{\tilde{\mu}_k}(\cdot,\cdot,\tilde{\boldsymbol{\pi}}^*) - V^{\tilde{\mu}_k}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) + V^{\tilde{\mu}_k}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) \\
&\quad - V^*(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) + V^*(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) - U^*\|_\infty \\
&\leq \|\bar{U}^{\tilde{\mu}_k}(\cdot,\cdot,\tilde{\boldsymbol{\pi}}^*) - V^{\tilde{\mu}_k}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*)\|_\infty + \|V^{\tilde{\mu}_k}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) - V^*(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*)\|_\infty \\
&\quad + \|V^*(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) - \bar{U}^*\|_\infty \\
&\leq \|\bar{U}^{\tilde{\mu}_k}(\cdot,\cdot,\tilde{\boldsymbol{\pi}}^*) - V^{\tilde{\mu}_k}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*)\|_\infty + \|V^{\tilde{\mu}_k}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) - V^*(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*)\|_\infty \\
&\quad + \epsilon_U(\gamma,\alpha,d_y,L_r,L_f,T) \qquad \text{(C.19)} \\
&\leq 2\epsilon_U(\gamma,\alpha,d_y,L_r,L_f,T) + \|V^{\tilde{\mu}_k}(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*) - V^*(\cdot,\cdot,J_1^*,\tilde{\boldsymbol{\pi}}^*)\|_\infty \qquad \text{(C.20)} \\
&\leq \xi_1 + 2\xi_2,
\end{aligned}$$

where (C.19) is due to Lemma 4.3.1, (C.20) is due to Proposition C.2.1 and Lemma C.2.7.

C.2.2 Proof of Lemma 4.3.3

The proof is a backward induction. Consider states (x, y) and $(\tilde{x}, \tilde{y})$. When $t = T - 1$, the difference between the two values is

$$\begin{aligned}
|\bar{J}_{T-1}(x,y) - J_{T-1}(\tilde{x},\tilde{y})| &= \big|g(x) - g(x^*) + \max_a(g(x^*) + h(y,a)) - \max_{\tilde{a}} r(\tilde{x},\tilde{y},\tilde{a})\big| \\
&\le \max_a \big|g(x) + h(y,a) - r(\tilde{x},\tilde{y},a)\big| \\
&\le \max_a \big|g(x) + h(y,a) - r(x,y,a)\big| + \max_a \big|r(x,y,a) - r(\tilde{x},\tilde{y},a)\big| \\
&\le \zeta + L_r(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2),
\end{aligned}$$

where the last inequality is from Property 4.2 and Assumption 4.3.1.

For period t, suppose the value difference is bounded by

$$\begin{aligned}
|\bar{J}_t(x,y) - J_t(\tilde{x},\tilde{y})| \le \sum_{i=0}^{T-t-1} \gamma^i \zeta + L_r \sum_{i=0}^{T-t-1} (\gamma L_f)^i (\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2) \\
+ \gamma L_r L_f \Big(\sum_{i=0}^{T-t-2} L_f^i \sum_{j=i}^{T-t-2} \gamma^j \Big) \|x - x^*\|_2.
\end{aligned}$$

Then for period $t-1$, the value difference can be expanded as

$$\begin{aligned}
&|\bar{J}_{t-1}(x,y) - J_{t-1}(\tilde{x},\tilde{y})| \\
&= \Big|\sum_{t=0}^{T-t} \gamma^t (g(x) - g(x^*)) + \max_{a\in\mathcal{A}} \mathbf{E}\big[g(x^*) + h(y,a) + \gamma \bar{J}_t(x^*, f_{\mathcal{Y}}(x^*,y,a,w))\big] \\
&\quad - \max_{\tilde{a}} \big(r(\tilde{x},\tilde{y},\tilde{a}) + \gamma \mathbf{E}[J_t(\tilde{x}, f_{\mathcal{Y}}(\tilde{x},\tilde{y},\tilde{a},w))]\big)\Big| \\
&\le \max_{a\in\mathcal{A}} \Big|\sum_{t=0}^{T-t} \gamma^t (g(x) - g(x^*)) + g(x^*) + h(y,a) + \gamma \mathbf{E}[\bar{J}_t(x^*, f_{\mathcal{Y}}(x^*,y,a,w))] \\
&\quad - r(\tilde{x},\tilde{y},a) - \gamma \mathbf{E}[J_t(\tilde{x}, f_{\mathcal{Y}}(\tilde{x},\tilde{y},a,w))]\Big| \\
&= \max_{a\in\mathcal{A}} \Big|\sum_{t=0}^{T-t} \gamma^t (g(x) - g(x^*)) + g(x^*) + h(y,a) - r(\tilde{x},\tilde{y},a) - \gamma \mathbf{E}[J_t(\tilde{x}, f_{\mathcal{Y}}(\tilde{x},\tilde{y},a,w))] \\
&\quad + \gamma \mathbf{E}[\bar{J}_t(x, f_{\mathcal{Y}}(x^*,y,a,w)) - \sum_{t=0}^{T-t-1} \gamma^t (g(x) - g(x^*))]\Big| \qquad \text{(C.21)} \\
&= \max_{a\in\mathcal{A}} \big|g(x) + h(y,a) + \gamma \mathbf{E}[\bar{J}_t(x, f_{\mathcal{Y}}(x^*,y,a,w))] - r(\tilde{x},\tilde{y},a) - \gamma \mathbf{E}[J_t(\tilde{x}, f_{\mathcal{Y}}(\tilde{x},\tilde{y},a,w))]\big|
\end{aligned}$$

$$
\begin{aligned}
&\leq \max_{a\in\mathcal{A}} |g(x) + h(y,a) - r(\tilde{x},\tilde{y},a)| \\
&\quad + \max_{a,w} \gamma \big|\mathbf{E}[\bar{J}_t(x, f_{\mathcal{Y}}(x^*,y,a,w))] - \mathbf{E}[J_t(\tilde{x}, f_{\mathcal{Y}}(\tilde{x},\tilde{y},a,w))]\big| \\
&\leq \zeta + L_r(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2) \\
&\quad + \gamma\Big(\sum_{i=0}^{T-t-1} \gamma^i \zeta + L_r \sum_{i=0}^{T-t-1} (\gamma L_f)^i L_f (\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2 + \|x-x^*\|_2) \\
&\quad + \gamma L_r L_f \big(\sum_{i=0}^{T-t-2} L_f^i \sum_{j=i}^{T-t-2} \gamma^j \big) \|x-x^*\|_2 \Big) \qquad \text{(C.22)} \\
&= \sum_{i=0}^{T-t} \gamma^i \zeta + L_r \sum_{i=0}^{T-t} (\gamma L_f)^i (\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2) + \gamma L_r L_f \Big(\sum_{i=0}^{T-t-1} L_f^i \sum_{j=i}^{T-t-1} \gamma^j \Big) \|x-x^*\|_2,
\end{aligned}
$$

where (C.21) is from (4.21), (C.22) is from the induction assumption and (4.3).

C.2.3 Proof of Proposition 4.3.5

$$
\begin{aligned}
|\bar{J}_t(x,y) - J_t(\tilde{x},\tilde{y})| \leq & \sum_{i=0}^{T-t-1} \gamma^i \zeta + L_r \sum_{i=0}^{T-t-1} (\gamma L_f)^i (\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2) \\
& + \gamma L_r L_f \Big(\sum_{i=0}^{T-t-2} L_f^i \sum_{j=i}^{T-t-2} \gamma^j \Big) \|x-x^*\|_2.
\end{aligned}
$$

The difference of the reward functions $\big|\mathbf{E}[\tilde{R}(s_0,a,J_1^*)] - \mathbf{E}[\bar{R}(s_0,a,\bar{J}_1^*)]\big|$ can be expanded as follows,

$$
\begin{aligned}
&\big|\mathbf{E}[\tilde{R}(s_0,a,J_1^*)] - \mathbf{E}[\bar{R}(s_0,a,\bar{J}_1^*)]\big| \\
&= \gamma \big|\mathbf{E}\big[J_1^*\big(f_{\mathcal{X}}(x_0,w_0), f_{\mathcal{Y}}^{\mu}(x_0,y_0,w_0)\big)\big] - \mathbf{E}\big[\bar{J}_1^*(f_{\mathcal{X}}(x_0,w_0), f_{\mathcal{Y}}^{\mu}(x_0,y_0,w_0))\big]\big| \\
&\leq \sum_{i=1}^{T-1} \gamma^i \zeta + \gamma^2 L_r L_f \Big(\sum_{i=0}^{T-3} L_f^i \sum_{j=i}^{T-3} \gamma^j \Big) \max_x \|x-x^*\|_2,
\end{aligned}
$$

where the inequality is by Lemma 4.3.3.

C.2.4 Proof of Theorem 4.4.1

Note that in this section, in violation of the notations in Sections 4.2 and 4.3, we use the same notation as in the exogenous slow state models in Sections 4.2 and 4.3.

Define the frozen-state hierarchical MDP for the endogenous slow state MDP, whose lower-level value function is

$$J_t(x, a_x, y) = \max_{a_y \in \mathcal{A}_\mathcal{Y}} \mathbf{E}\big[r(x, y, a_x, a_y) + \gamma J_{t+1}(x, a_x, f_\mathcal{Y}(x, y, a_y, w))\big], \tag{C.23}$$

and $J_T = 0$. Given the lower-level policies $\boldsymbol{\pi}$, the upper-level value at state $s_0 = (x_0, y_0)$ is:

$$V^*(x_0, y_0, J_1, \boldsymbol{\pi}) = \max_{a_x \in \mathcal{A}_\mathcal{X}, a_y \in \mathcal{A}_\mathcal{Y}} \mathbf{E}\big[\tilde{R}(s_0, \mu(s_0), J_1) + \gamma^T V^*(x_T, y_T, J_1, \boldsymbol{\pi})\big]. \tag{C.24}$$

Denote $\tilde{H}$ and $\tilde{H}^{\boldsymbol{\pi}}$ the Bellman operators of lower level of the frozen-state model, i.e.,

$$(\tilde{H} J_t)(x, a_x, y) = \max_{a_y \in \mathcal{A}_\mathcal{Y}} \mathbf{E}\big[r(x, y, a_x, a_y) + \gamma J_{t+1}(x, a_x, f_\mathcal{Y}(x, y, a_y, w))\big],$$

and

$$(\tilde{H}^{\boldsymbol{\pi}} J_t)(x, a_x, y) = \mathbf{E}\big[r(x, y, a_x, \pi_t(x, y))) + \gamma J_{t+1}(x, a_x, f_\mathcal{Y}^{\pi_t}(x^*, y, w))\big].$$

C.2.4.1 Additional Lemmas

Lemma C.2.8. *Suppose there exists $L_V > 0$ that for any state (x, y) and any pair $(\tilde{x}, \tilde{a}_x, \tilde{y})$, any value functions $V : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and $\tilde{V} : \mathcal{X} \times \mathcal{A}_{\mathcal{X}} \times \mathcal{Y} \to \mathbb{R}$, $|V(x, y) - \tilde{V}(\tilde{x}, \tilde{a}_x, \tilde{y})| \leq L_V \|(x, y) - (\tilde{x}, \tilde{y})\|_2$, then*

$$
\begin{aligned}
&|(H^t V)(x, y) - (\tilde{H}^t \tilde{V})(\tilde{x}, \tilde{a}_x, \tilde{y})| \\
&\leq (\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a_x - \tilde{a}_x\|_2)) L_r \sum_{i=0}^{t-1} \gamma^i \\
&\quad + (\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2) L_V \gamma^t + d_y(\alpha + 2)\big(L_r \sum_{i=1}^{t} i\gamma^i + L_V t\gamma^t\big).
\end{aligned}
$$

Proof. The proof is similar to Lemma C.2.2. The difference between $(H^t V)(x, y)$ and $(\tilde{H}^t \tilde{V})(\tilde{x}, \tilde{a}_x, \tilde{y})$ can be expanded as,

$$
\begin{aligned}
&|(H^t V)(x, y) - (\tilde{H}^t \tilde{V})(\tilde{x}, \tilde{a}_x, \tilde{y})| \\
&= \big|\max_a r(x, y, a) + \gamma \mathbf{E}[(H^{t-1} V)(x', y')] - \max_{b_y}\big(r(\tilde{x}, \tilde{y}, \tilde{a}_x, b_y) + \gamma \mathbf{E}[(\tilde{H}^{t-1} \tilde{V})(\tilde{x}, \tilde{a}_x, \tilde{y}')]\big)\big| \\
&\leq \max_{a_0 \in \mathcal{A}} \big|r(x, y, a_0) - r(\tilde{x}, \tilde{y}, \tilde{a}_x, a_{0,y})\big| \\
&\quad + \gamma \max_{x_1, y_1, \tilde{y}_1} \big|(H^{t-1} V)(x_1, y_1) - (\tilde{H}^{t-1} \tilde{V})(\tilde{x}, \tilde{a}_x, \tilde{y}_1)\big| \\
&\leq L_r(\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a_x - \tilde{a}_x\|_2) \\
&\quad + \gamma \max_{x_1, y_1, \tilde{y}_1} \big|(H^{t-1} V)(x_1, y_1) - (\tilde{H}^{t-1} \tilde{V})(\tilde{x}, \tilde{a}_x, \tilde{y}_1)\big| \\
&\leq \dots \\
&= (\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2 + \|a_x - \tilde{a}_x\|_2)) L_r \sum_{i=0}^{t-1} \gamma^i \\
&\quad + (\|x - \tilde{x}\|_2 + \|y - \tilde{y}\|_2) L_V \gamma^t + d_y(\alpha + 2)\big(L_r \sum_{i=1}^{t} i\gamma^i + L_V t\gamma^t\big), \qquad \text{(C.25)}
\end{aligned}
$$

where (C.25) is from Lemma C.1.1. □

Similar to Lemma 4.3.3, we have Lemma C.2.9 for the endogenous slow state MDP.

Lemma C.2.9. *The error in the lower level value function introduced by using nominal state approximation to the frozen-state hierarchical approximation is*

$$|\bar{J}_t(x,a_x,y) - J_t(\tilde{x},\tilde{a}_x,\tilde{y})| \leq \sum_{i=0}^{T-t-1} \gamma^i\zeta + L_r \sum_{i=0}^{T-t-1} (\gamma L_f)^i(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2 + \|a_x-\tilde{a}_x\|_2) \\ + \gamma L_r L_f\Big(\sum_{i=0}^{T-t-2} L_f^i \sum_{j=i}^{T-t-2}\gamma^j\Big) \max_{(x,a_x)} \|(x,a_x)-(x^*,a_x^*)\|_2.$$

C.2.4.2 Sketch of the Proof of Theorem 4.4.1 The proof is similar to the proof for Section 4.3.3. We introduce the sketch of the proof, and focus on the difference from Section 4.3.3. We first notice that in the case of endogenous slow state, Proposition 4.3.1 still holds, i.e., the optimal value function of the base model is Lipschitz:

$$|U^*(x,y) - U^*(\bar{x},\bar{y})| \leq \frac{L_r}{1-\gamma L_f}(\|x-\tilde{x}\|_2 + \|y-\tilde{y}\|_2).$$

Secondly, we prove a bound for the difference in the reward functions, $\big|\mathbf{E}[R(s_0,a,\boldsymbol{\pi}^*)] - \mathbf{E}[\bar{R}(s_0,a,J_1^*)]\big|$. The technique is the same as Appendix C.2. The only difference is that we use Lemma C.2.8 instead Lemma C.2.2, and Lemma C.2.9 instead of Lemma 4.3.3 in the proof. The difference between the two reward functions is

$$\begin{aligned}&\big|\mathbf{E}[R(s_0,a,\boldsymbol{\pi}^*)] - \mathbf{E}[\bar{R}(s_0,a,J_1^*)]\big| \\ &\leq d_y(\alpha+2)\big(L_r\sum_{i=1}^{t} i\gamma^i + \frac{L_r}{1-\gamma L_f}t\gamma^t\big) + \sum_{i=0}^{T-t-1}\gamma^i\zeta \\ &\quad + \gamma L_r L_f\Big(\sum_{i=0}^{T-t-2} L_f^i \sum_{j=i}^{T-t-2}\gamma^j\Big) \max_{(x,a_x)} \|(x,a_x)-(x^*,a_x^*)\|_2.\end{aligned}$$

Finally, we are able to show the expected loss $\mathcal{R}(\bar{\mu}^*,\bar{\boldsymbol{\pi}}^*,T)$ by using the new reward error,

$$\begin{aligned}&\mathcal{R}(\bar{\mu}^*,\bar{\boldsymbol{\pi}}^*,T) \\ &= \max_{x,y} \bar{U}^*(x,y) - \bar{U}^{\bar{\mu}^*}(x,y,\bar{\boldsymbol{\pi}}^*) \\ &\leq \frac{1}{(1-\gamma^T)^2}\Big(2\epsilon_r + (1+\gamma^T)\frac{L_r}{1-\gamma L_f}d\gamma^T\Big).\end{aligned}$$

C.2.5 Proof for Section 4.5

C.2.5.1 Proof of Lemma 4.5.1 Let $D_t = \Phi\big(\Phi^\dagger(J) - \Phi^\dagger(J')\big)$. Then, for any state s, we have

$$|D_t(s)| = \big|\boldsymbol{\phi}^\intercal(s)\big(\Phi^\dagger(J) - \Phi^\dagger(J')\big)\big|.$$

Select $\theta_1(s), \theta_1(s), \ldots, \theta_M(s) \in \mathbb{R}$ that satisfy Assumption 4.5.1, we have

$$\begin{aligned}
|D_t(s)| &= \big|\frac{\gamma'}{\gamma}\sum_{m=1}^{M}\theta_m(s)\boldsymbol{\phi}^\intercal(s_m)\big(\Phi^\dagger(J) - \Phi^\dagger(J')\big)\big| \\
&\leq \frac{\gamma'}{\gamma}\max_m\big|\boldsymbol{\phi}^\intercal(s_m)\big(\Phi^\dagger(J)) - \Phi^\dagger(J')\big)\big| \\
&\leq \frac{\gamma'}{\gamma}\max_m |D_t(s_m)| \\
&= \frac{\gamma'}{\gamma}\max_m |J(s_m) - J'(s_m)| \\
&\leq \frac{\gamma'}{\gamma}\|J - J'\|_\infty.
\end{aligned}$$

C.2.5.2 Proof of Lemma 4.5.2 Let $\epsilon' = \epsilon_{\mathrm{L}} + \delta$ for some $\delta > 0$. Choose $\bar{\boldsymbol{\omega}}_t \in \mathbb{R}^M$ such that $\|\bar{J}_t^* - \hat{J}_t(\bar{\boldsymbol{\omega}}_t)\|_\infty < \epsilon'$ for all t. Then,

$$\begin{aligned}
\|\hat{J}_t(\bar{\boldsymbol{\omega}}_t) - \Phi\bar{H}'(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty &= \|\Phi\bar{\boldsymbol{\omega}}_t - \Phi\Phi^\dagger \circ \tilde{H}\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty \\
&= \|\Phi\Phi^\dagger\Phi\bar{\boldsymbol{\omega}}_t - \Phi\Phi^\dagger \circ \tilde{H}\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty \\
&\leq \frac{\gamma'}{\gamma}\|\Phi\bar{\boldsymbol{\omega}}_t - \tilde{H}\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty \qquad \text{(C.26)} \\
&= \frac{\gamma'}{\gamma}\|\hat{J}_t(\bar{\boldsymbol{\omega}}_t) - \tilde{H}\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty \\
&\leq \frac{\gamma'}{\gamma}\|\hat{J}_t(\bar{\boldsymbol{\omega}}_t) - J_t^*\|_\infty + \frac{\gamma'}{\gamma}\|J_t^* - \tilde{H}\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty \\
&< \frac{\gamma'}{\gamma}\epsilon' + \frac{\gamma'}{\gamma}\|\tilde{H}J_{t+1}^* - \tilde{H}\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty \\
&\leq \frac{\gamma'}{\gamma}\epsilon' + \gamma'\|J_{t+1}^* - \hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty \\
&< \frac{\gamma'}{\gamma}\epsilon' + \gamma'\epsilon' \\
&= \frac{\gamma+1}{\gamma}\gamma'\epsilon',
\end{aligned}$$

where (C.26) is by Lemma 4.5.1. Let $\epsilon'' = \frac{\gamma+1}{\gamma}\gamma'\epsilon'$.

$$\begin{aligned}
\|\hat{J}_t(\bar{\boldsymbol{\omega}}_t) - \hat{J}_t(\boldsymbol{\omega}_t^*)\|_\infty &\le \|\hat{J}_t(\bar{\boldsymbol{\omega}}_t) - \Phi\bar{H}'(\bar{\boldsymbol{\omega}}_{t+1})\|_\infty + \|\Phi\bar{H}'(\bar{\boldsymbol{\omega}}_{t+1}) - \hat{J}_t(\boldsymbol{\omega}_t^*)\|_\infty \\
&< \epsilon'' + \|\Phi\Phi^\dagger \circ \tilde{H} \circ \hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1}) - \Phi\Phi^\dagger \circ \tilde{H} \circ \hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*)\|_\infty \\
&\le \epsilon'' + \frac{\gamma'}{\gamma}\|\tilde{H}\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1}) - \tilde{H}\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*)\|_\infty \\
&\le \epsilon'' + \gamma'\|\hat{J}_{t+1}(\bar{\boldsymbol{\omega}}_{t+1}) - \hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*)\|_\infty \\
&< \epsilon'' + \gamma'(\epsilon'' + \gamma'\|\hat{J}_{t+2}(\bar{\boldsymbol{\omega}}_{t+2}) - \hat{J}_{t+2}(\boldsymbol{\omega}_{t+2}^*)\|_\infty) \\
&< \dots \\
&< \epsilon'' \sum_{i=0}^{T-t-1} (\gamma')^i + (\gamma')^{T-t}\|\hat{J}_T(\bar{\boldsymbol{\omega}}_T) - \hat{J}_T(\boldsymbol{\omega}_T^*)\|_\infty \\
&= \frac{\gamma+1}{\gamma}\epsilon' \sum_{i=1}^{T-t}(\gamma')^i,
\end{aligned}$$

where the last equation is by letting $\boldsymbol{\omega}_T^* = \bar{\boldsymbol{\omega}}_T = \mathbf{0}$ since $J_T(s) = 0$ for all s. Therefore,

$$\begin{aligned}
\|J_t^* - \hat{J}_t(\boldsymbol{\omega}_t^*)\|_\infty &\le \|J_t^* - \hat{J}_t(\bar{\boldsymbol{\omega}}_t)\|_\infty + \|\hat{J}_t(\bar{\boldsymbol{\omega}}_t) - \hat{J}_t(\boldsymbol{\omega}_t^*)\|_\infty \\
&\le \epsilon'\big(1 + \frac{\gamma+1}{\gamma}\sum_{i=1}^{T-t}(\gamma')^i\big).
\end{aligned}$$

Since δ can be arbitrarily small, the proof is complete.

C.2.5.3 Proof of Lemma 4.5.3 For any vectors $(\bar{J}_1, \bar{J}_2, \dots, \bar{J}_{T-1})$, $\bar{H}\bar{J}_t = \bar{H}^{\boldsymbol{\pi}_{\bar{J}}}J_t$ for any $\bar{J}$, where $\boldsymbol{\pi}_{\bar{J}}$ is the greedy policy w.r.t. the vectors $(\bar{J}_1, \bar{J}_2, \dots, \bar{J}_{T-1})$.

$$\begin{aligned}
\|\bar{J}_t^* - \bar{J}_t^{\hat{\boldsymbol{\pi}}^*}\|_\infty &\le \|\bar{J}_t^* - \bar{H}\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*)\|_\infty + \|\bar{H}\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*) - \bar{J}_t^{\hat{\boldsymbol{\pi}}^*}\|_\infty \\
&= \|\bar{H}\bar{J}_{t+1}^* - \bar{H}\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*)\|_\infty + \|\bar{H}^{\hat{\boldsymbol{\pi}}^*}\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*) - \bar{H}^{\hat{\boldsymbol{\pi}}^*}\bar{J}_{t+1}^{\hat{\boldsymbol{\pi}}^*}\|_\infty \\
&\le \gamma\|\bar{J}_{t+1}^* - \hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*)\|_\infty + \gamma\|\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*) - \bar{J}_{t+1}^{\hat{\boldsymbol{\pi}}^*}\|_\infty \\
&\le \gamma\epsilon_{t+1}^{\text{bias}} + \gamma\|\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*) - \bar{J}_{t+1}^* + \bar{J}_{t+1}^* - \bar{J}_{t+1}^{\hat{\boldsymbol{\pi}}^*}\|_\infty \\
&\le \gamma\epsilon_{t+1}^{\text{bias}} + \gamma\|\hat{J}_{t+1}(\boldsymbol{\omega}_{t+1}^*) - \bar{J}_{t+1}^*\|_\infty + \gamma\|\bar{J}_{t+1}^* - \bar{J}_{t+1}^{\hat{\boldsymbol{\pi}}^*}\|_\infty \\
&\le 2\gamma\epsilon_{t+1}^{\text{bias}} + \gamma\|\bar{J}_{t+1}^* - \bar{J}_{t+1}^{\hat{\boldsymbol{\pi}}}\|_\infty \\
&< 2\gamma\epsilon_{t+1}^{\text{bias}} + \|\bar{J}_{t+1}^* - \bar{J}_{t+1}^{\hat{\boldsymbol{\pi}}}\|_\infty
\end{aligned}$$

$$
\begin{aligned}
&< \dots \\
&< 2\gamma \sum_{\tau=t+1}^{T-1} \epsilon_\tau^{\text{bias}}.
\end{aligned}
$$

C.2.5.4 Proof of Lemma 4.5.4

$$
\begin{aligned}
\|F'(\boldsymbol{\nu}) - F'(\boldsymbol{\nu}')\| &= \|\Phi^\dagger \circ F \circ \Phi(\boldsymbol{\nu}) - \Phi^\dagger \circ F \circ \Phi(\boldsymbol{\nu}')\| \\
&= \|\Phi\Phi^\dagger \circ F \circ \Phi(\boldsymbol{\nu}) - \Phi\Phi^\dagger \circ F \circ \Phi(\boldsymbol{\nu}')\|_\infty \\
&< \frac{\gamma'}{\gamma^T} \|F \circ \Phi(\boldsymbol{\nu}) - F \circ \Phi(\boldsymbol{\nu}')\|_\infty \\
&\leq \gamma' \|\Phi(\boldsymbol{\nu}) - \Phi(\boldsymbol{\nu}')\|_\infty \\
&= \gamma' \|\boldsymbol{\nu} - \boldsymbol{\nu}'\|.
\end{aligned}
$$

C.2.5.5 Proof of Lemma 4.5.5 Let $\epsilon' = \epsilon_{\text{U}}(\hat{J}_1, \hat{\boldsymbol{\pi}}) + \delta$ for some $\delta > 0$. Choose $\bar{\boldsymbol{\nu}} \in \mathbb{R}^M$ such that $\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty < \epsilon'$. Then,

$$
\begin{aligned}
&\|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \Phi F'(\bar{\boldsymbol{\nu}})\|_\infty \\
&= \|\Phi\bar{\boldsymbol{\nu}} - \Phi\Phi^\dagger \circ F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty \\
&= \|\Phi\Phi^\dagger\Phi\bar{\boldsymbol{\nu}} - \Phi\Phi^\dagger \circ F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty \\
&< \frac{\gamma'}{\gamma^T} \|\Phi\bar{\boldsymbol{\nu}} - F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty \qquad \text{(C.27)} \\
&= \frac{\gamma'}{\gamma^T} \|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty \\
&\leq \frac{\gamma'}{\gamma^T} \|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty + \frac{\gamma'}{\gamma^T} \|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty \\
&< \frac{\gamma'}{\gamma^T}\epsilon' + \frac{\gamma'}{\gamma^T} \|F\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty \\
&\leq \frac{\gamma'}{\gamma^T}\epsilon' + \gamma' \|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty \\
&< \frac{\gamma'}{\gamma^T}\epsilon' + \gamma'\epsilon' \\
&= \frac{\gamma^T + 1}{\gamma^T}\gamma'\epsilon',
\end{aligned}
$$

where (C.27) is by Lemma 4.5.1. Let $\epsilon'' = \frac{\gamma+1}{\gamma}\gamma'\epsilon'$.

$$\begin{aligned}
\|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty &\leq \|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \Phi F'(\bar{\boldsymbol{\nu}})\|_\infty + \|\Phi F'(\bar{\boldsymbol{\nu}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \\
&< \epsilon'' + \|\Phi\Phi^\dagger \circ F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \Phi\Phi^\dagger \circ F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \\
&< \epsilon'' + \frac{\gamma'}{\gamma^T}\|F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \\
&\leq \epsilon'' + \gamma'\|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty,
\end{aligned}$$

and it follows that

$$\|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \leq \frac{(1+\gamma^T)\gamma'}{(1-\gamma')\gamma^T}\epsilon'.$$

Therefore,

$$\begin{aligned}
&\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \\
&\leq \|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}})\|_\infty + \|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \bar{\boldsymbol{\nu}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty \\
&\leq \epsilon' + \frac{(1+\gamma^T)\gamma'}{(1-\gamma')\gamma^T}\epsilon' = \frac{\gamma^T + \gamma'}{\gamma^T(1-\gamma')}\epsilon'.
\end{aligned}$$

Since δ can be arbitrarily small, the proof is complete.

C.2.5.6 Proof of Lemma 4.5.6 The proof is similar to Lemma 4.5.3.

$$\begin{aligned}
&\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&\leq \|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty + \|F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&= \|F\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - F\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty + \|F^{\hat{\mu}^*}\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*) - F^{\hat{\mu}^*}\bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&\leq \gamma^T\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*)\|_\infty + \gamma^T\|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&\leq \gamma^T\epsilon^{\text{bias}} + \gamma^T\|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*) - \bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) + \bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&\leq \gamma^T\epsilon^{\text{bias}} + \gamma^T\|\hat{V}(\hat{J}_1, \hat{\boldsymbol{\pi}}, \boldsymbol{\nu}^*) - \bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty + \gamma^T\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \\
&\leq 2\gamma^T\epsilon^{\text{bias}} + \gamma^T\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty.
\end{aligned}$$

Therefore,

$$\|\bar{V}^*(\hat{J}_1, \hat{\boldsymbol{\pi}}) - \bar{V}^{\hat{\mu}^*}(\hat{J}_1, \hat{\boldsymbol{\pi}})\|_\infty \leq \frac{2\gamma^T}{1-\gamma^T}\epsilon^{\text{bias}}.$$

Bibliography

[1] Elizabeth Ablah, Eileen Scanlon, Kurt Konda, Annie Tinius, and Kristine M Gebbie. A large-scale points-of-dispensing exercise for first responders and first receivers in Nassau County, New York. *Biosecurity and Bioterrorism: Biodefense Strategy, Practice, and Science*, 8(1):25–35, 2010.

[2] CDC Media Relations. Still not enough naloxone where it's most needed, 2019.

[3] Morgan Godvin. The us faces a naloxone shortage at the worst possible time, 2021.

[4] The Economist. Opioid deaths in america reached new highs in the pandemic, 2021.

[5] Eva K Lee, Fan Yuan, Ferdinand H Pietz, Bernard A Benecke, and Greg Burel. Vaccine prioritization for effective pandemic response. *Interfaces*, 45(5):425–443, 2015.

[6] Rose A Rudd, Noah Aleshire, Jon E Zibbell, and R Matthew Gladden. Increases in drug and opioid overdose deaths—United States, 2000–2014. *American Journal of Transplantation*, 16(4):1323–1327, 2016.

[7] Chris Christie, Charlie Baker, Roy Cooper, Patrick J Kennedy, Bertha Madras, and Pam Bondi. The president's commission on combating drug addiction and the opioid crisis. *WhiteHouse.gov*, 2017.

[8] Jeffrey M Goodloe and Michael W Dailey. Should naloxone be available to all first responders? *Journal of Emergency Medical Services*, 2014.

[9] Jessica Rando, Derek Broering, James E Olson, Catherine Marco, and Stephen B Evans. Intranasal naloxone administration by police first responders is associated with decreased opioid overdose deaths. *The American Journal of Emergency Medicine*, 33(9):1201–1204, 2015.

[10] West Virginia Department of Health and Human Resources. DHHR begins distributing naloxone statewide for first responders, 2018.

[11] Meredith Cohn. Baltimore city running low on opioid overdose remedy. 2017.

[12] Centers for Disease Control and Prevention. Trends in number of COVID-19 cases and deaths in the US reported to CDC, by state/territory, 2021.

[13] Centers for Disease Control and Prevention. Risk for COVID-19 infection, hospitalization, and death by age group, 2021.

[14] Centers for Disease Control and Prevention. How CDC is making COVID-19 vaccine recommendations, 2021.

[15] Gabriele Neumann, Takeshi Noda, and Yoshihiro Kawaoka. Emergence and pandemic potential of swine-origin H1N1 influenza virus. *Nature*, 459(7249):931, 2009.

[16] World Health Organization. H1N1 in post-pandemic period. 2010.

[17] Tao Sheng Kwan-Gett, Atar Baer, and Jeffrey S Duchin. Spring 2009 H1N1 influenza outbreak in King County, Washington. *Disaster Medicine and Public Health Preparedness*, 3(S2):S109–S116, 2009.

[18] Novel Swine-Origin Influenza A (H1N1) Virus Investigation Team. Emergence of a novel swine-origin influenza A (H1N1) virus in humans. *New England Journal of Medicine*, 360(25):2605–2615, 2009.

[19] Centers for Disease Control and Prevention. 2009 H1N1 early outbreak and disease characteristics. 2009.

[20] Kunal J Rambhia, Matthew Watson, Tara Kirk Sell, Richard Waldhorn, and Eric Toner. Mass vaccination for the 2009 H1N1 pandemic: Approaches, challenges, and recommendations. *Biosecurity and Bioterrorism: Biodefense Strategy, Practice, and Science*, 8(4):321–330, 2010.

[21] Centers for Disease Control and Prevention. Vaccine against 2009 H1N1 influenza virus. 2009.

[22] Centers for Disease Control and Prevention. Widespread person-to-person outbreaks of hepatitis A across the United States. 2021.

[23] U.S. Department of Health and Human Services and Centers for Disease Control and Prevention. Vaccine storage and handling toolkit. 2018.

[24] Steven Nahmias. Simple approximations for a variety of dynamic leadtime lost-sales inventory models. *Operations Research*, 27(5):904–924, 1979.

[25] Albert Y Ha. Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Management Science*, 43(8):1093–1103, 1997.

[26] Esmail Mohebbi. Supply interruptions in a lost-sales inventory system with random lead time. *Computers & Operations Research*, 30(3):411–426, 2003.

[27] Paul Zipkin. Old and new methods for lost-sales inventory systems. *Operations Research*, 56(5):1256–1263, 2008.

[28] Marco Bijvank and Iris FA Vis. Lost-sales inventory theory: A review. *European Journal of Operational Research*, 215(1):1–13, 2011.

[29] Larissa Janssen, Thorsten Claus, and Jürgen Sauer. Literature review of deteriorating inventory models by key topics from 2012 to 2015. *International Journal of Production Economics*, 182:86–112, 2016.

[30] Andrew J Clark and Herbert Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science*, 6(4):475–490, 1960.

[31] Felipe K Tan. Optimal policies for a multi-echelon inventory problem with periodic ordering. *Management Science*, 20(7):1104–1111, 1974.

[32] Stephen C Graves. A multiechelon inventory model with fixed replenishment intervals. *Management Science*, 42(1):1–18, 1996.

[33] Fangruo Chen and Rungson Samroengraja. A staggered ordering policy for one-warehouse, multiretailer systems. *Operations Research*, 48(2):281–293, 2000.

[34] Geert-Jan Van Houtum, Alan Scheller-Wolf, and Jinxin Yi. Optimal control of serial inventory systems with fixed replenishment intervals. *Operations Research*, 55(4):674–687, 2007.

[35] Wei-Qi Zhou, Long Chen, and Hui-Ming Ge. A multi-product multi-echelon inventory control model with joint replenishment strategy. *Applied Mathematical Modelling*, 37(4):2039–2050, 2013.

[36] Christopher Grob. *Inventory Management in Multi-Echelon Networks: On the Optimization of Reorder Points*, volume 128. Springer, 2018.

[37] Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*, volume 3. Athena Scientific, 1996.

[38] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*, volume 1. MIT press Cambridge, 1998.

[39] Warren B Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, volume 703. John Wiley & Sons, 2007.

[40] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

[41] MVF Pereira and LMVG Pinto. Stochastic dual dynamic programming. *Mathematical Programming*, 52:359–375, 1991.

[42] Warren Powell, Andrzej Ruszczyński, and Huseyin Topaloglu. Learning algorithms for separable approximations of discrete stochastic optimization problems. *Mathematics of Operations Research*, 29(4):814–836, 2004.

[43] Juliana M Nascimento and Warren B Powell. An optimal approximate dynamic programming algorithm for the lagged asset acquisition problem. *Mathematics of Operations Research*, 34(1):210–237, 2009.

[44] Andrew B Philpott and Z Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.

[45] Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[46] Nils Löhndorf, David Wozabal, and Stefan Minner. Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. *Operations Research*, 61(4):810–823, 2013.

[47] Sumit Kunnumkal and Huseyin Topaloglu. Using stochastic approximation methods to compute optimal base-stock levels in inventory control problems. *Operations Research*, 56(3):646–664, 2008.

[48] Paul John Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, 1974.

[49] Ian H Witten. An adaptive optimal controller for discrete-time Markov environments. *Information and Control*, 34(4):286–295, 1977.

[50] Paul J Werbos. Approximate dynamic programming for real-time control and neural modeling. *Handbook of Intelligent Control*, pages 493–526, 1992.

[51] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 1008–1014, 2000.

[52] Bennett L Fox. Discretizing dynamic programs. *Journal of Optimization Theory and Applications*, 11(3):228–234, 1973.

[53] James C Bean, John R Birge, and Robert L Smith. Aggregation in dynamic programming. *Operations Research*, 35(2):215–220, 1987.

[54] Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. Reinforcement learning with soft state aggregation. In *Advances in Neural Information Processing Systems*, pages 361–368, 1995.

[55] Paul J Schweitzer, Martin L Puterman, and Kyle W Kindle. Iterative aggregation-disaggregation procedures for discounted semi-Markov reward processes. *Operations Research*, 33(3):589–605, 1985.

[56] Victoria CP Chen, David Ruppert, and Christine A Shoemaker. Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming. *Operations Research*, 47(1):38–53, 1999.

[57] Victoria CP Chen. Application of orthogonal arrays and MARS to inventory forecasting stochastic dynamic programs. *Computational Statistics & Data Analysis*, 30(3):317–341, 1999.

[58] Seyed Jamshid Mousavi, Kourosh Mahdizadeh, and Abbas Afshar. A stochastic dynamic programming model with fuzzy storage states for reservoir operations. *Advances in Water Resources*, 27(11):1105–1110, 2004.

[59] Hegazy Zaher and Taher Taha Zaki. Optimal control theory to solve production inventory system in supply chain management. *Journal of Mathematics Research*, 6(4):109, 2014.

[60] D Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3):415–419, 1975.

[61] Zhiyuan Ren and Bruce H Krogh. State aggregation in Markov decision processes. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 4, pages 3819–3824. IEEE, 2002.

[62] Benjamin Van Roy. Performance loss bounds for approximate value iteration with state aggregation. *Mathematics of Operations Research*, 31(2):234–244, 2006.

[63] Satoru Fujishige and Kazuo Murota. Notes on L-/M-convex functions and the separation theorems. *Mathematical Programming*, 88(1):129–146, 2000.

[64] Kazuo Murota and Akiyoshi Shioura. Extension of M-convexity and L-convexity to polyhedral convex functions. *Advances in Applied Mathematics*, 25(4):352–427, 2000.

[65] Yingdong Lu and Jing-Sheng Song. Order-based cost optimization in assemble-to-order systems. *Operations Research*, 53(1):151–169, 2005.

[66] Paul Zipkin. On the structure of lost-sales inventory models. *Operations Research*, 56(4):937–944, 2008.

[67] Woonghee Tim Huh and Ganesh Janakiraman. On the optimal policy structure in serial inventory systems with lost sales. *Operations Research*, 58(2):486–491, 2010.

[68] Zhan Pang, Frank Y Chen, and Youyi Feng. A note on the structure of joint inventory-pricing control with leadtimes. *Operations Research*, 60(3):581–587, 2012.

[69] Xiting Gong and Xiuli Chao. Optimal control policy for capacitated inventory systems with remanufacturing. *Operations Research*, 61(3):603–611, 2013.

[70] CHEN Xin. L-natural-convexity and its applications in operations. *Frontiers of Engineering Management*, 4(3):283–294, 2017.

[71] Sol Fanshel and James W Bush. A health-status index and its application to health-services outcomes. *Operations Research*, 18(6):1021–1066, 1970.

[72] George W Torrance, Warren H Thomas, and David L Sackett. A utility maximization model for evaluation of health care programs. *Health Services Research*, 7(2):118, 1972.

[73] Milton C Weinstein and William B Stason. Foundations of cost-effectiveness analysis for health and medical practices. *New England Journal of Medicine*, 296(13):716–721, 1977.

[74] Milton C Weinstein, Louise B Russell, Marthe R Gold, Joanna E Siegel, et al. *Cost-effectiveness in health and medicine*. Oxford University Press, 1996.

[75] Phillip O Coffin and Sean D Sullivan. Cost-effectiveness of distributing naloxone to heroin users for lay overdose reversal. *Annals of Internal Medicine*, 158(1):1–9, 2013.

[76] Sue Langham, Antony Wright, James Kenworthy, Richard Grieve, and William CN Dunlop. Cost-effectiveness of take-home naloxone for the prevention of overdose fatalities among heroin users in the United Kingdom. *Value in Health*, 21(4):407–415, 2018.

[77] Mahip Acharya, Divyan Chopra, Corey J Hayes, Benjamin Teeter, and Bradley C Martin. Cost-effectiveness of intranasal naloxone distribution to high-risk prescription opioid users. *Value in Health*, 23(4):451–460, 2020.

[78] Kazuo Murota. Discrete convex analysis. *Mathematical Programming*, 83(1):313–371, 1998.

[79] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[80] Harold Kushner and George Yin. *Stochastic Approximation and Recursive Algorithms and Applications*, volume 35. Springer-Verlag New York, 2003.

[81] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

[82] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

[83] Xin Chen, Zhan Pang, and Limeng Pan. Coordinating inventory control and pricing strategies for perishable products. *Operations Research*, 62(2):284–300, 2014.

[84] Open Data Pennsylvania. Overdose information network data CY January 2018 - current monthly county state police, 2021.

[85] Centers for Disease Control and Prevention. Understanding the epidemic, 2021.

[86] Christopher Ingraham. Heroin deaths surpass gun homicides for the first time, CDC data shows. *The Washington Post, December*, 8, 2016.

[87] Drug Enforcement Administration. 2015 national drug threat assessment summary. 2015.

[88] Drug Enforcement Administration. DEA releases 2015 national drug threat assessment: Heroin and painkiller abuse continue to concern. 2015.

[89] The Economist. Opioid deaths in America reached new highs in the pandemic, 2021.

[90] Robert M Kaplan and James W Bush. Health-related quality of life measurement for evaluation research and policy analysis. *Health Psychology*, 1(1):61, 1982.

[91] GoodRx. Evzio naloxone, 2021.

[92] Akhil Gupta, Naman Shukla, Lavanya Marla, Arinbjörn Kolbeinsson, and Kartik Yellepeddi. How to incorporate monotonicity in deep networks while preserving flexibility? *arXiv preprint arXiv:1909.10662*, 2019.

[93] Akhil Gupta, Lavanya Marla, Ruoyu Sun, Naman Shukla, and Arinbjörn Kolbeinsson. PenDer: Incorporating shape constraints via penalized derivatives. 2021.

[94] Dimitrios S Apostolopoulos, Liam Pedersen, Benjamin N Shamah, Kimberly Shillcutt, Michael D Wagner, and William L Whittaker. Robotic antarctic meteorite search: Outcomes. In *International Conference on Robotics and Automation*, volume 4, pages 4174–4179. IEEE, 2001.

[95] David Ferguson, Aaron Morris, Dirk Haehnel, Christopher Baker, Zachary Omohundro, Carlos Reverte, Scott Thayer, Charles Whittaker, William Whittaker, Wolfram Burgard, et al. An autonomous robotic system for mapping abandoned mines. In *Advances in Neural Information Processing Systems*, pages 587–594, 2004.

[96] Sebastian Thrun, Scott Thayer, William Whittaker, Christopher Baker, Wolfram Burgard, David Ferguson, Dirk Hahnel, D Montemerlo, Aaron Morris, Zachary Omohundro, et al. Autonomous exploration and mapping of abandoned mines. *Robotics & Automation Magazine*, 11(4):79–91, 2004.

[97] Larry Matthies, Erann Gat, Reid Harrison, Brian Wilcox, Richard Volpe, and Todd Litwin. Mars microrover navigation: Performance evaluation and enhancement. *Autonomous Robots*, 2(4):291–311, 1995.

[98] Daniel J Lizotte, Tao Wang, Michael H Bowling, and Dale Schuurmans. Automatic gait optimization with Gaussian process regression. In *International Joint Conference on Artifical Intelligence*, volume 7, pages 944–949, 2007.

[99] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1):5–23, 2016.

[100] Rafael Oliveira, Lionel Ott, Vitor Guizilini, and Fabio Ramos. Bayesian optimisation for safe navigation under localisation uncertainty. In *Robotics Research*, pages 489–504. Springer, 2020.

[101] Zhengkun Yi, Roberto Calandra, Filipe Veiga, Herke van Hoof, Tucker Hermans, Yilei Zhang, and Jan Peters. Active tactile object exploration with Gaussian processes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4925–4930. IEEE, 2016.

[102] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[103] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

[104] Henry C Herbol, Weici Hu, Peter Frazier, Paulette Clancy, and Matthias Poloczek. Efficient search of compositional space for hybrid organic–inorganic perovskites via Bayesian optimization. *NPJ Computational Materials*, 4(1):51, 2018.

[105] Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[106] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 2004–2012, 2013.

[107] Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Freeze-thaw Bayesian optimization. *arXiv preprint arXiv:1406.3896*, 2014.

[108] Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing Bayesian hyperparameter optimization via meta-learning. In *Association for the Advancement of Artificial Intelligence*, pages 1128–1135, 2015.

[109] Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *International Joint Conferences on Artificial Intelligence*, volume 15, pages 3460–8, 2015.

[110] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

[111] Matthias Poloczek, Jialei Wang, and Peter Frazier. Multi-information source optimization. In *Advances in Neural Information Processing Systems*, pages 4288–4298, 2017.

[112] Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial Intelligence and Statistics*, pages 528–536, 2017.

[113] Mickaël Binois, Jiangeng Huang, Robert B Gramacy, and Mike Ludkovski. Replication or exploration? sequential design for stochastic simulation experiments. *Technometrics*, 61(1):7–23, 2019.

[114] Peter I Frazier, Warren B Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

[115] Warren Scott, Peter Frazier, and Warren Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.

[116] Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch bayesian optimization. *Advances in neural information processing systems*, 29, 2016.

[117] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[118] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[119] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[120] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2753–2762, 2017.

[121] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pages 2377–2386, 2016.

[122] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[123] Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, pages 2701–2710. JMLR. org, 2017.

[124] Philippe Morere and Fabio Ramos. Bayesian RL for goal-only rewards. In *Conference on Robot Learning*, 2018.

[125] Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pages 5302–5311, 2018.

[126] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. JMLR. org, 2017.

[127] Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke van Hoof, and Doina Precup. A survey of exploration methods in reinforcement learning. *arXiv preprint arXiv:2109.00157*, 2021.

[128] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[129] Doina Precup, Richard S Sutton, and Satinder Singh. Theoretical results on reinforcement learning with temporally abstract options. In *European conference on machine learning*, pages 382–393. Springer, 1998.

[130] Kishor Jothimurugan, Osbert Bastani, and Rajeev Alur. Abstract value iteration for hierarchical reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1162–1170. PMLR, 2021.

[131] Xinlei Pan, Eshed Ohn-Bar, Nicholas Rhinehart, Yan Xu, Yilin Shen, and Kris M Kitani. Human-interactive subgoal supervision for efficient inverse reinforcement learning. *arXiv preprint arXiv:1806.08479*, 2018.

[132] Sujoy Paul, Jeroen van Baar, and Amit K Roy-Chowdhury. Learning from trajectories via subgoal discovery. *arXiv preprint arXiv:1911.07224*, 2019.

[133] Martin Stolle and Doina Precup. Learning options in reinforcement learning. In *International Symposium on abstraction, reformulation, and approximation*, pages 212–223. Springer, 2002.

[134] Amy McGovern and Andrew G Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. 2001.

[135] Sandeep Goel and Manfred Huber. Subgoal discovery for hierarchical reinforcement learning using learned policies. In *FLAIRS Conference*, pages 346–350, 2003.

[136] Shie Mannor, Ishai Menache, Amit Hoze, and Uri Klein. Dynamic abstraction in reinforcement learning via clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 71, 2004.

[137] Özgür Şimşek and Andrew G Barto. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *International Conference on Machine Learning*, page 95, 2004.

[138] Özgür Şimşek, Alicia P Wolfe, and Andrew G Barto. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd international conference on Machine learning*, pages 816–823, 2005.

[139] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[140] Jesse Zhang, Haonan Yu, and Wei Xu. Hierarchical reinforcement learning by discovering intrinsic options. *arXiv preprint arXiv:2101.06521*, 2021.

[141] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[142] Alexander Vezhnevets, Volodymyr Mnih, Simon Osindero, Alex Graves, Oriol Vinyals, John Agapiou, et al. Strategic attentive writer for learning macro-actions. *Advances in neural information processing systems*, 29, 2016.

[143] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[144] Kevin Frans, Jonathan Ho, Xi Chen, Pieter Abbeel, and John Schulman. Meta learning shared hierarchies. *arXiv preprint arXiv:1710.09767*, 2017.

[145] Vivek Veeriah, Tom Zahavy, Matteo Hessel, Zhongwen Xu, Junhyuk Oh, Iurii Kemaev, Hado P van Hasselt, David Silver, and Satinder Singh. Discovery of options via meta-learned subgoals. *Advances in Neural Information Processing Systems*, 34, 2021.

[146] Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *International Conference on Machine Learning*, volume 98, pages 463–471. Citeseer, 1998.

[147] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pages 278–287, 1999.

[148] Xiao Huang and John Weng. Novelty and reinforcement learning in the value system of developmental robots. In *2nd International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies, 2002.

[149] Frédéric Kaplan and Pierre-Yves Oudeyer. Maximizing learning progress: An internal reward system for development. In *Embodied Artificial Intelligence*, pages 259–270. Springer, 2004.

[150] Özgür Şimşek and Andrew G Barto. An intrinsic reward mechanism for efficient exploration. In *Proceedings of the 23rd international conference on Machine learning*, pages 833–840, 2006.

[151] Ana C Tenorio-Gonzalez, Eduardo F Morales, and Luis Villaseñor-Pineda. Dynamic reward shaping: Training a robot by voice. In *Ibero-American Conference on Artificial Intelligence*, pages 483–492. Springer, 2010.

[152] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, volume 2017, 2017.

[153] Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

[154] Guillaume Lample and Devendra Singh Chaplot. Playing FPS games with deep reinforcement learning. In *Association for the Advancement of Artificial Intelligence*, pages 2140–2146, 2017.

[155] Jonathan Sorg, Richard L Lewis, and Satinder P Singh. Reward design via online gradient ascent. In *Advances in Neural Information Processing Systems*, pages 2190–2198, 2010.

[156] Xiaoxiao Guo, Satinder Singh, Richard Lewis, and Honglak Lee. Deep learning for reward design to improve Monte Carlo tree search in ATARI games. In *International Joint Conference on Artificial Intelligence*, pages 1519–1525, 2016.

[157] Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In *Advances in Neural Information Processing Systems*, pages 4649–4659, 2018.

[158] Marc Pickett and Andrew G Barto. Policyblocks: An algorithm for creating useful macro-actions in reinforcement learning. In *International Conference on Machine Learning*, volume 19, pages 506–513, 2002.

[159] George Konidaris and Andrew Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *International Conference on Machine Learning*, pages 489–496. ACM, 2006.

[160] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022, 2007.

[161] Fernando Fernández, Javier García, and Manuela Veloso. Probabilistic policy reuse for inter-task transfer learning. *Robotics and Autonomous Systems*, 58(7):866–871, 2010.

[162] Marc Peter Deisenroth, Peter Englert, Jan Peters, and Dieter Fox. Multi-task policy search for robotics. In *International Conference on Robotics and Automation*, 2014.

[163] Finale Doshi-Velez and George Konidaris. Hidden parameter Markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *International Joint Conferences on Artificial Intelligence*, page 1432. NIH Public Access, 2016.

[164] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pages 357–368, 2017.

[165] Lerrel Pinto and Abhinav Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2161–2168. IEEE, 2017.

[166] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

[167] Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with PopArt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3796–3803, 2019.

[168] Nelson Vithayathil Varghese and Qusay H Mahmoud. A survey of multi-task deep reinforcement learning. *Electronics*, 9(9):1363, 2020.

[169] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[170] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[171] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.

[172] Bo Chen, Rui Castro, and Andreas Krause. Joint optimization and variable selection of high-dimensional gaussian processes. *arXiv preprint arXiv:1206.6396*, 2012.

[173] Josip Djolonga, Andreas Krause, and Volkan Cevher. High-dimensional gaussian process bandits. *Advances in neural information processing systems*, 26, 2013.

[174] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[175] Mojmir Mutny and Andreas Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. *Advances in Neural Information Processing Systems*, 31, 2018.

[176] Peter Frazier, Warren Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

[177] Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error bounds for Gaussian process regression with application to safe control. In *Advances in Neural Information Processing Systems*, pages 657–667, 2019.

[178] Carl Edward Rasmussen and Christopher K.Ĩ. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[179] Subhashis Ghosal, Anindya Roy, et al. Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 34(5):2413–2429, 2006.

[180] Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, pages 5267–5278, 2017.

[181] Scott Clark, Eric Liu, Peter Frazier, JiaLei Wang, Deniz Oktay, and Norases Vesdapunt. Moe: A global, black box optimization engine for real world metric optimization. `https://github.com/Yelp/MOE`, 2014.

[182] Jonas Močkus. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer, 1975.

[183] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[184] Dennis D Cox and Susan John. A statistical method for global optimization. In *International Conference on Systems, Man, and Cybernetics*, pages 1241–1246. IEEE, 1992.

[185] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, pages 1015–1022, 2010.

[186] J González. GPyOpt: A Bayesian optimization framework in Python. `http://github.com/SheffieldML/GPyOpt`, 2016.

[187] Tristan Deleu. Model-Agnostic Meta-Learning for Reinforcement Learning in PyTorch, 2018. Available at: https://github.com/tristandeleu/pytorch-maml-rl.

[188] Remi Lam, Karen Willcox, and David H Wolpert. Bayesian optimization with a finite budget: An approximate dynamic programming approach. *Advances in Neural Information Processing Systems*, 29, 2016.

[189] Javier González, Michael Osborne, and Neil Lawrence. Glasses: Relieving the myopia of bayesian optimisation. In *Artificial Intelligence and Statistics*, pages 790–799. PMLR, 2016.

[190] Shali Jiang, Daniel Jiang, Maximilian Balandat, Brian Karrer, Jacob Gardner, and Roman Garnett. Efficient nonmyopic bayesian optimization via one-shot multi-step trees. *Advances in Neural Information Processing Systems*, 33:18039–18049, 2020.

[191] Eric Lee, David Eriksson, David Bindel, Bolong Cheng, and Mike Mccourt. Efficient rollout strategies for bayesian optimization. In *Conference on Uncertainty in Artificial Intelligence*, pages 260–269. PMLR, 2020.

[192] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945, 2014.

[193] Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

[194] Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

[195] Raul Astudillo, Daniel Jiang, Maximilian Balandat, Eytan Bakshy, and Peter Frazier. Multi-step budgeted bayesian optimization with unknown evaluation costs. *Advances in Neural Information Processing Systems*, 34, 2021.

[196] Eric Hans Lee, David Eriksson, Valerio Perrone, and Matthias Seeger. A nonmyopic approach to cost-constrained bayesian optimization. In *Uncertainty in Artificial Intelligence*, pages 568–577. PMLR, 2021.

[197] Jia Yuan Yu and Shie Mannor. Arbitrarily modulated Markov decision processes. In *Proceedings of the 48h IEEE Conference on Decision and Control*, pages 2946–2953. IEEE, 2009.

[198] PS Ansell, Kevin D Glazebrook, José Nino-Mora, and M O'Keeffe. Whittle's index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 57(1):21–39, 2003.

[199] David B Brown and Martin B Haugh. Information relaxation bounds for infinite horizon Markov decision processes. *Operations Research*, 65(5):1355–1379, 2017.

[200] Dabeen Lee and Milan Vojnovic. Scheduling jobs with stochastic holding costs. *Advances in Neural Information Processing Systems*, 34, 2021.

[201] Shaolei Ren, Yuxiong He, and Fei Xu. Provably-efficient job scheduling for energy and fairness in geographically distributed data centers. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pages 22–31. IEEE, 2012.

[202] Zhou Zhou, Zhiling Lan, Wei Tang, and Narayan Desai. Reducing energy costs for ibm blue gene/p via power-aware job scheduling. In *Workshop on Job Scheduling Strategies for Parallel Processing*, pages 96–115. Springer, 2013.

[203] Hongzi Mao, Malte Schwarzkopf, Shaileshh Bojja Venkatakrishnan, Zili Meng, and Mohammad Alizadeh. Learning scheduling algorithms for data processing clusters. In *Proceedings of the ACM special interest group on data communication*, pages 270–288. 2019.

[204] Alon Halevy, Cristian Canton-Ferrer, Hao Ma, Umut Ozertem, Patrick Pantel, Marzieh Saeidi, Fabrizio Silvestri, and Ves Stoyanov. Preserving integrity in online social networks. *Communications of the ACM*, 65(2):92–98, 2022.

[205] Mohamed H Albadi and Ehab F El-Saadany. A summary of demand response in electricity markets. *Electric Power Systems Research*, 78(11):1989–1996, 2008.

[206] Cherrelle Eid, Elta Koliou, Mercedes Valles, Javier Reneses, and Rudi Hakvoort. Time-based pricing and electricity demand response: Existing barriers and next steps. *Utilities Policy*, 40:15–25, 2016.

[207] Kia Khezeli and Eilyan Bitar. An online learning approach to buying and selling demand response. *arXiv preprint arXiv:1707.07342*, 2017.

[208] Kia Khezeli, Weixuan Lin, and Eilyan Bitar. Learning to buy (and sell) demand response. *IFAC-PapersOnLine*, 50(1):6761–6767, 2017.

[209] Shuoyao Wang, Suzhi Bi, and Ying-Jun Angela Zhang. Demand response management for profit maximizing energy loads in real-time electricity market. *IEEE Transactions on Power Systems*, 33(6):6387–6396, 2018.

[210] Amy Hing-Ling Lau and Hon-Shiang Lau. The newsboy problem with price-dependent demand distribution. *IIE Transactions*, 20(2):168–175, 1988.

[211] Snigdha Banerjee and Ashish Sharma. Optimal procurement and pricing policies for inventory models with price and time dependent seasonal demand. *Mathematical and Computer Modelling*, 51(5-6):700–714, 2010.

[212] Xiangling Hu and Ping Su. The newsvendor's joint procurement and pricing problem under price-sensitive stochastic demand and purchase price uncertainty. *Omega*, 79:81–90, 2018.

[213] John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1-3):59–94, 1996.

[214] Hyeong Soo Chang, Pedram Jaefari Fard, Steven I Marcus, and Mark Shayman. Multitime scale markov decision processes. *IEEE Transactions on Automatic Control*, 48(6):976–987, 2003.

[215] Jnana Ranjan Panigrahi and Shalabh Bhatnagar. Hierarchical decision making in semiconductor fabs using multi-time scale Markov decision processes. In *2004 43rd IEEE Conference on Decision and Control*, volume 4, pages 4387–4392. IEEE, 2004.

[216] Shalabh Bhatnagar and J Ranjan Panigrahi. Actor-critic algorithms for hierarchical Markov decision processes. *Automatica*, 42(4):637–644, 2006.

[217] Chengjun Zhu, Jianzhong Zhou, Wei Wu, and Li Mo. Hydropower portfolios management via Markov decision process. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pages 2883–2888. IEEE, 2006.

[218] Wuthichai Wongthatsanekorn, Matthew J Realff, and Jane C Ammons. Multi-time scale Markov decision process approach to strategic network growth of reverse supply chains. *Omega*, 38(1-2):20–32, 2010.

[219] Matthew Jacobson, Nahum Shimkin, and Adam Shwartz. Piecewise stationary Markov decision processes, I: Constant gain. 1999.

[220] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(1-2):41–77, 2003.

[221] Ronald Edward Parr and Stuart Russell. *Hierarchical control and learning for Markov decision processes*. University of California, Berkeley Berkeley, CA, 1998.

[222] Thomas G Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

[223] Doina Precup. Temporal abstraction in reinforcement learning. *Ph. D. thesis, University of Massachusetts*, 2000.

[224] Bruce L Digney. Learning hierarchical control structures for multiple tasks and changing environments. In *Proceedings of the 5th International Conference on Simulation of Adaptive Behavior on From Animals to Animats 5*, pages 321–330, 1998.

[225] Anders Jonsson and Andrew G Barto. A causal approach to hierarchical decomposition of factored MDPs. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 401–408, 2005.

[226] Kamil Ciosek and David Silver. Value iteration with options and state aggregation. *Planning and Learning (PAL-15)*, page 1, 2015.

[227] Ishai Menache, Shie Mannor, and Nahum Shimkin. Q-cut-dynamic discovery of sub-goals in reinforcement learning. In *Proceedings of the 13th European Conference on Machine Learning*, pages 295–306, 2002.

[228] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[229] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial intelligence*, 121(1-2):49–107, 2000.

[230] Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. *Advances in Neural Information Processing Systems*, 27, 2014.

[231] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[232] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[233] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[234] Chaoqun Duan, Chao Deng, Abolfazl Gharaei, Jun Wu, and Bingran Wang. Selective maintenance scheduling under stochastic maintenance quality with multiple maintenance actions. *International Journal of Production Research*, 56(23):7160–7178, 2018.

[235] Jackson A Killian, Arpita Biswas, Sanket Shah, and Milind Tambe. Q-learning Lagrange policies for multi-action restless bandits. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 871–881, 2021.

[236] Dimitir P Bertsekas and Steven Shreve. *Stochastic optimal control: the discrete-time case*. 2004.

[237] Satinder P Singh and Richard C Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

[238] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Athena Scientific, 2012.