

# Bi-Level Online Provisioning and Scheduling with Switching Costs and Cross-Level Constraints

Jialei Liu

*Dept. of Electrical Engineering  
University at Buffalo, Buffalo, NY*  
jialeili@buffalo.edu

C. Emre Koksal

*Dept. of Electrical and Computer Engineering  
The Ohio State University, Columbus, OH*  
koksal.2@osu.edu

Ming Shi

*Dept. of Electrical Engineering  
University at Buffalo, Buffalo, NY*  
mshi24@buffalo.edu

**Abstract**—We study a bi-level online provisioning and scheduling problem motivated by network resource allocation, where provisioning decisions are made at a slow time scale while queue-/state-dependent scheduling is performed at a fast time scale. We model this two-time-scale interaction using an upper-level online convex optimization (OCO) problem and a lower-level constrained Markov decision process (CMDP). Existing OCO typically assumes stateless decisions and thus cannot capture MDP network dynamics such as queue evolution. Meanwhile, CMDP algorithms typically assume a fixed constraint threshold, whereas in provisioning-and-scheduling systems, the threshold varies with online budget decisions. To address these gaps, we study bi-level OCO-CMDP learning under switching costs (budget reprovisioning/system reconfiguration) and cross-level constraints that couple budgets to scheduling decisions. Our new algorithm solves this learning problem via several non-trivial developments, including a carefully designed dual feedback that returns the budget multiplier as sensitivity information for the upper-level update and a lower level that solves a budget-adaptive safe exploration problem via an extended occupancy-measure linear program. We establish near-optimal regret and high-probability satisfaction of the cross-level constraints.

**Index Terms**—bi-level provisioning and scheduling, two-time-scale decision making, online convex optimization (OCO), constrained Markov decision process (CMDP), switching costs, cross-level budget constraints, reinforcement learning (RL)

## I. INTRODUCTION

Modern networking systems, such as 5G/6G network and edge computing, must support heterogeneous services (e.g., video streaming, augmented reality (AR), virtual reality (VR), and autonomous driving) over shared wireless infrastructure [1]. A central operational challenge in these systems is online resource provisioning under uncertainty. In particular, a network operator must decide how much bandwidth/compute/energy to allocate to a slice over time, balancing quality of service (QoS) against operational expenditure (OpEx). Practical systems exhibit *multiple coupled time scales*. As illustrated in our motivating example online bandwidth provisioning with queue-aware scheduling (see Fig. 1 and Sec. III for details), the operator provisions an episode-level resource budget at a slow time scale (e.g., bandwidth reserved for a slice over seconds/minutes). This budget directly incurs OpEx and influences the QoS that can be achieved through scheduling. Within each episode, the scheduler performs state-dependent, packet-level actions at a fast time scale (e.g., allocating resource blocks every millisecond). These actions

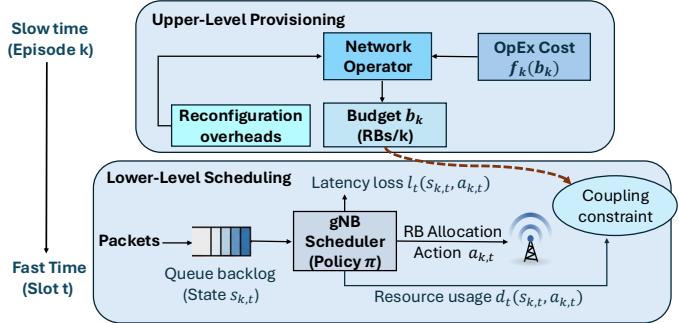


Fig. 1. Online bandwidth provisioning with queue-aware scheduling.

are *constrained* by the provisioned resource budget and aim to reduce latency and congestion, improving QoS [2], [3]. Moreover, changing the provisioning level across episodes is not free. It induces reconfiguration overheads (e.g., hardware retuning, virtualization/migration cost, and instability), which are naturally captured by *switching costs* [4], [5].

Online convex optimization (OCO) is a classical abstraction for sequential resource allocation and has been extensively studied under adversarial and stochastic settings [4]–[6]. Recent work incorporates switching costs and related operational constraints to penalize aggressive changes in decisions, which is attractive for network slicing. However, OCO with switching costs typically does not model endogenous Markovian state evolution nor provide Markovian-style safety under unknown dynamics. For example, the per-round loss depends on the current decision and does not capture the evolution of system states, while in networked systems, state variables such as queue lengths evolve endogenously. Consequently, provisioning decisions that appear cost-effective in the short term may cause persistent congestion and latency violations.

Stateful scheduling problems are naturally modeled as Markov decision processes (MDPs), and a rich literature develops provably efficient reinforcement learning (RL) algorithms [7]–[9]. Yet, existing constrained MDP (CMDP) works typically assume that the budget, i.e., the constraint threshold, is *fixed and exogenously specified* [8], [9]. However, in practice, the operator must learn what and how much budget to provision, while accounting for switching overhead. Thus, the constraint threshold itself is a primary decision variable.

Moreover, hierarchical RL (HRL) is a widely studied learning framework for multi-time-scale decision making [10]. However, its objective structure differs fundamentally from our resource provisioning problem. HRL typically optimizes a single, unified reward through task decomposition. In contrast, our problem involves a bi-level trade-off between provisioning cost and service performance. Moreover, HRL generally allows frequent switching between sub-policies.

Motivated by the above gaps, we propose a bi-level online optimization and learning framework that integrates *online budget provisioning* with *state-aware safe scheduling*. At the upper level, an OCO operator optimizes and provisions budgets  $\{b_k\}_{k=1}^K$  across episodes, incurring a convex service cost and a quadratic switching cost. At the lower level, the system evolves as an episodic CMDP under varying thresholds. Given the optimized provisioning budget  $b_k$ , the agent must learn a safe policy via RL under unknown dynamics. The two levels are coupled through an episode-wise budget constraint and an episode-wise objective that aggregates cross-level costs.

Designing an online algorithm for this coupled problem is technically challenging. (i) *The upper-level effective optimization loss is implicit and non-stationary.* Beyond the unknown convex service cost and switching penalty, it depends on the unknown optimal performance of the lower-level controller. To address this, we leverage a carefully designed dual variable of the budget constraint counterpart to construct an approximated subgradient for evaluating the upper-level sensitivity. (ii) *The lower-level CMDP faces a varying constraint threshold, since budget  $b_k$  changes.* This renders standard safe/constrained RL methods with fixed constraints inapplicable. We address this by designing a budget-adaptive exploration algorithm. In summary, our main contributions are as follows.

- We propose a new bi-level online optimization and learning algorithm for provisioning and scheduling with budget switching costs and cross-level budget constraints. At the upper level, we leverage dual multipliers returned by the lower-level (interpretable as the marginal value of extra budget), yielding a subgradient signal despite unknown stateful dynamics. At the lower level, we solve the episode-varying (decision-dependent) budget via budget-adaptive safe exploration and an extended occupancy-measure linear programming (LP), which both enforces feasibility and produces the dual feedback needed to couple the two levels.
- We establish near-optimal regret bounds with high-probability safety guarantees. Our analysis introduces a bi-level regret decomposition that explicitly quantifies how lower-level model-estimation and dual-signal errors propagate into the upper-level OCO updates, and it leverages the extended-LP strong duality to couple constraint satisfaction with sensitivity-based budget learning under switching costs.
- We instantiate the framework in a wireless network scenario, modeling the coupling between episode-level bandwidth provisioning and queue scheduling. Compared to decoupled baselines, our approach achieves a balance between cost and latency while maintaining strict physical feasibility.

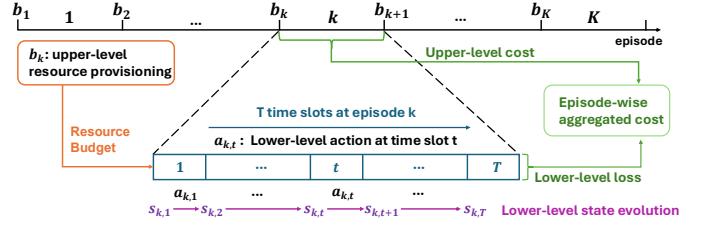


Fig. 2. Coupled two-time-scale decision making: at the beginning of episode  $k$ , the agent provisions a resource budget  $b_k$ ; within the episode, it follows a state-dependent policy selecting actions  $a_{k,t}$  over  $T$  time slots, incurring losses that aggregate to the episode cost and resource usage that must satisfy the cross-level budget constraint.

*Notations:*  $\Delta(\mathcal{X})$  denotes the probability simplex over a given set  $\mathcal{X}$ . The operator  $a \vee b$  denotes  $\max\{a, b\}$ . We use  $[T] = \{1, \dots, T\}$  and  $[x]^+ = \max\{0, x\}$ .

## II. PROBLEM FORMULATION

We study a sequential decision-making problem with *two coupled time scales*. An agent interacts with an environment over  $K$  episodes (slow time scale). Within each episode, the environment evolves over  $T$  time slots (fast time scale). In each episode  $k \in [K]$ , the agent chooses an *upper-level* budget decision  $b_k$  once per episode and a *lower-level* sequence of state-dependent actions  $\{a_{k,t}\}_{t=1}^T$  within the episode. The two levels are coupled through an episode-wise budget constraint (relating  $b_k$  to  $\{a_{k,t}\}_{t=1}^T$ ) and an episode-wise objective aggregating upper-level and lower-level costs, as depicted in Fig. 2.

### A. Upper-Level Budget Provisioning

At the slow time scale, at the beginning of each episode  $k \in [K]$ , the agent selects an upper-level budget  $b_k \in \mathcal{B} \triangleq [B_0, T]$ , which can represent the episode-level resource budget provisioned for a network slice (bandwidth, compute cycles, or energy) over seconds/minutes. Choosing  $b_k$  incurs an episode-dependent convex service cost  $f_k(b_k)$ , which is *not* revealed before the decision is made and is revealed at the end of episode  $k$ . As is typical in online convex optimization with switching costs [4], [5], [11], the service cost  $f_k(\cdot)$  could change arbitrarily, and it is  $\theta_f$ -strongly convex and differentiable on  $\mathcal{B}$  with bounded gradients  $\|\nabla f_k(\cdot)\| \leq F$ .

In addition, we consider the switching cost, i.e., changing the budget across episodes incurs an additional cost  $\alpha \|b_k - b_{k-1}\|^2$ , where  $b_0 = 0$ , and  $\alpha > 0$  quantifies reconfiguration overhead (e.g., signaling and control-plane costs, hardware retuning delays, and virtualization/migration overhead).

### B. Lower-Level Stateful Scheduling

At the fast time scale, within each episode, the environment evolves over  $t = 1, \dots, T$  time slots according to a finite-horizon (possibly time-inhomogeneous) MDP  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, \{l_t\}_{t=1}^T, \{d_t\}_{t=1}^T, \{P_t\}_{t=1}^T)$ . Here,  $\mathcal{S}$  and  $\mathcal{A}$  are the state and action spaces,  $l_t : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$  is the stage loss,  $d_t : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$  is the per-step resource consumption, and  $P_t(\cdot | s, a) \in \Delta(\mathcal{S})$  is the *unknown* transition kernel.

At each time  $t$  of episode  $k$ , the agent observes the current state  $s_{k,t}$  (e.g., queue backlogs and channel conditions), and then selects a (possibly randomized) scheduling/allocation action  $a_{k,t} \sim \pi_{k,t}(\cdot | s_{k,t})$ , where  $\pi_k^A = (\pi_{k,t})_{t=1}^T$  is a policy with  $\pi_{k,t} : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ . The agent then incurs a stage loss  $l_t(s_{k,t}, a_{k,t})$  (e.g., delay/backlog penalty), consumes resources  $d_t(s_{k,t}, a_{k,t})$  (e.g., used resource blocks/power), and the system evolves to next state  $s_{k,t+1} \sim P_t(\cdot | s_{k,t}, a_{k,t})$  due to unknown arrivals and service dynamics.

For compactness, we define the expected cumulative loss and consumption under policy  $\pi$  and model  $P$  as follows,

$$V_r(\pi; P) \triangleq \mathbb{E} \left[ \sum_{t=1}^T r_t(s_t, a_t) \mid \pi, P \right], \quad (1)$$

where  $r \in \{l, d\}$ , and the expectation is taken with respect to the randomness of actions under  $\pi$  and states under  $P$ .

### C. Cross-Level Coupling

The decisions in the two levels are coupled as follows.

1) **Coupling 1 (Cross-Level Budget Constraint):** Given upper-level budget  $b_k$ , the lower-level policy executed in episode  $k$  must satisfy the episode-wise budget constraint, i.e.,

$$V_d(\pi_k^A; P) \leq b_k. \quad (2)$$

We define the feasible policy set under budget  $b$  as

$$\Pi(b) \triangleq \{\pi^A : V_d(\pi^A; P) \leq b\}. \quad (3)$$

Then, (2) is equivalently  $\pi_k^A \in \Pi(b_k)$ . This is a *cross-level* constraint because its left-hand side is determined by lower-level actions while its right-hand side is the upper-level budget.

2) **Coupling 2 (Cross-Level Joint Cost):** Given  $(b_k, \pi_k^A)$ , we define the episode cost as follows,

$$C_k(b_k, \pi_k^A) \triangleq f_k(b_k) + \alpha \|b_k - b_{k-1}\|^2 + \beta V_l(\pi_k^A; P), \quad (4)$$

where  $\beta > 0$  weighs within-episode performance relative to provisioning and reconfiguration costs. The first two terms capture upper-level operating and switching overhead, and the last term captures the cumulative loss induced by stateful allocation/control within the episode, subject to  $\pi_k^A \in \Pi(b_k)$ .

### D. Online Optimization Objective and Challenges

The agent aims to minimize the cumulative cross-level joint cost over  $K$  episodes under the cross-level constraints, i.e.,

$$\min_{\{b_k, \pi_k^A\}_{k=1}^K} \sum_{k=1}^K C_k(b_k, \pi_k^A) \quad (5a)$$

$$\text{s.t. } b_k \in \mathcal{B}, \pi_k^A \in \Pi(b_k), \text{ for all } k \in [K]. \quad (5b)$$

Solving (5) online is non-trivial due to the two couplings. First, the upper-level decision must be made without knowing  $f_k$  in advance and its effective loss depends on the (learned) lower-level performance, since the value induced by  $\pi_k^A$  is only realized through stateful dynamics with unknown transitions. Second, the lower level faces a constraint whose threshold is episode-dependent (through  $b_k$ ), and thus requires safety guarantees that remain valid uniformly over the varying budget. Hence, traditional CMDP cannot be applied anymore.

### E. Performance Metric

We evaluate an online algorithm using the static regret, which compares against the best fixed budget and fixed lower-level policy in hindsight. Define the best static optimal solution

$$(b^*, \pi^{A,*}) \in \arg \min_{b \in \mathcal{B}, \pi^A \in \Pi(b)} \sum_{k=1}^K C_k(b, \pi^A), \quad (6)$$

where  $\pi^{A,*}$  could be non-stationary in an episode, i.e.,  $\pi^{A,*} = (\pi_t^*)_{t=1}^T$ . The static regret after  $K$  episodes is then defined as

$$\text{Reg}(K, b, \pi) \triangleq \sum_{k=1}^K [C_k(b_k, \pi_k^A) - C_k(b^*, \pi^{A,*})]. \quad (7)$$

A sublinear bound  $\text{Reg}(K, b, \pi) = o(K)$  implies that the average performance of the online algorithm converges to that of the best fixed budget-policy pair in hindsight.

### III. MOTIVATING EXAMPLE

The formulation in Sec. II captures network settings where a *slow* provisioning decision must be coordinated with *fast*, MDP-dependent scheduling, e.g., queue-aware stochastic network control and scheduling [2], [3], distributed low-complexity scheduling under interference constraints [12], and delay-sensitive wireless scheduling with queue information [13]. We instantiate the framework in **online bandwidth provisioning with queue-aware scheduling**, matching the abstraction in Fig. 1 and the two-time-scale timeline in Fig. 2.

a) *Scenario (two time scales):* Consider a next-generation NodeB (gNB) serving an ultra-reliable low-latency communications (URLLC) slice with stochastic packet arrivals in a radio access network (RAN). Slice provisioning/configuration is updated over coarse control intervals (seconds/minutes), while the medium access control (MAC) scheduler allocates resource blocks (RBs) at fine granularity (milliseconds) in response to queue and channel fluctuations.

b) *Upper level (provisioning with switching overhead):* At the beginning of episode  $k$ , the operator provisions a bandwidth budget  $b_k$ , e.g., total RB-slot units over  $T$  slots. Provisioning incurs an operating cost  $f_k(b_k)$  (spectrum leasing, activation energy, or opportunity cost relative to other slices), which may vary with network conditions and is revealed only after the episode. Reconfiguring the provisioned bandwidth across episodes induces overhead, modeled by  $\alpha \|b_k - b_{k-1}\|^2$ .

c) *Lower level (queue-aware scheduling under a budget):* Within episode  $k$ , over slots  $t = 1, \dots, T$ , the scheduler observes the state  $s_{k,t}$  (e.g., slice queue backlog) and selects an action  $a_{k,t}$  (RB allocation/transmission decision). The action incurs a stage loss  $l_t(s_{k,t}, a_{k,t})$  (delay/backlog penalty reflecting QoS) and consumes resources  $d_t(s_{k,t}, a_{k,t})$  (RB usage), while the queue evolves via arrivals and service dynamics. This statefulness is essential, since insufficient service can accumulate backlog and trigger persistent latency violations.

d) *Online optimization and learning with cross-level coupling:* The provisioned budget constrains scheduling through the episode-wise budget constraint (2), requiring the expected cumulative RB usage within episode to not exceed budget. Hence, the upper level must adapt  $b_k$  online under switching costs, while the performance of a given  $b_k$  is realized through a lower-level CMDP scheduler under unknown dynamics.

---

**Algorithm 1** Bi-Level Optimization and Learning (BLOL)

---

```

1: Initialization:  $b_1 \leftarrow B_0$ ; history  $\mathcal{H} \leftarrow \emptyset$ .
2: for  $k = 1$  to  $K$  do
3:   Execute BALDE (Algorithm 2) in episode  $k$  under
      budget  $b_k$  and obtain  $(\pi_k, \lambda_k, \mathcal{H})$ .
4:   if  $k < K_0$  then
5:      $b_{k+1} \leftarrow B_0$ 
6:   else
7:     Compute approximate subgradient according to
       (11) and update resource budget according to (12).
8:   end if
9: end for

```

---

#### IV. BI-LEVEL ALGORITHM DESIGN

We design an online algorithm for the problem in Sec. II.

##### A. Upper Level OCO via Lower Level Dual Feedback

The upper-level selects a budget sequence  $\{b_k\}_{k=1}^K$  to provision episode-wise resources while accounting for the unknown  $f_k$  and switching overhead. Moreover, the effective loss of  $b_k$  depends on the *best achievable lower-level performance* under the provisioned budget. This relationship is implicit because the operator lacks an analytical model of the packet arrival and scheduling dynamics, i.e., unknown MDP dynamics. We address these challenges as follows and see Algorithm 1.

1) *Reduce bi-level coupling to an “oracle” effective loss:* To isolate the upper-level design, we first analyze an *idealized* setting in which the lower-level scheduling policy can be optimized for any budget  $b_k$ . Define the corresponding cost

$$g_k(b_k) \triangleq f_k(b_k) + \beta L_k^*(b_k), \quad (8)$$

where  $f_k(b_k)$  is the upper-level service cost, and  $L_k^*(b_k)$  is the optimal lower-level scheduling loss under budget  $b_k$ , i.e.,

$$L_k^*(b_k) \triangleq \min_{\pi^A \in \Pi(b_k)} \mathbb{E}_{\pi^A} \left[ \sum_{t=1}^T l_t(s_t, a_t) \right]. \quad (9)$$

Under this *fictitious* oracle view, the upper level would be reduced to an ideal OCO problem over  $\mathcal{B}$ , i.e.,

$$\min_{b_k \in \mathcal{B}} \left\{ \sum_{k=1}^K g_k(b_k) + \sum_{k=1}^K \alpha \|b_k - b_{k-1}\|^2 \right\}. \quad (10)$$

2) *Dual-based subgradient approximation:* Directly optimizing (10) is still challenging because  $L_k^*(b_k)$  depends on the unknown lower-level MDP dynamics. Our key observation is that, by strong duality of the linear programming (LP) formulation of finite-horizon CMDPs, the optimal dual multiplier  $\lambda_k^*(b_k)$  of the budget constraint  $b_k$  provides the sensitivity of the optimal value with respect to the budget. Physically,  $\lambda_k^*(b_k)$  signal the upper level how much QoS could improve if more RBs were allocated. Thus,  $-\lambda_k^*(b_k)$  constitutes a valid subgradient of loss  $L_k^*(b_k)$  (see [?] for details). Specifically, for any  $b_k \in \mathcal{B}$ , the subgradient of the true loss is given by

$$\partial g_k(b_k) = \nabla f_k(b_k) + \beta \partial L_k^*(b_k) = \nabla f_k(b_k) - \beta \lambda_k^*(b_k),$$

However, since the true MDP dynamics are unknown, the true multiplier  $\lambda_k^*(b_k)$  is inaccessible. In our algorithm, we

---

**Algorithm 2** Budget-Adaptive Learning with Dual Estimation

---

```

1: Input: episode  $k$ , budget  $b_k$ , history  $\mathcal{H}$ , confidence level
    $\delta$ , baseline policy  $\pi_{\text{base}}$ .
2: Compute  $\hat{P}_{k,t}$  and  $\beta^p$  using  $\mathcal{H}$  according to (13-15).
3: Calculate estimates  $\bar{l}_{k,t}$  and  $\bar{d}_{k,t}$  according to (16-17).
4: if  $k \leq K_0$  then
5:    $\pi_k^A \leftarrow \pi_{\text{base}}$   $\triangleright$  Use safe baseline during warm-up
6:    $\lambda_k \leftarrow 0$   $\triangleright$  Dual variable is not used in warm-up
7: else
8:   Solve the extended LP (23) with budget  $b_k$ , and obtain
      optimal policy  $\pi_k^A$  and dual multiplier  $\lambda_k$  of the constraint.
9: end if
10: for  $t = 1$  to  $T$  do
11:   Observe state  $s_{k,t}$ , next execute action  $a_{k,t} \sim \pi_{k,t}(\cdot | s_{k,t})$ , and then observe loss  $l_t(s_{k,t}, a_{k,t})$ , consump-
      tion  $d_t(s_{k,t}, a_{k,t})$ , and next state  $s_{k,t+1}$ .
12:   Update history:  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(s_{k,t}, a_{k,t}, s_{k,t+1})\}$ .
13: end for
14: Output:  $(\pi_k, \lambda_k, \mathcal{H})$ .

```

---

approximates it with the empirical dual estimate  $\lambda_k$  returned by the lower-level scheduler. Then, the update direction is

$$\hat{h}_k \triangleq \nabla f_k(b_k) - \beta \lambda_k. \quad (11)$$

While  $\hat{h}_k$  is an approximation of the  $\partial L_k^*(b_k)$ , it acts as an exact subgradient for the *surrogate objective*  $\hat{g}_k(b_k) = f_k(b_k) + \beta \hat{L}_k(b_k)$ , where  $\hat{L}_k(b_k)$  is the surrogate lower-level optimal loss. This connection allows us to analyze the algorithm as performing projected subgradient descent on the surrogate functions  $\{\hat{g}_k\}$ , with the gap between true and surrogate objectives controlled by the lower-level learning guarantees, and it will appear in the upper-level regret bound.

3) *Switching cost control via stable OCO updates:* While the total episode cost includes the switching term  $\alpha \|b_k - b_{k-1}\|^2$ , our update rule operates on (approximate) subgradients of  $g_k$ . As we show in Lemma 2, a decaying step size yields stable budget trajectories with controlled path length, which in turn controls the cumulative switching overhead without requiring a switching regularizer in the update.

4) *Projected subgradient descent with a warm-up phase:* We adopt a warm-up period of  $K_0 = \tilde{\mathcal{O}}(S^2 AT^4/\gamma^2)$  episodes to allow the lower-level learner to collect sufficient samples for dual estimates, and see the proof sketch of Lemma 1.

- *Warm-up ( $k \leq K_0$ ):* set  $b_k = B_0$  and run the lower-level learner to explore the environment.
- *Online update ( $k > K_0$ ):* after observing  $f_k(\cdot)$  and receiving  $\lambda_k$  from the lower level, update the budget

$$b_{k+1} = \Pi_{\mathcal{B}} \left( b_k - \eta_k \hat{h}_k \right), \quad (12)$$

where the step-size is  $\eta_k = 1/[\theta_g(k - K_0 + 1)]$ .

##### B. Lower Level RL Under Upper Level Dynamic Budget

In episode  $k$ , given the budget  $b_k$ , the lower-level scheduler seeks a policy  $\pi_k$  (for brevity, we write  $\pi_k^A$  for  $\pi_k$  in this sub-

section) that minimizes the expected cumulative loss  $V_l(\pi_k; P)$  while satisfying the episode budget constraint  $V_d(\pi_k; P) \leq b_k$ .

To guarantee safety *uniformly* for dynamic constraint threshold and return a dual variable  $\lambda_k$ , we propose *Budget-Adaptive Learning with Dual Estimation* (BALDE), see Algorithm 2.

*1) Empirical estimates and confidence sets:* We focus on uncertainty in the packet arrival and channel dynamics (transition probabilities), and the results can be extended to unknown  $l_t$  and  $d_t$  via standard concentration arguments. For each state-action pair  $(s, a)$  at time slot  $t$ , define the visitation counts prior to episode  $k$  as  $n_{k,t}(s, a) = \sum_{k'=1}^{k-1} \mathbf{1}\{s_{k',t} = s, a_{k',t} = a\}$ ,  $n_{k,t}(s, a, s') = \sum_{k'=1}^{k-1} \mathbf{1}\{s_{k',t} = s, a_{k',t} = a, s_{k',t+1} = s'\}$ . At the start of episode  $k$ , the empirical transition estimate is

$$\hat{P}_{k,t}(s'|s, a) = n_{k,t}(s, a, s') / [n_{k,t}(s, a) \vee 1], \quad (13)$$

Then, we construct a confidence set  $\mathcal{P}_k$  centered at  $\hat{P}_{k,t}$ :

$$\mathcal{P}_k \triangleq \bigcap_{t \in [T], s \in \mathbb{S}, a \in \mathbb{A}} \mathcal{P}_{k,t}(s, a), \quad (14)$$

where, based on the empirical Bernstein inequality [14],  $\mathcal{P}_{k,t}(s, a) = \{P' : |P'_t(s'|s, a) - \hat{P}_{k,t}(s'|s, a)| \leq \beta_{k,t}^p(s, a, s') \text{ for all } s'\}$ , with radius

$$\beta_{k,t}^p(s, a, s') = \sqrt{\frac{4 \text{Var}(\hat{P}_{k,t}(s'|s, a)) L'}{n_{k,t}(s, a) \vee 1}} + \frac{14L'/3}{n_{k,t}(s, a) \vee 1}, \quad (15)$$

and  $L' = \log(\frac{2SATK}{\delta})$ . With probability at least  $1 - 2\delta$ , the true model  $P$  lies in  $\mathcal{P}_k$  for all  $k$  (see Lemma 1 for details).

*2) Pessimism for safety:* To ensure the scheduled resources do not exceed the budget  $b_k$  under model uncertainty, we inflate the constraint cost using a pessimistic penalty, i.e.,

$$\bar{l}_{k,t}(s, a) = d_t(s, a) + T\beta_{k,t}^p(s, a), \quad (16)$$

where  $\beta_{k,t}^p(s, a) \triangleq \sum_{s'} \beta_{k,t}^p(s, a, s')$ .  $\bar{l}_{k,t}$  enforces a conservative constraint that holds uniformly for all  $P' \in \mathcal{P}_k$ .

*3) Budget-aware optimism for efficient exploration:* Pure pessimism can severely restrict exploration, especially for rarely visited state-action pairs. We therefore introduce an optimistic adjustment to the scheduling loss, i.e.,

$$\bar{l}_{k,t}(s, a) = l_t(s, a) - [T^2 \beta_{k,t}^p(s, a)] / (b_k - b_{\text{base}}). \quad (17)$$

The scaling by  $(b_k - b_{\text{base}})^{-1}$  is crucial, since when the current budget  $b_k$  is tight (close to  $b_{\text{base}}$ ), the algorithm reduces aggressive exploration, and when  $b_k$  is larger, the learner can explore more aggressively while still maintaining feasibility.

*4) Optimization under extended LP:* Based on the construction above, we select the scheduling policy as follows,

$$\pi_k = \arg \min_{\{\pi': V_{\bar{l}_{k,t}}(\pi'; P') \leq b_k; P' \in \mathcal{P}_k\}} V_{\bar{l}_{k,t}}(\pi'; P'). \quad (18)$$

This formulation is optimistic in the objective (to learn low-latency scheduling) and pessimistic in the constraint (to ensure RB limits are respected). However, solving (18) efficiently and extracting the dual multiplier associated with the budget constraint are essential and non-trivial in our bi-level design.

To this end, we cast (18) as an *extended LP* in occupancy measures, which serves two roles: (i) it provides a

tractable solver for the lower-level under time-inhomogeneous dynamics, and (ii) it yields the *dual multiplier* estimate  $\lambda_k$  for the budget constraint, which acts as a sensitivity signal (approximately  $\partial_b L_k^*(b_k)$ ) required by the upper-level update. This primal-dual interface is critical not only for algorithm implementation but also for regret analysis, where  $\lambda_k$  connects lower-level learning error to upper-level optimization error. Specifically, for a policy  $\pi$  and transition kernels  $P = \{P_t\}_{t=1}^T$ , define the state-action occupancy measure  $w_t^\pi(s, a; P) \triangleq \Pr(s_t = s, a_t = a | \pi, P)$  [15]. The expected cumulative loss and resource consumption are linear, i.e.,

$$V_r^\pi(P) = \sum_{t=1}^T \sum_{s,a} w_t^\pi(s, a; P) r_t(s, a) = \langle r, w^\pi \rangle, \quad (19)$$

where  $r \in \{l, d\}$  and  $w^\pi = \{w_t^\pi\}_{t=1}^T$ . Any feasible occupancy measure must satisfy standard flow constraints. Let  $\mathcal{W}(P)$  denote the set of nonnegative  $w^\pi$  satisfying, for all  $s \in \mathcal{S}$ ,

$$\sum_a w_t^\pi(s, a) = \sum_{s', a'} P_{t-1}(s | s', a') w_{t-1}^\pi(s', a'). \quad (20)$$

Then, the CMDP under dynamic budget  $b_k$  is equivalent to

$$\min_{w^\pi \in \mathcal{W}(P)} \langle l, w^\pi \rangle, \text{ s.t. } \langle d, w^\pi \rangle \leq b_k. \quad (21)$$

An optimal policy can be obtained by normalization  $\pi_t^*(a | s) = w_t^*(s, a) / \sum_{a'} w_t^*(s, a')$  for nonzero denominators. In BALDE, the transition model is unknown and is optimized within a confidence set  $\mathcal{P}_k$ . To avoid bilinearity between  $P' \in \mathcal{P}_k$  and  $w^\pi$ , we introduce an augmented occupancy measure

$$q_t^\pi(s, a, s') \triangleq w_t^\pi(s, a) P'_t(s' | s, a), \quad (22)$$

which represents the joint occupancy of  $(s_t, a_t, s_{t+1})$ . This yields a linear program in  $q^\pi$  to represent the problem (18) :

$$\min_{q^\pi} \sum_{t,s,a,s'} q_t^\pi(s, a, s') \bar{l}_t(s, a) \quad (23a)$$

$$\text{s.t. } \sum_{t,s,a,s'} q_t^\pi(s, a, s') \bar{d}_t(s, a) \leq b_k, \quad (23b)$$

$$\sum_{a,s'} q_t^\pi(s, a, s') = \sum_{s'', a''} q_{t-1}^\pi(s'', a'', s), \quad (23c)$$

$$\hat{P}_{k,t}(s' | s, a) - \beta_{k,t}^p(s, a, s') \leq \frac{q_t(s, a, s')}{\sum_y q_t(s, a, y)} \quad (23d)$$

$$\leq \hat{P}_{k,t}(s' | s, a) + \beta_{k,t}^p(s, a, s'), \forall s, a, s', t, \quad (23d)$$

$$q_t(s, a, s') \geq 0, \forall s, a, s', t, \quad (23e)$$

where (23c) holds for all state  $s$  and (23d) encodes the transition-dynamics confidence set constraint  $P'_t(\cdot | s, a) \in \mathcal{P}_k$  in a linear form. Crucially, the dual variable associated with the budget constraint (23b) is exactly the multiplier  $\lambda_k$ , which the lower level returns to the upper level as a subgradient and sensitivity feedback for updating the budget  $b_k$ .

*5) Initial safety under dynamic budgets:* As is common in safe RL [8], [9], we assume access to a baseline safe scheduler  $\pi_{\text{base}}$  (e.g., a conservative rate-limiting policy) satisfying  $V_d(\pi_{\text{base}}; P) = b_{\text{base}}$  with  $b_{\text{base}} < B_0$ , which is used to guarantee feasibility in early learning episodes without any pre-samples. We only execute  $\pi_{\text{base}}$  during the warm-up phase.

## V. REGRET ANALYSIS AND BUDGET GUARANTEE

In this section, we establish the regret upper bound and cross-level budget constraint satisfaction guarantee for our algorithm. The main technical challenges lie in quantifying the error quantification from the lower-level reinforcement learning under varying threshold to the upper-level optimization updates under coupling and estimated sensitivity feedback. We provide proof sketches for the lemmas and theorems (due to page limits, see complete proofs in our technical report [?]).

### A. Regret Decomposition

Recall the total regret after  $K$  episodes  $\text{Reg}(K, b, \pi) \triangleq \sum_{k=1}^K C_k(b_k, \pi_k^A) - \sum_{k=1}^K C_k(b^*, \pi^{A,*})$ . For each episode  $k$  and a fixed upper-level decision  $b_k$ , define the *episode-wise optimal safe lower-level policy* under the true model  $P$ :

$$\pi_k^*(b_k) \in \arg \min_{\pi \in \Pi(b_k)} V_l(\pi; P), \quad (24)$$

where  $\Pi(b_k)$  is the feasible policy set induced by budget  $b_k$ . Using  $\pi_k^*(b_k)$  as an intermediate comparator, we can decompose the final regret as follows,

$$\begin{aligned} \text{Reg}(K, b, \pi) &= \underbrace{\sum_{k=1}^K \left( C_k(b_k, \pi_k^A) - C_k(b_k, \pi_k^*(b_k)) \right)}_{\text{(I) Lower-level BALDE regret under budget } b_k: \text{Reg}_{\text{LL}}(K, \pi)} \\ &+ \underbrace{\sum_{k=1}^K \left( C_k(b_k, \pi_k^*(b_k)) - C_k(b^*, \pi^{A,*}) \right)}_{\text{(II) Upper-level BLOL regret with switching costs: Reg}_{\text{UL}}(K, b)}. \end{aligned} \quad (25)$$

Term (I) in (25) represents the lower-level BALDE regret under varying budget  $b_k$ , i.e., the scheduling regret. By the definition of  $C_k$  in (4), for a fixed budget  $b_k$ , the upper-level service cost and switching cost diminishes to zero and we have

$$\text{Reg}_{\text{LL}}(K, \pi) = \beta \sum_{k=1}^K (V_l(\pi_k; P) - V_l(\pi_k^*(b_k); P)). \quad (26)$$

Thus,  $\text{Reg}_{\text{LL}}(K, \pi)$  is exactly the scaled learning regret of the lower-level algorithm BALDE under the budget sequence generated by the upper level. Term (II) in (25) represents the upper-level BLOL regret with switching costs, i.e., the provisioning regret. Using the oracle effective loss  $g_k(b) \triangleq f_k(b) + \beta L_k^*(b)$ , where  $L_k^*(b) = \min_{\pi \in \Pi(b)} V_l(\pi; P)$  (see (9)), the upper-level regret term  $\text{Reg}_{\text{UL}}(K, b)$  becomes

$$\sum_{k=1}^K (g_k(b_k) - g_k(b^*)) + \alpha \sum_{k=1}^K \|b_k - b_{k-1}\|^2. \quad (27)$$

### B. Lower-Level Regret Bound

We first bound the lower-level CMDP scheduling regret.

**Lemma 1** (Lower-level BALDE regret). *Fix  $\delta \in (0, 1)$ . With probability at least  $1 - 3\delta$ , executing BALDE (Algorithm 2) under budget sequence  $\{b_k\}_{k=1}^K$ , we have*

$$\sum_{k=1}^K \left( V_l(\pi_k; P) - V_l(\pi_k^*(b_k); P) \right) \leq \tilde{\mathcal{O}} \left( ST^3 \sqrt{AK} / \gamma \right),$$

where  $\gamma = B_0 - b_{\text{base}}$ .

Lemma 1 provides a *high-probability* regret guarantee for the lower-level scheduler under an *episode-varying* budget sequence  $\{b_k\}_{k=1}^K$ , which is crucial for coupling with the upper-level provisioning decisions. The bound scales as  $\tilde{\mathcal{O}}(\sqrt{K})$ , i.e.,

the average per-episode suboptimality decays as  $\tilde{\mathcal{O}}(1/\sqrt{K})$ . This  $\sqrt{K}$  dependence is optimal and matches the canonical statistical rate for learning under unknown Markov dynamics. The dependence on  $\gamma$  makes explicit a fundamental *safety-learning tradeoff*.  $\gamma$  is the minimum safety slack that allows the algorithm to enforce a pessimistically inflated constraint (to guarantee feasibility under uncertainty) while still retaining room for exploration. When  $\gamma$  is small, safe learners must behave conservatively, so a blow-up with  $1/\gamma$  is unavoidable [8], [9]. Finally, the  $T^3$  factor is characteristic of finite-horizon optimistic exploration with extended-LP-based planning. It reflects that estimation errors compound along length- $T$  trajectories and that additional factors appear when enforcing constraints pessimistically while maintaining optimism.

*Proof sketch of Lemma 1.* The proof follows an optimism-in-the-face-of-uncertainty argument together with pessimism for the constraint, but adapted to a time-varying budget  $b_k$ .

1) *Step 1 (High-probability event):* We construct confidence sets  $\mathcal{P}_k$  around empirical transitions and define a good event  $\mathcal{G}$  under which  $P \in \mathcal{P}_k$  for all  $k$ . Empirical Bernstein bounds then imply  $\Pr(\mathcal{G}) \geq 1 - 3\delta$ .

2) *Step 2 (Safety under a dynamic threshold):* We use a safe baseline policy  $\pi_{\text{base}}$  during warm-up phase  $K_0$  to ensure feasibility when uncertainty is high in early exploration. For  $k > K_0$ , the solver ensures safety by incorporating a cumulative transition bonus  $\epsilon_k^{\pi_k}(P_k) \triangleq T\mathbb{E}[\sum_{t=1}^T \beta_{k,t} | \pi_k, P_k]$ , we have  $V_d(\pi_k; P) \leq V_d(\pi_k; P_k) + \epsilon_k^{\pi_k}(P_k) = V_{\bar{d}}(\pi_k; P_k)$ . Since the solver ensures  $V_{\bar{d}}(\pi_k; P_k) \leq b_k$ , it follows that  $V_d(\pi_k; P) \leq b_k$  under event  $\mathcal{G}$  with a probability  $\Pr(\mathcal{G})$ .

3) *Step 3 (Regret bound):* We decompose the per-episode regret into two terms, i.e.,  $V_l(\pi_k; P) - V_l(\pi^*; P) = (V_l(\pi_k; P) - V_{\bar{l}}(\pi_k; P_k)) + (V_{\bar{l}}(\pi_k; P_k) - V_l(\pi^*; P))$ , where  $V_{\bar{l}}$  is the value under a pessimistic bonus-augmented loss  $\bar{l}_k = l - mT\bar{\beta}_k^p$ . We show that setting the scaling factor  $m \geq T/\gamma$  is sufficient to guarantee  $V_{\bar{l}}(\pi_k; P_k) \leq V_l(\pi^*; P)$ . In addition, the first error term  $V_l(\pi_k; P) - V_{\bar{l}}(\pi_k; P_k)$  in the decomposition can be bounded by relating  $V_{\bar{l}}$  back to  $V_l$  via the classic value difference lemma [7]–[9] and the cumulative bonus. Finally, summing over  $k$  and bounding the cumulative bonus terms  $\sum \epsilon_k$  by invoking the pigeon-hole principle yields the final regret bound of  $\tilde{\mathcal{O}}(T^3 S \sqrt{AK} / \gamma)$ .  $\square$

### C. Upper-Level Regret Bound

We now bound the upper-level OCO provisioning regret.

**Lemma 2** (Upper-level BLOL Regret with Switching Costs). *Surrogate effective cost  $\hat{g}_k(\cdot)$  is  $\theta_g$ -strongly convex on  $\mathcal{B}$  and has bounded subgradients  $\|\partial \hat{g}_k(b)\| \leq G$  for all  $b \in \mathcal{B}$  and all  $k$ . Let  $\{b_k\}$  be updated by projected subgradient descent using the approximate subgradient  $\hat{h}_k$  and step-size  $\eta_k = \frac{1}{\theta_g(k-K_0)}$  for  $k > K_0$ . Then, with probability at least  $1 - 3\delta$ , we have*

$$\text{Reg}_{\text{UL}}(K, b) \leq \underbrace{G^2 \log K / (2\theta_g) + M}_{\text{Surrogate regret and switching costs}} + \underbrace{\tilde{\mathcal{O}}(ST^3 \sqrt{AK} / \gamma)}_{\text{Value approximation error}},$$

where  $M = K_0G(T - B_0) + G^2\pi^2/(6\theta_g^2)$  collects constant overheads from the warm-up phase and the finite switching cost series, and  $\pi$  is the mathematical constant.

Lemma 2 isolates two phenomena. First, the term for surrogate regret and switching costs shows that, despite switching costs, the upper-level optimization error scales only as  $\tilde{\mathcal{O}}(\log K)$  under strong convexity. The key reason switching costs do not destroy the rate is stability, i.e., with  $\eta_k \propto 1/k$ , budget updates shrink over time, so the path length  $\sum_k \|b_k - b_{k-1}\|^2$  is bounded by a convergent series. Second, the term for value approximation error captures the penalty for optimizing the surrogate  $\hat{g}_k$  instead of the true  $g_k$ . This term scales as  $\tilde{\mathcal{O}}(\sqrt{K})$  and matches the lower-level learning regret, which confirms that the overall performance is dominated by the difficulty of learning dynamics and the resulting sensitivity.

*Proof Sketch of Lemma 2.* The core idea is to decompose the regret against the true objective  $g_k$  into the regret on the surrogate  $\hat{g}_k$  and the approximation error. Specifically, we have

$$\begin{aligned} \text{Reg}_{\text{UL}}(K, b) &= \underbrace{\sum_k (\hat{g}_k(b_k) - \hat{g}_k(b^*)) + \text{Switching costs}}_{\text{Surrogate regret and switching costs}} \\ &\quad + \underbrace{\sum_k (g_k(b_k) - \hat{g}_k(b_k)) - (g_k(b^*) - \hat{g}_k(b^*))}_{\text{Value approximation error}}. \end{aligned}$$

1) *Surrogate regret and switching costs:* Since  $\hat{h}_k$  is a valid subgradient of the strongly convex surrogate  $\hat{g}_k$  (as discussed in Sec. IV-A), the first term can be bounded as follows. With step size  $\eta_k \propto 1/k$ , the surrogate regret scales as  $\log K$ . The switching cost at step  $k$  is bounded by  $\|b_k - b_{k-1}\|^2 \leq (G\eta_{k-1})^2 \propto 1/k^2$ . Since the series  $\sum k^{-2}$  converges, the cumulative switching cost is bounded by a constant.

2) *Value approximation error:* The remaining terms capture the value difference between the true and surrogate objectives. By definition,  $g_k(b) - \hat{g}_k(b) = \beta(L_k^*(b) - \hat{L}_k(b))$ . Using the optimism lemma from the lower-level analysis (see [?]), we can show that  $\hat{L}_k(b)$  (the value of the optimistic surrogate) is close to  $L_k^*(b)$  (the true optimal value). The cumulative sum of these differences/errors is bounded by the lower-level regret bound, which thus yields the  $\tilde{\mathcal{O}}(\sqrt{K})$  term in the regret.  $\square$

#### D. Final Regret Bound and Safety Guarantee

Combining (25)-(27), Lemma 1, and Lemma 2, we obtain the final performance and safety guarantee.

**Theorem 1** (Total regret of our bi-level algorithm BLOL). *With probability at least  $1 - 3\delta$ , the regret of the proposed bi-level algorithm BLOL (Algorithm 1) satisfies*

$$\text{Reg}(K, \text{BLOL}) \leq \tilde{\mathcal{O}}\left(ST^3\sqrt{AK}/\gamma\right) + \tilde{\mathcal{O}}(\log K). \quad (28)$$

Theorem 1 establishes a near-optimal  $\tilde{\mathcal{O}}(\sqrt{K})$  regret, implying the average regret vanishes. Importantly, the bound separates the roles of the two levels: the upper level contributes only logarithmic regret (and bounded switching overhead) thanks to strong convexity and stable updates, while the

dominant  $\tilde{\mathcal{O}}(\sqrt{K})$  term comes from learning the unknown lower-level dynamics safely even under time-varying budgets. This clean separation clarifies the benefit of our primal-dual interface. The extended LP provides a dual variable signal that converts lower-level learning progress into correct sensitivity feedback for upper-level subgradient in budget updating.

**Theorem 2** (Safety Guarantee). *With probability at least  $1 - 3\delta$ , the algorithm BALDE (Algorithm 2) satisfies the budget constraint for all episodes  $k \in [K]$ , i.e.,  $V_d(\text{BALDE}; P) \leq b_k$ .*

Theorem 2 establishes the safety of our BALDE subroutine under dynamic/time-varying budget and constraint threshold. It guarantees that the lower-level scheduler strictly adheres to the resource limits imposed by the upper level.

*Proof sketch of Theorem 2.* For the warm-up phase ( $k \leq K_0$ ), safety is guaranteed by baseline policy  $\pi_{\text{base}}$ . For  $k > K_0$ , applying the value difference lemma and Hölder's inequality, the difference between the cost under true transition kernel  $P$  and that under  $P_k$  can be bounded as  $|V_d(\pi_k; P) - V_d(\pi_k; P_k)| \leq \mathbb{E}[\sum_{t=1}^T \|P_t - P_{k,t}\|_1 \|V_{d,t+1}^{\pi_k}\|_\infty] \leq TE[\sum_{t=1}^T \beta_{t,k}^p |\pi_k, P_k|] = \epsilon_k^{\pi_k}(P_k)$ , where  $V_{d,t}^\pi \triangleq \mathbb{E}[\sum_{\tau=t}^T d_\tau(s_\tau, a_\tau) \mid \pi, P]$ . Thus, we can always get  $V_d(\pi_k; P) \leq V_d(\pi_k; P_k) + \epsilon_k^{\pi_k}(P_k) = V_d(\pi_k; P_k) \leq b_k$ . Hence, the budget constraint is satisfied for all episodes with high probability.  $\square$

## VI. NUMERICAL RESULTS

We evaluate the proposed bi-level algorithm (Algorithm 1) on a queueing-based slice scheduling model under both Poisson arrivals (with mean  $\lambda = 1.12$  packets/slot) and real traffic traces MAWI [16] (aggregated into 10ms bins and linearly scaled to mean 1.12 while preserving burstiness). The goal is to jointly provision an episode-level bandwidth budget with reconfiguration overhead and schedule packet service within each episode under an episode-wise budget constraint.

### A. Experimental Configurations

1) *Lower level (queue scheduling learning):* The state  $s_t \in \{0, 1, \dots, S_{\max}\}$  is the queue backlog with  $S_{\max} = 10$ . The action  $a_t \in \{0, 1, 2\}$  is the number of RBs allocated in slot  $t$ . The queue evolves as  $s_{t+1} = \min\{S_{\max}, [s_t - a_t + A_t]^+\}$ , where  $[x]^+ = \max\{x, 0\}$  and  $A_t$  is the packet arrival. We use the latency/backlog stage loss  $l(s_t, a_t) = \mu + (1 - \mu)(s_t/S_{\max})^2$  and the per-slot resource consumption  $d(s_t, a_t) = a_t/2$ , where  $\mu = 0.1$ . We let the horizon  $T = 10$ .

2) *Upper level (budget provisioning optimization):* At the beginning of episode  $k$ , the provisioned budget is  $b_k \in [B_0, T]$ . The provisioning cost is  $f_k(b) = M_1 b + M_2(b - \rho_k)^2 + M_0(b - \rho_0)^2$ , where  $\rho_k = 5.0 + 0.5 \sin(2\pi k/2000) + \epsilon_k$ ,  $\epsilon_k \sim \mathcal{N}(0, 0.1^2)$ ,  $M_1 = 0.01$ ,  $M_2 = 0.05$ , and  $M_0 = 0.25$ . Across episodes we charge the switching cost  $\alpha \|b_k - b_{k-1}\|^2$  with  $\alpha = 0.5$ . We let the scaling parameter  $\beta = 1.0$ .

### B. Evaluation Metrics and Baselines

We report the cumulative *objective gap* relative to a optimal static solution and cumulative budget violation  $\text{Viol}(K) = \sum_{k=1}^K [V_d(\pi_k; P) - b_k]_+$  for all methods. We compare our

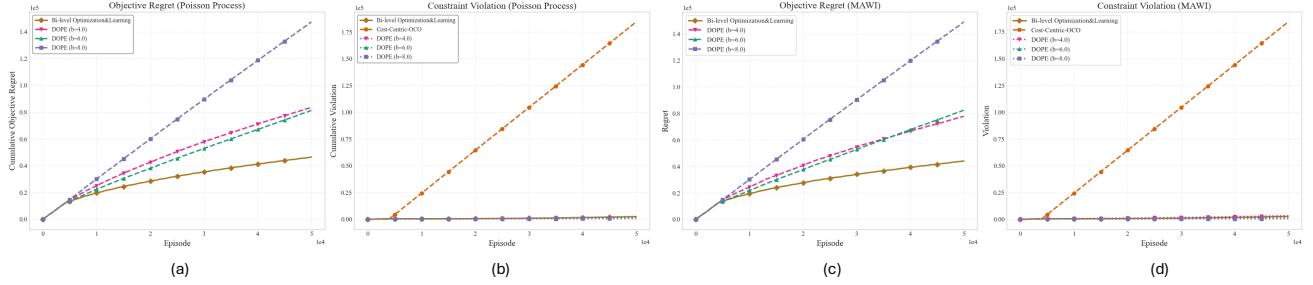


Fig. 3. Cumulative objective gap and cumulative budget violation under (a,b) Poisson arrivals and (c,d) MAWI traces.

algorithm with two baselines, fixed-budget DOPE [17] with  $b \in \{4, 6, 8\}$  (directly using DOPE with given fixed budget  $b$  and without adaptive upper-level provisioning) and fixed-scheduling OCO (directly optimizing  $f_k(\cdot) + \alpha \|b_k - b_{k-1}\|^2$  via OCO, while the lower level runs standard RL with  $Q$ -learning [7] and without dual feedback coupling).

### C. Results

We run  $K = 5 \times 10^4$  episodes. Fig. 3 compares objective gap and constraint violation. Across both Poisson and MAWI arrivals, our proposed bi-level algorithm achieves the smallest cumulative objective gap while maintaining near-zero violations, demonstrating that dual-coupled provisioning reduces cost without sacrificing safety. In contrast, fixed-budget DOPE incurs substantially larger objective gap due to non-adaptive provisioning, and the fixed-scheduling OCO is consistently worse than our proposed algorithm, indicating that ignoring lower-level sensitivity feedback leads to mis-provisioning under stateful dynamics and switching overhead.

## VII. CONCLUSION AND FUTURE WORK

We studied a bi-level online provisioning and scheduling problem with switching costs and cross-level budget constraints. The proposed framework integrates slow-time-scale resource provisioning (modeled as OCO with switching penalties) with fast-time-scale, state-dependent scheduling under varying constraint thresholds (modeled as CMDP under unknown dynamics). Our primal-dual bi-level algorithm couples the two levels through a dual sensitivity signal (the budget multiplier) returned by the lower level, enabling stable budget updates while maintaining safety during exploration. We established near-optimal static regret and high-probability satisfaction of the episode-wise budget constraints.

Future work includes the following directions. First, while the  $\tilde{\mathcal{O}}(\sqrt{K})$  dependence on the number of episodes is order-optimal, tightening the dependence on the horizon and problem parameters (e.g.,  $T$ , the safety slack  $\gamma$ , and state/action sizes) remains an important objective. Second, we will extend the analysis to dynamic benchmarks (e.g., tracking a time-varying comparator  $\{b_k^*\}$ ) to better capture highly non-stationary traffic and network conditions. Third, motivated by network slicing deployments, we plan to study multi-slice and distributed settings where provisioning decisions are coupled across slices (shared spectrum/compute) and control must operate under

limited signaling and partial observability, with scalable implementations of the extended LP subroutine.

## REFERENCES

- [1] S. Lin, M. Shi, A. Arora, R. Bassily, E. Bertino, C. Caramanis, K. Chowdhury, E. Ekici, A. Eryilmaz, S. Ioannidis *et al.*, “Leveraging synergies between ai and networking to build next generation edge networks,” in *2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2022, pp. 16–25.
- [2] X. Lin, N. Shroff, and R. Srikant, “A tutorial on cross-layer optimization in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, 2006.
- [3] L. B. Le, K. Jagannathan, and E. Modiano, “Delay analysis of maximum weight scheduling in wireless ad hoc networks,” in *2009 43rd Annual Conference on Information Sciences and Systems*. IEEE, 2009, pp. 389–394.
- [4] M. Shi, X. Lin, and S. Fahmy, “Competitive online convex optimization with switching costs and ramp constraints,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 876–889, 2021.
- [5] Y. Li, G. Qu, and N. Li, “Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit,” *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4761–4768, 2020.
- [6] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in *Proceedings of the 20th international conference on machine learning (icml-03)*, 2003, pp. 928–936.
- [7] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
- [8] M. Shi, Y. Liang, and N. Shroff, “A near-optimal algorithm for safe reinforcement learning under instantaneous hard constraints,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 31243–31268.
- [9] A. Roknalamouki, A. Ghosh, M. Shi, F. Nourzad, E. Ekici, and N. B. Shroff, “Provably efficient rl for linear mdps under instantaneous safety constraints in non-convex feature spaces,” *arXiv preprint arXiv:2502.18655*, 2025.
- [10] A. G. Barto and S. Mahadevan, “Recent advances in hierarchical reinforcement learning,” *Discrete event dynamic systems*, vol. 13, no. 4, pp. 341–379, 2003.
- [11] E. Hazan *et al.*, “Introduction to online convex optimization,” *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.
- [12] A. Gupta, X. Lin, and R. Srikant, “Low-complexity distributed scheduling algorithms for wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1846–1859, 2009.
- [13] B. Ji, C. Joo, and N. B. Shroff, “Delay-based back-pressure scheduling in multihop wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1539–1552, 2012.
- [14] A. Maurer and M. Pontil, “Empirical bernstein bounds and sample variance penalization,” *arXiv preprint arXiv:0907.3740*, 2009.
- [15] M. Shi, Y. Liang, and N. Shroff, “Near-optimal adversarial reinforcement learning with switching costs,” *arXiv preprint arXiv:2302.04374*, 2023.
- [16] “Mawi working group traffic archive,” <https://mawi.wide.ad.jp/mawi/>.
- [17] A. Bura, A. HasanzadeZonuzy, D. Kalathil, S. Shakkottai, and J.-F. Chamberland, “Dope: Doubly optimistic and pessimistic exploration for safe reinforcement learning,” *Advances in neural information processing systems*, vol. 35, pp. 1047–1059, 2022.