Decoupling Learning and Decision-Making: Breaking the $\mathcal{O}(\sqrt{T})$ Barrier in Online Resource Allocation with First-Order Methods

Wenzhi Gao [1] **Chunlin Sun** [1] **Chenyu Xue** [2] **Yinyu Ye** [1 3]

Abstract

Online linear programming plays an important role in both revenue management and resource allocation, and recent research has focused on developing efficient first-order online learning algorithms. Despite the empirical success of first-order methods, they typically achieve regret no better than $\mathcal{O}(\sqrt{T})$, which is suboptimal compared to the $\mathcal{O}(\log T)$ result guaranteed by the state-of-the-art linear programming (LP)-based online algorithms. This paper establishes several important facts about online linear programming, which unveils the challenge for first-order online algorithms to achieve beyond $\mathcal{O}(\sqrt{T})$ regret. To address this challenge, we introduce a new algorithmic framework which decouples learning from decision-making. For the first time, we show that first-order methods can achieve regret $\mathcal{O}(T^{1/3})$ with this new framework.

1. Introduction

This paper presents a new algorithmic framework to solve the online linear programming (OLP) problem. In this context, a decision-maker receives a sequence of resource requests with bidding prices, and makes irrevocable allocation decisions for these requests sequentially. The goal of OLP is to maximize the accumulated reward subject to a set of inventory constraints. OLP plays an important role in a wide range of applications, such as revenue management (Talluri et al., 2004), resource allocation (Katoh and Ibaraki, 1998), cloud computing (Hussain et al., 2013), and online advertising (Balseiro et al., 2022a).

[1]Institute for Computational and Mathematical Engineering, Stanford University [2]School of Information Management and Engineering, Shanghai University of Finance and Economics [3]Management Science & Engineering, Stanford University. Correspondence to: Wenzhi Gao <gwz@stanford.edu>, Chunlin Sun <chunlin@stanford.edu>, Chunyu Xue <xcy2721d@gmail.com>, Yinyu Ye <yyye@stanford.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Most state-of-the-art algorithms for OLP are dual linear programming (LP)-based (Agrawal et al., 2014; Kesselheim et al., 2014; Jiang et al., 2022; Li and Ye, 2022; Ma et al., 2022). More specifically, they require solving a sequence of LPs to make online decisions. However, the high computational cost of these LP-based methods prevents them from being applied in many time-sensitive or large-scale problems. For example, decisions have to be made instantaneously in online advertising (Balseiro et al., 2022a). This challenge motivates a recent line of research using first-order methods to address OLP (Li et al., 2020; Gao et al., 2023; Balseiro et al., 2022a;b). First-order methods are based on gradient information, and are more scalable and computationally efficient than LP-based methods.

Despite the advantage in computational efficiency, first-order methods are still not comparable to LP-based methods in terms of regret for many settings. Existing first-order OLP algorithms only achieve $\mathcal{O}(\sqrt{T})$ regret bound. The only exception is when the distribution of requests and bidding prices has finite support (Sun et al., 2020). Under the continuous support setting, it remains open:

Can first-order methods go beyond $\mathcal{O}(\sqrt{T})$ regret?

Contributions. This paper takes a first step towards answering this question with the following contributions:

- We characterize a dilemma empirically and theoretically in applying first-order methods to OLP. The dilemma interprets the difficulty in achieving regret and constraint violation better than $\mathcal{O}(\sqrt{T})$ with first-order methods, and also depicts the discrepancy between online learning and decision-making.
- To address the dilemma, we introduce a new online decision-making framework. The idea is to decouple the learning and decision-making procedures with two separate first-order methods and achieve better decision-making by efficiently combining them.
- With the help of this new framework, for the first time, we show that first-order OLP algorithms achieve $\mathcal{O}(T^{1/3})$ regret and constraint violation, which is so far the best result for first-order methods in OLP.

Table 1: Regret bounds in current OLP literature. $\log \log$ factors are ignored.

Paper	Setting and assumptions	Algorithm	Regret	Lower bound
(Li and Ye, 2022)	Bounded, continuous support, non-degeneracy	LP-based	$\mathcal{O}(\log T)$	Yes
(Bray, 2019)	Bounded, continuous support, non-degeneracy	LP-based	$\mathcal{O}(\log T)$	Yes
(Jiang et al., 2022)	Bounded, finite support of $\mathbf{a}_t$	LP-based	$\mathcal{O}(\log^2 T)$	Unknown
(Ma et al., 2022)	Bounded, continuous support, non-degeneracy	LP-based	$\mathcal{O}(\log T)$	Yes
(Chen et al., 2022)	Bounded, finite support, non-degeneracy	LP-based	$\mathcal{O}(1)$	Yes
(Li et al., 2020)	Bounded	First-order Subgradient	$\mathcal{O}(\sqrt{T})$	Yes
(Balseiro et al., 2022a)	Bounded	First-order Mirror Descent	$\mathcal{O}(\sqrt{T})$	Yes
(Gao et al., 2023)	Bounded	First-order Proximal Point	$\mathcal{O}(\sqrt{T})$	Yes
(Balseiro et al., 2022b)	Bounded	First-order and Momentum	$\mathcal{O}(\sqrt{T})$	Yes
(Sun et al., 2020)	Bounded, finite support, non-degeneracy	First-order Subgradient	$\mathcal{O}(T^{3/8})$	No ($\mathcal{O}(1)$)
This paper	Bounded, continuous support, non-degeneracy	First-order Subgradient	$\mathcal{O}(T^{1/3})$	No ($\mathcal{O}(\log T)$)

Related Literature. There is a vast amount of literature on OLP (Ma and Simchi-Levi, 2020; Mirrokni et al., 2012; Mahdian et al., 2012; Arlotto and Gurvich, 2019), and we review some recent developments that reach beyond $\mathcal{O}(\sqrt{T})$ regret in the stochastic input setting (**Table 1**). These algorithms mostly follow the same principle of making decisions based on the learned information: learning and decision-making are closely coupled with each other. We refer the interested readers to (Balseiro et al., 2023) for a more detailed review on OLP and relevant problems.

LP-based OLP Algorithms. Most LP-based OLP algorithms leverage the dual LP problem (Agrawal et al., 2014), with only a few exceptions (Kesselheim et al., 2014). Under assumptions of either non-degeneracy or finite support on resource requests and/or rewards, $\mathcal{O}(\log T)$ regret has been achieved under different settings. More specifically, Li and Ye (2022) establish the dual convergence of finite-horizon LP solution to the optimal dual solution to the underlying stochastic program. In the continuous support setting, $\mathcal{O}(\log T \log \log T)$ regret is achieved. Bray (2019) considers multi-secretary problem and establishes an $\mathcal{O}(\log T)$ regret result. Ma et al. (2022) consider the setting where a regularization term is imposed on the resource and also establish an $\mathcal{O}(\log T)$ result. Jiang et al. (2022) establish $\mathcal{O}(\log^2 T)$ regret without nondegeneracy, which assumes that the distribution of resource requests has finite support. Chen et al. (2022) consider the case where both resource requests and prices have finite support, and constant regret can be achieved in this case.

First-order OLP Algorithms. Early explorations of first-order OLP algorithms start from (Li et al., 2020) and (Balseiro et al., 2022a; Lobos et al., 2021), where $\mathcal{O}(\sqrt{T})$ regret is established using mirror descent and subgradient methods. Gao et al. (2023) show that proximal point update also achieves $\mathcal{O}(\sqrt{T})$ regret. Recently, Balseiro et al. (2022b) analyze a momentum variant of mirror descent and get $\mathcal{O}(\sqrt{T})$ regret. Under the finite support assumption, Sun et al. (2020) design a three-stage algorithm that achieves $\mathcal{O}(T^{3/8})$ regret. To our knowledge, this is the only instance of first-order OLP algorithm that goes beyond $\mathcal{O}(\sqrt{T})$.

Structure of the Paper. The rest of the paper is organized as follows. **Section 2** introduces the problem setup and main assumptions; In **Section 3**, we unveil a dilemma between online learning and decision-making; In **Section 4**, we present our framework that decouples learning from decision-making, and show that our framework achieves better regret than $\mathcal{O}(\sqrt{T})$. In **Section 5**, we conduct numerical experiments to validate our theoretical findings.

2. Problem Setup

Notations. Throughout the paper, we use $\|\cdot\|$ to denote Euclidean norm and $\langle \cdot, \cdot \rangle$ to denote Euclidean inner product. Bold letters notations $\mathbf{A}$ and $\mathbf{a}$ denote matrices and vectors respectively. Given a convex function f, its subdifferential is denoted by $\partial f(\mathbf{x})$ and $f'(\mathbf{x}) \in \partial f(\mathbf{x})$ is called a subgradient. $[\cdot]_+ = \max\{\cdot, 0\}$ denotes element-wise positive part function and $\mathbb{I}\{\cdot\}$ denotes the 0-1 indicator function. Given iteration count T, we use $\mathcal{O}(1/\sqrt{T}), \mathcal{O}(1/t)$ to denote stochastic gradient-based methods that use fixed stepsize and adaptive stepsize proportional to $1/t$ respectively.

2.1. Online Linear Programming and Duality

Consider an online resource allocation problem over time horizon $T \geq 1$: With an initial inventory of $m \geq 1$ resources $\mathbf{b} \in \mathbb{R}^m$ and average inventory $\mathbf{d} = \frac{\mathbf{b}}{T}$, at time $t = 1, \ldots, T$, a customer with order $(c_t, \mathbf{a}_t) \in \mathbb{R} \times \mathbb{R}^m$ arrives and requests resource $\mathbf{a}_t$ at bidding price c_t. Decision $x^t \in [0, 1]$ is made to either accept or reject the order. We receive reward c_t once the order is accepted.

Define reward vector $\mathbf{c} := (c_1, \ldots, c_T)^\top \in \mathbb{R}^T$ and request matrix $\mathbf{A} := (\mathbf{a}_1, \ldots, \mathbf{a}_T) \in \mathbb{R}^{m \times T}$. We can write the problem compactly as an "offline" LP:

$$\max_{\mathbf{0}\le\mathbf{x}\le\mathbf{1}} \quad \langle \mathbf{c}, \mathbf{x}\rangle \quad \text{subject to} \quad \mathbf{A}\mathbf{x} \le \mathbf{b}, \tag{P}$$

where $\mathbf{0}$ and $\mathbf{1}$ denote vectors of all zeros and ones, respectively. The dual problem of (P) is given by

$$\min_{(\mathbf{y},\mathbf{s})\ge\mathbf{0}} \ \langle \mathbf{b}, \mathbf{y}\rangle + \langle \mathbf{1}, \mathbf{s}\rangle \quad \text{subject to} \quad \mathbf{A}^\top \mathbf{y} + \mathbf{s} \ge \mathbf{c}. \tag{D}$$

According to (Li et al., 2020), we can eliminate $\mathbf{s} \ge \mathbf{0}, \mathbf{s} \ge \mathbf{c} - \mathbf{A}^\top \mathbf{y}$, and write (D) more compactly as

$$\min_{\mathbf{y}\ge\mathbf{0}} \quad f_T(\mathbf{y}) \coloneqq \tfrac{1}{T}\textstyle\sum_{t=1}^{T}[\langle \mathbf{d}, \mathbf{y}\rangle + [c_t - \langle \mathbf{a}_t, \mathbf{y}\rangle]_+], \tag{1}$$

and recall that $\mathbf{d} = \frac{\mathbf{b}}{T}$ is the average inventory. $f_T(\mathbf{y})$ can be viewed as the sample approximation of the function

$$f(\mathbf{y}) \coloneqq \mathbb{E}[f_T(\mathbf{y})] = \langle \mathbf{d}, \mathbf{y}\rangle + \mathbb{E}_{(c,\mathbf{a})}[[c - \langle \mathbf{a}, \mathbf{y}\rangle]_+], \tag{2}$$

if coefficient pairs $(c_t, \mathbf{a}_t)$ are drawn from some fixed distribution. Next, we respectively define

$$\mathbf{y}_T^\star \in \arg\min_{\mathbf{y}\ge\mathbf{0}} \ f_T(\mathbf{y}) \quad \text{and} \quad \mathbf{y}^\star \in \arg\min_{\mathbf{y}\ge\mathbf{0}} \ f(\mathbf{y})$$

and LP optimality conditions reveal the following connection between the primal-dual optimal solution pair,

$$x_t^\star \in \begin{cases} \{0\}, & c_t < \langle \mathbf{a}_t, \mathbf{y}_T^\star\rangle, \\ [0,1], & c_t = \langle \mathbf{a}_t, \mathbf{y}_T^\star\rangle, \\ \{1\}, & c_t > \langle \mathbf{a}_t, \mathbf{y}_T^\star\rangle. \end{cases} \tag{3}$$

This connection suggests that the primal optimal solution is largely determined by the dual through optimality conditions. This motivates dual-based algorithms for OLP.

2.2. Dual-based OLP Algorithms

Dual-based OLP algorithms work as follows:
Given an online learning algorithm

$$\mathbf{y}^+ = \mathcal{A}^t \coloneqq \mathcal{A}(\{(c_1, \mathbf{a}_1), \ldots, (c_t, \mathbf{a}_t)\}),$$

we maintain and update a dual solution sequence $\{\mathbf{y}^t\}_{t=1}^T$ in an online fashion; primal decisions $\{x^t\}_{t=1}^T$ are made simultaneously based on $\{\mathbf{y}^t\}_{t=1}^T$ and (3).

Algorithm 1 Dual-based OLP algorithm

Input: $\mathbf{y}^1, (\mathbf{A}, \mathbf{b}, \mathbf{c})$, algorithm $\mathcal{A}$
for t = 1 to T **do**

$$\mathbf{y}^{t+1} = \mathcal{A}^t \tag{4}$$

Make decision x^t according to (4) and (3)
end

Algorithm 1 illustrates the framework. The common choices of $\mathcal{A}$ are summarized as follows:

- LP-based method: Let $x^t = \mathbb{I}\{c_t \ge \langle \mathbf{a}_t, \mathbf{y}_t\rangle\}$, solve LP with data $\{(c_j, \mathbf{a}_j), j \le t\}$ and output $\mathbf{y}^{t+1} = \mathbf{y}_t^\star$.
- First-order method: Let $x^t = \mathbb{I}\{c_t \ge \langle \mathbf{a}_t, \mathbf{y}_t\rangle\}$, compute a stochastic subgradient of f at $\mathbf{y}^t$ and output $\mathbf{y}^{t+1}$:

$$\mathbf{g}_t = \mathbf{d} - \mathbf{a}_t x^t \in \partial_{\mathbf{y}=\mathbf{y}_t}[\langle \mathbf{d}, \mathbf{y}\rangle + [c_t - \langle \mathbf{a}_t, \mathbf{y}\rangle]_+],$$
$$\mathbf{y}^{t+1} = \arg\min_{\mathbf{y}\ge\mathbf{0}} \ \{\langle \mathbf{g}_t, \mathbf{y}\rangle + \tfrac{1}{2\alpha}\|\mathbf{y} - \mathbf{y}^t\|^2\}. \tag{5}$$

This paper focuses on the subgradient method. Variants of the subgradient method, including mirror descent (Balseiro et al., 2022a) and proximal point (Gao et al., 2023), have also been analyzed in the literature. Compared to the LP-based methods, first-order methods have much lower computational costs and memory requirements.

2.3. Performance Metric

Given the output of online algorithm $\hat{\mathbf{x}} = (x^1, \ldots, x^T)$, we define regret and constraint violation, respectively, to be

$$r(\hat{\mathbf{x}}) \coloneqq \max_{\mathbf{A}\mathbf{x}\le\mathbf{b}, \mathbf{0}\le\mathbf{x}\le\mathbf{1}} \ \langle \mathbf{c}, \mathbf{x}\rangle - \langle \mathbf{c}, \hat{\mathbf{x}}\rangle, \tag{6}$$
$$v(\hat{\mathbf{x}}) \coloneqq \|[\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}]_+\|. \tag{7}$$

These metrics are standard when measuring performance of online algorithms (Gao et al., 2023; Li et al., 2020).

2.4. Assumptions and Auxiliary Results

We make the following assumptions throughout the paper.

A1: $\{(c_t, \mathbf{a}_t)\}$ are generated i.i.d. from distribution $\mathcal{P}$.

A2: There exist constants $\bar{a}, \bar{c} > 0$ such that $\|\mathbf{a}\|_\infty \le \bar{a}$ and $|c| \le \bar{c}$ almost surely.

A3: The average resource $\mathbf{d} = \mathbf{b}/T$ satisfies

$$\underline{d} \cdot \mathbf{1} \le \mathbf{d} \le \bar{d} \cdot \mathbf{1},$$

where $0 < \underline{d} \le \bar{d}$.

A4: Second moment $\mathbb{E}[\mathbf{a}\mathbf{a}^\top]$ is positive definite with minimum eigenvalue λ_0.

A5: There exist $\lambda_1, \lambda_2 > 0$ such that for $(c, \mathbf{a}) \sim \mathcal{P}$,

$$\begin{aligned} &\lambda_1 |\langle \mathbf{a}, \mathbf{y} - \mathbf{y}^\star\rangle| \\ &\le |\mathbb{P}\{c \ge \langle \mathbf{a}, \mathbf{y}\rangle | \mathbf{a}\} - \mathbb{P}\{c \ge \langle \mathbf{a}, \mathbf{y}^\star\rangle | \mathbf{a}\}| \\ &\le \lambda_2 |\langle \mathbf{a}, \mathbf{y} - \mathbf{y}^\star\rangle| \end{aligned}$$

for all $\mathbf{y} \in \Xi_1 \coloneqq \{\mathbf{y} : \mathbf{y} \ge \mathbf{0}, \|\mathbf{y}\| \le \frac{\bar{c}}{\underline{d}} + 1\}$.

A6: $\mathbf{y}^\star$ satisfies $y_i^\star = 0$ if and only if

$$d_i - \mathbb{E}_{(c,\mathbf{a})}[a_i \mathbb{I}\{c > \langle \mathbf{a}, \mathbf{y}^\star\rangle\}] > 0$$

for all $i = 1, \ldots, m$.

The above assumptions are identical to Assumption 2 from Li and Ye (2022), except Ξ_1 is defined with respect to Euclidean norm for convenience of analysis. To our knowledge, Li and Ye (2022) require a stronger version of **A5** to get $\mathcal{O}(\log T \log\log T)$ regret, and so far, no first-order algorithm can reach regret beyond $\mathcal{O}(\sqrt{T})$ under **A1** to **A6**.

With the assumptions above, we immediately have the following auxiliary result from Li and Ye (2022).

Lemma 2.1 (Proposition 2 in (Li and Ye, 2022)). *Assume* **A1** *to* **A6** *and let* $\mu = \frac{\lambda_0\lambda_1}{2} > 0$, *then*

$$f(\mathbf{y}) - f(\mathbf{y}^\star) - \langle f'(\mathbf{y}^\star), \mathbf{y} - \mathbf{y}^\star\rangle \geq \tfrac{\mu}{2}\|\mathbf{y} - \mathbf{y}^\star\|^2, \quad (8)$$
$$\langle f'(\mathbf{y}^\star), \mathbf{y} - \mathbf{y}^\star\rangle \geq 0,$$

for all $\mathbf{y} \in \Xi_1$, *where* $f'(\mathbf{y}^\star) \in \partial f(\mathbf{y}^\star) = \{\nabla f(\mathbf{y}^\star)\}$. *Moreover,* $\mathbf{y}^\star \in \Xi_0 := \{\mathbf{y} : \mathbf{y} \geq \mathbf{0}, \|\mathbf{y}\| \leq \frac{\bar{c}}{\underline{d}}\}$ *is unique.*

Remark 2.1. Our results essentially require uniqueness of $\mathbf{y}^\star$ and the growth condition in **Lemma 2.1**. In other words, we can adopt the assumptions from other OLP literature which result in the same conditions (Bray, 2019; Chen et al., 2022; Jiang et al., 2022; Ma et al., 2022). Besides, suppose distribution $\mathcal{P}$ has finite support, then the structure of (D) guarantees the growth condition through polyhedral error bound (Yang and Lin, 2018). Therefore, our results can be extended to the finite-support setting if $\mathbf{y}^\star$ is unique.

Lemma 2.1 shows that the expected dual objective (2) has a unique optimal solution and exhibits quadratic growth, also known as semi-strong convexity (Yang and Lin, 2018). Since Li and Ye (2022) achieves $\mathcal{O}(\log T \log\log T)$ regret with a slightly stronger version of **A5** and **Lemma 2.1**, it is natural to expect that first-order methods also achieve regret better than $\mathcal{O}(\sqrt{T})$. However, in the next section, we present a dilemma, which almost prevents first-order OLP algorithms from achieving better performance.

3. Dilemma between Learning and Decision

In this section, we discuss a dilemma between learning and decision-making for first-order methods solving OLP. That is, a good estimation of $\mathbf{y}^\star$ benefits decision-making or achieving better regret and constraint violation than $\mathcal{O}(\sqrt{T})$ for some first-order algorithms (**Proposition 3.1**). To learn such a good estimation, one needs a slow updating speed in **Algorithm 1**, or a tiny stepsize α in the updating rule (5). However, such a tiny step size will prevent **Algorithm 1** from quickly updating the dual solution y^t as a response to new emerging reward and request pairs, resulting in a bad regret and constraint violation. Consequently, it seems impossible to break $O(\sqrt{T})$ regret through first-order algorithms, since they cannot have small and large stepsizes simultaneously.

3.1. Benefit in Learning Better

We start from discussing the benefit of learning a better dual solution for the dual-based online algorithm (**Algorithm 1**) by the following proposition.

Proposition 3.1. *Under* **A1** *to* **A6**, *given an estimate* $\mathbf{y}^1$ *of* $\mathbf{y}^\star$ *such that* $\|\mathbf{y}^1 - \mathbf{y}^\star\| = \mathcal{O}(1/T^\theta)$ *for some* $\theta \geq 0$, *then* ***Algorithm 1*** *with the subgradient update (5) and stepsize* $\mathcal{O}(T^{-\frac{2\theta'+1}{2\theta'+2}})$ *achieves* $\mathcal{O}(T^{\frac{1}{2\theta'+2}})$ *regret and constraint violation, where* $\theta' = \min\{\theta, 1/2\}$.

Proposition 3.1 shows that a better estimate of $\mathbf{y}^\star$ benefits the performance of OLP algorithms. In particular, when $\mathbf{y}^1 = \mathbf{y}^\star$, $\mathcal{O}(T^{1/3})$ regret and constraint violation are achieved at the same time. **Proposition 3.1** also reveals the significance of knowing the distribution $\mathcal{P}$. Specifically, Li and Ye (2022) show that even if the distribution $\mathcal{P}$ is given such that $\mathbf{y}^\star$ can be computed prior to the decision-making process, **Algorithm 1** with $\mathbf{y}^t = \mathbf{y}^\star$ for all t only achieves $\mathcal{O}(\sqrt{T})$ regret under **A1** to **A6**. This result seems to suggest that knowing $\mathcal{P}$ does not yield an improvement beyond $\mathcal{O}(\sqrt{T})$. However, **Proposition 3.1** points out that this knowledge at least helps obtain $\mathcal{O}(T^{1/3})$ regret, thereby opening up the possibility of achieving even better regret with knowledge of $\mathcal{P}$.

To understand the discrepancy between Li and Ye (2022) and **Proposition 3.1**, we remark that **Proposition 3.1** complements rather than conflicts with the results in Li and Ye (2022). In particular, $\mathbf{y}^\star$, the dual optimal solution for the expectation problem (2), can be different from $\mathbf{y}^\star_T$, the dual optimal solution of the realized problem (1), because of randomness. The fluctuations can result in $\mathcal{O}(\sqrt{T})$ regret and constraint violation. Compared to making decisions based on a fixed dual solution $\mathbf{y}^\star$, **Proposition 3.1** suggests that combining $\mathbf{y}^\star$ with SGD leads to better regret. In other words, dual solution has to be adjusted, to adapt to realized samples from the environment.

Although $\mathbf{y}^\star$ brings benefits for OLP, in practice, $\mathbf{y}^\star$ is generally unknown and needs to be learned along the horizon T. When one does not design a good learning algorithm, or when $\theta = 0$ in **Proposition 3.1**, the algorithm in **Proposition 3.1** cannot achieve better regret and constraint violation than $\mathcal{O}(\sqrt{T})$ by itself. This challenge also exists in the current literature about first-order methods for OLP. Specifically, the existing first-order methods used in OLP (Li et al., 2020; Balseiro et al., 2022a;b) are sub-optimal in learning $\mathbf{y}^\star$ and cannot break the $\mathcal{O}(\sqrt{T})$ barrier for regret and constraint violation. They share the same convergence rate as the first method listed in **Table 2** in learning $\mathbf{y}^\star$, but they are worse than other algorithms in **Table 2** which are designed to exploit the quadratic growth property (8) with a smaller stepsize. One natural idea is that, first-order methods, which are better at learning, also contributes to better decision-making. However, this intuition is incorrect.

3.2. Impossibility in Deciding Better

We now investigate the performance of **Algorithm 1** with first-order methods better at learning $\mathbf{y}^\star$, such as SGD with known μ, ASSG, and SADAGRAD in **Table 2** that has smaller stepsizes but a faster convergence rate compared with SGD in **Table 2**. Unfortunately, achieving regret better than $\mathcal{O}(\sqrt{T})$ remains impossible even with those first-order methods with faster convergence rates in learning $\mathbf{y}^\star$. Particularly, we illustrate this impossibility by considering the following one-dimensional linear programming problem (9), also known as the online multi-secretary problem:

$$\max_{0\leq x^t\leq 1} \sum_{t=1}^{T} c_t x^t \quad \text{subject to} \quad \sum_{t=1}^{T} x^t \leq \tfrac{T}{2}. \tag{9}$$

Here, c_t is sampled uniformly from $[0,1]$ for all $t = 1,\ldots,T$. We can compute $\mu = 1$, $y^\star = \frac{1}{2}$ is the median of the distribution of the objective, and $y^\star_T$ is the median of the realized samples $\{c_t\}_{t=1}^T$. However, their regret and constraint violation bounds still grow at order $\mathcal{O}(\sqrt{T})$.

To interpret this suboptimal performance of these gradient descent methods with a fast convergence rate but a small stepsize, one key factor is their limited capability in adaptability. Note that all these algorithms share a similarity that their stepsizes are $\mathcal{O}(1/T)$ in the last $\mathcal{O}(T)$ iterations. *This choice of stepsizes is required to guarantee fast convergence, but it simultaneously reduces the changing rate of the dual solution*. Consequently, it cannot adapt to the changes of the emerging samples in online decision-making. One specific example of lack of adaptability is that, once a mistake is made in estimating the dual solution (i.e., the gap between the estimated dual solution and $\mathbf{y}^\star$ is large), these algorithms cannot mitigate the estimation error for a long time. This estimation error will lead to suboptimal decisions thereafter, even until the end of the horizon. The following proposition theoretically illustrates this slow-updating issue for SGD with known μ.

Proposition 3.2. *Denote y^t as the estimated dual solution for the online secretary problem (9) at time t by SGD with known μ. If there exists $t_0 \geq T/10+1$ such that $y^{t_0} \geq y^\star + \frac{1}{\sqrt{T}}$, then $\mathbb{E}[y^t|y^{t_0}] \geq y^\star + \frac{1}{20\sqrt{T}}$ for all $t \geq t_0$.*

Proposition 3.2 tells that once SGD with known μ estimates the dual optimal solution with an $\mathcal{O}(1/\sqrt{T})$ error for one step, this error cannot be corrected and will be carried for all succeeding steps on expectation. More importantly, as shown in **Proposition 3.3**, this lack of adaptability also leads to suboptimal performance in decision-making, ruling out their possibility of achieving regret and constraint violation better than $\mathcal{O}(\sqrt{T})$.

Proposition 3.3. ***Algorithm 1*** *using SGD with known μ cannot achieve $\mathcal{O}(T^\beta)$ regret and constraint violation simultaneously for any $\beta < \frac{1}{2}$.*

We have shown that the discrepancy between online learning and decision-making: algorithms efficient in learning fail to achieve regret and constraint violation better than $\mathcal{O}(\sqrt{T})$ due to their small stepsizes, and weakness in adaptability. Conversely, the beginning of **Section 3** shows the importance of online learning in online decision-making: to achieve regret and constraint violation better than $\mathcal{O}(\sqrt{T})$, a good estimate of the dual optimal solution is required for first-order methods. As in **Proposition 3.1**, the proposed algorithm has a relatively larger stepsize to guarantee adaptability. However, this algorithm cannot learn a good dual solution on its own. These two sides depict a dilemma in first-order OLP algorithms: good decision-making requires a large stepsize and a good learned estimation, while good learning requires relatively a small stepsize. First-order methods cannot achieve these two aspects simultaneously.

Escaping the Horns of a Dilemma. The aforementioned dilemma seems discouraging for applying first-order methods in online linear programming or online decision-making. However, this dilemma is built on the assumption that one learns a sequence of dual solutions and makes decisions solely based on the same learned sequence. To address this challenge, in the next section, we introduce a two-path approach for online decision-making that maintains decision-making and learning paths independently. This new approach enjoys the strengths of both aspects, which leads to a first-order method achieving $\mathcal{O}(T^{1/3})$ regret and constraint violation without knowledge of $\mathbf{y}^\star$.

4. Decoupling Learning and Decision-Making

In this section, we present our algorithm framework. The dilemma we discussed in the previous section reveals a critical challenge: we might not be able to find a first-order method that is good at both learning and decision-making. However, the low cost of first-order methods opens up another way: instead of having to choose between learning and decision-making, it is possible to take the best of both worlds by using two different algorithms simultaneously: a learning algorithm $\mathcal{A}_L$ and a decision algorithm $\mathcal{A}_D$.

This simple idea yields a highly flexible framework:

1) We can choose $\mathcal{A}_L$ and $\mathcal{A}_D$ to be good learning and decision-making algorithms, respectively.

2) Information from $\mathcal{A}_L$ is flexibly incorporated into $\mathcal{A}_D$.

To illustrate the power of the framework, we show how its simple variant breaks the $\mathcal{O}(\sqrt{T})$ barrier of OLP. We also remark that the framework is broadly applicable to problems where learning and decision happen simultaneously.

Table 2: First-order methods for problems with quadratic growth property (8). SGD with μ refers to subgradient method with known growth parameter μ. Stepsizes of ASSG and SADAGRAD decay to $\mathcal{O}(1/T)$ in the last $\mathcal{O}(T)$ iterations; ASSG and SADAGRAD are parameter-free with respect to μ.

Algorithm	Stepsize	Convergence Rate $\|\mathbf{y}^k - \mathbf{y}^\star\|$	Parameter Free
SGD (Garrigos and Gower, 2023)	$\mathcal{O}(1/\sqrt{T})$	$\mathcal{O}(1/k^{1/4})$ for all $k = 1, \ldots, T$	Yes
SGD with known μ (Rakhlin et al., 2011)	$\mathcal{O}(1/(\mu t))$	$\mathcal{O}(1/\sqrt{k})$ for all $k = 1, \ldots, T$	No
Parameter-free ASSG (Xu et al., 2017)	$\mathcal{O}(1)$ to $\mathcal{O}(1/T)$	$\mathcal{O}(1/\sqrt{T}), k = T$	Yes
Parameter-free SADAGRAD (Chen et al., 2018)	$\mathcal{O}(1)$ to $\mathcal{O}(1/T)$	$\mathcal{O}(1/\sqrt{T}), k = T$	Yes

4.1. Algorithm Design

We are ready to introduce our algorithm, a realization of the aforementioned framework. We start by choosing $\mathcal{A}_D$ to be subgradient method and $\mathcal{A}_L$ to be any of the algorithms from **Table 2**. $\mathcal{A}_D$ and $\mathcal{A}_L$ generate two paths of dual sequences $\{\mathbf{y}^t\}_{t=1}^T$ and $\{\mathbf{y}_L^t\}_{t=1}^T$ respectively. $\mathcal{A}_D$ accesses information from $\mathcal{A}_L$ with a one-time *restart* strategy: we divide total horizon T into two phases: exploration and exploitation (**Figure 1** and **Algorithm 2**).

Exploration. During exploration phase from $t = 1$ to T_e, $\mathcal{A}_D$ and $\mathcal{A}_L$ run simultaneously but independently. Subgradient method $\mathcal{A}_D$ is equipped with stepsize α_e.

Exploitation. At $t = T_e + 1$, $\mathcal{A}_D$ restarts from $\mathbf{y}_L^{T_e+1}$ with a different stepsize α_p. Due to our simple one-time restart strategy, $\mathcal{A}_L$ stops after $t \geq T_e + 1$, $\mathbf{y}_L^t \equiv \mathbf{y}_L^{T_e+1}, t > T_e$.

Algorithm 2 Decoupling learning and decision-making

Input: $\mathbf{y}^1, (\mathbf{A}, \mathbf{b}, \mathbf{c}), \mathcal{A}_{\mathcal{L}}, \mathcal{A}_{\mathcal{D}}, \alpha_e, \alpha_p$
explore for t = 1 to T_e **do**
  $\mathbf{y}^{t+1} = \mathcal{A}_D^t$ with stepsize α_e and make decision x^t
  $\mathbf{y}_L^{t+1} = \mathcal{A}_L^t$
end
restart $\mathcal{A}_{\mathcal{D}}$ by $\mathbf{y}_L^{T_e+1}$
exploit for t = $T_e + 1$ to T **do**
  $\mathbf{y}^{t+1} = \mathcal{A}_D^t$ with stepsize α_p and make decision x^t
end

Remark 4.1. **Algorithm 2** is a simple realization of our framework, and the framework can be implemented very flexibly, for example, by using different $\mathcal{A}_D$ or with a multi-stage restart strategy.

4.2. Algorithm Analysis

The next two lemmas show the regret and violation using the aforementioned two-path two-phase algorithm.

Lemma 4.1 (Regret)**.** *Assuming* **A1** *to* **A6** *hold, we have*

$$\mathbb{E}[r(\hat{\mathbf{x}})] \leq \tfrac{m(\bar{a}+\bar{d})^2\alpha_e}{2} T_e + \tfrac{m(\bar{a}+\bar{d})^2\alpha_p}{2} T_p + \tfrac{\mathsf{R}}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|],$$

where $\mathsf{R} := \frac{\bar{c}}{\underline{d}} + \big[\frac{m(\bar{a}+\bar{d})^2}{2\underline{d}} + \sqrt{m}(\bar{a}+\bar{d})\big] \cdot \max\{\alpha_e, \alpha_p\}$.

Lemma 4.2 (Violation)**.** *Assuming* **A1** *to* **A6** *hold, we have*

$$\mathbb{E}[v(\hat{\mathbf{x}})] \leq \tfrac{\mathsf{R}}{\alpha_e} + \tfrac{1}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|],$$

where R *is defined in* ***Lemma 4.1***.

Remark 4.2. **Lemma 4.1** and **4.2** suggest that the distance to $\mathbf{y}^\star$ indeed plays a role in the bound. We also observe that:

1. T_e cannot be too large, since the best we can do in the exploration phase is $\mathcal{O}(\sqrt{T_e})$, and we cannot spend too much time in exploration.
2. The distance term is dominated by $\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\|$ rather than $\|\mathbf{y}^T - \mathbf{y}^\star\|$ alone. These two facts suggest that if overly small stepsize is used to reduce $\|\mathbf{y}^T - \mathbf{y}^\star\|$, the first term $\frac{1}{\alpha_p}\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\|$ will instead blow up, which aligns with our observation from **Section 3**.

In other words, after T_e steps of exploration drive $\mathbf{y}^{T_e}$ to an $o(1)$ proximity of $\mathbf{y}^\star$, the most "economical" strategy is to keep impetus and travel around within this neighborhood. Therefore, α_p should be determined based on the radius of the neighborhood around $\mathbf{y}^\star$.

After establishing enough intuitions, we now present instances of $(\mathcal{A}_L, T_e, \alpha_e, \alpha_p)$ in the theorem below.

Theorem 4.1. *Assuming* **A1** *to* **A6** *hold, then for all* $T \geq \frac{9m^2(\bar{a}+\bar{d})^4}{4\underline{d}^2}$, *we have:*

M0. *If* $T_e = 0$*, there is no exploration. With* $\alpha_p = T^{-1/2}$,
$$\mathbb{E}[r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})] \leq \mathcal{O}(\sqrt{T}).$$

M1. *If* $\mathcal{A}_L$ *is taken to be SGD with stepsize* $\mathcal{O}(1/\sqrt{T_e})$, *then with* $T_e = T^{4/5}, \alpha_e = T^{-2/5}, \alpha_p = T^{-3/5}$,
$$\mathbb{E}[r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})] \leq \mathcal{O}(T^{2/5}).$$

M2. *If* $\mathcal{A}_L$ *is taken to be SGD with stepsize* $\mathcal{O}(1/(\mu t))$, *then with* $T_e = T^{2/3}, \alpha_e = T^{-1/3}, \alpha_p = T^{-2/3}$,
$$\mathbb{E}[r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})] \leq \mathcal{O}(T^{1/3}).$$

M3. *If* $\mathcal{A}_L$ *is taken to be either ASSG or SADAGRAD, then with* $T_e = T^{2/3}, \alpha_e = T^{-1/3}, \alpha_p = T^{-2/3}$,
$$\mathbb{E}[r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})] \leq \mathcal{O}(T^{1/3}\log T).$$

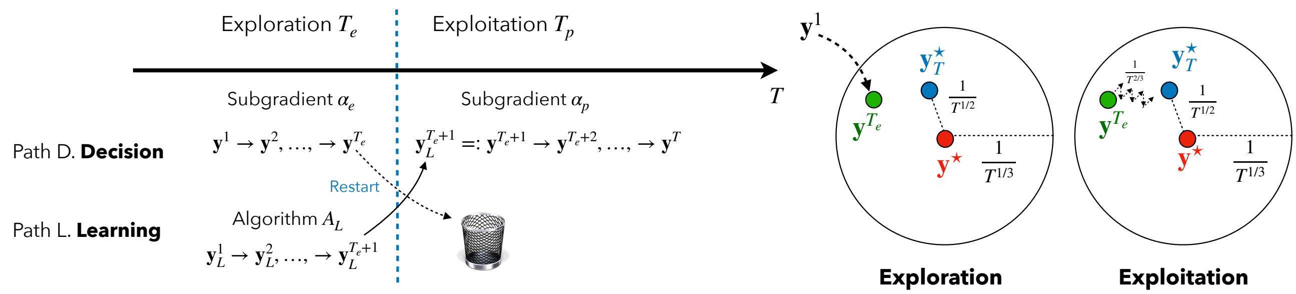

Figure 1: Left: two-path, two-phase framework. Right: illustration of restart, exploration phase sends $\mathbf{y}^{T_e}$ into a neighborhood of $\mathbf{y}^\star$, and in the exploitation phase, $\mathbf{y}^t$ stays in this neighborhood with not too tiny stepsize to make adjustments.

Based on the learning ability of $\mathcal{A}_L$, there exists some choice of $(T_e, \alpha_a, \alpha_p)$ that captures the best trade-off between exploration and exploitation. With this framework, the $\mathcal{O}(\sqrt{T})$ barrier in online LP is broken.

Discussion on μ. We make some remarks on the effect of parameter μ. First, as in strongly convex optimization, only an upper bound of μ is needed for the algorithm to work. Second, even if there is no way to estimate μ, we can use parameter-free algorithms discussed in **M3**, which will incur a $\log T$ factor but can exhibit much better dependency in constant. Finally, even if for some problems μ is close to 0 (as we will demonstrate in the experiment), our algorithm is still robust and exhibits $\mathcal{O}(\sqrt{T})$ regret empirically.

Technical Intuitions. Technically, our algorithm design follows the following intuitions:

1. Using theory of concentration (Li and Ye, 2022),
$$\mathbb{E}[\|\mathbf{y}^\star - \mathbf{y}_T^\star\|] \le \mathcal{O}(T^{-1/2}).$$
2. Using theory of stochastic first-order methods, $\mathbf{y}^t$, the sequence generated by SGD with fixed stepsize α, stay in a noise ball of radius $\mathcal{O}(\sqrt{\alpha})$ around $\mathbf{y}^\star$ if it starts from some initial point $\mathbf{y}^1$ such that $\|\mathbf{y}^1 - \mathbf{y}^\star\| \le \mathcal{O}(\sqrt{\alpha})$. In other words,
$$\mathbb{E}[\|\mathbf{y}^\star - \mathbf{y}^t\|] \le \mathcal{O}(\sqrt{\alpha})$$
3. The closer $\mathbf{y}^t$ is to $\mathbf{y}_T^\star$, the better regret we get.

With these intuitions, $\mathbf{y}^\star$ naturally acts as a bridge between $\mathbf{y}_T^\star$ and $\mathbf{y}^t$: we achieve improved regret so long as **1**). $\mathbf{y}^1$ is close to $\mathbf{y}^\star$ and **2**). SGD stepsize α is taken proportional to the initial distance. Particularly, since $\mathbf{y}_T^\star$ can appear anywhere within the $\mathcal{O}(T^{-1/2})$ size neighborhood around $\mathbf{y}^\star$, we have to remain adaptive to "catch up with" $\mathbf{y}_T^\star$ in this $\mathcal{O}(T^{-1/2})$ ball. Therefore, we should not take overly small α to avoid "over-fitting" $\mathbf{y}^\star$.

Lastly, in practice we do not have $\|\mathbf{y}^1 - \mathbf{y}^\star\| \le \mathcal{O}(\sqrt{\alpha})$ beforehand. This motivates us to consider first spending extra time learning (exploring), so that a good initial $\mathbf{y}^1$ can be obtained to adjust the algorithm. Further taking into account the dilemma from **Section 3**, we obtain the two-path and two-phase algorithm framework in this section. The intuition behind our algorithm design is simple but useful.

5. Numerical Experiments

This section conducts experiments to illustrate the performance and the theoretical results of our framework. In particular, we consider a benchmark algorithm from literature,

- **M0**: No exploration with $T_e = 0$ and $\alpha_p = T^{-1/2}$.

and two instances from **Theorem 4.1**.

- **M1**: $\mathcal{A}_L$ and $\mathcal{A}_D$ are both SGD with fixed stepsize $T_e^{-1/2}$ in the first $T_e = T^{4/5}$ time period.
- **M2**: $\mathcal{A}_L$ is SGD with $\mathcal{O}(1/(\mu t))$ stepsize and $\mathcal{A}_D$ is SGD with stepsize $T_e^{-1/2}$ in the first $T_e = T^{2/3}$ time. We always take $\mu = 1$, and do not tune it through the experiments.

Our experiment contains four parts. In the first part, we generate different distributions of $\{(c_t, \mathbf{a}_t)\}_{t=1}^T$ that satisfy the assumptions specified in **Section 2**, and assess the above algorithms' performance. In the second part, we turn to the distributions that violate at least one of the assumptions and discuss the performance of our algorithm. In the third part, we compare first-order and LP-based OLP algorithms in regret and running time. Finally, we justify the optimality of our stepsize choice in the fourth experiment.

5.1. Performance under Assumptions

In this part, we randomly generate $\{(c_t, \mathbf{a}_t)\}_{t=1}^T$ from distributions satisfying **A1** to **A6**. Three algorithms' performance is evaluated in terms of $r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})$.

We choose $m \in \{1, 5\}$ and T evenly spaced over $[10^2, 10^6]$ on log-scale. For each value of T, each algorithm's performance is averaged over 100 random trials under each distribution. For all the distributions, each d_i is sampled

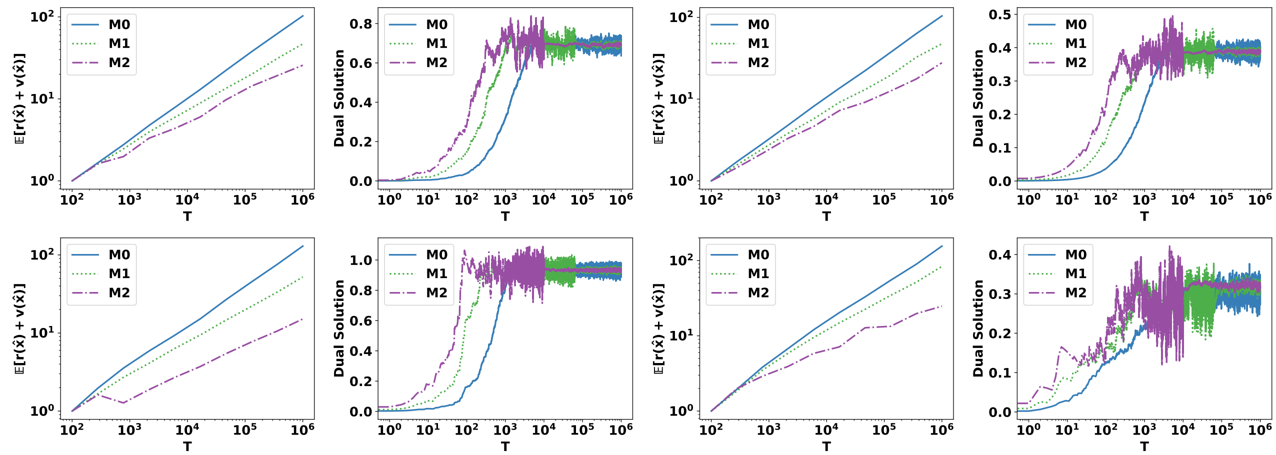

Figure 2: Growth of normalized $r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})$ and dual convergence of algorithms when the assumptions hold. Each pair of the left figure and the right figure is plotted based on the experiment on the same distribution.

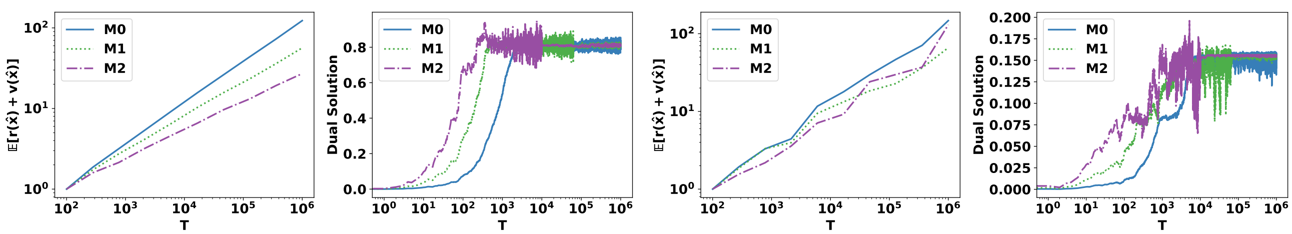

Figure 3: Growth of normalized $r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})$ of three tested algorithms when the assumptions are violated. The left two: normal distribution; the right two: discrete distribution.

i.i.d. from uniform distribution $\mathcal{U}[1/3, 2/3]$. $\{(c_t, \mathbf{a}_t)\}_{t=1}^T$ is generated in the following way: **1)**. For the first distribution (Li et al., 2020), we take $m = 1$, and sample each a_{it} and c_t i.i.d. from $\mathcal{U}[0, 2]$; **2)**. For the second distribution (Li and Ye, 2022), we let $m = 1$ and each $a_{it} = 1$, and randomly sample each c_t i.i.d. from $\mathcal{U}[0, 1]$; **3)**. For the third distribution (Li and Ye, 2022), we randomly generate a_{it} from the truncated standard Cauchy distribution with the location parameter 1 and the threshold ± 10, and let $c_t = \sum_{i=1}^m a_{it} - \varepsilon_t$ with ε_t from $\mathcal{U}[0, m]$; **4)**. For the fourth distribution, we let $m = 5$, sample a_{it} and c_t i.i.d. from the $\mathcal{U}[1, 6]$ and $\mathcal{U}[0, 3]$, respectively.

For each distribution and algorithm, we normalize the average of $r + v$ by its minimal empirical value and plot its growth behavior with respect to T. Then we fix $T = 10^6$ and plot the convergence of $\{y^t\}_{t=1}^T$ for the case $m = 1$ and the last coordinate $\{y_m^t\}_{t=1}^T$ of $\{\mathbf{y}^t\}_{t=1}^T$ for the case $m = 5$. **Figure 2** clearly suggests that **M1** and **M2** have better order of performance compared to **M0**, which is consistent with our theory. Meanwhile, the dual solution in **M0** converges slower than in **M1**, **M2**, and exhibits more oscillation around the optimal dual solution $\mathbf{y}^\star$.

5.2. Performance when Assumptions are Violated

In the second part, we turn to distributions that violate at least one of the assumptions from **A1** to **A6**. We choose m, T and generate $\{d_i\}$ as in the first experiment. We generate $\{(c_t, \mathbf{a}_t)\}_{t=1}^T$ as follows. **1)**. For the first distribution (Li et al., 2020), we take $m = 1$, and each a_{it} is generated from normal distribution $\mathcal{N}(1, 1)$. We let each $c_t = \sum_{i=1}^m a_{it} - \varepsilon_t$ with ε_t from $\mathcal{U}[0, m]$. This distribution violates **A2**. **2)**. The second distribution has finite support and violates **A5**. More specifically, we choose $m = 1$ and randomly generate 10 different pairs of $(c_t, \mathbf{a}_t)$. Each c_t and each element in $\mathbf{a}_t$ is sampled from $\mathcal{U}[0, 1]$. After obtaining $\{(c_t, \mathbf{a}_t)\}_{t=1}^T$, at each time period t, we sample $(c_t, \mathbf{a}_t)$ from these pairs uniformly.

As in the first experiment, **Figure 3** plots the growth of normalized $r + v$ and convergence of $\{y^t\}_{t=1}^T$. **M1** and **M2** perform better than **M0** under the first distribution, with the order still being $\mathcal{O}(T^{1/3})$. Even for the discrete distribution, **M1** and **M2** still exhibit slightly better performance than **M0**. This further shows the robustness of our methods.

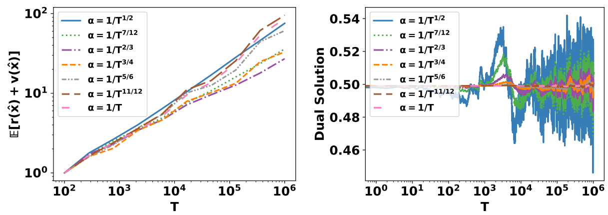

Figure 4: Left: y-axis: normalized $r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})$ under different choices of α_p. x-axis: different T. Right: Dual convergence under different α_p.

5.3. Comparing First-order and LP-based Methods

In this section, we compare first-order OLP algorithms with an LP-based method in both regret and running time. We choose **M0**, **M1**, **M2** and an LP-based algorithm **MLP** from Li and Ye (2022, Algorithm 3). **MLP** solves an LP of t columns at time t, and it achieves $\mathcal{O}(\log T)$ regret. We generate instances according to the first distribution in **Section 5.1** with $m = 2$ and $T \in \{10^3, 10^4, 10^5\}$. For each (m, T) pair, we average the result over 10 experiments.

Table 3: Comparison of first-order and LP-based methods.

T	Algorithm	Avg. Regret	Avg. Time(s)
10^3	**M0**	12.37	< 0.001
	M1	7.04	< 0.001
	M2	4.18	< 0.001
	MLP	3.82	0.95
10^4	**M0**	38.24	< 0.01
	M1	23.46	< 0.01
	M2	13.83	< 0.01
	MLP	4.12	37.5
10^5	**M0**	123.03	6.3×10^{-2}
	M1	47.86	6.4×10^{-2}
	M2	24.00	6.4×10^{-2}
	MLP	5.91	4742.9

Table 3 summarizes the comparison between first-order and LP-based methods. Although the LP-based method demonstrates superior performance in terms of regret, it requires significantly more computational time compared to the first-order method. For instance, **MLP** takes over one hour when $T = 10^5$, whereas the first-order method **M2** only needs 0.064 seconds. Meanwhile, **M2** shows up to 5x improvement over **M0**, with an almost negligible computational cost. Therefore, our proposed framework effectively balances efficiency and regret performance.

5.4. Validation of Stepsize Choice

Our last experiment serves as a validation of our theoretical analysis. **Theorem 4.1** shows that there exists an optimal choice of α_p and for **M2** the choice is $T^{-2/3}$. To verify this, we search $\alpha_p = T^{-\beta}, \beta \in \{\frac{1}{2}, \frac{7}{12}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}, \frac{11}{12}, 1\}$ and record performance of algorithms under different stepsizes. We also plot the dual convergence behavior for $T = 10^6$ as in previous subsections. **Figure 4** suggests that the best choice of α_p exactly appears around $T^{-2/3}$, which verifies our theoretical findings.

6. Conclusion

In this paper, we unveil the dilemma of first-order online linear programming algorithms. We develop a novel first-order online learning framework that decouples learning and decision-making, which for the first time achieves better than $\mathcal{O}(\sqrt{T})$ regret under continuous distribution. Our new framework and analysis provide new insights for online linear programming, which we believe can motivate new algorithms and better theoretical guarantees for online decision-making problems.

Acknowledgement and Disclosure of Funding

The authors are grateful to the Area Chairs and the anonymous reviewers for their constructive comments. This research is partially supported by National Natural Science Foundation of China (Grant 72394360, 72394364).

Impact Statement

This paper proposes new algorithms for online linear programming problems and shows improved results for first-order online algorithms. We believe that our work has no negative social impact.

References

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

Alessandro Arlotto and Itai Gurvich. Uniformly bounded regret in the multisecretary problem. *Stochastic Systems*, 9(3):231–260, 2019.

Santiago R Balseiro, Haihao Lu, and Vahab Mirrokni. The best of many worlds: Dual mirror descent for online allocation problems. *Operations Research*, 2022a.

Santiago R Balseiro, Haihao Lu, Vahab Mirrokni, and Balasubramanian Sivan. From online optimization to PID controllers: Mirror descent with momentum. *arXiv preprint arXiv:2202.06152*, 2022b.

Santiago R Balseiro, Omar Besbes, and Dana Pizarro. Survey of dynamic resource-constrained reward collection problems: Unified model and analysis. *Operations Research*, 2023.

Robert L Bray. Logarithmic regret in multisecretary and online linear programming problems with continuous valuations. *arXiv e-prints*, pages arXiv–1912, 2019.

Guanting Chen, Xiaocheng Li, and Yinyu Ye. An improved analysis of lp-based control for revenue management. *Operations Research*, 2022.

Zaiyi Chen, Yi Xu, Enhong Chen, and Tianbao Yang. Sadagrad: Strongly adaptive stochastic gradient methods. In *International Conference on Machine Learning*, pages 913–921. PMLR, 2018.

Wenzhi Gao, Dongdong Ge, Chunlin Sun, and Yinyu Ye. Solving linear programs with fast online learning algorithms. In *International Conference on Machine Learning*, pages 10649–10675. PMLR, 2023.

Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

Hameed Hussain, Saif Ur Rehman Malik, Abdul Hameed, Samee Ullah Khan, Gage Bickler, Nasro Min-Allah, Muhammad Bilal Qureshi, Limin Zhang, Wang Yongji, Nasir Ghani, et al. A survey on resource allocation in high performance distributed computing systems. *Parallel Computing*, 39(11):709–736, 2013.

Jiashuo Jiang, Will Ma, and Jiawei Zhang. Degeneracy is OK: Logarithmic Regret for Network Revenue Management with Indiscrete Distributions. *arXiv*, 2022. doi: 10.48550/arxiv.2210.07996.

Naoki Katoh and Toshihide Ibaraki. Resource allocation problems. *Handbook of Combinatorial Optimization: Volume1–3*, pages 905–1006, 1998.

Thomas Kesselheim, Andreas Tönnis, Klaus Radke, and Berthold Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 303–312, 2014.

Xiaocheng Li and Yinyu Ye. Online linear programming: Dual convergence, new algorithms, and regret bounds. *Operations Research*, 70(5):2948–2966, 2022.

Xiaocheng Li, Chunlin Sun, and Yinyu Ye. Simple and fast algorithm for binary integer and online linear programming. *Advances in Neural Information Processing Systems*, 33:9412–9421, 2020.

Alfonso Lobos, Paul Grigas, and Zheng Wen. Joint online learning and decision-making via dual mirror descent. In *International Conference on Machine Learning*, pages 7080–7089. PMLR, 2021.

Wanteng Ma, Ying Cao, Danny H K Tsang, and Dong Xia. Optimal Regularized Online Convex Allocation by Adaptive Re-Solving. *arXiv*, 2022. doi: 10.48550/arxiv.2209.00399.

Will Ma and David Simchi-Levi. Algorithms for online matching, assortment, and pricing with tight weight-dependent competitive ratios. *Operations Research*, 68 (6):1787–1803, 2020.

Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Online optimization with uncertain information. *ACM Transactions on Algorithms (TALG)*, 8(1): 1–29, 2012.

Vahab S Mirrokni, Shayan Oveis Gharan, and Morteza Zadimoghaddam. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1690–1701. SIAM, 2012.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

Rui Sun, Xinshang Wang, and Zijie Zhou. Near-optimal primal-dual algorithms for quantity-based network revenue management. *arXiv preprint arXiv:2011.06327*, 2020.

Kalyan T Talluri, Garrett Van Ryzin, and Garrett Van Ryzin. *The theory and practice of revenue management*, volume 1. Springer, 2004.

Yi Xu, Qihang Lin, and Tianbao Yang. Stochastic convex optimization: Faster local growth implies faster global convergence. In *International Conference on Machine Learning*, pages 3821–3830. PMLR, 2017.

Tianbao Yang and Qihang Lin. RSG: Beating subgradient method without smoothness and strong convexity. *The Journal of Machine Learning Research*, 19(1):236–268, 2018.

Appendix

Table of Contents

Structure of the Appendix The appendix is organized as follows. In **Section A**, we introduce auxiliary results for the discussion of our framework; **Section B** presents our two-path two-phase algorithm in a more general framework; **Section C, D** prove the main results in our paper. **Section E** presents additional experiment results.

A. Auxiliary Results

In this section, we provide auxiliary results and definitions that will help present our main results. We start by assuming the existence of a dual learning algorithm with θ-convergence rate.

A.1. θ-Convergence Rate of Dual Learning Algorithms

Proposition A.1. *A dual learning algorithm $\mathcal{A}_t = \mathcal{A}(\{(c_j, \mathbf{a}_j), j \leq t\})$ has θ-convergence rate if at least with probability $1 - \frac{1}{T}$, it outputs $\mathcal{A}_T = \mathbf{y}^T$ such that*

$$\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|] \leq \frac{c_1 + c_2 \log T}{T^\theta},$$

where $c_1, c_2 > 0$ are constants independent of T.

Given a learning algorithm, **Proposition A.1** can be verified through its convergence rate. Here are some examples.

Table 4: Summary of convergence rates

Algorithm	First-order	Knowledge of μ	θ	c_2
LP-resolving (Li and Ye, 2022)	No	No	$1/2$	> 0
God's perspective	–	–	∞	$= 0$
Subgradient with stepsize $\alpha_t \equiv 1/\sqrt{T}$	Yes	No	$1/4$	$= 0$
Subgradient with stepsize $\alpha_t = 1/(\mu t)$	Yes	Yes	$1/2$	$= 0$
Parameter-free ASSG (Xu et al., 2017)	Yes	No	$1/2$	> 0
Parameter-free SADAGRAD (Chen et al., 2018)	Yes	No	$1/2$	$= 0$

If $\theta = 0$, we say that the learning algorithm does not converge. On the other hand, by $\theta = \infty$, we mean the algorithm outputs exactly $\mathbf{y}^\star$. This happens, for example, if the distribution $\mathcal{P}$ is known. From now we assume the existence of a dual learning algorithm with convergence rate $\theta > 0$.

A7: There exists a dual learning algorithm with convergence rate $\theta > 0$.

A.2. Verification of Convergence Rates

In this subsection, we verify the choice of θ, c_1, c_2 for the aforementioned algorithms.

A.2.1. LP-RESOLVING

In LP-resolving, we have $\mathbf{y}^T = \mathbf{y}_T^\star$ and we have the following result.

Lemma A.1 ((Li and Ye, 2022))**.** *Assume that* **A1** *to* **A6** *hold and that $T \geq \max\{m, 3\}$. Then there exists some constant $C > 0$ such that*

$$\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|] \leq \frac{C\sqrt{(m \log m + 1) \log \log T}}{\sqrt{T}}$$

Lemma A.2. *LP-resolving satisfies $\theta = 1/2$.*

A.2.2. SUBGRADIENT

Analysis of subgradient is similar to that in strongly convex case (Lacoste-Julien et al., 2012), where, at each iteration

$$\mathbf{y}^{t+1} = \Pi_{\mathbb{R}_+^m \cap \Xi_0}(\mathbf{y}^t - \alpha_t \mathbf{g}^t) \tag{10}$$

for some stepsize α_t and $\Pi_\mathcal{S}(\cdot)$ denotes orthogonal projection onto set $\mathcal{S}$. Recall that in **Lemma 2.1** we have $\Xi_0 = \{\mathbf{y} : \mathbf{y} \geq \mathbf{0}, \|\mathbf{y}\| \leq \frac{\bar{c}}{\underline{d}}\}$. Unless specified, we will use $\Pi(\cdot)$ to denote $\Pi_{\mathbb{R}_+^m \cap \Xi_0}(\cdot)$ to simplify notation. The following lemma characterizes the behavior of subgradient method.

Lemma A.3. *Assume that A1 to A6 hold, then if $\alpha_t < \mu$, subgradient update (10) satisfies*

$$\mathbb{E}_t[\|\mathbf{y}^{t+1} - \mathbf{y}^\star\|^2] \leq (1 - \alpha_t \mu)\|\mathbf{y}^t - \mathbf{y}^\star\|^2 + \alpha_t^2 m(\bar{a} + \bar{d})^2$$

for all $t \geq 1$, where $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot|\{(c_j, \mathbf{a}_j), j \leq t\}]$ denotes the conditional expectation on history.

Plugging in different choices of α_t, a telescopic sum completes the proof.

Lemma A.4 (Subgradient with constant stepsize)**.** *Under the same assumptions as **Lemma A.3**, if $\alpha_t \equiv \alpha < 1/\mu$, then*

$$\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|^2] \leq \frac{\Delta^2}{\mu \alpha T} + \frac{m(\bar{a} + \bar{d})^2}{\mu}\alpha,$$

where $\Delta = \|\mathbf{y}_1 - \mathbf{y}^\star\|$. Taking $\alpha_t = 1/\sqrt{T}$ gives

$$\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|^2] \leq \frac{\Delta^2 + m(\bar{a} + \bar{d})^2}{\mu\sqrt{T}}.$$

Therefore, subgradient with constant stepsize satisfies $\theta = 1/4$ and $c_2 = 0$.

Lemma A.5 (Subgradient with $1/(\mu t)$ stepsize)**.** *Under the same assumptions as **Lemma A.3**, if $\alpha_t = 2/(\mu(t+1))$, then*

$$\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|^2] \leq \frac{4m(\bar{a} + \bar{d})^2}{\mu^2 T}.$$

Therefore, subgradient with $\mathcal{O}(1/(\mu t))$ stepsize satisfies $\theta = 1/2$ and $c_2 = 0$.

A.2.3. PARAMETER-FREE ALGORITHMS

In this subsection, we discuss the convergence rate of two parameter-free algorithms (Xu et al., 2017; Chen et al., 2018) from literature. These two algorithms employ a double-loop multi-stage restart strategy and work without knowledge of μ. We assume that T is sufficiently large and is taken exactly after some restart iteration.

Lemma A.6 (ASSG)**.** *For ASSG, we have $\theta = 1/2$ and $c_2 > 0$.*

Lemma A.7 (SADAGRAD)**.** *For SADAGRAD we have $\theta = 1/2$ and $c_2 = 0$.*

A.3. Proof of Results in Section A

Proof of Lemma A.2. Taking $\theta = 1/2, c_2 = C\sqrt{m \log m + 1}$ completes the proof.

Proof of Lemma A.3. We successively deduce that

$$\begin{aligned} \|\mathbf{y}^{t+1} - \mathbf{y}^\star\|^2 &= \|\Pi(\mathbf{y}^t - \alpha_t \mathbf{g}^t) - \mathbf{y}^\star\|^2 && (11) \\ &\leq \|\mathbf{y}^t - \alpha_t \mathbf{g}^t - \mathbf{y}^\star\|^2 \\ &= \|\mathbf{y}^t - \mathbf{y}^\star\|^2 - 2\alpha_t \langle \mathbf{y}^t - \mathbf{y}^\star, \mathbf{g}^t \rangle + \alpha_t^2 \|\mathbf{g}^t\|^2 \\ &\leq \|\mathbf{y}^t - \mathbf{y}^\star\|^2 - 2\alpha_t \langle \mathbf{y}^t - \mathbf{y}^\star, \mathbf{g}^t \rangle + \alpha_t^2 m(\bar{a} + \bar{d})^2, && (12) \end{aligned}$$

where (11) uses the non-expansiveness of the projection operator; (12) uses

$$\|\mathbf{d} - \mathbb{I}\{c_k > \langle \mathbf{a}_k, \mathbf{y} \rangle\}\mathbf{a}_k\|^2 \leq m(\bar{a} + \bar{d})^2.$$

Since $\mathbb{E}[\mathbf{g}^t] \in \partial f(\mathbf{y}^t)$, we have, by convexity of f, that

$$-2\langle \mathbf{y}^t - \mathbf{y}^\star, \alpha_t \mathbb{E}[\mathbf{g}^t] \rangle \leq -2\alpha_t (f(\mathbf{y}^t) - f(\mathbf{y}^\star))$$

and we invoke **Lemma 2.1** to get

$$f(\mathbf{y}^t) - f(\mathbf{y}^\star) \geq f(\mathbf{y}^t) - f(\mathbf{y}^\star) - \langle \nabla f(\mathbf{y}^\star), \mathbf{y}^t - \mathbf{y}^\star \rangle \geq \tfrac{\mu}{2}\|\mathbf{y}^t - \mathbf{y}^\star\|^2.$$

Conditioned on the history and taking expectation, we have

$$\begin{aligned}
&\|\mathbf{y}^t - \mathbf{y}^\star\|^2 - 2\mathbb{E}[\langle \mathbf{y}^t - \mathbf{y}^\star, \alpha_t \mathbf{g}^t \rangle] + \alpha_t^2 m(\bar{a} + \bar{d})^2 \\
\leq\ & \|\mathbf{y}^t - \mathbf{y}^\star\|^2 - 2\alpha_t (f(\mathbf{y}^t) - f(\mathbf{y}^\star)) + \alpha_t^2 m(\bar{a} + \bar{d})^2 \\
\leq\ & \|\mathbf{y}^t - \mathbf{y}^\star\|^2 - \alpha_t \mu \|\mathbf{y}^t - \mathbf{y}^\star\|^2 + \alpha_t^2 m(\bar{a} + \bar{d})^2 \\
=\ & (1 - \alpha_t \mu)\|\mathbf{y}^t - \mathbf{y}^\star\|^2 + \alpha_t^2 m(\bar{a} + \bar{d})^2.
\end{aligned}$$

and this completes the proof.

Proof of Lemma A.4. Unrolling the recursion from **Lemma A.3** till $\mathbf{y}^1$, we have

$$\begin{aligned}
\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|^2] &\leq (1 - \mu\alpha)\mathbb{E}[\|\mathbf{y}^T - \mathbf{y}^\star\|^2] + \alpha^2 m(\bar{a} + \bar{d})^2 \\
&\leq (1 - \mu\alpha)^T \|\mathbf{y}^1 - \mathbf{y}^\star\|^2 + \textstyle\sum_{j=0}^{T-1} \alpha^2 m(\bar{a} + \bar{d})^2 (1 - \mu\alpha)^j \qquad (13) \\
&\leq (1 - \mu\alpha)^T \|\mathbf{y}^1 - \mathbf{y}^\star\|^2 + \tfrac{m(\bar{a}+\bar{d})^2}{\mu}\alpha \\
&\leq \tfrac{1}{\mu\alpha T}\|\mathbf{y}^1 - \mathbf{y}^\star\|^2 + \tfrac{m(\bar{a}+\bar{d})^2}{\mu}\alpha \\
&= \tfrac{\Delta^2}{\mu\alpha T} + \tfrac{m(\bar{a}+\bar{d})^2}{\mu}\alpha, \qquad (14)
\end{aligned}$$

where (13) uses the relation $\sum_{j=0}^{T-1}(1 - \mu\alpha)^j = \frac{1-(1-\mu\alpha)^T}{\mu\alpha} \leq \frac{1}{\mu\alpha}$ and (14) is by $(1 - \mu\alpha)^T \leq \frac{1}{1+\mu\alpha T} \leq \frac{1}{\mu\alpha T}$ since $\mu\alpha < 1$. This completes the proof.

Proof of Lemma A.5. Using **Lemma A.3** and with our choice $\alpha_t = \frac{2}{\mu(t+1)}$,

$$\begin{aligned}
\mathbb{E}_t[\|\mathbf{y}^{t+1} - \mathbf{y}^\star\|^2] &\leq (1 - \alpha_t \mu)\|\mathbf{y}^t - \mathbf{y}^\star\|^2 + \alpha_t^2 m(\bar{a} + \bar{d})^2 \\
&= \frac{t-1}{t+1}\|\mathbf{y}^t - \mathbf{y}^\star\|^2 + \frac{4m(\bar{a} + \bar{d})^2}{\mu^2 (t+1)^2}.
\end{aligned}$$

Multiply both sides by $(t + 1)^2$, we get

$$(t+1)^2 \mathbb{E}_t[\|\mathbf{y}^{t+1} - \mathbf{y}^\star\|^2] \leq (t^2 - 1)\|\mathbf{y}^t - \mathbf{y}^\star\|^2 + \frac{4m(\bar{a} + \bar{d})^2}{\mu^2} \qquad (15)$$

$$4\mathbb{E}_1[\|\mathbf{y}^2 - \mathbf{y}^\star\|^2] \leq \frac{4m(\bar{a} + \bar{d})^2}{\mu^2} \qquad (16)$$

Re-arranging the terms,

$$(t+1)^2 \mathbb{E}_t[\|\mathbf{y}^{t+1} - \mathbf{y}^\star\|^2] - t^2 \|\mathbf{y}^t - \mathbf{y}^\star\|^2 \leq \frac{4m(\bar{a} + \bar{d})^2}{\mu^2}.$$

Taking expectation over all the randomness and telescoping from $t = 2$ to T, with (16) added, gives

$$\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|^2] \leq \frac{4m(\bar{a} + \bar{d})^2 T}{\mu^2 (T+1)^2} \leq \frac{4m(\bar{a} + \bar{d})^2}{\mu^2 T}$$

and this completes the proof.

Proof of Lemma A.6 Using Theorem 3 of (Xu et al., 2017), we have, with probability at least $1 - \delta$, that the number of iterations to obtain $f(\mathbf{y}^T) - f(\mathbf{y}^\star) \leq \varepsilon$ is bounded by

$$K = C \cdot \tfrac{2}{\varepsilon} \log(\tfrac{1}{\delta}) \cdot \left[\log_2 \left(\tfrac{2}{\varepsilon}[f(\mathbf{y}^1) - f(\mathbf{y}^\star)] \right) + 1 \right]$$

for some $C > 0$. Given $T > 0$, we take $\varepsilon = \frac{M \log^2 T}{T}$, $\delta = 1/T$ and define

$$M := C \cdot [2 \log_2(2\sqrt{m}(\bar{a} + \bar{d})\tfrac{\bar{c}}{\underline{d}}) + 4].$$

Then we successively deduce that

$$\begin{aligned} K &= C(\log_2\big(\tfrac{2[f(\mathbf{y}^1)-f(\mathbf{y}^\star)]}{\varepsilon}\big)+1)\log(\tfrac{1}{\delta})\tfrac{2}{\varepsilon} \\ &= C[\log_2(2[f(\mathbf{y}^1)-f(\mathbf{y}^\star)])+\log_2(\tfrac{T}{M\log^2 T})]\log T(\tfrac{2T}{M\log^2 T})+C\cdot\log T(\tfrac{2T}{M\log^2 T}) \\ &\leq C[\log_2(2\sqrt{m}(\bar{a}+\bar{d})\tfrac{\bar{c}}{\underline{d}})+\log_2 T]\tfrac{2T}{M\log T}+\tfrac{2CT}{M\log T} \\ &\leq 2C\log_2(2\sqrt{m}(\bar{a}+\bar{d})\tfrac{\bar{c}}{\underline{d}})\tfrac{T}{M}+\tfrac{4CT}{M} \\ &= C[2\log_2(2\sqrt{m}(\bar{a}+\bar{d})\tfrac{\bar{c}}{\underline{d}})+4]\tfrac{T}{M}\leq T, \end{aligned} \tag{17}$$

where (17) uses the relation

$$\begin{aligned} f(\mathbf{y}^1)-f(\mathbf{y}^\star) &\leq \sqrt{m}(\bar{a}+\bar{d})\|\mathbf{y}^\star-\mathbf{y}^1\| \\ &= \sqrt{m}(\bar{a}+\bar{d})\|\mathbf{y}^\star\| \\ &\leq \sqrt{m}(\bar{a}+\bar{d})\tfrac{\bar{c}}{\underline{d}} \end{aligned}$$

Therefore, we have, with probability at least $1-\frac{1}{T}$ that

$$\|\mathbf{y}^T-\mathbf{y}^\star\|\leq\sqrt{\tfrac{2}{\mu}[f(\mathbf{y}^T)-f(\mathbf{y}^\star)]}\leq\sqrt{\tfrac{2M\log^2 T}{\mu T}}=\sqrt{\tfrac{2M}{\mu}}\tfrac{\log T}{\sqrt{T}}$$

and this completes the proof.

Proof of Lemma A.7 By Theorem 3 of (Chen et al., 2018), there exists some $C,\lambda>0$ that

$$\mathbb{E}[f(\mathbf{y}^T)-f(\mathbf{y}^\star)]\leq\varepsilon$$

after $C\lambda/\varepsilon$ iterations. Taking $\varepsilon=C\lambda/T$, we deduce that

$$\frac{\mu}{2}\mathbb{E}[\|\mathbf{y}^T-\mathbf{y}^\star\|^2]\leq\mathbb{E}[f(\mathbf{y}^T)-f(\mathbf{y}^\star)]\leq\frac{C\lambda}{T}$$

and $\mathbb{E}[\|\mathbf{y}^T-\mathbf{y}^\star\|]\leq\sqrt{\frac{2C\lambda}{\mu T}}$. This completes the proof.

B. Algorithm Design and Analysis

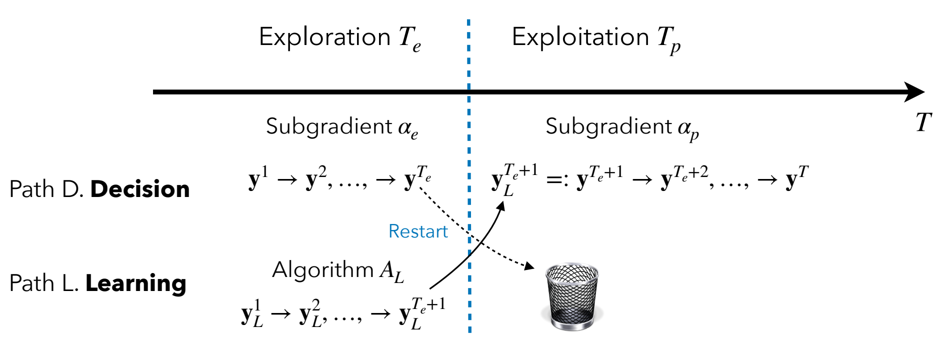

Figure 5: Two-path and two-phase decision for online linear programming

In this section, we present our results in a general framework. Our algorithm design contains two phases: exploration (**E**) and exploitation (**P**). See **Figure 5** for an illustration.

Exploration. This phase starts from time horizon $t = 1$ to $t = T_e$. Two (first-order) methods simultaneously maintain and update two dual sequences, which we call **Path D**ecision and **Path L**earning. **Path D** maintains sequence $\{\mathbf{y}^t\}$ that is used for *decision*, and we restrict **Path D** to use subgradient update with stepsize α_e

$$x^t = \mathbb{I}\{c_t > \langle \mathbf{a}_t, \mathbf{y}^t\rangle\}$$
$$\mathbf{y}^{t+1} = [\mathbf{y}^t - \alpha_e(\mathbf{d} - \mathbf{a}_t x^t)]_+.$$

From our discussion in **Section A**, subgradient has convergence rate $\theta = 1/4$. In contrast, **Path L** should be equipped with a learning algorithm with $\theta \geq 1/4$, which aims to output the best possible $\mathbf{y}_L^{T_e+1}$ when the exploration phase ends. Our analysis allows $T_e = 0$ or $T_e = T$ to encompass the analysis of one-phase algorithms.

Exploitation. At time horizon $T_e + 1$, the online algorithm enters "exploitation" phase, where we no longer maintain **Path L** and restarts **Path D** with $\mathbf{y}_L^{T_e+1}$. In the exploitation phase, **Path D** still uses subgradient method but with a different stepsize α_p till the end of horizon T.

Table 5: Notations of algorithm design

Notation	Meaning
$\mathcal{A}_L$	**Path L** learning algorithm
θ	Convergence rate of $\mathcal{A}_L$
α_e	Stepsize of subgradient in exploration
α_p	Stepsize of subgradient in exploitation
$\mathbf{y}^t$	t-th dual iteration for *decision algorithm*
$\mathbf{y}_L^t$	t-th dual iteration in exploration for *learning algorithm* $\mathcal{A}_L$
T	Total decision horizon T
T_e	Time of transition from exploration to exploitation
T_p	$T - T_e$

Our analysis is done in two steps. In B.1, we focus on the behavior of dual iterations in **Path D**. In B.2, we analyze the regret and constraint violation bounds for our two-phase algorithm. We show that given θ, $(T_e, \alpha_e, \alpha_p)$ can be optimally determined to improve algorithm performance.

B.1. Dual Convergence

In this section, we analyze the behavior of $\{\mathbf{y}^t\}$ along **Path D**, where iterate by $\mathbf{y}^{t+1} = [\mathbf{y}^t - \alpha_t(\mathbf{d} - \mathbf{a}_t x^t)]_+$ and

1. before restart, we take stepsize α_p
2. after restart from $\mathbf{y}^{T_e+1}$, we take stepsize α_e

where $\mathbf{y}^{T_e+1} \in \Xi_0$. The following lemma characterizes almost sure boundedness of the dual sequence $\{\mathbf{y}^t\}$.

Lemma B.1. *Assume that A1 to A6 hold. Let $\{\mathbf{y}^t\}$ be generated by **Path D**, then*

$$\|\mathbf{y}^t\| \leq \frac{\bar{c}}{\underline{d}} + \frac{m(\bar{a}+\bar{d})^2\alpha_e}{2\underline{d}} + \alpha_e\sqrt{m}(\bar{a}+\bar{d}), \text{ for all } t \leq T_e,$$
$$\|\mathbf{y}^t\| \leq \frac{\bar{c}}{\underline{d}} + \frac{m(\bar{a}+\bar{d})^2\alpha_p}{2\underline{d}} + \alpha_p\sqrt{m}(\bar{a}+\bar{d}), \text{ for all } t \geq T_e + 1$$

almost surely. In other words, $\mathbf{y}^t \in \Xi_1$ for all $\alpha_p \leq \frac{2\underline{d}}{3m(\bar{a}+\bar{d})^2}$ almost surely.

The next lemma will be used to obtain a stronger dual convergence result for $\{\mathbf{y}^t\}, t \geq T_e + 1$.

Lemma B.2. *Assume that A1 to A6 hold, then*

$$\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|^2 | \mathbf{y}^{T_e+1}] \leq \frac{\|\mathbf{y}_e^{T_e+1} - \mathbf{y}^\star\|^2}{\mu\alpha_p T_p} + \frac{m(\bar{a}+\bar{d})^2}{\mu}\alpha_p$$

Proof of Lemma B.1. The first relation follows immediately from Lemma 5 of (Gao et al., 2023) using $\mathbf{y}^1 = \mathbf{0}$, while the second relation uses the fact $\mathbf{y}^{T_e+1} = \mathbf{y}_L^{T_e+1} \in \Xi_0$ and that $\|\mathbf{y}^{T_e+1}\| \leq \frac{\bar{c}}{\underline{d}}$. To see $\mathbf{y}^t \in \Xi_1$, we successively deduce that, for $\alpha_p \leq \frac{2\underline{d}}{3m(\bar{a}+\bar{d})^2}$, that

$$\frac{m(\bar{a}+\bar{d})^2\alpha_p}{2\underline{d}} + \alpha_p\sqrt{m}(\bar{a}+\bar{d}) = \frac{1}{3} + \frac{2\underline{d}}{3\sqrt{m}(\bar{a}+\bar{d})} \leq \frac{1}{3} + \frac{2(\bar{a}+\bar{d})}{3\sqrt{m}(\bar{a}+\bar{d})} \leq 1$$

and this completes the proof.

Proof of Lemma B.2. The result is a direct consequence of **Lemma A.3**, where we consider $\mathbf{y}^{T_e+1}$ as the starting point.

B.2. Performance Analysis of the Algorithm

In this section, we conduct the performance analysis of our algorithm. With the auxiliary results in hand, the proof focuses on finding a proper trade-off between T_e, α_p, α_d based on θ. To simplify notation, we define

$$\mathsf{R} := \frac{\bar{c}}{\underline{d}} + \left[\frac{m(\bar{a}+\bar{d})^2}{2\underline{d}} + \sqrt{m}(\bar{a}+\bar{d})\right] \cdot \max\{\alpha_e, \alpha_p\} \tag{18}$$

and from **Lemma B.1** we know that $\|\mathbf{y}^t\| \leq \mathsf{R}$ almost surely. Though we have not formally chosen α_e, α_p, they will be set such that $\mathsf{R} = \frac{\bar{c}}{\underline{d}} + o(1) = \mathcal{O}(1)$.

The following lemma analyzes the regret of the whole algorithm.

Lemma B.3. *Assuming A1 to A6, we have*

$$\mathbb{E}[r(\hat{\mathbf{x}})] \leq \frac{m(\bar{a}+\bar{d})^2\alpha_e}{2}T_e + \frac{m(\bar{a}+\bar{d})^2\alpha_p}{2}T_p + \frac{\mathsf{R}}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|].$$

The next lemma analyzes the constraint violation of the whole algorithm.

Lemma B.4. *Assuming A1 to A6, we have*

$$\mathbb{E}[v(\hat{\mathbf{x}})] \leq \frac{\mathsf{R}}{\alpha_e} + \frac{1}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|].$$

Putting things together, we take a trade-off between T_e, α_p, α_d and get the following result.

Theorem B.1. *Under the same assumptions as **Lemma B.3** as well as A7, let $\theta' = \min\{\theta, 1/2\}$. Then for all $T \geq \frac{9m^2(\bar{a}+\bar{d})^4}{4\underline{d}^2}$. If we choose*

$$T_e^\star = T^{\frac{1}{\theta'+1}}, \quad \alpha_e^\star = T^{-\frac{1}{2(\theta'+1)}}, \quad \alpha_p^\star = T^{-\frac{2\theta'+1}{2(\theta'+1)}},$$

Then

$$\mathbb{E}[r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})] \leq \mathcal{O}(T^{\frac{1}{2(\theta'+1)}}).$$

Proof of Lemma B.3. We first deduce that

$$\begin{aligned}
\mathbb{E}[r(\hat{\mathbf{x}})] &= \mathbb{E}[\langle \mathbf{c}, \mathbf{x}^\star\rangle - \langle \mathbf{c}, \hat{\mathbf{x}}\rangle] \\
&= \mathbb{E}[f_T(\mathbf{y}_T^\star) - \langle \mathbf{c}, \hat{\mathbf{x}}\rangle] && (19)\\
&\leq \mathbb{E}[f_T(\mathbf{y}^\star) - \langle \mathbf{c}, \hat{\mathbf{x}}\rangle] && (20)\\
&= Tf(\mathbf{y}^\star) - \mathbb{E}[\langle \mathbf{c}, \hat{\mathbf{x}}\rangle] \\
&\leq \mathbb{E}\left[\sum_{t=1}^T f(\mathbf{y}^t) - \langle \mathbf{c}, \hat{\mathbf{x}}\rangle\right] \\
&= \sum_{t=1}^T \mathbb{E}[\langle \mathbf{d}, \mathbf{y}^t\rangle + [c_t - \langle \mathbf{a}_t, \mathbf{y}^t\rangle]_+ - c_t x^t] && (21)\\
&= \sum_{t=1}^T \mathbb{E}[\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle],
\end{aligned}$$

where (19) uses strong duality of LP; (20) uses the fact $\mathbf{y}^\star$ is a feasible solution and that $\mathbf{y}_T^\star$ is the optimal solution to the sample LP; (21) uses the definition of $f(\mathbf{y})$ and that $(c_t, \mathbf{a}_t)$ are i.i.d. generated.

Recall that given $\alpha \in \{\alpha_p, \alpha_d\}$,

$$\begin{aligned}
\|\mathbf{y}^{t+1}\|^2 - \|\mathbf{y}^t\|^2 &= \|[\mathbf{y}^t - \alpha(\mathbf{d} - \mathbf{a}_t x^t)]_+\|^2 - \|\mathbf{y}^t\|^2 \\
&\leq \|\mathbf{y}^t - \alpha(\mathbf{d} - \mathbf{a}_t x^t)\|^2 - \|\mathbf{y}^t\|^2 \qquad (22)\\
&= -2\alpha\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle + \alpha^2\|\mathbf{d} - \mathbf{a}_t x^t\|^2 \\
&\leq -2\alpha\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle + m(\bar{a} + \bar{d})^2\alpha^2, \qquad (23)
\end{aligned}$$

where (22) uses $\|[\mathbf{x}]_+\| \leq \|\mathbf{x}\|$ and (23) uses **A2**, **A3**. A simple re-arrangement gives

$$\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle \leq \frac{m(\bar{a} + \bar{d})^2\alpha}{2} + \frac{\|\mathbf{y}^t\|^2 - \|\mathbf{y}^{t+1}\|^2}{2\alpha} \qquad (24)$$

Now we decompose regret according to two phases

$$\begin{aligned}
\mathbb{E}[r(\hat{\mathbf{x}})] &= \sum_{t=1}^{T}\mathbb{E}[\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle] \\
&= \sum_{t=1}^{T_e}\mathbb{E}[\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle] + \sum_{t=T_e+1}^{T}\mathbb{E}[\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle] \\
&=: r_e + r_p
\end{aligned}$$

and we bound two parts of regret respectively. For r_e, we have

$$\begin{aligned}
r_e &= \sum_{t=1}^{T_e}\mathbb{E}[\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle] \\
&\leq \frac{m(\bar{a} + \bar{d})^2\alpha_e}{2}T_e + \sum_{t=1}^{T_e}\frac{\|\mathbf{y}^t\|^2 - \|\mathbf{y}^{t+1}\|^2}{2\alpha_e} \qquad (25)\\
&\leq \frac{m(\bar{a} + \bar{d})^2\alpha_e}{2}T_e + \frac{\|\mathbf{y}^1\|^2 - \mathbb{E}[\|\mathbf{y}^{T_e+1}\|^2]}{2\alpha_e} \qquad (26)\\
&\leq \frac{m(\bar{a} + \bar{d})^2\alpha_e}{2}T_e,
\end{aligned}$$

where (25) uses relation (24) and in (26) we use $\mathbf{y}^1 = \mathbf{0}$. For r_p, we have

$$\begin{aligned}
r_p &= \sum_{t=T_e+1}^{T}\mathbb{E}[\langle \mathbf{d} - \mathbf{a}_t x^t, \mathbf{y}^t\rangle] \\
&\leq \frac{m(\bar{a} + \bar{d})^2\alpha_p}{2}T_p + \sum_{t=T_e+1}^{T}\frac{\mathbb{E}[\|\mathbf{y}^t\|^2] - \mathbb{E}[\|\mathbf{y}^{t+1}\|^2]}{2\alpha_p} \qquad (27)\\
&= \frac{m(\bar{a} + \bar{d})^2\alpha_p}{2}T_p + \frac{\mathbb{E}[\|\mathbf{y}^{T_e+1}\|^2] - \mathbb{E}[\|\mathbf{y}^{T+1}\|^2]}{2\alpha_p} \\
&= \frac{m(\bar{a} + \bar{d})^2\alpha_p}{2}T_p + \frac{\mathbb{E}[\langle \mathbf{y}^{T_e+1} + \mathbf{y}^{T+1}, \mathbf{y}^{T_e+1} - \mathbf{y}^{T+1}\rangle]}{2\alpha_p} \qquad (28)\\
&\leq \frac{m(\bar{a} + \bar{d})^2\alpha_p}{2}T_p + \frac{\mathsf{R}}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^{T+1}\|] \qquad (29)\\
&\leq \frac{m(\bar{a} + \bar{d})^2\alpha_p}{2}T_p + \frac{\mathsf{R}}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|], \qquad (30)
\end{aligned}$$

where (27) again uses relation (24); (29) uses the Cauchy's inequality

$$\langle \mathbf{y}^{T_e+1} + \mathbf{y}^{T_e+1}, \mathbf{y}^{T_e+1} - \mathbf{y}^{T+1} \rangle \leq \|\mathbf{y}^{T_e+1} + \mathbf{y}^{T+1}\| \cdot \|\mathbf{y}^{T_e+1} - \mathbf{y}^{T+1}\|$$

and almost sure boundedness of iterations derived from **Lemma B.1**:

$$\|\mathbf{y}^{T_e+1} + \mathbf{y}^{T+1}\| \leq \|\mathbf{y}^{T+1}\| + \|\mathbf{y}^{T_e+1}\| \leq 2\mathsf{R}.$$

Finally (30) is obtained from the triangle inequality

$$\|\mathbf{y}^{T_e+1} - \mathbf{y}^{T+1}\| = \|\mathbf{y}^{T_e+1} - \mathbf{y}^\star + \mathbf{y}^\star - \mathbf{y}^{T+1}\| \leq \|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|.$$

Summing up two bounds on r_e and r_p completes the proof.

Proof of Lemma B.4. For constraint violation, recall that

$$\mathbb{E}[v(\hat{\mathbf{x}})] = \mathbb{E}[\|[\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}]_+\|] = \mathbb{E}\left[\left\|\left[\sum_{t=1}^{T}(\mathbf{a}_t x^t - \mathbf{d})\right]_+\right\|\right]$$

and that

$$\mathbf{y}^{t+1} = [\mathbf{y}^{t+1} - \alpha(\mathbf{d} - \mathbf{a}_t x^t)]_+ \geq \mathbf{y}^t - \alpha(\mathbf{d} - \mathbf{a}_t x^t).$$

A re-arrangement gives, for $\alpha \in \{\alpha_e, \alpha_p\}$, that

$$\mathbf{a}_t x^t \leq \mathbf{d} + \frac{1}{\alpha}(\mathbf{y}^{t+1} - \mathbf{y}^t). \tag{31}$$

Now we decompose $\sum_{t=1}^{T}(\mathbf{a}_t x^t - \mathbf{d})$ by

$$\begin{aligned}
\sum_{t=1}^{T}(\mathbf{a}_t x^t - \mathbf{d}) &= \sum_{t=1}^{T_e}(\mathbf{a}_t x^t - \mathbf{d}) + \sum_{t=T_e+1}^{T}(\mathbf{a}_t x^t - \mathbf{d}) \\
&\leq \frac{1}{\alpha_e}\sum_{t=1}^{T_e}(\mathbf{y}^{t+1} - \mathbf{y}^t) + \frac{1}{\alpha_p}\sum_{t=T_e+1}^{T}(\mathbf{y}^{t+1} - \mathbf{y}^t) \qquad (32)\\
&= \frac{1}{\alpha_e}(\mathbf{y}^{T_e+1} - \mathbf{y}^1) + \frac{1}{\alpha_p}(\mathbf{y}^{T+1} - \mathbf{y}^{T_e+1}) \\
&= \frac{1}{\alpha_e}\mathbf{y}^{T_e+1} + \frac{1}{\alpha_p}(\mathbf{y}^{T+1} - \mathbf{y}^{T_e+1}),
\end{aligned}$$

where (32) uses (31). Now we deduce that

$$\begin{aligned}
\mathbb{E}[\|[\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}]_+\|] &\leq \mathbb{E}\left[\left\|\frac{1}{\alpha_e}\mathbf{y}^{T_e+1} + \frac{1}{\alpha_p}(\mathbf{y}^{T+1} - \mathbf{y}^{T_e+1})\right\|\right] \\
&\leq \frac{1}{\alpha_e}\mathbb{E}[\|\mathbf{y}^{T_e+1}\|] + \frac{1}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^{T_e+1}\|] \\
&\leq \frac{\mathsf{R}}{\alpha_e} + \frac{1}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|], \qquad (33)
\end{aligned}$$

where (33) again uses triangle inequality and this completes the proof.

Proof of Theorem B.1. For all $T \geq \frac{9m^2(\bar{a}+\bar{d})^4}{4\underline{d}^2}$, we know

$$\alpha_p^\star = T^{-\frac{2\theta'+1}{2(\theta'+1)}} \leq T^{-1/2} \leq \frac{2\underline{d}}{3m(\bar{a}+\bar{d})^2}$$

since $\frac{2\theta'+1}{2(\theta'+1)} \leq 0.5$. Therefore $\alpha_p \leq \frac{2\underline{d}}{3m(\bar{a}+\bar{d})^2}$ and by **Lemma B.1** we know, for all t, that $\mathbf{y}^t \in \Xi_1$ almost surely.

Next we consider the sum of regret and violation by summing up **Lemma B.3** and **Lemma B.4**.

$$\begin{aligned}\mathbb{E}[r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})] \leq\ & \frac{m(\bar{a}+\bar{d})^2\alpha_e}{2}T_e + \frac{m(\bar{a}+\bar{d})^2\alpha_p}{2}T_p + \frac{\mathsf{R}}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|] \\ & + \frac{\mathsf{R}}{\alpha_e} + \frac{1}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|] \\ =\ & V_e + V_p,\end{aligned}$$

where we define

$$\begin{aligned}V_e &:= \frac{m(\bar{a}+\bar{d})^2\alpha_e}{2}T_e + \frac{\mathsf{R}}{\alpha_e} \\ V_p &:= \frac{m(\bar{a}+\bar{d})^2\alpha_p}{2}T_p + \frac{\mathsf{R}+1}{\alpha_p}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\| + \|\mathbf{y}^{T+1} - \mathbf{y}^\star\|].\end{aligned}$$

Next we invoke **A7**, where we have, with probability at least $1 - 1/T$, that

$$\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\|] \leq \frac{c_1 + c_2 \log T_e}{T_e^\theta} \leq \frac{c_1 + c_2 \log T}{T_e^\theta}$$

Denote the event $\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\|] \leq \frac{c_1+c_2\log T}{T_e^\theta}$ to be E, we have

$$\begin{aligned}\mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\|] &= \mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\||E] \cdot \mathbb{P}\{E\} + \mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\||\bar{E}] \cdot \mathbb{P}\{\bar{E}\} \\ &\leq \mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\||E] + \mathbb{E}[\|\mathbf{y}^{T_e+1} - \mathbf{y}^\star\||\bar{E}] \cdot \mathbb{P}\{\bar{E}\} \\ &\leq \frac{c_1 + c_2\log T}{T_e^\theta} + \frac{2\mathsf{R}}{T} \qquad (34) \\ &\cong \frac{\log T}{T_e^\theta} + \frac{2\mathsf{R}}{T},\end{aligned}$$

where (34) uses **Lemma B.1** and **Lemma 2.1**. Similarly we can deduce that

$$\begin{aligned}\mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\|] &= \mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\||E] \cdot \mathbb{P}\{E\} + \mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\||\bar{E}] \cdot \mathbb{P}\{\bar{E}\} \\ &\leq \mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\||E] + \mathbb{E}[\|\mathbf{y}^{T+1} - \mathbf{y}^\star\||\bar{E}] \cdot \mathbb{P}\{\bar{E}\} \\ &\leq \tfrac{c_1+c_2\log T}{\sqrt{\mu}\sqrt{T_p}T_e^\theta}\alpha_p^{-1/2} + \sqrt{\tfrac{m(\bar{a}+\bar{d})^2}{\mu}}\alpha_p^{1/2} + \tfrac{2\mathsf{R}}{T} \qquad (35) \\ &\cong \tfrac{\log T}{\sqrt{\mu}\sqrt{T_p}T_e^\theta}\alpha_p^{-1/2} + \sqrt{\tfrac{m}{\mu}}\alpha_p^{1/2} + \tfrac{2\mathsf{R}}{T}\end{aligned}$$

where in (35) we invoke **Lemma B.2** and the inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. Then we have

$$V_p \lesssim \frac{m(\bar{a}+\bar{d})^2\alpha_p}{2}T_p + (\log T)T_e^{-\theta}\alpha_p^{-1} + \frac{\log T}{\sqrt{\mu}}T_p^{-1/2}T_e^{-\theta}\alpha_p^{-3/2} + \sqrt{\frac{m}{\mu}}\alpha_p^{-1/2} + \frac{4\mathsf{R}}{\alpha_p T}.$$

And notice that α_e only appears in

$$V_e = \frac{m(\bar{a}+\bar{d})^2\alpha_e}{2}T_e + \frac{\mathsf{R}}{\alpha_e}$$

and $\alpha_e = \sqrt{\frac{2\mathsf{R}}{m(\bar{a}+\bar{d})^2 T_e}} = \mathcal{O}(\frac{1}{\sqrt{T_e}})$ minimizes V_p, giving $V_p = \mathcal{O}(\sqrt{T_e})$. Therefore, we fix $\alpha_e = T_e^{-1/2}$ and put all the things together to choose T_p, α_e, α_p. Ignoring all the constants and $\log T$ terms, we have

$$V_p + V_e \lesssim T_e^{1/2} + T_p\alpha_p + T_e^{-\theta}\alpha_p^{-1} + T_p^{-1/2}T_e^{-\theta}\alpha_p^{-3/2} + \alpha_p^{-1/2} + \alpha_p^{-1}T^{-1}$$

We now do case analysis.

Case 1. If $\theta = 0$, then learning algorithm does not converge. $T_p\alpha_p + \alpha_p^{-1} \geq 2\sqrt{T_p}$ and $\sqrt{T_e} + \sqrt{T_p} \geq \sqrt{T_e + T_p} = \sqrt{T}$. In this case choosing $T_e = T, T_p = 0, \alpha_e = \sqrt{T}$ gives

$$V_p + V_e \leq \mathcal{O}(\sqrt{T}).$$

Case 2. If $0 < \theta < 1/2$, then learning algorithm converges. Without loss of generality, we take $T_e = T^\beta, \beta \in (0, 1)$ and successively deduce that

$$\begin{aligned}
&V_p + V_e \\
\lesssim\ & T_e^{1/2} + T_p\alpha_p + T_e^{-\theta}\alpha_p^{-1} + T_p^{-1/2}T_e^{-\theta}\alpha_p^{-3/2} + \alpha_p^{-1/2} + \alpha_p^{-1}T^{-1} \\
\cong\ & T^{\beta/2} + T(1 - T_eT^{-1})\alpha_p + T^{-\theta\beta}\alpha_p^{-1} + T^{-1/2}(1 - T_eT^{-1})^{-1/2}T^{-\theta\beta}\alpha_p^{-3/2} + \alpha_p^{-1/2} + \alpha_p^{-1}T^{-1} \\
\cong\ & T^{\beta/2} + \alpha_pT + T^{-\theta\beta}\alpha_p^{-1} + \alpha_p^{-3/2}T^{-1/2-\beta\theta} + \alpha_p^{-1/2} + \alpha_p^{-1}T^{-1}
\end{aligned}$$

Now assume $\alpha_p = T^{-\lambda}$, we have

$$V_p + V_e \lesssim T^{\beta/2} + T^{1-\lambda} + T^{\lambda-\theta\beta} + T^{\frac{3}{2}\lambda - 1/2 - \beta\theta} + T^{\frac{\lambda}{2}} + T^{\lambda - 1}$$

and this reduces to an optimization problem

$$\begin{aligned}
\min_{\lambda,\beta} \quad & \max\{\tfrac{\beta}{2}, 1-\lambda, \lambda - \theta\beta, \tfrac{3}{2}\lambda - \tfrac{1}{2} - \beta\theta, \tfrac{\lambda}{2}, \lambda - 1\} \\
\text{subject to} \quad & 0 \leq \beta \leq 1 \\
& \lambda \geq 0
\end{aligned}$$

and solving the problem gives the following parameter setting

$$\begin{aligned}
\beta^\star(\theta) &= \tfrac{1}{\theta+1} \\
\lambda^\star(\theta) &= 1 - \tfrac{1}{2(\theta+1)} \\
\alpha_e^\star(\theta) &= T^{-\frac{1}{2(\theta+1)}} \\
\alpha_p^\star(\theta) &= T^{-\frac{2\theta+1}{2(\theta+1)}} \\
T_e^\star(\theta) &= T^{\frac{1}{\theta+1}} \\
V_p + V_e &\lesssim \mathcal{O}(T^{\frac{1}{2(\theta+1)}})
\end{aligned}$$

Case 3. If $\theta > 1/2$, we have $\alpha_e^\star = T^{-1/3}, \alpha_p^\star = T^{-2/3}, T_e^\star = T^{2/3}$ and $V_p + V_e = \mathcal{O}(T^{1/3})$.

Putting all the results together, we have

$$\begin{aligned}
\theta' &= \min\{\theta, 1/2\} \\
T_e^\star(\theta) &= T^{\frac{1}{\theta'+1}} \\
\alpha_e^\star(\theta) &= T^{-\frac{1}{2(\theta'+1)}} \\
\alpha_p^\star(\theta) &= T^{-\frac{2\theta'+1}{2(\theta'+1)}}
\end{aligned}$$

and $V_p + V_e = \mathcal{O}(T^{\frac{1}{2(\theta'+1)}})$. Adding back $\log T$ terms, this completes the proof.

C. Proof of Main Results in Section 3

C.1. Proof of Proposition 3.1

Invoke **Theorem B.1**, plug in $\theta' = \min\{\theta, 1/2\}$ and this completes the proof.

C.2. Proof of Proposition 3.2

First, we establish the update rule formula for $\mathbb{E}[y^{t+1}]$ in terms of $\mathbb{E}[y^t]$. Specifically, we have

$$\mathbb{E}[y^{t+1}] = \mathbb{E}[[y^t - \tfrac{1}{t}(\tfrac{1}{2} - \mathbb{I}\{c_t > y^t\})]_+] \tag{36}$$
$$\geq \mathbb{E}[y^t - \tfrac{1}{t}(\tfrac{1}{2} - \mathbb{I}\{c_t > y^t\})] \tag{37}$$
$$\geq \mathbb{E}[y^t - \tfrac{1}{t}y^t + \tfrac{1}{2t}] \tag{38}$$

where (36) is obtained by the update rule of subgradient, (37) uses Jensen's inequality, and (38) is obtained by the fact that c_t is independent of y^t and it is drawn uniformly from $[0, 1]$. Indeed, we have

$$\mathbb{E}[\mathbb{I}\{c_t > y^t\}] = \mathbb{E}[\mathbb{E}[\mathbb{I}\{c_t > y^t\}|y^t]] = \mathbb{E}[\textstyle\int_0^1 \mathbb{I}\{c > y^t\}\mathrm{d}c|y^t] = \mathbb{E}[1 - y^t].$$

Subtracting $t/2$ from both sides and multiplying both sides the the inequality by t, we have

$$t(\mathbb{E}[y^{t+1}] - \tfrac{1}{2}) \geq (t-1)(\mathbb{E}[y^t] - \tfrac{1}{2}), \quad \text{for all } t = 1, \ldots, T.$$

Next we condition on the value of y^{t_0} and

$$t(\mathbb{E}[y^{t+1}|y^{t_0}] - \tfrac{1}{2}) \geq (t_0 - 1)(y^{t_0} - \tfrac{1}{2}). \tag{39}$$

Thus, given $y^{t_0} > y^\star + \frac{1}{\sqrt{T}} = \frac{1}{2} + \frac{1}{\sqrt{T}}$ for some t_0, we have

$$t(\mathbb{E}[y^{t+1}|y^{t_0}] - \tfrac{1}{2}) \geq (t_0 - 1)(y^{t_0} - \tfrac{1}{2}) \geq \tfrac{t_0 - 1}{\sqrt{T}}, \tag{40}$$

As a result, when $t_0 \geq \frac{T}{10} + 1$, (40) implies

$$\mathbb{E}[y^{t+1}|y^{t_0}] \geq \tfrac{1}{2} + \tfrac{t_0 - 1}{t \times \sqrt{T}} \geq \tfrac{1}{2} + \tfrac{1}{10\sqrt{T}},$$

since we assume $t_0 \geq T/10 + 1$. This completes the proof.

C.3. Proof of Proposition 3.3

Based on (Rakhlin et al., 2011), there exists some universal constant $c > 0$ such that with probability no less than $1 - 1/T^4$, $|y^t - y^\star| \leq c \log T/\sqrt{T}$ for all $t \geq t_0$, where $y^\star = \frac{1}{2}$ and $t_0 = \mathcal{O}(\log T)$. Thus, without loss of generality, we assume

$$y^t \in [\tfrac{1}{4}, \tfrac{3}{4}], \text{ and } y^{t+1} = y^t - \tfrac{1}{t}(\tfrac{1}{2} - \mathbb{I}\{c_t > y^t\}) \tag{41}$$

for all $t \geq t_0$ by setting a new random initialization $y^{t_0} \in [1/4, 3/4]$ and ignoring the all decision steps before the t_0 step. In the following, we show that **Algorithm 1** with SGD and known μ must have $\Omega(T^{1/2})$ regret or constraint violation for any initialization y^{t_0}. We first calculate $\mathbb{E}[y^t - \frac{1}{2}]$ and $\mathbb{E}[(y^t - \frac{1}{2})^2]$ similar to the proof of **Proposition 3.2**. Specifically, for $\mathbb{E}[y^t - 1/2]$, we have

$$\mathbb{E}[y^{t+1}|y^t] = \big(1 - \tfrac{1}{t}\big)y^t + \frac{1}{2t},$$

which implies

$$\mathbb{E}[y^{t+1} - \tfrac{1}{2}|y^{t_0}] = \frac{t_0 - 1}{t}(y^1 - \tfrac{1}{2}) + \tfrac{1}{2}, \tag{42}$$

Also, similarly, for $\mathbb{E}[(y^t - 1/2)^2]$ we have under assumption (41)

$$\begin{aligned}
\mathbb{E}[(y^{t+1} - \tfrac{1}{2})^2|y^t] &= \mathbb{E}[(y^t - \tfrac{1}{t}(\tfrac{1}{2} - \mathbb{I}\{c_t > y^t\}) - \tfrac{1}{2})^2|y^t] \\
&= (1 - \tfrac{1}{t})^2(y^t - \tfrac{1}{2})^2 + \tfrac{1}{4t^2} - \tfrac{1}{t^2}(y^t - \tfrac{1}{2})^2 \\
&\geq (1 - \tfrac{1}{t})^2(y^t - \tfrac{1}{2})^2 + \tfrac{1}{4t^2} - \tfrac{c}{t^3},
\end{aligned}$$

which implies

$$\mathbb{E}[(y^{t+1}-\tfrac{1}{2})^2|y^t] \geq \frac{(t_0-1)^2}{t^2}(y^{t_0}-\tfrac{1}{2})^2 + \frac{1}{4t} - \frac{c\log t + t_0}{t^2}. \tag{43}$$

Combining (42) and (43), we then can compute

$$\begin{aligned}
\mathbb{E}\Big[\Big(\sum_{t=t_0}^{T}\mathbb{I}\{c_t>y^t\}-\tfrac{T-t_0+1}{2}\Big)^2\Big] &= \sum_{t=t_0}^{T}\mathbb{E}[(\mathbb{I}\{c_t>y^t\}-\tfrac{1}{2})^2] + 2\sum_{t_0\leq i<j\leq T}\mathbb{E}[(\mathbb{I}\{c_j>y^j\}-\tfrac{1}{2})(\mathbb{I}\{c_i>y^i\}-\tfrac{1}{2})] \\
&= \tfrac{T-t_0}{4} + 2\sum_{t_0\leq i<j\leq T}\mathbb{E}[(\mathbb{I}\{c_j>y^j\}-\tfrac{1}{2})(\mathbb{I}\{c_i>y^i\}-\tfrac{1}{2})] \\
&= \tfrac{T-t_0}{4} + 2\sum_{t_0\leq i<j\leq T}\tfrac{i-1}{j-1}\mathbb{E}[(y^i-\tfrac{1}{2})^2] - \tfrac{i-1}{4i(j-1)} \\
&\geq \tfrac{T-t_0}{4} - 2\sum_{t_0\leq i<j\leq T}\tfrac{c\log T+t_0}{(i-1)^2} \\
&= \Omega(T).
\end{aligned} \tag{44}$$

In addition, since $|y^t-\frac{1}{2}| \leq \frac{c}{\sqrt{T}}$, by Hoeffding's inequality, we have with probability no less than $1-\frac{1}{T^2}$

$$\Big|\sum_{t=t_0}^{T}\mathbb{I}\{c_t>y^t\} - \frac{T-t_0+1}{2}\Big| = \mathcal{O}(\sqrt{T}\log T).$$

Consequently, by (44), we have

$$\mathbb{E}\Big[\Big|\sum_{t=t_0}^{T}\mathbb{I}\{c_t>y^t\} - \frac{T-t_0+1}{2}\Big|\Big] = \Omega(\frac{\sqrt{T}}{\log T}). \tag{45}$$

This is the summation of constraint violation and constraint (resource) leftover, and thus, the summation of constraint violation and the regret must be no less than $\Omega(\sqrt{T}/\log T)$.

C.4. Proposition of Disability in Mistake Correction

In this section we still consider the multi-secretary problem (9), and we show that other algorithms listed in **Table 2** with small stepsize also suffer from slow updating. In particular, **Lemma C.1** reveals that all gradient-descent-based algorithms listed in with high estimation accuracy and low learning rate can change no more than $\mathcal{O}(\frac{1}{T^{1/2}})$ within $\mathcal{O}(T^\alpha)$ steps for all $\alpha\in(0,1)$.

Lemma C.1. *Denote y^t as the estimated dual price for the online secretary problem (9) at time t. Suppose (i) $|y^{t+1}-y^t|\leq\frac{1}{t}$, (ii) $\mathbb{E}[|y^t-y^\star|]\leq\frac{1}{t}$, and (iii) $y_t\leq 1$ for all $t=1,\ldots,T$. Then, with probability no less than $1-\frac{2}{T^3}$,*

$$|y^t-y^{t+k}| \leq \frac{2\sqrt{k}}{t}\cdot\log T \leq \frac{8\sqrt{k}}{T}\cdot\log T$$

for all $t\geq\frac{T}{2}$ and $0\leq k\leq T-t$.

Here, the condition assumed in **Lemma C.1** are abstracted from algorithms listed in **Table 2**: Condition (i) assumes that the learning rate is $\mathcal{O}(1/t)$; Condition (ii) assumes the estimation error of the optimal dual price can be bounded by $1/t$, which correspond to the convergence rate of optimizing strongly convex functions with stochastic gradient descent algorithms; Condition (iii) assumes that bounteousness of the estimated dual price. These conditions are satisfied by fast algorithms in **Table 2** regardless of some universal constants.

Proof of Lemma C.1. This is a direct application of Hoeffding's inequality. Specifically, based on Hoeffding's inequality, during steps t to $t+k$, there are $\frac{1}{2}k\pm 2\sqrt{k}\log T$ acceptances and rejections. As a result, we have

$$|y^{t+k}-y^t| \leq \sum_{s=t}^{t+\mathcal{O}(\sqrt{k})}\frac{1}{s} = \frac{4\sqrt{k}\log T}{t}.$$

Plugging in $t\geq\frac{T}{2}$, we complete the proof.

D. Proof of Main Results in Section 4

The main results in the paper can be directly obtained as special cases of **Theorem B.1**.

D.1. Proof of Lemma 4.1

In view of **Lemma B.3**, we complete the proof.

D.2. Proof of Lemma 4.2

In view of **Lemma B.4**, we complete the proof.

D.3. Proof of Theorem 4.1

In view of **Theorem B.1**, we complete the proof by taking $\theta = \{0, \frac{1}{4}, \frac{1}{2}\}$ respectively.

E. Additional Experiments

In this section, we provide some supplementary experiments to further demonstrate the superior performance of our proposed framework. We still evaluate the performance of the two instances of the proposed framework, i.e., **M1** and **M2**, and compare them with the no-exploration algorithm **M0**.

E.1. More Choices of m

In this subsection, we focus on the case that there are more than one type of resources. To demonstrate, we let $m = 5$. We conduct experiments on 4 distributions. The first 3 distributions are generated in the same way as those in **Section 5.1**. The last distribution is generated as follows. We sample c_t i.i.d. from uniform distribution $\mathcal{U}[0, 5]$, and sample a_{it} i.i.d. such that $a_{it} - 5$ follows the beta distribution $\mathcal{B}(1, 8)$. Each d_i is still sampled i.i.d. from uniform distribution $\mathcal{U}[1/3, 2/3]$. Note that all distributions satisfy Assumption **A1** to **A6**.

The results are presented in **Figure 6**, and in the same way as **Figure 2**, with the only difference that we plot the first coordinate $\mathbf{y}_1^t$ to demonstrate the convergence behavior of the sequence $\{\mathbf{y}^t\}_{t=1}^T$. It clearly shows that, over all 4 distributions, **M1** and **M2** has better order of regret and constraint violation than **M0**, and the dual solution sequence of **M1** and **M2** converge much faster than the one of **M0**. This demonstrates the superior and robust performance of our framework.

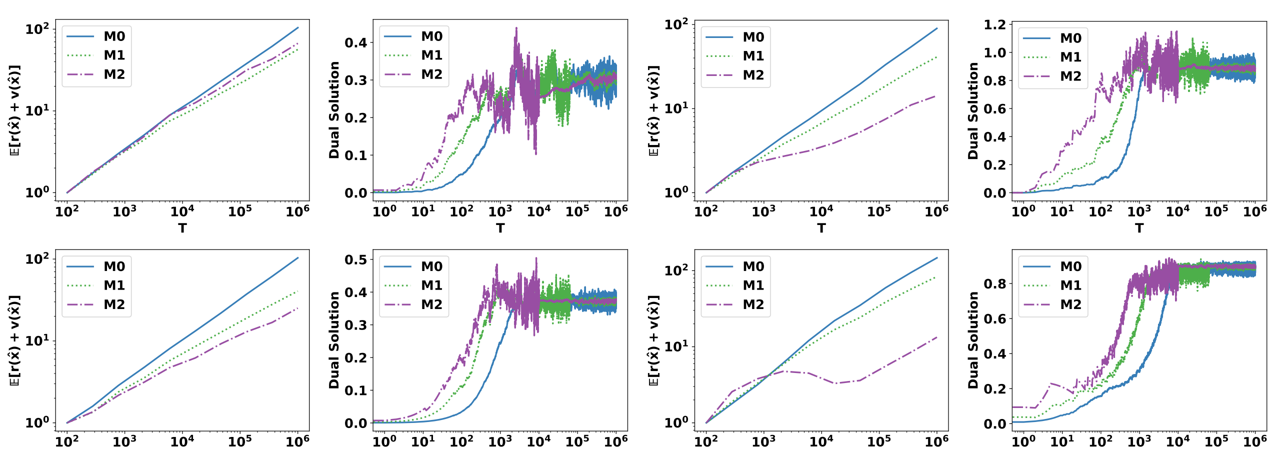

Figure 6: Growth of normalized $r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})$ and dual convergence of tested algorithms. Each pair of the left figure and the right figure is plotted based on the experiment on the same distribution.

E.2. More Experiments on Distributions which Violate Assumptions

In this subsection, we provide more experiments on distributions that violate the assumptions in **Section 2**. We generate 2 different distributions, and take $m = 1$. For the first distribution, we generate c_t i.i.d. according to the uniform distribution $\mathcal{U}[1,6]$, and a_{it} i.i.d. such that $a_{it} - 3$ satisfies exponential distribution with parameter m. For the second distribution, we consider the discrete distribution. Specifically, we randomly generate 5 different pairs of $(c_t, \mathbf{a}_t)$. Each a_{it} is sampled i.i.d. from normal distribution $\mathcal{N}(1,2)$, and each $c_t = \sum_{i=1}^m a_{it} - \varepsilon_t$ with ε_t from $\mathcal{U}[0,m]$. After obtaining these pairs of $\{(c_t, \mathbf{a}_t)\}$, at each time period t, we sample $(c_t, \mathbf{a}_t)$ from them with the same probability. **Figure 7** plots the growth of normalized $r + v$ and the convergence behavior of $\{y^t\}_{t=1}^T$. It shows that even the assumptions are violated, **M1** and **M2** still enjoy better performance than **M0**.

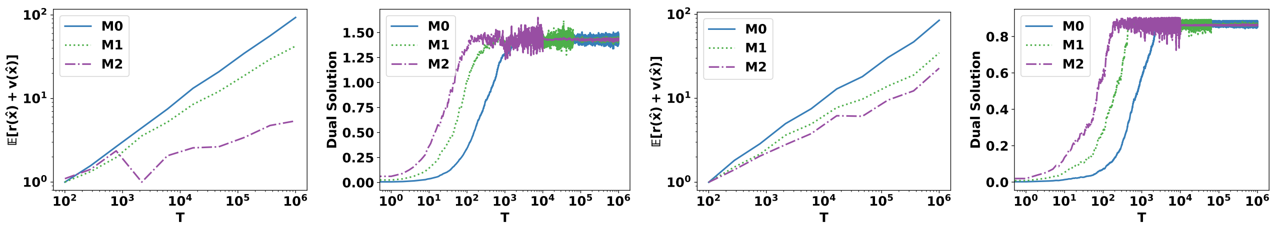

Figure 7: Growth of normalized $r(\hat{\mathbf{x}}) + v(\hat{\mathbf{x}})$ and dual convergence of tested algorithms when the assumptions are violated. Each pair of the left figure and the right figure is plotted based on the experiment on the same distribution.

References in the Appendix

Z. Chen, Y. Xu, E. Chen, and T. Yang. Sadagrad: Strongly adaptive stochastic gradient methods. In *International Conference on Machine Learning*, pages 913–921. PMLR, 2018.

S. Lacoste-Julien, M. Schmidt, and F. Bach. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012.

X. Li and Y. Ye. Online linear programming: Dual convergence, new algorithms, and regret bounds. *Operations Research*, 70(5):2948–2966, 2022.

A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

Y. Xu, Q. Lin, and T. Yang. Stochastic convex optimization: Faster local growth implies faster global convergence. In *International Conference on Machine Learning*, pages 3821–3830. PMLR, 2017.