Efficient Online Mirror Descent Stochastic Approximation for Multi-Stage Stochastic Programming

Junhui Zhang[1] and Patrick Jaillet[1,2]

[1]Operations Research Center, MIT
[2]Department of Electrical Engineering and Computer Science, MIT

Abstract

We study the unconstrained and the minimax saddle point variants of the convex multi-stage stochastic programming problem, where consecutive decisions are coupled through the objective functions, rather than through the constraints. Based on the analysis of deterministic mirror descent algorithms with inexact gradients, we introduce the idea of *stochastic conditional gradient oracles*, a multi-stage analog of the stochastic gradient oracles used in (classical) stochastic programming. We show one approach to construct such oracles and prove the convergence of the (accelerated) mirror descent stochastic approximation, both in expectation and with high probability. To further reduce the oracle complexity, we view the problem from a *semi-online* perspective, where the stage t decision variables are constructed s stages in advance, instead of before stage 1. We show that the delay in decision making allows an asynchronous implementation of the mirror descent stochastic approximation algorithms. By avoiding computing solutions for scenarios that are inconsistent with information available during stage t, the complexity is reduced from exponential to linear in the number of stages.

1 Introduction

Sequential decision making has found applications in a variety of real-life problems: from classical ones such as power system management [5, 29, 33] and inventory control [22, 40], to more modern ones such as data center management [9, 27, 28] and online resource allocation [2, 42, 43]. In this work, we study *sequential* decision making in *stochastic* environments under a *semi-online* framework.

To model the sequential revelation of information and the underlying stochasticity, we adopt a multi-stage stochastic programming (MSSP) formulation [40]. Informally, MSSP divides the decision-making process into stages ($t = 1, \ldots, T$), and at stage t, the value of the variable x_t needs to be determined based only on the information known by stage t. The goal is to minimize the costs subject to coupled constraints, both of which are random functions of $x_1, \ldots, x_T$. Due to its wide applications, MSSP has been studied in many prior works [12, 33, 40]. However, existing algorithms either suffer from restrictive assumptions such as stage-wise independent randomness [12, 22], or have exponential (in T) complexity [23, 39].

We study the unconstrained and the minimax saddle point variants of the convex multi-stage stochastic programming problems, where consecutive decisions are coupled through the objective functions, rather than through the constraints. Motivated by the analysis of (deterministic, inexact) mirror descent algorithms, we propose the *stochastic conditional gradient oracle*, a generalization of the stochastic gradient oracles in the classical stochastic programming literature [21, 31]. We

provide one sampling approach to construct such oracles, and show that mirror descent stochastic approximations with these oracles converge in expectation and with high probability.

To further improve the oracle and memory efficiency, we consider a semi-online framework, where the decision x_t only needs to be made at stage $\max(1, t-s)$, i.e., s stages in advance. We propose an online updating mechanism, which delays the updates (i.e., mirror descent steps) of future decision variables. Taking advantage of the decomposability of mirror descent updates across stages and scenarios, we show that the delay does not change the outputs of the algorithms, thus maintaining their convergence properties. Moreover, the information gained during the delay helps the algorithms avoid computing solutions for scenarios inconsistent with the available information, thus improving the complexity from exponential to linear in T.

We demonstrate the effectiveness and robustness of our mirror descent stochastic approximations by applying them to a tracking problem and a revenue management problem.

1.1 Related works

In the classical setup of stochastic programming (SP), the objective function is $F(x) = \mathbb{E}[f(x, w)]$, where $w \in \Omega$ and $(\Omega, \mathcal{F}, \mathbb{P})$ is the underlying probability space. Multi-stage stochastic programming (with T stages) is a generalization of SP to cases where information about $w = (w_1, \ldots, w_T)$ is revealed sequentially. That is, at stage t, w_t is revealed and the stage t variable $x_t \in \mathbb{R}^{n_t}$ needs to be determined based on $w_{1:t}$. The goal is to minimize $\sum_{t=1}^{T} f_t(x_t, w_{1:t})$ subject to the constraints $\mathbf{g}_t(x_{t-1}, x_t, w_{1:t}) \leq \mathbf{0}$ for all t.

Treating the stage t decision as a random variable $X_t : \Omega \to \mathbb{R}^{n_t}$, the information constraint requires that X_t should be measurable w.r.t. $\mathcal{F}_t = \sigma(w_{1:t})$, the σ-algebra generated by the first t components of w. This *non-anticipativity constraint* can be formulated as the linear constraint that $\mathbb{E}[X_t|\mathcal{F}_t] = X_t$, and algorithms such as the augmented Lagrangian method can be applied [35, 36, 37]. Alternatively, MSSP can be formulated using *scenario trees*, where nodes in layer-t of a scenario tree represent different information available by stage t. These two formulations are equivalent ways to model MSSP (with finite scenarios) [40], and our mirror descent algorithms suggest that algorithms designed for the non-anticipativity constraint formulation also admit a scenario tree interpretation.

Algorithms for MSSP. Mirror descent stochastic approximation has been well studied for stochastic programming problems where the objective functions are of the (simple) form $F(x) = \mathbb{E}[f(x, w)]$ [17, 18, 21, 31, 40]. In the multi-stage setting, to apply stochastic approximation type of algorithms, one needs to have access to (potentially stochastic, biased) first order oracles for the cost-to-go functions. [23] proposes the dynamic stochastic approximation (DSA) algorithm, which solves backwardly for approximate subgradients of the cost-to-go functions at the query points, and then applies inexact primal-dual updates. To find a $T\epsilon$ suboptimal *first stage* solution, DSA has complexity $O(\epsilon^{-2T})$ for convex objectives and $O(\epsilon^{-T})$ for strongly convex objectives. Another well known algorithm for MSSP is the stochastic dual dynamic programming (SDDP) [1, 16, 19, 33], which is a cutting-plane based algorithm designed for MSSP where the randomness in different stages are independent: $f_t, \mathbf{g}_t$ depend on w_t but not $w_{1:(t-1)}$, and $w_1, \ldots, w_T$ are independent. [19, 20] show that to find an $O(T\epsilon)$ suboptimal solution for a scenario tree which has at most d children per node, SDDP needs $O(Td^T\epsilon^{-\max_{t=1,\ldots,T} n_t})$ forward-backward iterations, where each iteration involves (approximately) solving T convex optimization problems (of dimension $O(\max_{t=1,\ldots,T} n_t)$).

In addition, [38, 39, 41] show that for sample average approximation type of algorithms for MSSP, the sample complexity has an *upper bound* which is exponential in T. In fact, [11, 14] show that even approximating the solution of 2-stage stochastic programs is $\#P$-hard for a sufficiently high accuracy. In this work, we propose asynchronous versions of mirror descent stochastic

approximations, which have linear in T complexity in the semi-online setting.

Online optimization. Sequential decision making is also studied through the perspective of online optimization, where the unknown part (i.e., future) of the objectives could be potentially chosen by an adversary, and the performance is compared to the optimal *offline* solution. Well-known algorithms include online mirror descent [15, 45], online restarted gradient descent [4], and online primal-dual approach [7], to name a few. In this work, we consider the semi-online case, where the solution is constructed in an online fashion and is compared against the optimal *online* solution. Our efficient online implementation is motivated by a recent line of research on smoothed online convex optimization [3, 13, 24, 25, 26, 27, 44], where the cost at stage t is the sum of a stage cost which depends only on x_t, and a switching cost $\|x_t - x_{t-1}\|$ or $\|x_t - x_{t-1}\|^2$. As a comparison, our algorithms can be applied to (convex or saddle point) problems with general couplings between consecutive decision variables, and do not require strong convexity (as required in [25, 26]).

1.2 Contributions

We make the following contributions to the multi-stage stochastic programming literature.

- Based on a pathwise analysis of mirror descent algorithms for the multi-stage stochastic programming problems (MS-Unconstrained) and (MS-Saddle) (Section 3), we propose the stochastic conditional gradient oracle, a multi-stage analog of the stochastic gradient oracle in stochastic programming literature (Section 4.1). We propose a sampling algorithm that constructs such conditional oracles robust to distribution misspecification (Section 4.2).

- We propose (accelerated) mirror descent stochastic approximation algorithms for the multi-stage problems and prove their convergence, both in expectation and with high probability, with potentially *biased* stochastic conditional gradient oracles (Section 4.3). Compared with existing algorithms for multi-stage stochastic programming, our algorithms do not assume stage-wise independent randomness. In addition, our algorithms do not require knowledge of T, the total number of stages, when setting the parameters.

- To reduce the oracle and space complexity, we propose an efficient online implementation of the (accelerated) mirror descent stochastic approximation algorithms (Section 5). The overall algorithm, when applied to a scenario tree with at most d children per node and when decisions are made s stages in advance, achieves oracle complexity $O(Td^s 2^{1/\epsilon^2})$ and space complexity $O((d^s\epsilon^{-2} + \epsilon^{-4}) \max_{t=1,\dots,T} n_t)$ to find an ϵT-suboptimal solution for convex objectives.

In addition, our results can be applied to more general information constraints: we assume that the decisions are made based on information in $\mathcal{G}_t$ which is a relaxation of the baseline $\mathcal{F}_t = \sigma(w_1, \dots, w_t) \subset \mathcal{G}_t$.

2 Setup

We consider T-stage stochastic programming problems with and without constraints, on a discrete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ where $\Omega = [K]$ is finite, $\mathcal{F}$ is the power set of Ω, and $p_k = \mathbb{P}[w = k] > 0$ for all $k = 1, 2, \dots, K$. For convenience, for a sub-σ-algebra $\mathcal{G} \subset \mathcal{F}$ and a random variable $X : \Omega \to \mathbb{R}^{n_0}$ for some $n_0 \in \mathbb{N}$, by $X \in \mathcal{G}$ we mean X is measurable w.r.t. $\mathcal{G}$; given a set $S \subset \mathbb{R}^{n_0}$, by $X \in \mathcal{G} \cap S$ we mean $X \in \mathcal{G}$ and $X(w) \in S$ for all $w \in \Omega$; for $f : \mathbb{R}^{n_0} \times \Omega \to \mathbb{R}$ and $X : \Omega \to \mathbb{R}^{n_0}$, we use $f(X) : \Omega \to \mathbb{R}$ to denote $f(X)(w) = f(X(w), w)$.

In addition, recall that a function $f : \Omega \times \mathbb{R}^{n_0} \to \overline{\mathbb{R}}$ is random lower-semicontinuous (lsc) w.r.t. $\mathcal{G} \subset \mathcal{F}$ if $\mathrm{epi}_f : \Omega \to \mathbb{R}^{n_0} \times \overline{\mathbb{R}}$ defined as $\mathrm{epi}_f(w) = \mathrm{epi}_{f(w,\cdot)}$ is closed valued and measurable w.r.t. $\mathcal{G}$. For its properties, see [34, 40].

2.1 Multi-stage unconstrained programming

By a multi-stage unconstrained programming (MS-Unconstrained) problem, we mean the following:

$$\inf_{X_1 \in \mathcal{G}_1 \cap \mathcal{X}_1} \cdots \inf_{X_T \in \mathcal{G}_T \cap \mathcal{X}_T} \mathbb{E}[f(X_{1:T})], \tag{MS-Unconstrained}$$
$$f(x_1, \ldots, x_T, w) := f_1(x_1, w) + \sum_{t=2}^{T} f_t(x_{t-1}, x_t, w),$$

where $f_t : \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{n_t} \times \Omega \to \mathbb{R}$ ($n_0 = 0$) and $\mathcal{X}_t \subset \mathbb{R}^{n_t}$ for all $t = 1, \ldots, T$. In addition, $\{\Omega, \emptyset\} = \mathcal{G}_1 \subset \cdots \subset \mathcal{G}_T \subset \mathcal{F}$ is a filtration that represents the information available at each stage. For convenience, $n = \sum_{t=1}^{T} n_t$. To ensure that $\mathcal{G}_t$ contains all the information necessary to evaluate f_t at stage t, we make the following assumption.

Assumption 2.1. *There exists a baseline filtration $\{\Omega, \emptyset\} = \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_T \subset \mathcal{F}$ such that for all t, $\mathcal{F}_t \subset \mathcal{G}_t$, and f_t is random lsc w.r.t. $\mathcal{F}_t$.*

We also make the following assumption regarding the objective functions and the constraints.

Assumption 2.2. *For each t, $f_t(\cdot, \cdot, w)$ is a convex function in (x_{t-1}, x_t) and is differentiable for all $w \in \Omega$, and $\mathcal{X}_t \subset \mathbb{R}^{n_t}$ is a nonempty compact convex subset. In addition, $X^*_{1:T}$ is a solution to* (MS-Unconstrained).

Remark. Our Assumption 2.1 (and Assumption 2.3 below) generalizes the information constraints in the classical setups of MSSP, where the sample space Ω consists of $w = (w_1, \ldots, w_T)$, and the information available at stage t is $\mathcal{F}_t = \sigma(w_{1:t})$. As an example, if $\mathcal{G}_t = \mathcal{F}$ for all t, then full information is available for all stages. As another example, for $l \in \mathbb{N}$, having l-step look-ahead into the future objectives and constraints can be modeled using $\mathcal{G}_t = \mathcal{F}_{\min(t+l,T)}$.

2.2 Multi-stage saddle point problem

By a multi-stage saddle point problem (MS-Saddle), we mean the following:

$$\inf_{X_1 \in \mathcal{G}_1 \cap \mathcal{X}_1} \cdots \inf_{X_T \in \mathcal{G}_T \cap \mathcal{X}_T} \mathbb{E}[\tilde{\phi}(X_{1:T})] \tag{MS-Saddle}$$
$$\tilde{\phi}(x_1, \ldots, x_T, w) := \tilde{\phi}_1(x_1, w) + \sum_{t=2}^{T} \tilde{\phi}_t(x_{t-1}, x_t, w)$$
$$\tilde{\phi}_t(x_{t-1}, x_t, w) := \sup_{y_t \in \mathcal{Y}_t} \phi_t(x_{t-1}, x_t, y_t, w), \quad t = 1, \ldots, T$$

where $\phi_t : \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{n_t} \times \mathbb{R}^{m_t} \times \Omega \to \mathbb{R}$ ($n_0 = 0$), $\mathcal{X}_t \in \mathbb{R}^{n_t}$ and $\mathcal{Y}_t \in \mathbb{R}^{m_t}$ for all t. For convenience, $m = \sum_{t=1}^{T} m_t$, and we denote

$$\phi(x_1, y_1, \ldots, x_T, y_T, w) := \phi_1(x_1, y_1, w) + \sum_{t=2}^{T} \phi_t(x_{t-1}, x_t, y_t, w).$$

Similar to the above setting for the unconstrained problem, we make the following measurability assumption.

Assumption 2.3. *There exists a baseline filtration $\{\Omega, \emptyset\} = \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_T \subset \mathcal{F}$ such that for all t, $\mathcal{F}_t \subset \mathcal{G}_t$, and $\pm\phi_t$ and $\tilde{\phi}_t$ are random lsc w.r.t. $\mathcal{F}_t$.*

We also make the following assumption regarding the objective functions and the constraints.

Assumption 2.4. *For each t, $\phi_t(\cdot, \cdot, \cdot, w)$ is differentiable, convex in (x_{t-1}, x_t) and concave in y_t for all $w \in \Omega$, and $\mathcal{X}_t \subset \mathbb{R}^{n_t}$ and $\mathcal{Y}_t \subset \mathbb{R}^{m_t}$ are nonempty compact convex subsets. In addition, $X^*_{1:T}$ is a solution to* (MS-Saddle)*, with a corresponding $Y^*_{1:T}$ such that $Y^*_t \in \mathcal{G}_t \cap \mathcal{Y}_t$ and $\tilde{\phi}_t(X^*_{t-1}(w), X^*_t(w), w) = \phi_t(X^*_{t-1}(w), X^*_t(w), Y^*_t(w), w)$ for all t and all w.*

The problem (MS-Saddle) is well defined under Assumptions 2.3, in the sense that $\tilde{\phi}_t$ is random lsc w.r.t. $\mathcal{G}_t$ for all t. We provide further discussions of these assumptions in Appendix A.

Recall that in the classical MSSP formulation [40], consecutive decision variables are coupled through the constraints: for each scenario $w \in \Omega$, the problem is

$$\min_{x_1 \in \mathcal{X}_1, \ldots, x_T \in \mathcal{X}_T} \sum_{t=1}^{T} h_t(x_t, w), \quad s.t.\ g_1(x_1, w) \le \mathbf{0},\ g_t(x_{t-1}, x_t, w) \le \mathbf{0},\ t = 2, \ldots, T \tag{1}$$

where $h_t : \mathbb{R}^{n_t} \times \Omega \to \mathbb{R}$ and each component of $g_t : \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{n_t} \times \Omega \to \mathbb{R}^{m_t}$ ($n_0 = 0$) are convex and satisfy similar conditions as Assumption 2.1.

Under regularity conditions (such as Proposition 3.6 in [40] for linear constraints), (1) can be reformulated as a saddle point problem MS-Saddle, with $\phi_1(x_1, y_1, w) = f_1(x_1, w) + \langle y_1, g_1(x_1, w)\rangle$ and for $t = 2, \ldots, T$, $\phi_t(x_{t-1}, x_t, y_t, w) = f_t(x_t, w) + \langle y_t, g_t(x_{t-1}, x_t, w)\rangle$, with the caveat that $\mathcal{Y}_t = \mathbb{R}^{m_t}_{\ge 0}$ is unbounded. Nevertheless, additional structural assumptions on the problem (1) could lead to (implicit) bounds on the norms of dual solutions [23]. Our revenue management experiment in Section 6.2 is an example (see Appendix C for additional details).

2.3 Modeling the information constraints

In this work, we use two (equivalent) approaches to model the information constraints that a variable X is measurable w.r.t. some sub-σ-algebra $\mathcal{G} \subset \mathcal{F}$.

Pathwise approach. Since K is finite, a random vector $X : \Omega \to \mathbb{R}^{n_0}$ such that $X \in \mathcal{F}$ can be viewed as a vector $X = [X(1)|\cdots|X(K)] \in \mathbb{R}^{n \times K}$. Thus, given $\mathcal{G} \subset \mathcal{F}$, the information constraint that $X \in \mathcal{G}$ can be equivalently represented as $X\mathsf{P}_\mathcal{G} = X$, where $\mathsf{P}_\mathcal{G} \in \mathbb{R}^{\Omega \times \Omega}$ is defined as $\mathsf{P}_\mathcal{G}(i, j) = \mathbb{E}[\mathbf{1}[w = i]|\mathcal{G}](j)$ for $i, j \in \Omega$. $\mathsf{P}_\mathcal{G}$ represents the distribution $\mathbb{P}$ conditioned on $\mathcal{G}$, and useful properties for this matrix are presented in Lemma B.1. With this notation, the constraints in (MS-Unconstrained) are equivalent to $X_t\mathsf{P}_t = X_t$ and $X_t(w) \in \mathcal{X}_t$ for all $w \in \Omega$, for all t. And similarly for (MS-Saddle). We abbreviate $\mathsf{P}_{\mathcal{G}_t} = \mathsf{P}_t$ for simplicity.

Scenario trees. In the scenario tree formulation, layers correspond to stages, and nodes correspond to different information that has been revealed. More precisely, for a sub-σ-algebra $\mathcal{G} \subset \mathcal{F}$, we use $\Omega_\mathcal{G}$ to denote the associated partition of Ω. Thus, $\mathcal{G} = 2^{\Omega_\mathcal{G}}$ is the power set of $\Omega_\mathcal{G}$. For convenience, we use $[w]_\mathcal{G}$ to denote the partition that w lies in in $\Omega_\mathcal{G}$. For a random variable $X : \Omega \to \mathbb{R}^{n_0}$ which is measurable w.r.t. $\mathcal{G}$, we use $X_\mathcal{G} : \Omega_\mathcal{G} \to \mathbb{R}^{n_0}$ to denote the *reduced* random variable, where $X_\mathcal{G}([w]_\mathcal{G}) := X(w)$ for all $w \in \Omega$, and we abbreviate it as $X([w]_\mathcal{G})$. Similarly, if a function $f : \mathbb{R}^{n_0} \times \Omega \to \mathbb{R}$ is such that for any $x \in \mathbb{R}^{n_0}$, $f(x, \cdot)$ is $\mathcal{G}$ measurable, then $f(x, [w]_\mathcal{G}) = f(x, w)$ for all $w \in \Omega$. We abbreviate $[w]_{\mathcal{G}_t} = [w]_t$ and $\Omega_{\mathcal{G}_t} = \Omega_t$ for simplicity. In addition, we denote $\pi_t([w]_t)$ as the distribution over $\Omega_{t+1}([w]_t) = \{[w']_{t+1},\ w' \in [w]_t\}$, i.e. the distribution over the children of $[w]_t$.

Comparing the two approaches. The two approaches are equivalent for the setup we consider (see Figure 1 for an example). Thus, one might wonder why we adopt two approaches

instead of one. We would like to point out that when implementing numerical algorithms, trees might be more memory efficient. Consider a random variable $X : \Omega \to \mathbb{R}$ which is measurable w.r.t. $\mathcal{G} \subset \mathcal{F}$. To store such a variable X, there is no need to store it as a vector in $\mathbb{R}^K$. Instead, one only needs to store the reduced variable $X_{\mathcal{G}} : \Omega_{\mathcal{G}} \to \mathbb{R}$, which is a vector in $\mathbb{R}^{|\Omega_{\mathcal{G}}|}$. For instance, for $\mathcal{G} = \{\emptyset, \Omega\}$ and $\Omega_{\mathcal{G}} = \{\Omega\}$, we have $|\Omega_{\mathcal{G}}| = 1$ and so only one number in $\mathbb{R}$ needs to be stored. On the other hand, as will be presented in Section 4.3, there are two sources of randomness in the problem: from the multi-stage problem itself, and from the sampling process to construct the stochastic (conditional) gradients. Thus, the pathwise perspective allows a more formal treatment of the randomness, especially the (in)dependence between the samples and the constructed X_t involved.

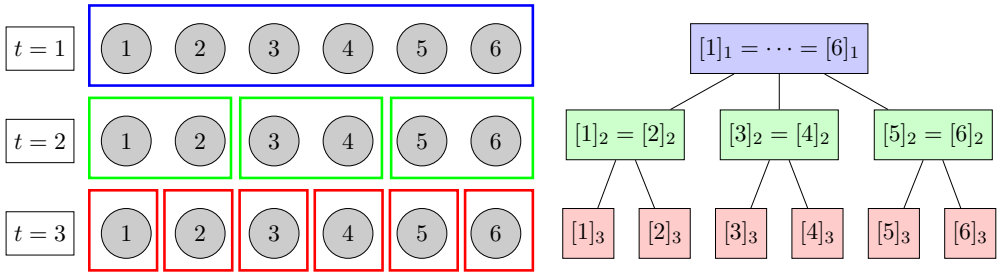


Figure 1: Example probability space with $\Omega = \{1, \ldots, 6\}$, $\Omega_1 = \{\{1, \ldots, 6\}\}$, $\Omega_2 = \{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$, and $\Omega_3 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$. Left: pathwise representation where boxes are partitions based on $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$. Right: tree representation.

3 Mirror descent: a deterministic perspective

We first give a brief review of mirror descent [6, 21, 32]. We equip $\mathbb{R}^n$ with the Euclidean inner product $\langle \cdot, \cdot \rangle$ and a norm $\|\cdot\|$ not necessarily induced by the inner product. Recall that for a convex compact subset $\mathcal{Q} \subset \mathbb{R}^n$, $v : \mathcal{Q} \to \mathbb{R}$ is a distance generating function [21, 31] for $\mathcal{Q}$ if v is convex and continuous on $\mathcal{Q}$, and

$$\mathcal{Q}^o := \{x \in \mathcal{Q} | \exists g \in \mathbb{R}^n,\ x \in \operatorname*{argmin}_{x' \in \mathcal{Q}} v(x') + \langle g, x' \rangle\}$$

is a convex set, and restricted to $\mathcal{Q}^o$, v is continuously differentiable and 1-strongly convex, i.e.:

$$\langle x' - x, \nabla v(x') - \nabla v(x) \rangle \geq \|x' - x\|^2.$$

Then, initialized at some $x^{(0)} \in \mathcal{Q}^o$, mirror descent updates the variables iteratively for $l = 0, 1, \ldots$

$$x^{(l+1)} = \operatorname*{argmin}_{x' \in \mathcal{Q}} \gamma_l \langle g^{(l)}, x' \rangle + D_v(x', x^{(l)}), \tag{2}$$

where $D_v(y, x) := v(y) - v(x) - \langle \nabla v(x), y - x \rangle$ is the Bregman divergence induced by v and $g^{(l)} \in \mathbb{R}^n$ is a subgradient of the objective function, and $\gamma_l \geq 0$ is usually chosen based on properties of the objective function f and the set $\mathcal{X}$.

In Section 3.1, we treat (MS-Unconstrained) and (MS-Saddle) as problems in $\mathbb{R}^{n \times K}$ and $\mathbb{R}^{(n+m) \times K}$ respectively, and show that with suitable inner products, norms, and the distance generating functions, the update (2) is efficient. In Section 3.2, we propose and analyze the (accelerated) mirror descent updates for problems (MS-Unconstrained) and (MS-Saddle) with *inexact* gradient information. In Section 3.3, we provide a scenario tree perspective.

3.1 Mirror descent updates for all scenarios

We first reformulate (MS-Unconstrained) as a convex optimization problem in $\mathbb{R}^{n\times K}$ with the objective function

$$F(X) := \mathbb{E}[f(X_{1:T}(w), w)], \quad \forall X = (X_1, \ldots, X_T) \in \mathbb{R}^{n\times K}.$$

For the constraints, we define $\overline{\mathcal{X}}_t \subset \mathbb{R}^{n_t\times K}$ as

$$\overline{\mathcal{X}}_t := \mathcal{G}_t \cap \mathcal{X}_t = \{X_t \in \mathbb{R}^{n_t\times K},\ X_t\mathsf{P}_t = X_t,\ X_t(w) \in \mathcal{X}_t\ \forall w \in \Omega\}.$$

In addition, $\mathcal{X} := \prod_{t=1}^T \mathcal{X}_t$ and $\overline{\mathcal{X}} = \prod_{t=1}^T \overline{\mathcal{X}}_t$. Thus, the information constraint in (MS-Unconstrained) is equivalent to $X \in \overline{\mathcal{X}}$. We make the following assumption about the inner products, norms, and distance generating functions in $\mathbb{R}^n$.

Assumption 3.1. *We equip $\mathbb{R}^n$ with the Euclidean inner product $\langle\cdot,\cdot\rangle$. For all t, $\mathbb{R}^{n_t}$ is equipped with the norm $\|\cdot\|$ not necessarily induced by the Euclidean inner product, and $v_t : \mathcal{X}_t \to \mathbb{R}$ is a distance generating function which is 1-strongly convex w.r.t. the norm $\|\cdot\|$ such that $\mathcal{X}_t$ admits easy projection using D_{v_t}. We equip $\mathbb{R}^n$ with the norm $\|x_{1:T}\|^2 = \sum_{t=1}^T \|x_t\|^2$ and $\mathcal{X}$ with the distance generating function $v(x_{1:T}) := \sum_{t=1}^T v_t(x_t)$.*

From $\mathbb{R}^{n_t}$ to $\mathbb{R}^{n_t\times K}$, we use the following induced norm and inner product:

$$\langle X_t, X_t'\rangle := \mathbb{E}[\langle X_t(w), X_t'(w)\rangle], \quad \|X_t\|^2 = \mathbb{E}[\|X_t(w)\|^2].$$

Then the dual norm satisfies that $\|Y_t\|_*^2 = \mathbb{E}[\|Y_t(w)\|_*^2]$ (Lemma B.2). Further, we use $V_t : \mathcal{X}_t^K \to \mathbb{R}$ and $D_{V_t} : \mathcal{X}_t^K \times (\mathcal{X}_t^o)^K \to \mathbb{R}$ to denote

$$V_t(X_t) = \mathbb{E}[v_t(X_t)], \quad D_{V_t}(X_t, X_t') = \mathbb{E}[D_{v_t}(X_t(w), X_t'(w))],$$

thus V_t is 1-strongly convex. We further extend the above definitions to $\mathbb{R}^{n\times K}$ through $\|X\|^2 = \sum_{t=1}^T \|X_t\|^2$, $V(X) = \sum_{t=1}^T V_t(X_t)$, and similarly for the norm and D_V.

With these decomposable assumptions on the distance generating functions, inner products, and norms, we have the following Lemma 3.1, which implies that the Bregman projections [1] are still easy.

Lemma 3.1. *Under Assumption 3.1. For $X = (X_1, \ldots, X_T) \in \prod_{t=1}^T \mathcal{G}_t \cap \mathcal{X}_t^o$ and $G = (G_1, \ldots, G_T) \in \prod_{t=1}^T \mathcal{F} \cap \mathbb{R}^{n_t}$, the following is well defined*

$$X^+ = \operatorname*{argmin}_{X'\in\overline{\mathcal{X}}} \langle G, X'\rangle + D_V(X', X).$$

The following holds for all t:

$$X_t^+ = \operatorname*{argmin}_{X_t'\in\mathcal{G}_t\cap\mathcal{X}_t} \langle G_t\mathsf{P}_t, X_t'\rangle + D_{V_t}(X_t', X_t),$$

$$X_t^+(w) = \operatorname*{argmin}_{x_t\in\mathcal{X}_t} \langle G_t\mathsf{P}_t(w), x_t\rangle + D_{v_t}(x_t, X_t(w)),\ \forall w \in \Omega.$$

In addition, for any $X_t' \in \mathcal{G}_t \cap \mathcal{X}_t$,

$$\langle G_t\mathsf{P}_t, X_t - X_t'\rangle - \frac{1}{2}\|G_t\mathsf{P}_t\|_*^2 \le D_{V_t}(X_t', X_t) - D_{V_t}(X_t', X_t^+).$$

[1] The distance generating functions defined in [21] and [31] are for the standard Euclidean inner product $\langle\cdot,\cdot\rangle$, but over the $n_t \times K$ dimensional space, the inner product is the *expected* inner product. Thus, V_t is not a distance generating function for $\overline{\mathcal{X}}_t$ (in the sense of [21] and [31]). Nevertheless, we still call V_t a distance generating function and D_{V_t} the Bregman divergence.

Lemma 3.1 suggests that the projection onto $\overline{\mathcal{X}}$ is decomposable, and that only the $G_t\mathsf{P}_t = \mathbb{E}[G_t|\mathcal{G}_t]$ component matters. The proof of Lemma 3.1 is deferred to Appendix B.1.

For the saddle problem (MS-Saddle), we denote $\mathcal{Z}_t = \mathcal{X}_t \times \mathcal{Y}_t$, and set $\overline{\mathcal{Y}}_t$, $\overline{\mathcal{Z}}_t$, $\overline{\mathcal{Y}}$, $\overline{\mathcal{Z}}$ similarly as above. In addition, we denote $\Phi(Z_{1:T}) = \mathbb{E}[\phi(X_{1:T}, Y_{1:T})]$ and

$$\frac{\tilde{\partial}\phi}{\partial z_t}(z,w) = \begin{bmatrix} \frac{\partial}{\partial x_t}\phi(z,w) \\ -\frac{\partial}{\partial y_t}\phi(z,w) \end{bmatrix}, \quad \frac{\tilde{\partial}\phi}{\partial z}(z,w) = \begin{bmatrix} \frac{\partial}{\partial x}\phi(z,w) \\ -\frac{\partial}{\partial y}\phi(z,w) \end{bmatrix}.$$

Assumption 3.2. *We equip $\mathbb{R}^{n+m}$ with the Euclidean inner product $\langle\cdot,\cdot\rangle$. For all t, $\mathbb{R}^{n_t}$ ($\mathbb{R}^{m_t}$) are equipped with the norm $\|\cdot\|$ not necessarily induced by the Euclidean inner product, and $v_t : \mathcal{X}_t \to \mathbb{R}$ ($u_t : \mathcal{Y}_t \to \mathbb{R}$) is a distance generating function which is 1-strongly convex w.r.t. the norm $\|\cdot\|$ such that $\mathcal{X}_t$ ($\mathcal{Y}_t$) admits easy projecting using D_{v_t} (D_{u_t}). We equip $\mathbb{R}^{n+m}$ with the norm $\|z_{1:T}\|^2 = \sum_{t=1}^T(\|x_t\|^2 + \|y_t\|^2)$ and $\mathcal{Z}$ with the distance generating function $w(z_{1:T}) := \sum_{t=1}^T(v_t(x_t) + u_t(y_t))$.*

Results similar to Lemma 3.1 can be shown for $Z_t \in \mathcal{G}_t \cap (\mathcal{X}_t^o \times \mathcal{Y}_t^o)$ and $G_t \in \mathcal{F} \cap \mathbb{R}^{m_t+n_t}$.

3.2 A deterministic view of mirror descent

In this section, we present the mirror descent algorithms [21, 31] and analyze their non-asymptotic convergence properties with *inexact* gradient information – the accumulation of gradient *inexactness* allows easy adaptation of the methods and their performance to stochastic approximation type of methods, which we explore in Section 4.

3.2.1 Mirror descent for (MS-Unconstrained)

Consider the mirror descent algorithm applied to (MS-Unconstrained) with the distance generating function $V(X_{1:T}) := \sum_{t=1}^T V_t(X_t)$ and initialization $X_1^{(0)} \in \mathcal{G}_1 \cap \mathcal{X}_1^o, \ldots, X_T^{(0)} \in \mathcal{G}_T \cap \mathcal{X}_T^o$:

$$X_{1:T}^{(l+1)} = \operatorname*{argmin}_{(X_1,\ldots,X_T)\in\overline{\mathcal{X}}} \langle \gamma_l G_{1:T}^{(l)}, X_{1:T}\rangle + D_V(X_{1:T}, X_{1:T}^{(l)}), \quad l = 0, 1, \ldots \tag{3}$$

where $\gamma_l \geq 0$, and $G_t^{(l)} : \Omega \to \mathbb{R}^{n_t}$ for $t = 1, 2, \ldots, T$ are (approximate) gradients.

Lemma 3.2. *Assume that Assumptions 2.1, 2.2, and 3.1 hold. Consider the update in* (3)*, then for $\overline{X}_t^{(L)} := \frac{\sum_{l=0}^L \gamma_l X_t^{(l)}}{\sum_{l=0}^L \gamma_l}$, we have*

$$\begin{aligned} &\mathbb{E}[f(\overline{X}_{1:T}^{(L)})] - \mathbb{E}[f(X_{1:T}^*)] \\ &\leq \frac{D_V(X_{1:T}^*, X_{1:T}^{(0)}) + \frac{1}{2}\sum_{l=0}^L \gamma_l^2 \sum_{t=1}^T \|G_t^{(l)}\mathsf{P}_t\|_*^2 - \sum_{l=0}^L \gamma_l \langle \Delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}^*\rangle}{\sum_{l=0}^L \gamma_l}, \end{aligned}$$

where $\Delta_t^{(l)} = \left(G_t^{(l)} - \frac{\partial}{\partial x_t} f(X_{1:T}^{(l)})\right)\mathsf{P}_t$.

Proof of Lemma 3.2. By Lemma 3.1, for any $X_t' \in \mathcal{G}_t \cap \mathcal{X}_t$,

$$\langle \gamma_l G_t^{(l)}\mathsf{P}_t, X_t^{(l)} - X_t'\rangle \leq D_{V_t}(X_t', X_t^{(l)}) - D_{V_t}(X_t', X_t^{(l+1)}) + \frac{1}{2}\gamma_l^2 \|G_t\mathsf{P}_t\|_*^2.$$

In addition, since $X_t^{(l)}$ and X_t' are both $\mathcal{G}_t$ measurable, by definition of $\Delta_t^{(l)}$, we have

$$\langle \Delta_t^{(l)}, X_t^{(l)} - X_t'\rangle = \langle G_t^{(l)}\mathsf{P}_t, X_t^{(l)} - X_t'\rangle - \langle \frac{\partial}{\partial x_t} f(X_{1:T}^{(l)}), X_t^{(l)} - X_t'\rangle.$$

Thus, the above can be rewritten as

$$\begin{aligned}&\langle \gamma_l \frac{\partial}{\partial x_t} f(X_{1:T}^{(l)}), X_t^{(l)} - X_t' \rangle \\ &\le D_{V_t}(X_t', X_t^{(l)}) - D_{V_t}(X_t', X_t^{(l+1)}) + \frac{1}{2}\gamma_l^2 \|G_t \mathsf{P}_t\|_*^2 - \gamma_l \langle \Delta_t^{(l)}, X_t^{(l)} - X_t' \rangle.\end{aligned}$$

Summing over l and t, and using $D_{V_t} \ge 0$, we get

$$\begin{aligned}\sum_{l=0}^{L} \gamma_l \langle \frac{\partial}{\partial x} f(X_{1:T}^{(l)}), X_{1:T}^{(l)} - X_{1:T}' \rangle \le & D_V(X_{1:T}', X_{1:T}^{(0)}) + \frac{1}{2}\sum_{l=0}^{L} \gamma_l^2 \sum_{t=1}^{T} \|G_t \mathsf{P}_t\|_*^2 \\ & - \sum_{l=0}^{L} \gamma_l \langle \Delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}' \rangle.\end{aligned} \tag{4}$$

By convexity of $f(\cdot, w)$ we have for any $w \in \Omega$,

$$\sum_{l=0}^{L} \gamma_l \langle \frac{\partial}{\partial x} f(X_{1:T}^{(l)}(w)), X_{1:T}^{(l)}(w) - X_{1:T}'(w) \rangle \ge (\sum_{l=0}^{L} \gamma_l)(f(\overline{X}_{1:T}^{(L)}(w), w) - f(X_{1:T}'(w), w)).$$

Finally, taking $X_t' = X_t^*$, and taking expectation w.r.t. w and dividing by $(\sum_{l=0}^{L} \gamma_l)$ gives the result. □

3.2.2 Mirror descent for (MS-Saddle)

Consider the mirror descent algorithm applied to (MS-Saddle) with the distance generating function $W(Z_{1:T}) := \sum_{t=1}^{T} V_t(X_t) + U_t(Y_t)$ for $Z_t = (X_t, Y_t)$ with initialization $Z_1^{(0)} \in \mathcal{G}_1 \cap (\mathcal{X}_1^o \times \mathcal{Y}_1^o), \ldots, Z_T^{(0)} \in \mathcal{G}_T \cap (\mathcal{X}_T^o \times \mathcal{Y}_T^o)$:

$$Z_{1:T}^{(l+1)} = \operatorname*{argmin}_{(Z_1,\ldots,Z_T) \in \overline{\mathcal{Z}}} \langle \gamma_l G_{1:T}^{(l)}, Z_{1:T} \rangle + D_W(Z_{1:T}, Z_{1:T}^{(l)}), \quad l = 0, 1, \ldots, \tag{5}$$

where $\gamma_l \ge 0$, and $G_t^{(l)} : \Omega \to \mathbb{R}^{n_t + m_t}$ for $t = 1, 2, \ldots, T$ are (approximate) gradients.

Lemma 3.3. *Assume that Assumptions 2.3, 2.4, and 3.2 hold. Consider the update in* (5)*, then for* $\overline{Z}_t^{(L)} := \frac{\sum_{l=0}^{L} \gamma_l Z_t^{(l)}}{\sum_{l=0}^{L} \gamma_l}$ *and any* $Z \in \overline{\mathcal{Z}}$*,*

$$\begin{aligned}&\mathbb{E}[\phi(\overline{X}_{1:T}^{(L)}, Y_{1:T}) - \phi(X_{1:T}, \overline{Y}_{1:T}^{(L)})] \\ &\le \frac{2D_W(Z_{1:T}, Z_{1:T}^{(0)}) + \sum_{l=0}^{L} \sum_{t=1}^{T} \frac{\gamma_l^2}{2}(\|G_t^{(l)} \mathsf{P}_t\|_*^2 + \|\Delta_t^{(l)}\|_*^2) - \gamma_l \langle Z_t^{(l)} - \tilde{Z}_t^{(l)}, \Delta_t^{(l)} \rangle}{\sum_{l=0}^{L} \gamma_l},\end{aligned}$$

where $\Delta_t^{(l)} = \left(G_t^{(l)} - \frac{\tilde{\partial}}{\partial z_t} \phi(Z_{1:T}^{(l)})\right) \mathsf{P}_t$*. The sequence* $\tilde{Z}_t^{(0)} = Z_t^{(0)}$

$$\tilde{Z}_t^{(l+1)} = \operatorname*{argmin}_{Z_t' \in \overline{\mathcal{Z}}_t} \langle -\gamma_l \Delta_t^{(l)}, Z_t' \rangle + D_{W_t}(Z_t', \tilde{Z}_t^{(l)}), \ l = 0, 1, \ldots, L-1.$$

Proof of Lemma 3.3. First, by a similar argument as in the proof of Lemma 3.2 (4), we get for any $Z_t \in \mathcal{G}_t \cap \mathcal{Z}_t$,

$$\sum_{l=0}^{L} \gamma_l \langle Z_t^{(l)} - Z_t, \frac{\tilde{\partial}}{\partial z_t} \phi(Z_{1:T}^{(l)}) \rangle \le D_{W_t}(Z_t, Z_t^{(0)}) + \sum_{l=0}^{L} \frac{\gamma_l^2}{2} \|G_t^{(l)} \mathsf{P}_t\|_*^2 - \gamma_l \langle Z_t^{(l)} - Z_t, \Delta_t^{(l)} \rangle.$$

Now consider the sequence $\tilde{Z}_t^{(0)} = Z_t^{(0)}$ for all t, and $\tilde{Z}_t^{(l+1)}$ defined in the statement of the lemma, then by an argument similar to Lemma 3.1 we have for any $Z_t \in \mathcal{G}_t \cap \mathcal{Z}_t$,

$$-\gamma_l \langle \tilde{Z}_t^{(l)} - Z_t, \Delta_t^{(l)} \rangle \le \frac{\gamma_l^2}{2} \|\Delta_t^{(l)}\|_*^2 + D_{W_t}(Z_t, \tilde{Z}_t^{(l)}) - D_{W_t}(Z_t, \tilde{Z}_t^{(l+1)}).$$

Thus we get

$$\begin{aligned} &\sum_{l=0}^{L} \gamma_l \langle Z_t^{(l)} - Z_t, \frac{\tilde{\partial}}{\partial z_t} \phi(Z_{1:T}^{(l)}) \rangle \\ &\le 2 D_{W_t}(Z_t, Z_t^{(0)}) + \sum_{l=0}^{L} \frac{\gamma_l^2}{2} (\|G_t^{(l)} \mathsf{P}_t\|_*^2 + \|\Delta_t^{(l)}\|_*^2) - \gamma_l \langle Z_t^{(l)} - \tilde{Z}_t^{(l)}, \Delta_t^{(l)} \rangle. \end{aligned} \tag{6}$$

Since $\phi(x_{1:T}, y_{1:T}, w)$ is convex in $x_{1:T}$ and concave in $y_{1:T}$, we have for any $Z_t : \Omega \to \mathbb{R}^{m_t + n_t}$ for $t = 1, \dots, T$,

$$\mathbb{E}[\phi(\overline{X}_{1:T}^{(L)}, Y_{1:T}) - \phi(X_{1:T}, \overline{Y}_{1:T}^{(L)})] \le \left(\sum_{l=0}^{L} \gamma_l \right)^{-1} \left(\sum_{l=0}^{L} \gamma_l \sum_{t=1}^{T} \langle Z_t^{(l)} - Z_t, \frac{\tilde{\partial}}{\partial z} \phi(Z_{1:T}^{(l)}) \rangle \right). \tag{7}$$

Finally, summing (6) over t and together with (7) give the result. □

3.2.3 Accelerated Mirror Descent for smooth (MS-Unconstrained)

With additional assumptions such as smoothness and/or strong convexity of the objective functions, classical convex optimization can be accelerated, with rates of convergence better than $1/\sqrt{l}$. More precisely, assume that $\mathbb{R}^n$ is equipped with the Euclidean inner product $\langle \cdot, \cdot \rangle$ and the *induced norm* $\| \cdot \|$. For the problem (MS-Unconstrained), consider the following updates.

$$\begin{aligned} \overline{X}_{1:T}^{(l)} &= \operatorname*{argmin}_{X_{1:T} \in \overline{\mathcal{X}}} \langle G_{1:T}^{(l)}, X_{1:T} - X_{1:T}^{(l)} \rangle + \frac{(1+\gamma) L_2}{2} \|X_{1:T} - X_{1:T}^{(l)}\|^2 \\ \underline{X}_{1:T}^{(l)} &= \operatorname*{argmin}_{X_{1:T} \in \overline{\mathcal{X}}} (1+\gamma) L_2 D_V(X_{1:T}, X_{1:T}^{(0)}) \\ &\qquad + \sum_{l'=0}^{l} \alpha_{l'} (\langle G_{1:T}^{(l')}, X_{1:T} - X_{1:T}^{(l')} \rangle + \frac{(1-\theta)\mu}{2} \|X_{1:T} - X_{1:T}^{(l')}\|^2) \\ X_{1:T}^{(l+1)} &= \tau_l \underline{X}_{1:T}^{(l)} + (1 - \tau_l) \overline{X}_{1:T}^{(l)} \end{aligned} \tag{8}$$

where $A_l = \sum_{l'=0}^{l} \alpha_{l'}$ and $\tau_l = \frac{\alpha_{l+1}}{A_{l+1}}$, and for some $\gamma \ge 0$ and $\theta \in [0, 1]$,

$$\alpha_0 = 1, \ (1+\gamma) L_2 + (1-\theta) \mu A_l = \frac{(1+\gamma) L_2 \alpha_{l+1}^2}{A_{l+1}}.$$

Lemma 3.4. *Assume that Assumptions 2.2, 3.1, and 2.1 hold, and that $f(\cdot, w)$ satisfies the following condition for all w*

$$\frac{\mu}{2}\|x - x'\|^2 \le f(x', w) - f(x, w) - \langle \nabla f(x, w), x' - x\rangle \le \frac{L_2}{2}\|x - x'\|^2, \quad \forall x, x' \in \mathcal{X}, \tag{9}$$

and that the projection step in (8) *can be computed efficiently, then with $\gamma = 1$ and $\theta = 1/2$, the following holds*

$$F(\overline{X}_{1:T}^{(l)}) - F^* \le A_l^{-1}(2L_2 D_V(X_{1:T}^*, X_{1:T}^{(0)}) + \sum_{l'=0}^{l} \overline{\Delta}^{(l')}),$$

where for $t = 1, \dots, T$ and $l = 0, 1 \dots, L$, $\Delta_t^{(l)} = \left(G_t^{(l)} - \frac{\partial}{\partial x_t} f(X_{1:T}^{(l)})\right) \mathsf{P}_t$, and denoting $A_{-1} = 0$,

$$\overline{\Delta}^{(l')} = \begin{cases} A_{l'} \frac{\|\Delta^{(l')}\|^2}{2L_2} + \langle \Delta^{(l')}, \alpha_{l'}(X_{1:T}^* - X_{1:T}^{(l')}) + A_{l'-1}(\overline{X}_{1:T}^{(l'-1)} - X_{1:T}^{(l')})\rangle & \mu = 0 \\ (\frac{\alpha_{l'}}{\mu} + \frac{A_{l'}}{2\mu} + \frac{A_{l'}}{2L_2})\|\Delta^{(l')}\|^2 & \mu > 0 \end{cases}.$$

Notice that for $\mu = 0$, θ does not affect the trajectory of (8), and so the bound in Lemma 3.4 holds under all θ. The proof of Lemma 3.4 is deferred to Appendix B.2.

Remark. [18] proves that the (accelerated) mirror descent stochastic approximation algorithms achieve the optimal convergence rate. Their algorithms can be applied to objective functions that also have a non-smooth component, and work for general norms, and their smoothness-strong-convexity condition is on the function $\mathbb{E}[f(x, w)]$. However, the strong convexity assumption in [18] is stronger: instead of $\frac{\mu}{2}\|x - x'\|^2$, the LHS of the first $\le$ in (9) is replaced by $\mu D_v(x', x)$. Since D_v is 1-strongly convex, $\mu D_v(x', x) \ge \frac{\mu}{2}\|x' - x\|^2$. In addition, their results hold only for *unbiased* gradient oracles, while we allow a bias of b_t for stage-t gradient oracles and explicate the dependency of the suboptimality on the bias.

3.3 Implementation from the scenario tree perspective

As discussed in Section 2.3, multi-stage problems can also be interpreted from the scenario tree perspective. For instance, suppose $X_t : \Omega \to \mathbb{R}^{n_t}$ is a random variable such that X_t is measurable w.r.t. $\mathcal{G}_t$, then instead of storing all K vectors $X_t(1), \dots, X_t(K)$, one can store the reduced variable $(X_t)_{\mathcal{G}_t} : \mathcal{G}_t \to \mathbb{R}^{n_t}$ where $(X_t)_{\mathcal{G}_t}([w]_t) = X_t(w)$ for all w (and we use the abbreviation $(X_t)_{\mathcal{G}_t}([w]_t) = X_t([w]_t)$). Thus, the total number of $\mathbb{R}^{n_t}$ vectors stored will be $|\Omega_t|$, the number of nodes in layer t of the scenario tree.

As a result, the (accelerated) mirror descent updates can be performed in this *reduced* space. Indeed, for (MS-Unconstrained), applying Lemma 3.2to the update in (3), we get that the update can be decomposed as

$$X_t^{(l+1)}([w]_t) = \operatorname*{argmin}_{x_t \in \mathcal{X}_t} \langle \gamma_l G_t^{(l)} \mathsf{P}_t([w]_t), x_t\rangle + D_{v_t}(x_t, X_t^{(l)}([w]_t)). \tag{10}$$

Similarly, for the saddle point problem (MS-Saddle),

$$Z_t^{(l+1)}([w]_t) = \operatorname*{argmin}_{z_t \in \mathcal{Z}_t} \langle \gamma_l G_t^{(l)} \mathsf{P}_t([w]_t), z_t\rangle + D_{w_t}(z_t, Z_t^{(l)}([w]_t)). \tag{11}$$

For the accelerated mirror descent algorithm, first notice that instead of storing the entire sequence of updates $X_{1:T}^{(0:l)}$ and $G_{1:T}^{(0:l)}$, we only need to store the cumulative gradients $\overline{G}_{1:T}^{(l)}$ defined below

$$\overline{G}_{1:T}^{(l)} = \sum_{l'=0}^{l} \alpha_{l'}(G_{1:T}^{(l')} - (1 - \theta)\mu X_{1:T}^{(l')}) - (1 + \gamma)L_2 \nabla v(X_{1:T}^{(0)}). \tag{12}$$

By a similar argument as in Lemma 3.1, the update for $\underline{X}_{1:T}^{(l)}$ can be decomposed stage-wise.

Thus, in the scenario tree interpretation, the updates (8) are equivalent to

$$\begin{aligned}
\overline{X}_t^{(l)}([w]_t) &= \operatorname*{argmin}_{x_t \in \mathcal{X}_t} \langle G_t^{(l)} \mathsf{P}_t([w]_t), x_t \rangle + \frac{(1+\gamma)L_2}{2} \|x_t - X_t^{(l)}([w]_t)\|^2 \\
\overline{G}_t^{(l)}([w]_t) &= \overline{G}_t^{(l-1)}([w]_t) + \alpha_l (G_t^{(l)} \mathsf{P}_t([w]_t) - (1-\theta)\mu X_t^{(l)}([w]_t)) \\
\underline{X}_t^{(l)}([w]_t) &= \operatorname*{argmin}_{x_t \in \mathcal{X}_t} (1+\gamma) L_2 v_t(x_t) + \frac{A_l(1-\theta)\mu}{2} \|x_t\|^2 + \langle \overline{G}_t^{(l)}([w]_t), x_t \rangle \\
X_t^{(l+1)}([w]_t) &= \tau_l \underline{X}_t^{(l)}([w]_t) + (1-\tau_l)\overline{X}_t^{(l)}([w]_t)
\end{aligned} \tag{13}$$

4 Mirror descent stochastic approximation

To implement the mirror descent algorithms, in addition to the updates (10), (11), or (13), one still needs to decide what $G_t^{(l)}$ is. Two factors should be taken into consideration. First, Lemmas 3.2, 3.3, and 3.4 suggest that the suboptimalities (in terms of the function values or minimax gaps) of the outputs depend on $\Delta_t^{(l)}$, which for (MS-Unconstrained) is

$$\Delta_t^{(l)} = G_t^{(l)} \mathsf{P}_t - \frac{\partial}{\partial x_t} f(X_{1:T}^{(l)}) \mathsf{P}_t.$$

Thus, one might hope to design $G_t^{(l)}$ such that $\Delta_t^{(l)} = 0$. Indeed, simply setting $G_t^{(l)} = \frac{\partial}{\partial x_t} f(X_{1:T}^{(l)})$ achieves this. However, the second factor is that given $G_t^{(l)}$, during the actual mirror descent updates (e.g. 10), one needs to compute $G_t^{(l)} \mathsf{P}_t$, the projection of $G_t^{(l)}$ to the non-anticipativity subspace corresponding to $\mathcal{G}_t$. Since this projection using P_t requires taking the conditional expectation of $G_t^{(l)}$, which might be computationally inefficient [31], one might hope to design $G_t^{(l)}$ which is measurable w.r.t. $\mathcal{G}_t$, and then the projection using P_t can be omitted.

Taking these two factors into consideration, it appears that $G_t^{(l)} \approx \frac{\partial}{\partial x_t} f(X_{1:T}^{(l)}) \mathsf{P}_t$ is a desirable choice. That is, one needs to estimate the *conditional expectation* of the gradient $\mathbb{E}[\frac{\partial}{\partial x_t} f | \mathcal{G}_t]$. Indeed, the decision X_t is made based on information contained in $\mathcal{G}_t$, and it makes sense to use the best approximation of the gradient subject to this information constraint, i.e. the gradient conditioned on $\mathcal{G}_t$.

In Section 4.1, we formally propose the *stochastic conditional gradient oracle*, the multi-stage counter part of stochastic gradient oracle. In Section 4.2, we discuss how to construct these stochastic gradients. In Section 4.3, we analyze mirror descent stochastic approximation algorithms with these stochastic conditional gradients.

4.1 Multi-stage equivalent of stochastic gradients

Recall that $(\Omega, \mathcal{F}, \mathbb{P})$ is the underlying probability space of the multi-stage problem. To model the randomness in (one sample of) the stochastic gradient (for stage t variables), we use the probability space $(\Xi_t, \mathcal{H}_t, \mu_t)$. Thus, the joint space $(\Omega, \mathcal{F}, \mathbb{P}) \otimes ((\Xi_t, \mathcal{H}_t, \mu_t)^{\otimes l})$ represents l i.i.d. stochastic gradients, independent of $(\Omega, \mathcal{F}, \mathbb{P})$. With some abuse of notation, we denote $\mathbb{E}[\cdot] = \mathbb{E}[\cdot | \mathcal{H}_t^{(\otimes l)}]$, i.e. the expectation w.r.t. $(\Omega, \mathcal{F}, \mathbb{P})$, and $\tilde{\mathbb{E}}[\cdot] = \mathbb{E}[\cdot | \mathcal{F}]$, i.e. the expectation w.r.t. the randomness in l independent stochastic gradients. Similarly for $\mathbb{P}$ and $\tilde{\mathbb{P}}$.

Definition 4.1. *For the problem* (MS-Unconstrained) *and $b_t, \sigma_t \geq 0$, we say $\mathcal{O}_t : \overline{\mathcal{X}} \times \Xi_t \to \mathbb{R}^{n_t \times K}$ is a (b_t, σ_t)-stochastic conditional gradient oracle for X_t if for any $(X_1, X_2, \ldots, X_T) \in \overline{\mathcal{X}}$ and $\xi_t \in \Xi_t$, $\mathcal{O}_t(X_{1:T}, \xi_t)$ is a random vector in $\mathbb{R}^{n_t}$ satisfying the following three conditions:*

$$\mathcal{O}_t(X_{1:T}, \xi_t) \in \mathcal{G}_t, \quad \forall \xi_t \in \Xi_t \tag{14a}$$

$$\left\| \tilde{\mathbb{E}}\left[\mathcal{O}_t(X_{1:T}, \xi_t)(w)\right] - \left(\frac{\partial}{\partial x_t} f(X_{1:T})\mathsf{P}_t\right)(w) \right\|_* \leq b_t, \quad \forall w \in \Omega, \tag{14b}$$

$$\tilde{\mathbb{E}}[\|\mathcal{O}_t(X_{1:T}, \xi_t)(w) - \tilde{\mathbb{E}}[\mathcal{O}_t(X_{1:T}, \xi_t')(w)]\|_*^2] \leq \sigma_t^2, \quad \forall w \in \Omega. \tag{14c}$$

In addition, for some $\overline{\sigma}_t > 0$, we say $\mathcal{O}_t$ is $\overline{\sigma}_t$-concentrated if

$$\tilde{\mathbb{E}}\left[\exp\left(\left\|\mathcal{O}_t(X_{1:T}, \xi_t)(w) - \tilde{\mathbb{E}}\left[\mathcal{O}_t\left(X_{1:T}, \xi_t'\right)(w)\right]\right\|_*^2 / \overline{\sigma}_t^2\right)\right] \leq \exp(1), \quad \forall w \in \Omega. \tag{15}$$

Stochastic conditional gradient oracles for Z_t for the saddle point problem (MS-Saddle) can be defined similarly, with $\overline{\mathcal{X}}$ replaced by $\overline{\mathcal{Z}}$, n_t replaced by n_t+m_t, X_t replaced by Z_t, and $\frac{\partial}{\partial x_t} f(X_{1:T})\mathsf{P}_t$ replaced by $\frac{\tilde{\partial}}{\partial z_t}\phi(Z_{1:T})\mathsf{P}_t$.

Remark. Due to the measurability conditions (14a), in the scenario tree representation,

$$\mathcal{O}_t(X_{1:T}, \xi_t)([w]_t) = \mathcal{O}_t(X_{1:T}, \xi_t)(w), \quad \forall w \in \Omega.$$

Thus, instead of constructing/storing K vectors in $\mathbb{R}^{n_t}$, only $|\Omega_{\mathcal{G}_t}|$ vectors are needed.

Remark. Notice that the same ξ_t can be used for all scenarios $w \in \Omega$. As a result, $\mathcal{O}_t(X_{1:T}, \xi_t)(w)$ and $\mathcal{O}_t(X_{1:T}, \xi_t)(w')$ could be correlated. However, our convergence results in Section 4.3 do not require that the stochastic conditional gradients are independent *across scenarios*.

4.2 Constructing conditional stochastic gradients

In classical stochastic programming problems where the objective function is $F(x) = \mathbb{E}[f(x, w)]$, the canonical approach to construct a stochastic gradient is to take the sampling space to be $(\Xi, \mathcal{H}, \mu) = (\Omega, \mathcal{F}, \mathbb{P})$, the probability space corresponding to the stochastic programming problem itself. Then, $\mathcal{O}(x, \xi) := \nabla f(x, \xi)$ for $\xi \in \Xi = \Omega$ is an unbiased stochastic gradient oracle.

For the multi-stage setting, the stochastic gradient is a random vector living in dimension $\mathbb{R}^{n_t \times K}$ which satisfies the measurability condition ((14a)) and the moment conditions ((14b) and (14c)). Below in Lemma 4.1, we propose one approach to construct the above-mentioned stochastic conditional gradient.

For the probability space, we set $\Xi_t = [0, 1]$, $\mathcal{H}_t = \mathcal{B}([0, 1])$ the Borel set for $[0, 1]$ and μ_t is the uniform distribution on $[0, 1]$. Thus, ξ_t is a random variable uniformly distributed in $[0, 1]$. Next, for any finite set $\tilde{\Omega}$, we define a sampling function $R_{\tilde{\Omega}} : [0, 1] \times \Delta(\tilde{\Omega}) \to \tilde{\Omega}$ where $\Delta(\tilde{\Omega})$ denotes all probability distributions over $\tilde{\Omega}$:

$$R_{\tilde{\Omega}}(r, \nu) = \min\{k \in \tilde{\Omega}, \quad \sum_{k'=1}^{k} \nu_{k'} \geq r\}. \tag{16}$$

That is, $[0, 1]$ is divided into intervals of length $\nu_1, \ldots, \nu_{|\tilde{\Omega}|}$ and R returns the index of the interval r lies in. In particular, $R(\xi_t, \nu)$ has distribution ν.

In addition, recall that $\pi_t([w]_t)$ is the distribution over the child nodes of $[w]_t$, i.e. $\Omega_{t+1}([w]_t) = \{[w']_{t+1},\ w' \in [w]_t\}$. For two distributions μ, ν defined over $[K_0]$, recall that their total variation distance is denoted as $TV(\mu, \nu) = \frac{1}{2}\sum_{k=1}^{K_0} |\mu(k) - \nu(k)|$.

Lemma 4.1. *For $t \le T-1$ and $\|\frac{\partial}{\partial x_t} f(x,w)\|_* \le L_{1,t}$ for all $x \in \mathcal{X}$ and $w \in \Omega$. With the deterministic function $R_{\tilde{\Omega}}(\cdot,\cdot)$ defined in (16) and the space $(\Xi_t, \mathcal{H}_t, \mu_t)$ defined above, for the objective function* (MS-Unconstrained)*, first, sample a child node of $[w]_t$ using distribution $\hat{\pi}_t([w]_t)$*

$$[w_c]_{t+1} = R_{\Omega_{t+1}([w]_t)}(\xi_t, \hat{\pi}_t([w]_t)).$$

Then the conditional gradient is evaluated along the path $[w_c]_{t+1}$. That is, denoting $x_{t-1} = X_{t-1}([w]_{t-1})$, $x_t = X_t([w]_t)$, and $x_{t+1} = X_{t+1}([w_c]_{t+1})$

$$\mathcal{O}_t(X_{1:T}, \xi_t)([w]_t) = \frac{\partial}{\partial x_t} f_t(x_{t-1}, x_t, [w]_t) + \frac{\partial}{\partial x_t} f_{t+1}(x_t, x_{t+1}, [w_c]_{t+1}). \tag{17}$$

The above constructed $\mathcal{O}_t(X_{1:T})$ is a (b_t, σ_t)-stochastic conditional gradient oracle with

$$b_t = \max_{[w]_t \in \Omega_t} 2L_{1,t} \cdot TV(\hat{\pi}([w]_t, \pi([w]_t)),\ \sigma_t^2 = \sup_{X_{1:T} \in \overline{\mathcal{X}}} \max_{[w]_t \in \Omega_t} \sigma_t^2(X_{1:T}, [w]_t)$$

where $[w_c]_{t+1} \sim \hat{\pi}_t([w]_t)$ and

$$\sigma_t^2(X_{1:T}, [w]_t) = \mathbb{E}\left[\|\frac{\partial f_{t+1}}{\partial x_t}(X_t([w]_t), X_{t+1}([w_c]_{t+1}), [w_c]_{t+1}) - g([w]_t)\|_*^2\right]$$
$$g([w]_t) = \mathbb{E}[\frac{\partial f_{t+1}}{\partial x_t}(X_t([w]_t), X_{t+1}([w_c]_{t+1}), [w_c]_{t+1})].$$

It's $\overline{\sigma}_t$-concentrated for

$$\overline{\sigma}_t^2 = \sup_{X_{1:T} \in \overline{\mathcal{X}}} \max_{w \in \Omega} \|\frac{\partial f_{t+1}}{\partial x_t}(X_t([w]_t), X_{t+1}([w]_{t+1}), [w]_{\mathcal{F}_{t+1}}) - g([w]_t)\|_*^2.$$

Similar constructions and results hold for (MS-Saddle).

Proof of Lemma 4.1. The measurability assumption holds since for any ξ_t, the output (17) is the same for $w, w' \in \Omega$ if $[w]_t = [w']_t$. For the bias and variance, notice that for each fixed $[w]_t$, we have $x_{t-1}, x_t, [w]_t$ fixed, and so the first term in the RHS of (17) is fixed. The only randomness comes from the second term. In addition, $[w_c]_{t+1}$ follows the distribution $\hat{\pi}_t([w]_t)$. Thus, the bias condition follows from the assumption that $\|\frac{\partial}{\partial x_t} f\|_* \le L_{1,t}$, and the second moment condition (14c) and the concentration condition (15) hold with the given σ_t and $\overline{\sigma}_t$. □

Remark. For $t = T$, $\frac{\partial f}{\partial x_T}(X_{1:T}(w), w) = \frac{\partial f_T}{\partial x_T}(X_{T-1}(w), X_T(w), w)$. Since $X_{T-1} \in \mathcal{G}_{T-1} \subset \mathcal{G}_T$ and $f_T(x_{t-1}, x_t, \cdot)$ is $\mathcal{F}_t$ measurable for any x_{t-1}, x_t, we have

$$\frac{\partial f_T}{\partial x_T}(X_{T-1}(w), X_T(w), w) = \frac{\partial f_T}{\partial x_T}(X_{T-1}([w]_{T-1}), X_T([w]_T), [w]_T).$$

Thus, the conditional gradient can be calculated exactly without any sampling.

Remark. Suppose $\hat{\pi}_t([w]_t) = \pi_t([w]_t)$, and so the child node is sampled according to the true distribution over the child nodes, then $b_t = 0$. The above Lemma 4.1 implies that the stochastic conditional gradient oracle is robust to model misspecification. Together with the convergence results for the stochastic approximation type of algorithms below (e.g. Theorem 4.1), where the suboptimalities depend linearly on b_t, we see that the overall mirror descent stochastic approximation algorithms with the above sampling mechanism are also robust to model misspecification.

Remark. In Definition 4.1, we assume that $b_t, \sigma_t, \overline{\sigma}_t$ are constants that do not depend on the query point $X_{1:T}$. In fact, it's not hard to generalize the definitions such that $b_t, \sigma_t, \overline{\sigma}_t$ also depend on query points. This allows for a more refined control of the suboptimality in Theorems 4.1, 4.2, and 4.3 presented in the next section.

4.3 Performance of mirror descent stochastic approximation

In this section, we analyze the convergence properties of the aforementioned mirror descent updates, where $G_t^{(l)}$ is the output of $\mathcal{O}_t$, a stochastic conditional gradient oracle not necessarily the one constructed in Lemma 4.1. Below, we assume that $\xi_{1:T}^{(0:(L-1))}$ is sampled from $\otimes_{t=1}^T(\Xi_t, \mathcal{H}_t, \mu_t)^{\otimes L}$.

Theorem 4.1. *Assume that Assumptions 2.1, 2.2 and 3.1 hold. Assume that $\|\frac{\partial}{\partial x}f(x,w)\|_* \le L_1$ for all $x \in \mathcal{X}$ and $w \in \Omega$, $\|x - x'\| \le D$ for all $x, x' \in \mathcal{X}$, $D_V(X_{1:T}^*, X_{1:T}^{(0)}) \le \tilde{D}^2$, and $\mathcal{O}_t$ is a (b_t, σ_t)-stochastic conditional gradient oracle for some constants $b_t, \sigma_t \ge 0$ for all t, and denote $b^2 = \sum_{t=1}^T b_t^2$, $\sigma^2 = \sum_{t=1}^T \sigma_t^2$. Consider the update (3) with $G_t^{(l)} = \mathcal{O}_t(X_{1:T}^{(l)}, \xi_t^{(l)})$.*

With $\gamma_l = \sqrt{\frac{\tilde{D}^2}{(L+1)(L_1^2+2\sigma^2+2b^2)}}$,

$$\tilde{\mathbb{E}}[\mathbb{E}[f(\overline{X}_{1:T}^{(L)})]] - \mathbb{E}[f(X_{1:T}^*)] \le 2\sqrt{\frac{\tilde{D}^2(2\sigma^2 + 2b^2 + L_1^2)}{L+1}} + bD.$$

If $\mathcal{O}_t$ is also $\overline{\sigma}_t$-concentrated for all t, and $\overline{\sigma}^2 = \sum_{t=1}^T \overline{\sigma}_t^2$, then with $\gamma_l = \gamma = \sqrt{\frac{\tilde{D}^2}{(L+1)(2(1+\lambda)\overline{\sigma}^2+2b^2+L_1^2)}}$, we get

$$\tilde{\mathbb{P}}[\mathbb{E}[f(\overline{X}_{1:T}^{(L)})] - \mathbb{E}[f(X_{1:T}^*)] \ge \Lambda] \le 2\exp(-\lambda),$$

where

$$\Lambda = 2\sqrt{\frac{\tilde{D}^2(2(1+\lambda)\overline{\sigma}^2 + 2b^2 + L_1^2)}{L+1}} + 2\gamma\sqrt{\lambda(L+1)} \cdot \overline{\sigma}D + bD.$$

Proof of Theorem 4.1. For convenience, we denote

$$\delta_t^{(l)}(w) = \mathcal{O}_t(X_{1:T}^{(l)}, \xi_t^{(l)})(w) - \tilde{\mathbb{E}}[\mathcal{O}_t(X_{1:T}^{(l)}, \xi_t^{(l)})(w)].$$

Since $(a+b)^2 \le 2(a^2+b^2)$, $\|\frac{\partial}{\partial x}f(x)\|_* \le L_1$, and (14b),

$$\begin{aligned}
\sum_{t=1}^T \|G_t^{(l)}\mathsf{P}_t(w)\|_*^2 &\le 2\sum_{t=1}^T \|\mathcal{O}_t(X_{1:T}^{(l)}, \xi_t^{(l)})(w) - \frac{\partial}{\partial x_t}f(X_{1:T}^{(l)}(w))\|_*^2 + 2L_1^2 \\
&\le 4\sum_{t=1}^T \|\mathcal{O}_t(X_{1:T}^{(l)}, \xi_t^{(l)})(w) - \tilde{\mathbb{E}}[\mathcal{O}_t(X_{1:T}^{(l)}, \xi_t)(w)]\|_*^2 + 4b^2 + 2L_1^2 \\
&= 4\sum_{t=1}^T \|\delta_t^{(l)}(w)\|_*^2 + 4b^2 + 2L_1^2. \qquad (18)
\end{aligned}$$

For the in expectation result, using condition (14c), we get

$$\sum_{t=1}^T \tilde{\mathbb{E}}[\|G_t^{(l)}\mathsf{P}_t(w)\|_*^2] \le 4\sigma^2 + 4b^2 + 2L_1^2.$$

In addition, using (14b) condition again, we get

$$\begin{aligned}
&|\tilde{\mathbb{E}}[\langle \Delta_{1:T}^{(l)}(w), X_{1:T}^{(l)}(w) - X_{1:T}^*(w)\rangle | \xi_{1:T}^{(0:(l-1))}]| \\
&\le |\langle \tilde{\mathbb{E}}[\delta_{1:T}^{(l)}(w) | \xi_{1:T}^{(0:(l-1))}], X_{1:T}^{(l)}(w) - X_{1:T}^*(w)\rangle| + bD = bD.
\end{aligned}$$

Thus, from Lemma 3.2, we have

$$\tilde{\mathbb{E}}[\mathbb{E}[f(\overline{X}_{1:T}^{(L)})]] - \mathbb{E}[f(X_{1:T}^*)] \leq \frac{\tilde{D}^2 + \sum_{l=0}^{L} \gamma_l^2 (2\sigma^2 + 2b^2 + L_1^2)}{\sum_{l=0}^{L} \gamma_l} + bD,$$

which gives the first claim.

For the high-probability result, first, from (18) we have

$$\sum_{l=0}^{L} \gamma_l^2 \sum_{t=1}^{T} \|G_t^{(l)} \mathsf{P}_t\|_*^2 \leq \sum_{l=0}^{L} \gamma_l^2 (4b^2 + 2L_1^2) + 4 \sum_{l=0}^{L} \gamma_l^2 \|\delta_{1:T}^{(l)}\|_*^2.$$

By the condition (15) and together with Jensen's inequality (applied to $\exp(\cdot)$), we have

$$\exp\left(\|\delta_t^{(l)}\|_*^2 / \overline{\sigma}_t^2\right) = \exp\left(\left(\sum_{w \in \Omega} p_w \|\delta_t^{(l)}(w)\|_*^2\right) / \overline{\sigma}_t^2\right) \leq \sum_{w \in \Omega} p_w \exp\left(\|\delta_t^{(l)}(w)\|_*^2 / \overline{\sigma}_t^2\right).$$

Thus, we get

$$\begin{aligned}
\tilde{\mathbb{E}}\left[\exp\left(\|\delta_{1:T}^{(l)}\|_*^2 / \overline{\sigma}^2\right) | \xi_{1:T}^{0:(l-1)}\right] &= \tilde{\mathbb{E}}\left[\exp\left(\sum_{t=1}^{T} \frac{\overline{\sigma}_t^2}{\overline{\sigma}^2} \cdot \|\delta_t^{(l)}\|_*^2 / \overline{\sigma}_t^2\right) | \xi_{1:T}^{0:(l-1)}\right] \\
&\leq \sum_{t=1}^{T} \frac{\overline{\sigma}_t^2}{\overline{\sigma}^2} \cdot \tilde{\mathbb{E}}\left[\exp\left(\|\delta_t^{(l)}\|_*^2 / \overline{\sigma}_t^2\right) | \xi_{1:T}^{0:(l-1)}\right] \leq \exp(1). \qquad (19)
\end{aligned}$$

Using Jensen's inequality (applied to $\exp(\cdot)$) again, we get

$$\tilde{\mathbb{E}}\left[\exp\left(\frac{\sum_{l=0}^{L} \gamma_l^2 \|\delta_{1:T}^{(l)}\|_*^2}{\overline{\sigma}^2 \cdot \sum_{l'=0}^{L} \gamma_{l'}^2}\right)\right] \leq \exp(1).$$

Thus, we have for any $\lambda > 0$,

$$\tilde{\mathbb{P}}[\sum_{l=0}^{L} \gamma_l^2 \|\delta_{1:T}^{(l)}\|_*^2 \geq (1+\lambda)(\sum_{l'=0}^{L} \gamma_{l'}^2) \cdot \overline{\sigma}^2] \leq \exp(-\lambda). \qquad (20)$$

In addition, by condition 14b, we have

$$\sum_{l=0}^{L} \gamma_l \langle \Delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}^* \rangle \geq -\sum_{l=0}^{L} \gamma_l \cdot bD + \sum_{l=0}^{L} \gamma_l \cdot \langle \delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}^* \rangle.$$

Since $\|X_{1:T}^{(l)} - X_{1:T}^*\|^2 \leq D^2$, by Cauchy-Schwarz inequality, we get

$$|\langle \delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}^* \rangle| \leq \sum_{t=1}^{T} \|\delta_t^{(l)}\|_* \cdot \|X_t^{(l)} - X_t^*\| \leq \|\delta_{1:T}^{(l)}\|_* \cdot D.$$

Using (19), we get

$$\tilde{\mathbb{E}}\left[\exp\left(\langle \delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}^* \rangle^2 / (\overline{\sigma} D)^2\right) | \xi_{1:T}^{0:(l-1)}\right] \leq \tilde{\mathbb{E}}\left[\exp\left(\|\delta_{1:T}^{(l)}\|_*^2 / \overline{\sigma}^2\right) | \xi_{1:T}^{0:(l-1)}\right] \leq 1.$$

A similar argument as in the proof of Proposition 3.2 in [31] gives that for any $\lambda > 0$,

$$\tilde{\mathbb{E}}[\exp(-\lambda \sum_{l=0}^{L} \gamma_l \cdot \langle \delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}^* \rangle)] \leq \exp(\lambda^2 (\sum_{l=0}^{L} \gamma_l^2) \cdot \overline{\sigma}^2 \cdot D^2).$$

Thus for we have

$$\tilde{\mathbb{P}}[-\sum_{l=0}^{L} \gamma_l \langle \delta_{1:T}^{(l)}, X_{1:T}^{(l)} - X_{1:T}^* \rangle \geq \lambda \overline{\sigma} D \sqrt{\sum_{l=0}^{L} \gamma_l^2}] \leq \exp(-\lambda^2/4). \tag{21}$$

Combining (20) and (21) with Lemma 3.2, we have

$$\tilde{\mathbb{P}}[\mathbb{E}[f(\overline{X}_{1:T}^{(L)})] - \mathbb{E}[f(X_{1:T}^*)] \geq \Lambda_0] \leq \exp(-\lambda_1) + \exp(-\lambda_2^2/4),$$

where

$$\Lambda_0 = \frac{\tilde{D}^2 + \sum_{l=0}^{L} \gamma_l^2 (2(1+\lambda_1)\overline{\sigma}^2 + 2b^2 + L_1^2) + \lambda_2 \overline{\sigma} D \sqrt{\sum_{l=0}^{L} \gamma_l^2}}{\sum_{l=0}^{L} \gamma_l} + bD.$$

The result follows from taking $\lambda = \lambda_1 = \lambda_2^2/4$. □

Remark. As a special case, suppose $\tilde{D} = \tilde{D}_0 \sqrt{T}$, $D = D_0 \sqrt{T}$, $L_1 = L_{1,0}\sqrt{T}$, $\sigma_t = \sigma_0$ and $b_t = b_0$ for all t, then the suggested $\gamma_l = \sqrt{\frac{\tilde{D}_0^2}{(L+1)(L_{1,0}^2 + 2\sigma_0^2 + 2b_0^2)}}$,

$$\tilde{\mathbb{E}}[\mathbb{E}[f(\overline{X}_{1:T}^{(L)})]] - \mathbb{E}[f(X_{1:T}^*)] \leq 2T \sqrt{\frac{\tilde{D}_0^2 (2\sigma_0^2 + 2b_0^2 + L_{1,0}^2)}{L+1}} + b_0 D_0 T.$$

Thus, to get $T\epsilon$ suboptimality (in expectation), one needs to ensure that $b_0 D_0 = O(\epsilon)$ and set $L = O(1/\epsilon^2)$, which is independent of the number of stages T. In particular, γ_l does not depend on T, the total number of stages.

Theorem 4.2. *Assume that Assumptions 2.3, 2.4 and 3.2 hold. Assume that* $\|\frac{\tilde{\partial}}{\partial z}\phi(z,w)\|_* \leq L_1$ *for all* $z \in \mathcal{Z}$ *and* $w \in \Omega$, $\|z - z'\| \leq D$ *for all* $z, z' \in \mathcal{Z}$, $D_W(Z_{1:T}, Z_{1:T}^{(0)}) \leq \tilde{D}^2$ *for all* $Z_{1:T} \in \overline{Z}$, *and* $\mathcal{O}_t$ *is a* (b_t, σ_t)*-stochastic conditional gradient oracle for some constants* $b_t, \sigma_t \geq 0$ *for all* t, *and denote* $b^2 = \sum_{t=1}^{T} b_t^2$, $\sigma^2 = \sum_{t=1}^{T} \sigma_t^2$. *Consider the update* (3) *with* $G_t^{(l)} = \mathcal{O}_t(Z_{1:T}^{(l)}, \xi_t^{(l)})$.

With $\gamma_l = \sqrt{\frac{2\tilde{D}^2}{(L+1)(3\sigma^2 + 3b^2 + L_1^2)}}$, *for any* $Z_{1:T} \in \overline{\mathcal{Z}}$,

$$\tilde{\mathbb{E}}[\mathbb{E}[\phi(\overline{X}_{1:T}^{(L)}, Y_{1:T}) - \phi(X_{1:T}, \overline{Y}_{1:T}^{(L)})]] \leq 2\sqrt{\frac{\tilde{D}^2(6\sigma^2 + 6b^2 + 2L_1^2)}{L+1}} + bD.$$

If $\mathcal{O}_t$ *is also* $\overline{\sigma}_t$*-concentrated for all* t, *and* $\overline{\sigma}^2 = \sum_{t=1}^{T} \overline{\sigma}_t^2$, *then with* $\gamma_l = \gamma = \sqrt{\frac{2\tilde{D}^2}{(L+1)(3(1+\lambda)\overline{\sigma}^2 + 3b^2 + L_1^2)}}$, *we get*

$$\tilde{\mathbb{P}}[\mathbb{E}[\phi(\overline{X}_{1:T}^{(L)}, Y_{1:T}) - \phi(X_{1:T}, \overline{Y}_{1:T}^{(L)})] < \Lambda, \ \forall Z \in \overline{\mathcal{Z}}] \geq 1 - 2\exp(-\lambda),$$

where

$$\Lambda = 2\sqrt{\frac{\tilde{D}^2(6(1+\lambda)\overline{\sigma}^2 + 6b^2 + 2L_1^2)}{L+1}} + 2\gamma\sqrt{\lambda(L+1)} \cdot \overline{\sigma} D + bD.$$

The proof of Theorem 4.2 is similar to the proof of Theorem 4.1 and uses Lemma 3.3. We omit it due to space constraints.

For the accelerated mirror descent (8), a similar argument as above applied to Lemma 3.4 gives the following results.

Theorem 4.3. *Assume that Assumptions 2.1, 2.2, and 3.1 hold, that $f(\cdot, w)$ satisfies (9) for all w, and that the projection step in (8) can be computed efficiently. Further assume that for all t, $D_V(X^*_{1:T}, X^{(0)}_{1:T}) \le \tilde{D}^2$, and $\mathcal{O}_t$ is a (b_t, σ_t)-stochastic conditional gradient oracle for some constants $b_t, \sigma_t \ge 0$, and denote $b^2 = \sum_{t=1}^T b_t^2$, $\sigma^2 = \sum_{t=1}^T \sigma_t^2$. Consider the update (8) with $G_t^{(l)} = \mathcal{O}_t(X^{(l)}_{1:T}, \xi_t^{(l)})$ and $\gamma = 1$, $\theta = 1/2$.*

If $\mu = 0$, $\|\frac{\partial}{\partial x} f(x, w)\|_* \le L_1$ for all $x \in \mathcal{X}$ and $w \in \Omega$, $\|x - x'\| \le D$ for all $x, x' \in \mathcal{X}$. Then we have

$$\tilde{\mathbb{E}}[\mathbb{E}[f(\overline{X}^{(l)}_{1:T})]] - \mathbb{E}[f(X^*_{1:T})] \le \frac{8L_2\tilde{D}^2}{(l+1)(l+2)} + \frac{2}{3}(l+3)(\frac{b^2+\sigma^2}{2L_2} + bD).$$

If $\mu > 0$, then for $\rho = (1 + \frac{1}{4}\sqrt{\frac{\mu}{L_2}})^{-2} \le 1 - \frac{3}{16}\sqrt{\frac{\mu}{L_2}}$,

$$\tilde{\mathbb{E}}[\mathbb{E}[f(\overline{X}^{(l)}_{1:T})]] - \mathbb{E}[f(X^*_{1:T})] \le \rho^l \cdot 2L_2\tilde{D}^2 + (b^2+\sigma^2) \cdot (\frac{3}{2\mu} + \frac{1}{2L_2}) \cdot \frac{1}{1-\rho}.$$

Proof of Theorem 4.3. For the first claim, recall that by Lemma 3.4

$$\overline{\Delta}^{(l')} = A_{l'} \frac{\|\Delta^{(l')}\|^2}{2L_2} + \langle \Delta^{(l')}, \alpha_{l'}(X^*_{1:T} - X^{(l')}_{1:T}) + A_{l'-1}(\overline{X}^{(l'-1)}_{1:T} - X^{(l')}_{1:T})\rangle.$$

Thus, since $\mathcal{O}_t$ is a (b_t, σ_t)-stochastic conditional gradient oracle,

$$\tilde{\mathbb{E}}[\overline{\Delta}^{(l')} | \xi^{0:(l'-1)}_{1:T}] \le \frac{A_{l'}}{2L_2} \sum_{t=1}^T (b_t^2 + \sigma_t^2) + A_{l'} bD.$$

The result follows from $\frac{(l+1)(l+2)}{4} \le A_l \le \frac{(l+1)(l+2)}{2}$ for all l (Lemma B.5) and Lemma 3.4.

For the second claim, by Lemma 3.4, with $\gamma = 1, \mu > 0, \theta = 1/2$,

$$(\frac{\alpha_{l'}}{\mu} + \frac{A_{l'}}{2\mu} + \frac{A_{l'}}{2L_2})^{-1} \tilde{\mathbb{E}}[\overline{\Delta}^{(l')} | \xi^{(0:(l'-1)}_{1:T}] = \tilde{\mathbb{E}}[\|\Delta^{(l')}\|^2 | \xi^{(0:(l'-1))}_{1:T}] \le \sum_{t=1}^T (b_t^2 + \sigma_t^2).$$

By convexity of $s \to (1+s)^{-2}$ on $[0, \infty)$, we have $(1+s)^{-2} \le 1 - 3s/4$ for all $0 \le s \le 1$, and so $\rho = (1 + \frac{1}{4}\sqrt{\frac{\mu}{L_2}})^{-2} \le 1 - \frac{3}{16}\sqrt{\frac{\mu}{L_2}}$. The result follows from Lemma 3.4 and Lemma B.5: we have $A_l \ge (1 + \frac{1}{4}\sqrt{\frac{\mu}{L_2}})^{2l} = \rho^{-l}$ and

$$\sum_{l'=0}^l \frac{A_{l'}}{A_l} \le \sum_{l'=0}^l \rho^{l-l'} \le \sum_{l'=0}^\infty \rho^{l'} \le \frac{1}{1-\rho}.$$

□

Following a similar argument as the proof in Theorem 4.1, it's easy to show that the updates (8) also converge with high probability.

5 Efficient online implementation

The mirror descent stochastic approximation algorithms presented in Section 4.3 converge to optimal solutions under mild assumptions on the problems and the stochastic conditional gradient oracles. The output, after running L iterations, is a set of T random variables (for (MS-Unconstrained)): $X_1, X_2, \ldots, X_T$ where $X_t \in \mathcal{G}_t$ satisfies the information constraint for all t. However, even for scenario trees where each non-leaf node has only 2 children, the "effective" number of variables in the last stage X_t is 2^{T-1}, and so even for constant L, the oracle complexity (the total number of calls of $\mathcal{O}_t$) is exponential in T.

This difficulty motivates us to consider a *semi-online* setting, where both the input and output are sequential. That is, suppose that the (true) scenario is $w^* \in \Omega$, then at stage $t = 1, \ldots, T$,

- sequential input: the decision maker is given $[w^*]_t$, the node in the t-th layer of the scenario tree where $w^* \in [w^*]_t$;
- sequential output: the decision maker needs to decide $X_{t+s}([w]_{t+s})$ for all $[w]_{t+s}$ consistent with $[w^*]_t$, i.e., for nodes in layer $t+s$ that are in the subtree rooted at $[w^*]_t$. (If $t = 1$, $\{X_{t'}([w]_{t'}),\ [w]_{t'} \subset [w^*]_1\}$ for $t' = 1, \ldots, s+1$ are needed.)

Above, $s \in \{0, 1, \ldots, T\}$ represents the number of stages that X_t needs to be made in advance. As special cases, $s = T$ is the classical offline setting where all X_t's have to be made at the beginning, while $s = 0$ is the online setting, where X_t only needs to be made at stage t. $s \in \{0, 1, \ldots, T\}$ interpolates between these two cases.

With this semi-online setup, the sequentially revealed information narrows down the probability space and helps reduce the number of (effective) decision variables. However, even if $s = 0$ and one only wants an approximately optimal *first stage decision* X_1, Dynamic Stochastic Approximation ([23]), the best known stochastic approximation algorithm for multi-stage stochastic programming problems, has complexity which is exponential in T. This is due to the *nested* iterations when computing approximate subgradients of the cost-to-go functions.

In fact, the *naive* implementation of our (accelerated) gradient descents, which computes all of $\overline{X}_{1:T}^{(L)}$ before stage 1, also suffers from this inefficiency. In the following, we propose an approach which can take advantage of the sequentially revealed information to reduce the complexity. On a high level, due to the decomposability of the updates across stages and scenarios, these updates can be implemented asynchronously. Importantly, updates can be delayed. It is the information gained during the delay that allows us to *early stop* updating $X_t(w)$ for w that is inconsistent with the current available information.

5.1 A state perspective

Before presenting our asynchronous updates, we first make the following reformulation. For each node $[w]_t$ in the scenario tree, we associate it with a state vector $S_t \in \mathcal{S}_t \subset \mathbb{R}^{\tilde{n}_t}$, consisting of the decision variables and any auxiliary information such as momentum and/or ergodic mean. We use $\psi_{\mathsf{query}}(S_t)$ and $\psi_{\mathsf{output}}(S_t)$ to denote the next query point and the output based on the state S_t.

In addition, we associate each node with an update function $\mathcal{A}_t^{(l)} : \mathcal{S}_t \times \mathbb{R}^{n_t} \to \mathcal{S}_t$. Assuming that the mirror descent algorithm is initialized with the states $S_t^{(0)}([w]_t)$ for all t and all $[w]_t$, then during the l-th iteration, the update is the following: for all t and all nodes $[w]_t$, the updated state for $[w]_t$ depends on its previous state, as well as the (stochastic) first-order information $G_t^{(l)}$:

$$S_t^{(l+1)}([w]_t) = \mathcal{A}_t^{(l)}(S_t^{(l)}([w]_t), G_t^{(l)}([w]_t)), \tag{22}$$

Suppose the stochastic oracle in Lemma 4.1 is used, then for $t \le T-1$, $G_t^{(l)}([w]_t)$ is constructed by sampling a child node $[w_c]_{t+1}$ based on the distribution over the children of $[w]_t$, and for $t = T$, there is no sampling:

$$G_t^{(l)}([w]_t) = \begin{cases} \frac{\partial}{\partial x_t} f_t(Q_1, Q_2, [w]_t) + \frac{\partial}{\partial x_t} f_{t+1}(Q_2, Q_3, [w_c]_{t+1}) & t \le T-1 \\ \frac{\partial}{\partial x_T} f_T(Q_1, Q_2, [w]_t) & t = T \end{cases}. \tag{23}$$

Above, Q_1, Q_2, Q_3 are defined as below:

$$Q_1 = \psi_{\mathsf{query}}(S_{t-1}^{(l)}([w]_{t-1})),\ Q_2 = \psi_{\mathsf{query}}(S_t^{(l)}([w]_t)), Q_3 = \psi_{\mathsf{query}}(S_{t+1}^{(l)}([w_c]_{t+1})).$$

Importantly, during the l-th update for the state variable $S_t([w]_t)$, all needed is the l-th iteration state of the following nodes: its parent node $[w]_{t-1}$, itself $[w]_t$, and the sampled child node $[w_c]_{t+1}$. Thus, information such as the l-th iteration state of $[w]_{t+2}$, i.e. its grandchild nodes, is not needed. Figure 2 gives an example of updating the root node for 3 iterations.

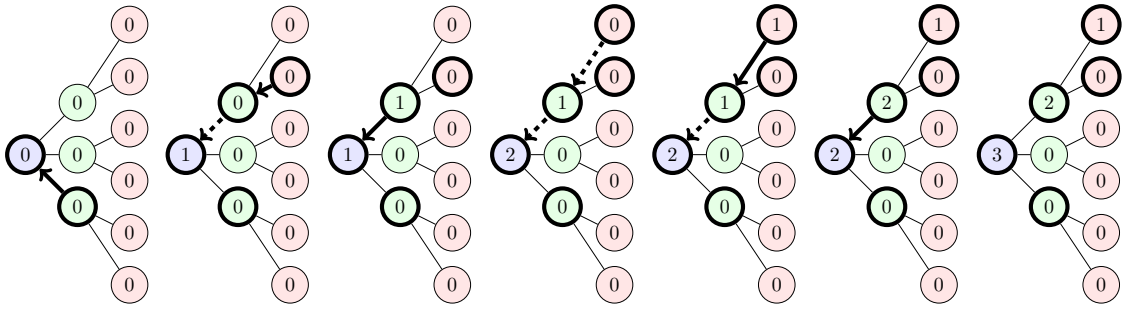


Figure 2: Denoting the nodes as $w_{1,1}$ for layer 1, $w_{2,1:3}$ for layer 2, and $w_{3,1:6}$ for layer 3 where $w_{2,1}$ and $w_{3,1}$ are the top ones. Solid arrows are immediate updates and dashed arrows are planned updates. The numbers represent the number of updates that has been applied to each node. Only nodes with bold boundaries are visited. In this example, the update order is $S_1^{(1)}(w_{1,1})$, $S_2^{(1)}(w_{2,1})$, $S_1^{(2)}(w_{1,1})$, $S_3^{(1)}(w_{3,1})$, $S_2^{(2)}(w_{2,1})$, $S_1^{(3)}(w_{1,1})$.

For the three mirror descent updates as presented in Section 4.3, we can choose the states and updates as follows.

- *Mirror descent update* (10). We can take $\mathcal{S}_t = \mathcal{X}_t \times \mathcal{X}_t$ and $S_t([w]_t) = (X_t([w]_t), \overline{X}_t([w]_t))$ where $\overline{X}_t$ is the weighted average as defined in Lemma 3.2. Then $\psi_{\mathsf{query}}(S_t([w]_t)) = X_t([w]_t)$ and $\psi_{\mathsf{output}}(S_t([w]_t)) = \overline{X}_t([w]_t)$, and $\mathcal{A}_t^{(l)}$ updates the states according to (10).
- *Mirror descent update* (11). Similarly, we can take $\mathcal{S}_t = \mathcal{Z}_t \times \mathcal{Z}_t$ for the update (11) for the problem (MS-Saddle).
- *Accelerated mirror descent update* (13). The state can be $\mathcal{S}_t = \mathcal{X}_t \times \mathbb{R}^{n_t}$ and $S_t([w]_t) = (X_t([w]_t), \overline{G}_t([w]_t))$ where $\overline{G}_t$ is the cumulative gradient defined in (12). Then $\psi_{\mathsf{query}}(S_t([w]_t)) = X_t([w]_t)$ and $\psi_{\mathsf{output}}(S_t([w]_t)) = \overline{X}_t([w]_t)$ computed using $X_t([w]_t)$ and (13), and $\mathcal{A}_t^{(l)}$ updates the states according to (13).

5.2 Lazy update, efficient update

In this section, we state the updating mechanism to compute $S_t^{(l)}([w]_t)$ for some $t = 1, \ldots, T$, $l \in \{1, \ldots\}$, and the node $[w]_t$ in layer t. To formalize the discussion on the memory requirement

during the update, we assume that all processes during an algorithm have access to a shared memory space denoted as Memory, which admits the following operations: $\mathsf{read}(x)$, $\mathsf{write}(x)$, $\mathsf{del}(x)$, which reads, writes, and deletes the value of x from Memory. When some information x or some set $\mathcal{S}$ is in Memory, we abbreviate it as $x \in \mathsf{Memory}$ and $\mathcal{S} \subset \mathsf{Memory}$.

We assume that all the initializations and the $\xi_t^{(l)}$ used in constructing the conditional stochastic gradients are stored in the shared memory. That is, $\mathsf{Memory}^{(init)} \subset \mathsf{Memory}$ throughout the algorithm, where $\mathsf{Memory}^{(init)} := \{S_{1:T}^{(0)}([w]_t),\ w \in \Omega\} \cup \{\xi_{1:T}^{(0:L)}\}$. Then, to compute $S_t^{(l)}$, we consider the following $\mathcal{P}_{t,l}$ (Algorithm 1). We show its validity, and oracle and space complexity in Lemma 5.1.

Algorithm 1 Procedure $\mathcal{P}_{t,l}$

Require: $[w]_t$ is given, $\mathsf{Memory}^{(init)} \subset \mathsf{Memory}$, and if $t \geq 2$, $S_{t-1}^{(0:(l-1))}([w]_{t-1}) \in \mathsf{Memory}$.
Ensure: $S_t^{(0:l)}([w]_t) \in \mathsf{Memory}$.
  if $t = T$ **then**
    for $l' = 1, \ldots, l$ **do**
      $\mathsf{read}(S_T^{(l'-1)}([w]_t), S_{T-1}^{(l'-1)}([w]_{t-1}))$, compute $G_T^{(l'-1)}([w]_t)$ using (23)
      $S_T^{(l')}([w]_t) \leftarrow \mathcal{A}_T^{(l'-1)}(S_T^{(l'-1)}([w]_t), G_T^{(l'-1)}([w]_t))$, $\mathsf{write}(S_T^{(l')}([w]_t))$
    end for
  else
    for $l' = 1, \ldots, l$ **do**
      $\mathsf{read}(\xi_t^{(l'-1)})$, sample $[w_c]_{t+1}$, apply $\mathcal{P}_{t+1,l'-1}$ to the node $[w_c]_{t+1}$
      $\mathsf{read}(S_t^{(l'-1)}([w]_t), S_{t+1}^{(l'-1)}([w_c]_{t+1}))$, and if $t \geq 2$, $\mathsf{read}(S_{t-1}^{(l'-1)}([w]_{t-1}))$
      Compute $G_t^{(l'-1)}([w]_t)$ using (23)
      $S_t^{(l')}([w]_t) \leftarrow \mathcal{A}_t^{(l'-1)}(S_t^{(l'-1)}([w]_t), G_t^{(l'-1)}([w]_t))$
      $\mathsf{del}(S_{t+1}^{(1:(l'-1)}([w_c]_{t+1}))$, $\mathsf{write}(S_t^{(l')}([w]_t))$
    end for
  end if

Lemma 5.1. *The updates in Algorithm 1 are valid: when $\mathcal{P}_{t+1,l'-1}$ is applied,* Memory *contains all the required information.*

The oracle complexity for the stochastic conditional gradient and the proximal update is no more than 2^l.

Denoting the shared memory space at the start and the end of the procedure $\mathcal{P}_{t,l}$ as $\overline{\mathsf{Memory}}_0$ and $\overline{\mathsf{Memory}}_1$ respectively, then $\overline{\mathsf{Memory}}_1 = \overline{\mathsf{Memory}}_0 \cup \{S_t^{(1:l)}[w]_t\}$. Further assuming that all states S_t can be stored with B bits, then $\mathsf{Memory} \setminus \overline{\mathsf{Memory}}_0$ requires no more than $O(l^2 B)$ bits of space throughout the procedure $\mathcal{P}_{t,l}$.

Proof of Lemma 5.1. For the first claim, if $t = T$, $\mathcal{P}_{t,l}$ iteratively computes $S_T^{(1)}([w]_t), \ldots, S_T^{(l)}([w]_t)$, and indeed, for $l' = 1, \ldots, l$, since by assumption, Memory contains $S_{T-1}^{(l'-1)}([w]_{T-1})$, and $S_T^{(l'-1)}([w]_t)$ has been computed and stored in Memory in the previous iteration, $S_T^{(l')}([w]_t)$ can be computed. If $l > 0$ and $t < T$, then $\mathcal{P}_{t+1,l'-1}$ is applied, and indeed, $S_t^{(0:(l'-2))}([w]_t)$ has been computed in the previous iterations and are stored in Memory. For $l' = 1, \ldots, l$, when computing $S_t^{(l')}([w]_t)$, $S_t^{(l'-1)}([w]_t)$ has been computed in previous iteration, $S_{t+1}^{(l'-1)}([w_c]_{t+1})$ has been computed by $\mathcal{P}_{t+1,l'-1}$, and if $t \geq 2$, $S_{t-1}^{(l'-1)}([w]_{t-1})$ is in Memory by assumption.

For the second claim, first, notice that for procedure $\mathcal{P}_{t,l}$, the oracle calls involved are the following: l calls to $\mathcal{O}_t$, and if $l > 0, t < T$, then all oracle calls during $\mathcal{P}_{t+1,l'}$ for $l' = 0, \ldots, l-1$. Denoting an upper bound on the number of oracle calls by $\mathcal{P}_{t',l}$ for all t' by a_l. Then we can take $a_0 = 0$ and $a_1 = 1$. In addition, we have

$$a_l \le l + \sum_{l'=0}^{l-1} a_{l'}, \quad l = 2, 3, \ldots.$$

Simple induction then shows that $a_l \le 2^l - 1$ for all $l = 0, 1, \ldots$.

For the third claim, the first part can be proven by induction on (l, t) where the base case is $l = 0$ and $t = T$: if $l = 0$ then $\overline{\mathsf{Memory}}_0 = \mathsf{Memory}$; if $t = T$, then only lB bits of information (i.e. $S_t^{(1:l)}([w]_t)$) is written to Memory. If $l > 0, t < T$, since by inductive hypothesis, $\mathcal{P}_{t+1,l'-1}$ only adds $S_{t+1}^{(1:(l'-1))}([w]_t)$ to Memory, which is deleted by end of iteration l', and so after the l'-th iteration only $S_t^{(l')}([w]_t)$ is added to Memory. For the second part, the statement is true for $l = 0$ and $t = T$. For $l > 0, t < T$, notice that $\mathcal{P}_{t+1,0}$, $\mathcal{P}_{t+1,1}$, $\ldots$,$\mathcal{P}_{t+1,l-1}$ are run in series, and at the end of iteration l', all intermediate results are deleted. Denoting the maximum (over all t') additional space needed by $\mathcal{P}_{t',l'}$ as $b_{l'}$: we can take $b_0 = 0$, $b_1 = B$, then we have

$$b_l \le lB + \max_{l'=0,\ldots,l-1} b_{l'}.$$

Then induction gives $b_l \le Bl(l+1)/2 = O(l^2 B)$. □

Remark. In the space requirement above, we focus on the size of Memory, the shared memory space, and omit the *working memory* needed to apply $\mathcal{A}_t^{(l)}$ (e.g., computing the partial derivatives for a child node or computing the Bregman projection). We justify this by pointing out that usually these operations are also memory efficient: the extra space for the computation is on the same order as the space needed to store the states.

Remark. In procedure $\mathcal{P}_{t,l}$, it is possible that the samples $[w_c]_{t+1}$ are the same for two different l_1', l_2', but we don't store the updates during $\mathcal{P}_{t+1,l_1'}$ and $\mathcal{P}_{t+1,l_2'}$. This is justified if $d >> L$ since it's unlikely that the two samples are the same. However, if d is comparable with L, then storing the intermediate updates could potentially make the updates more gradient-oracle efficient.

5.3 Efficient online updating mechanism

To compute the sequence $S_t^{(L)}$ in an online fashion, the overall updates become Algorithm 2, where at time t, we apply $\mathcal{P}_{t+s,L}$ to the node $[w']_{t+s}$ for all $[w']_{t+s} \subset [w^*]_t$. Its correctness and complexity is stated in Theorem 5.1.

Theorem 5.1. *The updates in Algorithm 2 are valid: when $\mathcal{P}_{t',L}$ is applied, all the needed information is available. Denoting the shared memory space at the start of Algorithm 2 as $\overline{\mathsf{Memory}}_0$ and at the end of stge t as $\overline{\mathsf{Memory}}_t$, then*

$$\overline{\mathsf{Memory}}_t = \overline{\mathsf{Memory}}_0 \cup \{S_{t'}^{(1:L)}([w']_{t'}),\ [w']_{t'} \subset [w^*]_t,\ t' = t, \ldots, t+s\}. \tag{24}$$

That is, $\overline{\mathsf{Memory}}_t \setminus \overline{\mathsf{Memory}}_0$ contains L states for nodes in the subtree rooted at $[w^*]_t$ of depth $s+1$.

Assuming that all nodes have at most d child nodes, then the oracle complexity for the stochastic conditional gradient and the proximal update is $O(T2^L d^s)$.

Further assuming that all states S_t can be stored with B bits, then $\mathsf{Memory} \setminus \overline{\mathsf{Memory}}_0$ is no more than $O(d^s L^2 B)$ throughout the algorithm.

Algorithm 2 Lazy update

Require: At $t = 1, \dots, T-s$, $[w^*]_t$ is given. At $t = 1$, $\mathsf{Memory}^{(init)} \subset \mathsf{Memory}$. s, the number of stages decision variables are needed in advance. L, the number of updates required.

Ensure: At (the end of) $t = 1, \dots, T-s$, $S_{t'}^{(L)}([w']_{t'}) \in \mathsf{Memory}$ for $[w']_{t'} \subset [w^*]_{\mathcal{G}_t}$ and $t' = t, \dots, t+s$.

for $t = 1, \dots, T-s$ **do**
  if $t = 1$ **then**
    for $t' = 1, \dots, s+1$ **do**
      Apply $\mathcal{P}_{t',L}$ to $[w']_{t'}$ for all $[w']_{t'} \subset [w^*]_1$
    end for
  else
    $\mathsf{del}(S_{t'}^{(1:L)}([w']_{t'})$ for all $[w']_{t'} \subset [w^*]_{t-1} \setminus [w^*]_t$, $t' = t, \dots, t+s-1$
    Apply $\mathcal{P}_{t+s,L}$ to $[w']_{t+s}$ for all $[w']_{t+s} \subset [w^*]_t$
    $\mathsf{del}(S_{t-1}^{(1:L)}([w^*]_{t-1}))$
  end if
end for

Proof of Theorem 5.1. For the first claim, when $t = 1$, $\mathcal{P}_{1,L}$ can be called since $\mathsf{Memory}^{(init)} \subset \mathsf{Memory}$. If $s > 0$, $\mathcal{P}_{2,L}$ can be called since by Lemma 5.1, after $\mathcal{P}_{1,L}$ is applied, Memory contains all $S_1^{(0:L)}([w^*]_1)$. Similarly can show that $\mathcal{P}_{t',L}$ can be applied for $t' = 1, \dots, s+1$. By the end of stage 1, (24) follows from Lemma 5.1. Now suppose the first claim is true for stage $1, \dots, t-1$, then at the beginning of stage t, the memory space is $\overline{\mathsf{Memory}}_{t-1}$. Notice that applying $\mathcal{P}_{t+s,L}$ to the node $[w']_{t+s}$ where $[w']_{t+s} \subset [w^*]_t$ requires $S_{t+s-1}^{(0:L)}([w']_{t+s-1})$, which is in $\overline{\mathsf{Memory}}_{t-1}$ and has not been deleted. Thus, $\mathcal{P}_{t+s,L}$ can be applied, and adds the L states of nodes in layer $s+1$ of the subtree rooted at $[w^*]_t$ (Lemma 5.1). The delete operation removes those states inconsistent with $[w^*]_t$ and the states for $t-1$. Thus, the first claim holds for t. By induction, it holds for all $t = 1, \dots, T-s$.

For the second claim, notice that the number of nodes that need the L-th updates is upper bounded by

$$d^0 + \cdots + d^s + (T-s-1)d^s \le Td^s.$$

For each L-th update, Algorithm 2 applies $\mathcal{P}_{\cdot,L}$, which requires at most 2^L oracles by Lemma 5.1. Thus, the total number of oracles is upper bounded by $O(Td^s 2^L)$. Since each gradient oracle is used in one proximal update, the number of updates is also $O(T2^L d^s)$.

For the third claim, By (24), $\overline{\mathsf{Memory}}_t \setminus \overline{\mathsf{Memory}}_0$ contains L states for at most $\sum_{t'=1}^{s+1} d^{t'-1} = O(d^s)$ nodes, and so $\overline{\mathsf{Memory}}_t \setminus \overline{\mathsf{Memory}}_0$ requires at most $O(LB \cdot d^s)$ bits. Notice that at stage t, $\mathcal{P}_{t',L}$ is applied for at most $O(d^s)$ nodes in series, and each application adds at most $O(LB)$ bits to Memory, and requires at most $O(L^2 B)$ bits of additional memory by Lemma 5.1, we have $\mathsf{Memory} \setminus \overline{\mathsf{Memory}}_t$ is no more than $O(LBd^s + L^2 B)$ throughout stage t. Thus, $\mathsf{Memory} \setminus \overline{\mathsf{Memory}}_0$ requires no more than $O(LBd^s + L^2 B)$ bits. □

Remark. Due to the asynchronous updates in our Algorithm 2, the initialization $S_{t'}^{(0)}$ and the sampling distribution for $[w]_{t'}$ are needed at stage $\ge t' - s - L$, and so they can be given in an online fashion.

Finally, combining the in-expectation guarantees from Theorem 4.1 with the Algorithm 2, we have the following convergence guarantee.

Corollary 5.1. *Assume that the assumptions in Theorem 4.1 hold, that the updates* (3) *are applied with the stochastic conditional oracle in Lemma 4.1. Then the implementation in Algorithm 2 achieves the following for* $X_t^{(L)}(w) = \psi_{\mathsf{output}}(S_t^{(L)}(w))$

$$\tilde{\mathbb{E}}[\mathbb{E}[f(X_{1:T}^{(L)})]] - \mathbb{E}[f(X_{1:T}^*)] \le 2\sqrt{\frac{\tilde{D}^2(2\sigma^2 + 2b^2 + L_1^2)}{L+1}} + bD.$$

In addition, with $L = \epsilon^{-2}$*, the oracle complexity is* $O(Td^s 2^{1/\epsilon^2})$ *while the space complexity is* $O(\epsilon^{-2}Bd^s + \epsilon^{-4}B)$*. Further assuming* $B = O(\max_{t=1,\dots,T} n_t)$ *and taking* $s = 0$*, the oracle and the proximal step complexity is* $O(T2^{1/\epsilon^2})$ *and the space complexity is* $O(\epsilon^{-4}\max_{t=1,\dots,T} n_t)$*.*

Similar in expectation results hold for (5) and (8). Note that the high probability guarantee in Theorems 4.1 and 4.2 is *high probability w.r.t. the randomness in the stochastic conditional gradients*, not w.r.t. the randomness in $(\Omega, \mathcal{F}, \mathbb{P})$. To obtain high probability bounds on the suboptimality

$$f(\psi_{\mathsf{output}}(S_1^{(L)}(w)), \psi_{\mathsf{output}}(S_2^{(L)}(w)), \dots, \psi_{\mathsf{output}}(S_T^{(L)}(w)), w) - f(X_{1:T}^*(w), w),$$

one needs to bound the deviations in Lemma 3.2 in a "pathwise" manner, not in expectation, which we leave to future work.

6 Numerical experiments

We apply our mirror descent stochastic approximation (3) and (8) to a smoothed online convex optimization problem, and (5) to a revenue management problem.

Our experiment results demonstrate that the proposed mirror descent stochastic approximation algorithms converge in a variety of settings: convex optimization with and without strong convexity, and saddle point problems. In addition, they admit the following advantages: applicable even when the randomness across stages is *correlated*, robustness against *misspecified* sampling distribution in constructing the gradients, efficiency for large T.

All experiments are implemented using Python and run on MacBook Air with the M3 chip.

6.1 Smoothed online convex optimization

We consider a smoothed online convex optimization problem, modeled as (MS-Unconstrained) with

$$f_t(x_{t-1}, x_t, w) := h(\|x_t - (\theta_t + \epsilon_t(w))\|_2) + \frac{1}{2}\|x_t - x_{t-1}\|_2^2. \tag{25}$$

The objective function (25) appears in tracking problems [25], where the goal is to decide the positions $x_1, \dots, x_T$ in order to minimize the distance from the moving target $(\theta_t + \epsilon_t(w)$ at stage $t)$ and the moving cost (functions of $\|x_t - x_{t-1}\|_2$).

In the experiment, we take $x_0 = \mathbf{0}$ and $h : [0, \infty) \to \mathbb{R}$ is either the strongly convex quadratic cost $h_{\text{quad}}(s) = s^2/2$, or the huber cost $h_{\text{huber}}(s) = \begin{cases} s^2/2 & |s| \le 1 \\ |s| - 1/2 & |s| > 1 \end{cases}$, which is convex but not strongly convex. $n_t = 10$ and $\mathcal{X}_t = \{x_t \in \mathbb{R}^{n_t},\ \|x_t\|_2 \le 10\}$ is the ball with radius 10 for all t. For $i = 1, \dots, 10$, $\theta_{t,i} = 7.5\sin(2\pi \cdot (1 + \frac{i-1}{100})t)$ is a known sequence.

For the randomness, $w = (w_1, \dots, w_T)$ and the information at stage t is $\mathcal{G}_t = \sigma(w_{1:t})$. Thus, a node in the t-layer of the tree can be specified by $(w_1, \dots, w_t)$. We take $\pi_t([w]_t)$, the *true*

distribution of the child nodes, to be uniform. The actual noise sequence $(\epsilon_1, \ldots, \epsilon_T)$ follows an auto-regressive process with the discount factor $\rho = 0.8$: $\epsilon_1(w) = w_1$, and $\epsilon_t(w) = \rho\epsilon_{t-1}(w) + w_t$ for $t \geq 2$.

Misspecified distributions. We sample using $\hat{\pi}_t([w]_t) = (1-\delta)\pi_t([w]_t) + \delta d_t([w]_t)$ where $d_t([w]_t)$ is a probability distribution and $\delta \in [0, 1]$. Thus, d_t represents the misspecification of the children distribution. We consider 2 types of perturbation d_t: for $d_t^{(0)}$, each coordinate is sampled uniformly in $[0, 1]$ then normalized such that the sum is 1. For $d_t^{(1)}$, one coordinate is picked uniformly at random and set to 1, and all rests are set to 0. Thus, type $d_t^{(1)}$ can be viewed as a more adversarial perturbation. We generate one d_t for each node independently.

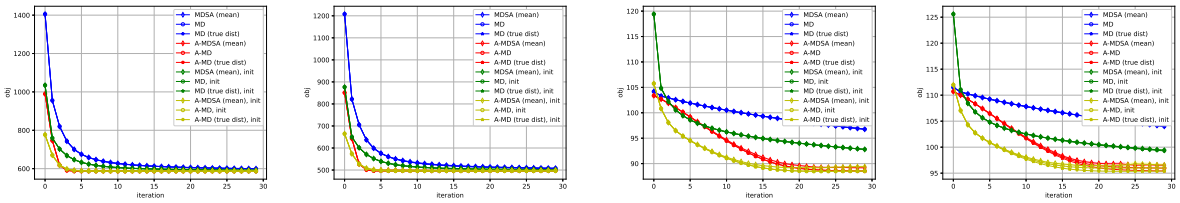
Algorithms setup. We use $v_t(x_t) = \frac{1}{2}\|x_t\|_2^2$ as the distance generating functions. The stochastic gradients are constructed using Lemma 4.1, under the *misspecified* distribution $\hat{\pi}_t$. For the accelerated updates (8), we use the α_l, A_l as defined in (8) with $\gamma = 1$ and $\theta = 1/2$. For $h = h_{\text{quad}}$, we take $\mu = 1$ and $L_2 = 3$. For $h = h_{\text{huber}}$, we take $\mu = 0$ and $L_2 = 3$.

Additional experiments show that the performance of (A-)MD(SA) algorithms is similar to that presented below, under a variety of parameter settings for δ, ρ, and α when $h = \alpha h_{\text{quad}}$ and $h = \alpha h_{\text{huber}}$. We omit these results due to space constraints.

6.1.1 $T = 5$ and each non-leaf node has 10 child nodes

We generate Ω in the following manner: we first sample one $w_1 \sim_{i.i.d.} \mathcal{N}(\mathbf{0}, 16I)$, then we generate 10 $w_2 \sim_{i.i.d.} \mathcal{N}(\mathbf{0}, 16I)$ independent of w_1, then for each (w_1, w_2), we generate 10 $w_3 \sim_{i.i.d.} \mathcal{N}(\mathbf{0}, 16I)$ independent of (w_1, w_2) and so on.

For the updates (3), we take the step size $\gamma_l = 1/\sqrt{L}$. We test two types of initialization: $X_t^{(0)} = \mathbf{0}$ and $X_t^{(0)} = \theta_t$. In Figures 3, we present the 5-run average with $X_t^{(0)} = \mathbf{0}$ (MDSA (mean)) and $X_t^{(0)} = \theta_t$ (MDSA (mean), init), with the mean $\pm$ 1 standard deviation in a lighter color[2]. In addition, we also run (3) with the *exact* gradient computed under $\hat{\pi}_t$ ((MD) and (MD, init)) and under π_t ((MD (true dist)) and (MD (true dist), init)). We present the results for the accelerated updates, with the same setup. We point out that the first output of the accelerated updates is given by (8), and so is not necessarily $X_t^{(0)}$.



(a) $(0.1, d_t^{(0)})$, h_{quad} (b) $(0.5, d_t^{(1)})$, h_{quad} (c) $(0.1, d_t^{(0)})$, h_{huber} (d) $(0.5, d_t^{(1)})$, h_{huber}

Figure 3: Parameters: (δ, d_t), h. Objective values for smooth online convex optimization.

Effects of stochastic gradients. As expected, with the stochastic gradient oracle (MDSA and A-MDSA), the total running time is approximately $80 \sim 90$ seconds, while with the exact gradient ((A-)MD, (A-)MD (true dist)), the total running time is approximately $140 \sim 150$ seconds. Thus, the running time is significantly reduced if the stochastic gradient is used, without compromising the convergence by too much. In terms of the convergence of objective values, notice that one key difference between stochastic gradients and full gradients is the noise term σ_t, which is 0 for

[2]The variation is very small, and so the region is almost invisible.

full gradients, and which depends on the variance of the gradient for stochastic gradients. This is reflected by Figure 3 (c) and (d), where A-MDSA is converging to a *slightly* larger value than A-MD and A-MD (true dist) for both initialization.

Correlation between randomness. For both the quadratic loss and the huber loss, we test our algorithms for $\rho = 0.8$. With this correlated sequence ϵ_t, as shown by all setups in Figure 3, all of our algorithms converge within 30 iterations, or show a trend of convergence (MD(SA) for huber loss). Indeed, our theoretical results do not assume independence between randomness in different stages.

Robustness against misspecification. Comparing the results for A-MD and A-MD (true dist) for both initializations in Figure 3 (d), we see that A-MD is converging to a *slightly* larger value than A-MD (true dist). However, in all other settings, the bias in the sampling distribution does not have a noticeable effect on the convergence.

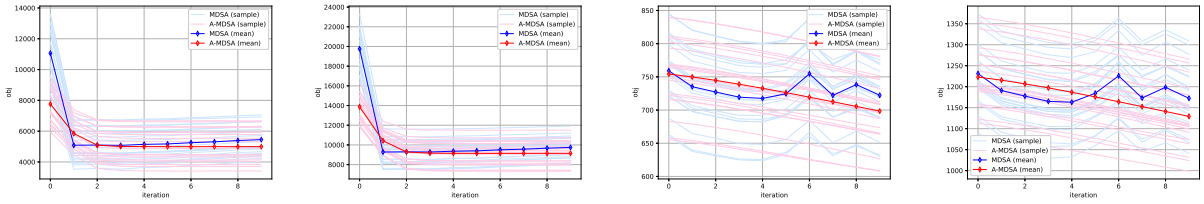
Strong convexity. Comparing the results for h_{quad} and h_{huber}, we see that for both the accelerated and the non-accelerated updates, strongly convex objectives converge faster for all our algorithms. For the update (8), this agrees with our theoretical results; for the update (3), this suggests that the suboptimality in Theorem 4.1 could be loose, and tighter bounds could be attained with strong convexity.

Acceleration. As expected, in all our settings, when the initializations are the same, A-MD(SA) converges faster than MD(SA): for h_{quad}, A-MD(SA) converges in ~ 5 iterations, MD(SA) converges in ~ 15 iterations; for h_{huber}, A-MD(SA) converges in ~ 20 iterations, MD(SA) does not reach convergence in 30 iterations.

6.1.2 $T = 30, 50$ and each non-leaf node has 50 child nodes

We test our updates (3) and (8) using the lazy updates in Algorithm 2 for the smoothed online convex optimization problem with $T = 30, 50$, where each non-leaf node has $d = 50$ child nodes, and $s = 0$. We take $\delta = 0.2$ and $\delta_t^{(0)}$ as the perturbation to the sampling distribution. In addition, if the procedures $\mathcal{P}_{t,l}$ and $\mathcal{P}_{t,l'}$ are applied to the same node, the sampling distributions are perturbed by two independent random $d_t^{(0)}$'s.

For (3), we use $\gamma = 3/\sqrt{L}$, and we use $X_t^{(0)} = \mathbf{0}$ as initialization. To generate all $(w_1, \ldots, w_T)$, we first sample d vectors $\tilde{w}_0, \ldots, \tilde{w}_{d-1} \sim_{i.i.d.} \mathcal{N}(\mathbf{0}, 16I)$. Then we number nodes using the path from the root: a node numbered as $(0, k_1, \ldots, k_{t-1})$ means it's the $(k_{t-1}+1)$-th child of the node $(0, k_1, \ldots, k_{t-2})$. Here (0) is the root, $k_{t'} \in \{0, \ldots, d-1\}$, and $t = 1, \ldots, T$. Then, we associate the node $(0, k_1, \ldots, k_{t-1})$ with the randomness where $w_{t'} = \tilde{w}_{k_{t'}}$ $(k_0 = 0)$. Notice that $w_1, \ldots, w_T$ are independent, but the sequence $(\epsilon_1, \ldots, \epsilon_T)$ is still correlated.



(a) $T = 30$, h_{quad}. (b) $T = 50$, h_{quad}. (c) $T = 30$, h_{huber}. (d) $T = 50$, h_{huber}.

Figure 4: $\delta = 0.2$, $\delta_t^{(0)}$, and $T = 30, 50$.

Our experiments show that even when T and the number of child nodes are large, our algorithms

are very efficient: the running time for the entire T stages is approximately 20 seconds for $T = 30$, and approximately 45 seconds for $T = 50$.

In Figure 4, we present $f(\overline{X}_1^{(l)}(w^{(i)}), \overline{X}_2^{(l)}(w^{(i)}), \ldots, \overline{X}_T^{(l)}(w^{(i)}), w^{(i)})$ and their averages, where $w^{(1)}, \ldots, w^{(20)}$ are i.i.d. sampled scenarios. As revealed in Figure 4, the quadratic loss converges within 5 iterations, while the huber loss does not reach convergence within 10 iterations, but the objective values show a decreasing trend. These agree with our theoretical results in Corollary 5.1.

6.2 Revenue management

We consider the following revenue management problem:

$$\max_{x_1 \in [0,1]} \cdots \max_{x_T \in [0,1]} \sum_{t=1}^{T} c_t x_t, \quad s.t. \sum_{t=1}^{T} \mathbf{a}_t x_t \leq \mathbf{b}_0.$$

Here $\mathbf{b}_0 \in (0, \infty)^M$ denotes the budget, and $c_t \geq 0$, $\mathbf{a}_t \geq \mathbf{0}$ denote the revenue and the resource consumed when $x_t = 1$. If all pairs $(c_t, \mathbf{a}_t)$ are known, then the problem is a linear programming problem. However, when $(c_t, \mathbf{a}_t)$ are random variables which are revealed at stage t, the problem becomes a sequential decision making process with stochasticity, which we model using the framework of (MS-Saddle).

Precisely, let $\mathbf{b}_0(w) = \mathbf{b}_0 \in (0, \infty)^M$ denote the initial budget (which is the same for all scenarios w) and $\mathcal{X}_t = [0, 1] \times \{\tilde{\mathbf{b}} \in \mathbb{R}^M, \ \mathbf{0} \leq \tilde{\mathbf{b}} \leq \mathbf{b}\}$, then we consider the following problem:

$$\begin{aligned} &\max_{(X_1, \mathbf{b}_1) \in \mathcal{G}_1 \cap \mathcal{X}_1} \cdots \max_{(X_T, \mathbf{b}_T) \in \mathcal{G}_T \cap \mathcal{X}_T} \sum_{t=1}^{T} \mathbb{E}[c_t([w]_t) X_t([w]_t)] \\ &s.t. \ \mathbf{a}_t([w]_t) X_t([w]_t) + \mathbf{b}_t([w]_t) \leq \mathbf{b}_{t-1}([w]_{t-1}), \ \forall \ w, t, \end{aligned} \tag{26}$$

where $w = (w_1, \ldots, w_T)$ with $w_t := (\mathbf{a}_t, c_t)$ and $\mathcal{G}_t := \sigma(w_1, \ldots, w_t)$.

Thus, the above problem is a linear programming problem, with decision variables $(X_t, \mathbf{b}_t)([w]_t)$ for all $[w]_t \in \Omega_{\mathcal{G}_t}$ and all t. In particular, strong duality holds, and (26) is equivalent to the saddle point problem for $\mathcal{Y}_t = [0, \infty)^M$ with

$$\phi_t(x_{t-1}, \mathbf{b}_{t-1}, x_t, \mathbf{b}_t, \mathbf{y}_t, w) := -c_t([w]_t) x_t + \langle \mathbf{y}_t, \mathbf{a}_t([w]_t) x_t + \mathbf{b}_t - \mathbf{b}_{t-1} \rangle. \tag{27}$$

In the above problem, $\mathcal{Y}_t$ is unbounded. Nevertheless, under additional assumptions, it's possible to show that there exists an optimal dual solution whose norm has a known upper bound (Appendix C).

Tree setup. We take $M = 10$ and $B = 2T \cdot \mathbf{1} \in \mathbb{R}^M$, and for each non-leaf node, the distribution over of its child nodes are the same, i.e. π_t is the uniform distribution.

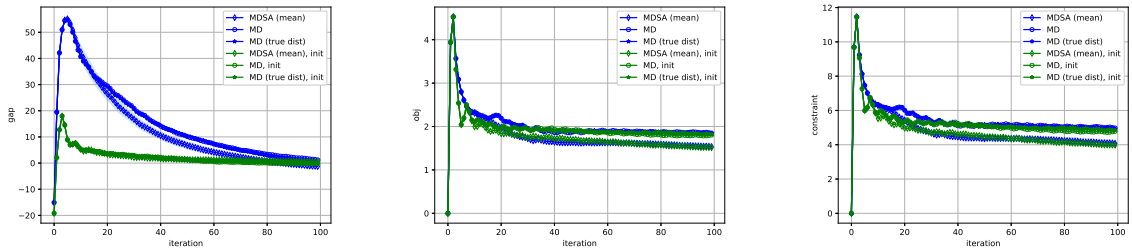
Algorithms setup. The sampling distributions by MDSA are perturbed by $d_t^{(0)}$, with $\delta = 0.2$. For the feasibility set of the dual $\mathbf{y}_t$, instead of using $[0, \infty)^M$, we use $\mathcal{Y}_t' = [0, 5]^M$. All $x_t, \mathbf{y}_t$ are initialized to be 0 and $\mathbf{0}$, while for $\mathbf{b}_t$, we consider two types: $\mathbf{b}_t^{(0)} = \mathbf{0}$, and $\mathbf{b}_t^{(0)} = B(1 - t/T)$. To differentiate them, we add "init" in the legends of the second type.

We consider problems where $T = 5$ and each non-leaf node has 10 child nodes. We randomly generate an instance of the problem, where the $\mathbf{a}_t$ and c_t are sampled uniformly in $[1, 5]^M$ and $[0, 2]$ respectively, and the sampling is independent for all nodes.

Evaluation. We use $\overline{Z}_{1:T}^*$, the 500 iteration output of MD (with $\gamma_l = \gamma/\sqrt{500}$), as an approximation to the solution to the saddle point problem. The algorithms are evaluated through the

(approximate) gaps[3], objective function values, and the total budgets spent.

We test MDSA and (MDSA, init) for 5 runs.For MDSA, the running time is ~ 220 seconds, while for MD, the running time is ~ 390 seconds. We present the results with $\gamma = 5$ in Figure 5.



(a) gap (b) objective (c) constraint

Figure 5: Revenue management, $\gamma = 5$. Gap: $\mathbb{E}[\phi(\overline{X}^{(l)}_{1:T}, \overline{Y}^{*}_{1:T})] - \mathbb{E}[\phi(\overline{X}^{*}_{1:T}, \overline{Y}^{(l)}_{1:T})]$, objective: $\mathbb{E}[\sum_{t=1}^{T} c_t \overline{X}^{(l)}_t]$, constraint: $\|\mathbb{E}[\sum_{t=1}^{T} \mathbf{a}_t \overline{X}^{(l)}_t]\|_\infty$.

Convergence of the (approximate) gap $\mathbb{E}[\phi(\overline{X}^{(l)}_{1:T}, \overline{Y}^{*}_{1:T})] - \mathbb{E}[\phi(\overline{X}^{*}_{1:T}, \overline{Y}^{(l)}_{1:T})]$. In Figure 5(a), for both initializations, the term $\mathbb{E}[\phi(\overline{X}^{(l)}_{1:T}, \overline{Y}^{*}_{1:T})] - \mathbb{E}[\phi(\overline{X}^{*}_{1:T}, \overline{Y}^{(l)}_{1:T})]$ converges to 0 when the exact gradients are used (MD and MD(true dist)), and to a slightly negative value when stochastic gradients are used (MDSA). This agrees with the *upper bound* for $\max_{Y_{1:T} \in \overline{\mathcal{Y}}} \mathbb{E}[\phi(\overline{X}^{(l)}_{1:T}, Y_{1:T})] - \min_{X_{1:T} \in \overline{\mathcal{X}}} \mathbb{E}[\phi(X_{1:T}, \overline{Y}^{(l)}_{1:T})]$ in Theorem 4.2. To understand why the (approximate) gap can be negative, notice that

$$
\begin{aligned}
&\mathbb{E}[\phi(\overline{X}^{(l)}, \overline{Y}^{*})] - \mathbb{E}[\phi(\overline{X}^{*}, \overline{Y}^{(l)})] = A_1 - A_2,\\
&A_1 = \mathbb{E}[\phi(\overline{X}^{(l)}, \overline{Y}^{*})] - \min_{X \in \overline{\mathcal{X}}} \mathbb{E}[\phi(X, \overline{Y}^{*})] + \max_{Y_{1:T} \in \overline{\mathcal{Y}}} \mathbb{E}[\phi(\overline{X}^{*}, Y)] - \mathbb{E}[\phi(\overline{X}^{*}, \overline{Y}^{(l)})],\\
&A_2 = \max_{Y_{1:T} \in \overline{\mathcal{Y}}} \mathbb{E}[\phi(\overline{X}^{*}, Y)] - \min_{X \in \overline{\mathcal{X}}} \mathbb{E}[\phi(X_{1:T}, \overline{Y}^{*})].
\end{aligned}
$$

where we omit all subscripts $1:T$ (e.g. $\overline{X}^{(l)}_{1:T} = \overline{X}^{(l)}$). Thus, although $A_1 \geq 0$, due to the suboptimality of $(\overline{X}^{*}_{1:T}, \overline{Y}^{*}_{1:T})$, the term A_2 could be negative. In fact, by Lemma 3.3 and Theorem 4.2, with $\gamma_l = \gamma/\sqrt{500}$, $\tilde{\mathbb{E}}[A_2] = O(\frac{D^2}{\gamma} + \gamma L_1^2)/\sqrt{500}$.

Objective values $\mathbb{E}[\sum_{t=1}^{T} c_t \overline{X}^{(l)}_t]$ and constraints $\|\mathbb{E}[\sum_{t=1}^{T} \mathbf{a}_t \overline{X}^{(l)}_t]\|_\infty$. From Figure 5 (b) and (c), we see that for all settings, during the first 2 iterations, the objective values are increasing, and at iteration 2 the constraints are violated[4]. After that, objective values and the constraints are converging to values that do not depend on the initialization or the bias in the gradient sampling distribution. However, the values depend on whether the stochastic gradients are used. This suggests that with this γ, the terms $O(\gamma\sigma^2)$ and $O(D^2/\gamma)$ are relatively balanced, with the latter having a slightly larger effect on the performance.

[3]The approximate optimal solution $\overline{Z}^{*}_{1:T}$ used is the one corresponding to the same initialization. That is, the blue curves in Figure 5(a) are evaluated using the 500-th iteration output initialized at $\mathbf{b}^{(0)}_t = \mathbf{0}$, and the green curves are evaluated using that but initialized at $\mathbf{b}^{(0)}_t = B(1 - t/T)$.

[4]The violation of the constraints *in expectation* implies that there exists at least one scenario such that the constraints are violated.

7 Conclusion

In this work, we study the unconstrained (MS-Unconstrained) and the saddle point (MS-Saddle) variant of the multi-stage stochastic programming problems. We show the convergence of the (accelerated) mirror descent stochastic approximation with stochastic conditional gradient oracles. To further reduce the complexity, we consider a semi-online framework where the updates of mirror descent stochastic approximation are made in an asynchronous fashion and are based on the sequentially revealed information about the underlying scenario, which reduces the complexity from exponential to linear in T.

Below, we point out three directions of future works. First, our Corollary 5.1 is an *in expectation* guarantee for the suboptimality, where the expectation is taken over w and the randomness in the gradient oracles. It would be interesting to have guarantees along each sample path under w, i.e., where the expectation is taken over the randomness in the gradient oracles only. Second, the suggested step sizes γ_l for (3) and (5) and the sequence α_l for (8) depend on prior information about the objective functions, the sets $\mathcal{X}_t, \mathcal{Y}_t$, and the oracles. It would be interesting to develop adaptive algorithms which do not require such prior information. Third, our algorithms are closely related to the classical (accelerated) mirror descent algorithms [31, 10]. It would be interesting to explore the multi-stage analog of alternative assumptions and algorithms that can be applied to other families of problems, e.g. problems where the constraint sets are unbounded [8, 30].

Acknowledgements

This work was funded by the Office of Naval Research grant N00014-24-1-2470.

References

[1] Shabbir Ahmed, Lingquan Ding, and Alexander Shapiro. *A Python package for multi-stage stochastic programming.*

[2] Santiago R. Balseiro, Haihao Lu, and Vahab Mirrokni. "The Best of Many Worlds: Dual Mirror Descent for Online Allocation Problems". In: *Operations Research* 71.1 (2023), pp. 101–119.

[3] Nikhil Bansal, Anupam Gupta, Ravishankar Krishnaswamy, Kirk Pruhs, Kevin Schewior, and Cliff Stein. "A 2-Competitive Algorithm For Online Convex Optimization With Switching Costs". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*. Ed. by Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim. Vol. 40. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015, pp. 96–109.

[4] Omar Besbes, Yonatan Gur, and Assaf Zeevi. "Non-Stationary Stochastic Optimization". In: *Operations Research* 63.5 (2015), pp. 1227–1244.

[5] Arnab Bhattacharya, Jeffrey P. Kharoufeh, and Bo Zeng. "Managing Energy Storage in Microgrids: A Multistage Stochastic Programming Approach". In: *IEEE Transactions on Smart Grid* 9.1 (2018), pp. 483–496.

[6] Sébastien Bubeck. "Convex Optimization: Algorithms and Complexity". In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357.

[7] Niv Buchbinder and Joseph (Seffi) Naor. "The Design of Competitive Online Algorithms via a Primal–Dual Approach". In: *Foundations and Trends® in Theoretical Computer Science* 3.2–3 (2009), pp. 93–263.

[8] Antonin Chambolle and Thomas Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". In: *Journal of Mathematical Imaging and Vision* 40.1 (May 2011), pp. 120–145.

[9] Niangjun Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. "Using Predictions in Online Optimization: Looking Forward with an Eye on the Past". In: *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. SIGMETRICS '16. Antibes Juan-les-Pins, France: Association for Computing Machinery, 2016, pp. 193–206.

[10] Olivier Devolder, François Glineur, and Yurii Nesterov. *First-order methods with inexact oracle: the strongly convex case*. Tech. rep. Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), May 2013.

[11] Martin Dyer and Leen Stougie. "Computational complexity of stochastic programming problems". In: *Mathematical Programming* 106.3 (May 2006), pp. 423–432.

[12] Christian Füllner and Steffen Rebennack. *Stochastic dual dynamic programming and its variants – a review*.

[13] Gautam Goel and Adam Wierman. "An Online Algorithm for Smoothed Regression and LQR Control". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Ed. by Kamalika Chaudhuri and Masashi Sugiyama. Vol. 89. Proceedings of Machine Learning Research. New York: PMLR, 16–18 Apr 2019, pp. 2504–2513.

[14] Grani A. Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann. "A comment on "computational complexity of stochastic programming problems"". In: *Mathematical Programming* 159.1 (Sept. 2016), pp. 557–569.

[15] Elad Hazan. "Introduction to Online Convex Optimization". In: *Found. Trends Optim.* 2.3–4 (Aug. 2016), pp. 157–325.

[16] Caleb Ju and Guanghui Lan. *Dual dynamic programming for stochastic programs over an infinite horizon*. 2023.

[17] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. "Solving Variational Inequalities with Stochastic Mirror-Prox Algorithm". In: *Stochastic Systems* 1.1 (2011), pp. 17–58.

[18] Guanghui Lan. "An optimal method for stochastic composite optimization". In: *Mathematical Programming* 133.1 (June 2012), pp. 365–397.

[19] Guanghui Lan. "Complexity of stochastic dual dynamic programming". In: *Mathematical Programming* 191.2 (Feb. 2022), pp. 717–754.

[20] Guanghui Lan. "Correction to: Complexity of stochastic dual dynamic programming". In: *Mathematical Programming* 194.1 (July 2022), pp. 1187–1189.

[21] Guanghui Lan. *First-order and Stochastic Optimization Methods for Machine Learning*. Cham, Switzerland: Springer Cham, 2020.

[22] Guanghui Lan and Alexander Shapiro. "Numerical Methods for Convex Multistage Stochastic Optimization". In: *Foundations and Trends® in Optimization* 6.2 (2024), pp. 63–144.

[23] Guanghui Lan and Zhiqiang Zhou. "Dynamic stochastic approximation for multi-stage stochastic optimization". In: *Mathematical Programming* 187.1 (May 2021), pp. 487–532.

[24] Yingying Li, Xin Chen, and Na Li. "Online optimal control with linear dynamics and predictions: algorithms and regret analysis". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[25] Yingying Li and Na Li. "Leveraging predictions in smoothed online convex optimization via gradient-based algorithms". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[26] Yingying Li, Guannan Qu, and Na Li. "Online Optimization With Predictions and Switching Costs: Fast Algorithms and the Fundamental Limit". In: *IEEE Transactions on Automatic Control* 66.10 (2021), pp. 4761–4768.

[27] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan L. H. Andrew. "Online algorithms for geographical load balancing". In: *2012 International Green Computing Conference (IGCC)*. 2012, pp. 1–10.

[28] Minghong Lin, Adam Wierman, Lachlan L. H. Andrew, and Eno Thereska. "Dynamic right-sizing for power-proportional data centers". In: *2011 Proceedings IEEE INFOCOM*. 2011, pp. 1098–1106.

[29] Runzhao Lu, Tao Ding, Boyu Qin, Jin Ma, Xin Fang, and Zhaoyang Dong. "Multi-Stage Stochastic Programming to Joint Economic Dispatch for Energy and Reserve With Uncertain Renewable Energy". In: *IEEE Transactions on Sustainable Energy* 11.3 (2020), pp. 1140–1151.

[30] Renato D. C. Monteiro and B. F. Svaiter. "On the Complexity of the Hybrid Proximal Extragradient Method for the Iterates and the Ergodic Mean". In: *SIAM Journal on Optimization* 20.6 (2010), pp. 2755–2787.

[31] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. "Robust Stochastic Approximation Approach to Stochastic Programming". In: *SIAM Journal on Optimization* 19.4 (2009), pp. 1574–1609.

[32] Arkadi Nemirovski. "Prox-Method with Rate of Convergence O(1/t) for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems". In: *SIAM Journal on Optimization* 15.1 (2004), pp. 229–251.

[33] M. V. F. Pereira and L. M. V. G. Pinto. "Multi-stage stochastic optimization applied to energy planning". In: *Mathematical Programming* 52.1 (May 1991), pp. 359–375.

[34] Roger J. B. Wets R. Tyrrell Rockafellar. *Variational Analysis*. Berlin, Germany: Springer Science & Business Media, 2009.

[35] R. T. Rockafellar and R. J.-B. Wets. "Nonanticipativity and L1-martingales in stochastic optimization problems". In: *Stochastic Systems: Modeling, Identification and Optimization, II*. Ed. by Roger J.- B. Wets. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, pp. 170–187.

[36] R. T. Rockafellar and Roger J.-B. Wets. "Scenarios and Policy Aggregation in Optimization Under Uncertainty". In: *Mathematics of Operations Research* 16.1 (1991), pp. 119–147.

[37] R. Tyrrell Rockafellar and Roger J-B Wets. "Stochastic variational inequalities: single-stage to multistage". In: *Math. Program.* 165.1 (Sept. 2017), pp. 331–360.

[38] Alexander Shapiro. "Inference of statistical bounds for multistage stochastic programming problems". English. In: *Mathematical Methods of Operations Research* 58.1 (Sept. 2003). Place: Heidelberg Publisher: Springer Nature B.V., pp. 57–68.

[39] Alexander Shapiro. "On complexity of multistage stochastic programs". In: *Operations Research Letters* 34.1 (2006), pp. 1–8.

[40] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory, Third Edition*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2021.

[41] Alexander Shapiro and Arkadi Nemirovski. "On Complexity of Stochastic Programming Problems". In: *Continuous Optimization: Current Trends and Modern Applications*. Ed. by Vaithilingam Jeyakumar and Alexander Rubinov. Boston, MA: Springer US, 2005, pp. 111–146.

[42] Alberto Vera and Siddhartha Banerjee. "The Bayesian Prophet: A Low-Regret Framework for Online Decision Making". In: *Management Science* 67.3 (2021), pp. 1368–1391.

[43] Yaqi Xie, Will Ma, and Linwei Xin. "The Benefits of Delay to Online Decision-Making". In: *SSRN Electronic Journal* (Nov. 2024).

[44] Runyu Zhang, Yingying Li, and Na Li. "On the Regret Analysis of Online LQR Control with Predictions". In: *2021 American Control Conference (ACC)*. 2021, pp. 697–703.

[45] Martin Zinkevich. "Online convex programming and generalized infinitesimal gradient ascent". In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. Washington, DC, USA: AAAI Press, 2003, pp. 928–935.

A Discussion on the assumptions

In Assumption 2.3, it's explicitly assumed that $\tilde{\phi}_t$ is random lsc w.r.t. $\mathcal{F}_t$. In fact, this is implied by the assumption that $\pm\phi_t$ are random lsc w.r.t. $\mathcal{F}_t$. Indeed, let $\widehat{\mathcal{Y}}_t \subset \mathcal{Y}_t$ be a countable dense subset of $\mathcal{Y}_t$. By Proposition 14.44 in [34], we have $\widehat{\phi}_t(x_{t-1}, x_t, w) := \sup_{y_t \in \widehat{\mathcal{Y}}_t} \phi_t(x_{t-1}, x_t, y_t, w)$ is random lsc w.r.t. $\mathcal{G}_t$. Since $-\phi_t$ is convex and thus continuous (Theorem 2.35 [34]) in y_t, and $\widehat{\mathcal{Y}}_t$ is dense in $\mathcal{Y}_t$, we have $\widehat{\phi}_t(x_{t-1}, x_t, w) = \tilde{\phi}_t(x_{t-1}, x_t, w)$ and thus $\tilde{\phi}_t$ is random lsc w.r.t. $\mathcal{G}_t$.

In Assumption 2.4, the assumption that $Y_t^* \in \mathcal{G}_t \cap \mathcal{Y}_t$ can be relaxed to $Y_t^* \in \mathcal{F} \cap \mathcal{Y}_t$. Indeed, suppose there exists $\widetilde{Y}_t \in \mathcal{F} \cap \mathcal{Y}_t$ such that $\tilde{\phi}_t(X_{t-1}^*(w), X_t^*(w), w) = \phi_t(X_{t-1}^*(w), X_t^*(w), \widetilde{Y}_t(w), w)$ for all w. Since $-\phi_t$ is random lsc w.r.t. $\mathcal{G}_t$, and since $X_{t-1}^*, X_t^* \in \mathcal{G}_t$, by Proposition 14.45 in [34], $(w, y_t) \to -\phi_t(X_{t-1}^*(w), X_t^*(w), y_t, w)$ is random lsc w.r.t. $\mathcal{G}_t$. Then by Theorem 14.37 in [34], there exists $Y_t^* \in \mathcal{G}_t$ satisfying the requirement in the Assumption 2.4.

B Appendix for Section 3

Lemma B.1. *For any random variables $X : \Omega \to \mathbb{R}^{n_0}$ and $\mathcal{G} \subset \mathcal{F}$, we have $X\mathsf{P}_{\mathcal{G}} = \mathbb{E}[X|\mathcal{G}]$.*

Proof of Lemma B.1. Notice that $X(w) = \sum_{i\in\Omega} X(i)\mathbf{1}[w = i]$, then for all $j \in \Omega$,

$$(X\mathsf{P}_{\mathcal{G}})(j) = \sum_{i\in\Omega} X(i)\mathsf{P}_{\mathcal{G}}(i, j) = \sum_{i\in\Omega} X(i)\mathbb{E}[\mathbf{1}[w = i]|\mathcal{G}](j)$$
$$= \mathbb{E}[\sum_{i\in\Omega} X(i)\mathbf{1}[w = i]|\mathcal{G}](j) = \mathbb{E}[X(w)|\mathcal{G}](j).$$

□

Lemma B.2. *The dual norm satisfies $\|Y_t\|_*^2 = \sum_{w\in\Omega} p_w \|Y_t(w)\|_*^2$.*

Proof of Lemma B.2. Notice that for any $Y_t : \Omega \to \mathbb{R}^{n_t}$,

$$\sup_{X_t:\Omega\to\mathbb{R}^{n_t},\ \|X_t\|\le 1} \langle Y_t, X_t\rangle = \sup_{X_t:\Omega\to\mathbb{R}^{n_t},\ \sum_{w\in\Omega} p_w\|X_t(w)\|^2\le 1} \sum_{w\in\Omega} p_w\langle Y_t(w), X_t(w)\rangle$$
$$= \sup_{s(w)\ge 0,\ \sum_{w\in\Omega} p_w s(w)^2\le 1} \sum_{w\in\Omega} p_w \sup_{\|X_t(w)\|=s(w)} \langle Y_t(w), X_t(w)\rangle$$
$$= \sup_{s(w)\ge 0,\ \sum_{w\in\Omega} p_w s(w)^2\le 1} \sum_{w\in\Omega} p_w s(w)\|Y_t(w)\|_*$$
$$= (\sum_{w\in\Omega} p_w\|Y_t(w)\|_*^2)^{1/2}$$

where we make the change of variable $s(w) = \|X_t(w)\|$, and the last step is by Cauchy-Schwarz. □

B.1 Lemmas regarding the Bregman projection

Proof of Lemma 3.1. Notice that $X' \to \langle G, X'\rangle + D_V(X', X)$ can be viewed as a continuous function over $\mathcal{X}^K$, and since each $\mathcal{X}_t$ is convex compact, the set $\overline{\mathcal{X}}_t$ is a convex compact subset in $\mathbb{R}^{n_t\times K}$, and so $\overline{\mathcal{X}} = \prod_{t=1}^T \overline{\mathcal{X}}_t$ is a convex compact subset. Thus, the argmin is attained. It's also unique since D_V is strongly convex. Thus, X^+ is well defined.

Since $V = \sum_{t=1}^T V_t$ and $\overline{\mathcal{X}} = \prod_{t=1}^T \mathcal{G}_t \cap \mathcal{X}_t$ are decomposable w.r.t t, for $X^+ = (X_1^+, \ldots, X_T^+)$, we have

$$X_t^+ = \operatorname*{argmin}_{X_t'\in\mathcal{G}_t\cap\mathcal{X}_t} \langle G_t, X_t'\rangle + D_{V_t}(X_t', X_t).$$

Notice that for any $X' \in \overline{\mathcal{X}}$, we have $X'_t \in \mathcal{G}_t$, and so by Lemma B.1 $X'_t\mathsf{P}_t = X'_t$, thus for $G_t : \Omega \to \mathbb{R}^{n_t}$,

$$\langle G_t, X'_t\rangle = \langle G_t, X'_t\mathsf{P}_t\rangle = \langle G_t\mathsf{P}_t, X'_t\rangle.$$

Thus, we prove the first claim

$$X_t^+ = \operatorname*{argmin}_{X'_t\in\mathcal{G}_t\cap\mathcal{X}_t} \langle G_t\mathsf{P}_t, X'_t\rangle + D_{V_t}(X'_t, X_t).$$

For the second claim, we first define the function $h : \mathcal{X}_t \times \Omega \to \mathbb{R}$ as

$$h(x_t, w) := \langle G_t\mathsf{P}_t(w), x_t\rangle + D_{v_t}(x_t, X_t(w)).$$

By a similar argument as above, $\operatorname{argmin}_{x_t\in\mathcal{X}_t} h(x_t, w)$ is well defined and unique. In addition, it's easy to check that h is random lsc w.r.t. $\mathcal{G}_t$. Thus, by Theorem 14.37 in [34], $w \to \operatorname{argmin}_{x_t\in\mathcal{X}_t} h(x_t, w)$ is measurable w.r.t. $\mathcal{G}_t$. In particular,

$$\operatorname*{argmin}_{X'_t\in\mathcal{F}\cap\mathcal{X}_t} \langle G_t\mathsf{P}_t, X'_t\rangle + D_{V_t}(X'_t, X_t) = \operatorname*{argmin}_{X'_t\in\mathcal{G}_t\cap\mathcal{X}_t} \langle G_t\mathsf{P}_t, X'_t\rangle + D_{V_t}(X'_t, X_t),$$

and since $\mathcal{F}$ is the power set of $[K]$, the LHS can be decoupled, i.e.

$$X''_t \in \operatorname*{argmin}_{X'_t\in\mathcal{F}\cap\mathcal{X}_t} \langle G_t\mathsf{P}_t, X'_t\rangle + D_{V_t}(X'_t, X_t) \iff X''_t(w) \in \operatorname*{argmin}_{x_t\in\mathcal{X}_t} h(x_t, w)\ \forall w \in \Omega$$

which proves the second claim.

The last claim can be proved by a similar argument as Lemma 3.4 in [21]. □

B.2 Acceleration based on inexact oracle

First, we reformulate (MS-Unconstrained) as the following

$$\min_{X_{1:T}\in\mathcal{Q}} F(X_{1:T}) := \mathbb{E}[f(X_{1:T})], \tag{28}$$

where $\mathcal{Q} = \overline{\mathcal{X}}$. Recall that we equip $\mathbb{R}^n$ with the standard Euclidean inner product, and $\mathbb{R}^{n\times K}$ with the inner product $\langle X_{1:T}, X'_{1:T}\rangle = \mathbb{E}[\sum_{t=1}^T \langle X_t, X'_t\rangle]$. In this section, we assume that all norms are the norms corresponding to the inner products. For simplicity, we abbreviate $X^{(l)}_{1:T}$ as $x^{(l)}$, $\overline{X}^{(l)}_{1:T}$ as $y^{(l)}$, and $\underline{X}^{(l)}_{1:T}$ as $z^{(l)}$. In addition, we denote

$$\phi_l(x) := (1+\gamma)L_2V(x) + \sum_{l'=0}^{l} \alpha_{l'}(F(x^{(l')}) + \langle G^{(l')}, x - x^{(l')}\rangle + \frac{(1-\theta)\mu}{2}\|x - x^{(l')}\|^2)$$

Lemma B.3. *Under the assumptions in Lemma 3.4,*

$$\frac{\mu}{2}\|x - x'\|^2 \le F(x') - F(x) - \langle \nabla F(x), x' - x\rangle \le \frac{L_2}{2}\|x - x'\|^2, \quad \forall x, x' \in \mathcal{Q}. \tag{29}$$

Assume that $\nabla V(x^{(0)}) = \mathbf{0}$, then for all $l \ge 0$, we have $A_lF(y^{(l)}) \le \psi_l(z^{(l)}) + E_l$, where $E_l = \sum_{l'=0}^l A_{l'}\delta_{l'}$.

$$\delta_0 = \langle G^{(0)} - \nabla F(x^{(0)}), x^{(0)} - y^{(0)}\rangle - \frac{\gamma L_2}{2}\|y^{(0)} - x^{(0)}\|^2,$$

and for $l \ge 0$, $\delta_{l+1} = \hat{\delta}_{l+1} + (1-\tau_l)\tilde{\delta}_{l+1}$ where

$$\hat{\delta}_{l+1} = \langle \nabla F(x^{(l+1)}) - G^{(l+1)}, y^{(l+1)} - x^{(l+1)}\rangle - \frac{\gamma L_2}{2}\|y^{(l+1)} - x^{(l+1)}\|^2,$$

$$\tilde{\delta}_{l+1} = \langle G^{(l+1)} - \nabla F(x^{(l+1)}), y^{(l)} - x^{(l+1)}\rangle - \frac{\mu}{2}\|y^{(l)} - x^{(l+1)}\|^2.$$

The proof of Lemma B.3 follows closely [10]. However, [10] considers the case where the gradient inexactness is upper bounded by a constant δ, independent of the query point and the iteration number. In the proof below, we explicitly track the accumulation of the error at each stage.

Proof of Lemma B.3. (29) follows from (9).

For $l = 0$, first, notice that since $\alpha_0 = 1$ and V is 1-strongly convex, we have

$$\begin{aligned}\psi_0(x) &= (1+\gamma)L_2 V(x) + F(x^{(0)}) + \langle G^{(0)}, x - x^{(0)}\rangle + \frac{(1-\theta)\mu}{2}\|x - x^{(0)}\|^2 \\ &\geq \frac{(1+\gamma)L_2}{2}\|x - x^{(0)}\|^2 + F(x^{(0)}) + \langle G^{(0)}, x - x^{(0)}\rangle\end{aligned}$$

Thus, we have

$$\begin{aligned}\psi_0(z^{(0)}) = \min_{x\in\mathcal{Q}} \psi_0(x) &\geq \min_{x\in\mathcal{Q}} \frac{(1+\gamma)L_2}{2}\|x - x^{(0)}\|^2 + F(x^{(0)}) + \langle G^{(0)}, x - x^{(0)}\rangle \\ &= \frac{(1+\gamma)L_2}{2}\|y^{(0)} - x^{(0)}\|^2 + F(x^{(0)}) + \langle G^{(0)}, y^{(0)} - x^{(0)}\rangle \\ &= \frac{L_2}{2}\|y^{(0)} - x^{(0)}\|^2 + F(x^{(0)}) + \langle \nabla F(x^{(0)}), y^{(0)} - x^{(0)}\rangle - \delta_0\end{aligned}$$

where $\delta_0 = \langle G^{(0)} - \nabla F(x^{(0)}), x^{(0)} - y^{(0)}\rangle - \frac{\gamma L_2}{2}\|y^{(0)} - x^{(0)}\|^2$. Assume now that the statement holds for some $l \geq 0$. By the update of $z^{(l)}$, we have

$$\langle (1+\gamma)L_2\nabla V(z^{(l)}) + \sum_{l'=0}^{l} \alpha_{l'} G^{(l')} + (1-\theta)\mu\alpha_{l'}(z^{(l)} - x^{(l')}), x - z^{(l)}\rangle \geq 0, \quad \forall x \in \mathcal{Q}.$$

Then, the strong convexity of V implies that

$$\begin{aligned}L_2 V(x) &\geq L_2 V(z^{(l)}) + \langle L_2\nabla V(z^{(l)}), x - z^{(l)}\rangle + \frac{L_2}{2}\|x - z^{(l)}\|^2 \\ &\geq L_2 V(z^{(l)}) + \frac{L_2}{2}\|x - z^{(l)}\|^2 \\ &\quad - (1+\gamma)^{-1}\langle \sum_{l'=0}^{l} \alpha_{l'} G^{(l')} + (1-\theta)\mu\alpha_{l'}(z^{(l)} - x^{(l')}), x - z^{(l)}\rangle\end{aligned}$$

Thus we have

$$\begin{aligned}\psi_{l+1}(x) \geq\ & (1+\gamma)L_2 V(z^{(l)}) + \frac{(1+\gamma)L_2}{2}\|x - z^{(l)}\|^2 \\ & - \langle \sum_{l'=0}^{l} \alpha_{l'} G^{(l')} + (1-\theta)\mu\alpha_{l'}(z^{(l)} - x^{(l')}), x - z^{(l)}\rangle \\ & + \sum_{l'=0}^{l} \alpha_{l'}(F(x^{(l')}) + \langle G^{(l')}, x - x^{(l')}\rangle + \frac{(1-\theta)\mu}{2}\|x - x^{(l')}\|^2) \\ & + \alpha_{l+1}(F(x^{(l+1)}) + \langle G^{(l+1)}, x - x_{l+1}\rangle + \frac{(1-\theta)\mu}{2}\|x - x^{(l+1)}\|^2)\end{aligned}$$

Using

$$\langle z^{(l)} - x^{(l')}, z^{(l)} - x\rangle = \frac{1}{2}\|z^{(l)} - x^{(l')}\|^2 + \frac{1}{2}\|z^{(l)} - x\|^2 - \frac{1}{2}\|x - x^{(l')}\|^2,$$

we have

$$\begin{aligned}\psi_{l+1}(x) \geq\ & (1+\gamma)L_2 V(z^{(l)}) + \frac{(1+\gamma)L_2 + A_l(1-\theta)\mu}{2}\|x - z^{(l)}\|^2 \\ & + \sum_{l'=0}^{l} \alpha_{l'}(F(x^{(l')}) + \langle G^{(l')}, z^{(l)} - x^{(l')}\rangle + \frac{(1-\theta)\mu}{2}\|z^{(l)} - x^{(l')}\|^2) \\ & + \alpha_{l+1}(F(x^{(l+1)}) + \langle G^{(l+1)}, x - x^{(l+1)}\rangle + \frac{(1-\theta)\mu}{2}\|x - x^{(l+1)}\|^2) \\ =\ & \psi_l(z^{(l)}) + \frac{(1+\gamma)L_2 + A_l(1-\theta)\mu}{2}\|x - z^{(l)}\|^2 \\ & + \alpha_{l+1}(F(x^{(l+1)}) + \langle G^{(l+1)}, x - x^{(l+1)}\rangle + \frac{(1-\theta)\mu}{2}\|x - x^{(l+1)}\|^2).\end{aligned}$$

By induction hypothesis,

$$A_l^{-1}(\psi_l(z^{(l)}) + E_l) \geq F(y^{(l)}) \geq F(x^{(l+1)}) + \langle G^{(l+1)}, y^{(l)} - x^{(l+1)}\rangle - \tilde{\delta}_{l+1}$$

where

$$\tilde{\delta}_{l+1} = \langle G^{(l+1)} - \nabla F(x^{(l+1)}), y^{(l)} - x^{(l+1)}\rangle - \frac{\mu}{2}\|y^{(l)} - x^{(l+1)}\|^2.$$

Thus we have

$$\begin{aligned}\psi_{l+1}(x) \geq\ & A_l(F(x^{(l+1)}) + \langle G^{(l+1)}, y^{(l)} - x^{(l+1)}\rangle - \tilde{\delta}_{l+1}) - E_l \\ & + \frac{(1+\gamma)L_2 + A_l(1-\theta)\mu}{2}\|x - z^{(l)}\|^2 \\ & + \alpha_{l+1}(F(x^{(l+1)}) + \langle G^{(l+1)}, x - x^{(l+1)}\rangle + \frac{(1-\theta)\mu}{2}\|x - x^{(l+1)}\|^2) \\ \geq\ & A_{l+1}F(x^{(l+1)}) + \alpha_{l+1}\langle G^{(l+1)}, x - z^{(l)}\rangle - A_l\tilde{\delta}_{l+1} - E_l \\ & + \frac{(1+\gamma)L_2 + A_l(1-\theta)\mu}{2}\|x - z^{(l)}\|^2\end{aligned}$$

where the last equality is by noticing that

$$A_l(y^{(l)} - x^{(l+1)}) + \alpha_{l+1}(x - x^{(l+1)}) = \alpha_{l+1}(x - z^{(l)}).$$

Therefore, we have

$$\begin{aligned}\psi_{l+1}(z^{(l+1)}) \geq & A_{l+1}F(x^{(l+1)}) - A_l\tilde{\delta}_{l+1} - E_l \\ & + A_{l+1} \cdot \min_{x\in\mathcal{Q}} \tau_l\langle G^{(l+1)}, x - z^{(l)}\rangle + \frac{(1+\gamma)L_2\tau_l^2}{2}\|x - z^{(l)}\|^2.\end{aligned}$$

Notice that defining $y = \tau_l x + (1-\tau_l)y^{(l)}$, we get $y - x^{(l+1)} = \tau_l(x - z^{(l)})$ and defining $\mathcal{Q}' = \tau_l\mathcal{Q} + (1-\tau_l)y^{(l)} \subset \mathcal{Q}$, we have

$$\begin{aligned}& \min_{x\in\mathcal{Q}} \tau_l\langle G^{(l+1)}, x - z^{(l)}\rangle + \frac{(1+\gamma)L_2\tau_l^2}{2}\|x - z^{(l)}\|^2 \\ =\ & \min_{x\in\mathcal{Q}'}\langle G^{(l+1)}, y - x^{(l+1)}\rangle + \frac{(1+\gamma)L_2}{2}\|y - x^{(l+1)}\|^2 \\ \geq\ & \min_{x\in\mathcal{Q}}\langle G^{(l+1)}, y - x^{(l+1)}\rangle + \frac{(1+\gamma)L_2}{2}\|y - x^{(l+1)}\|^2.\end{aligned}$$

Thus, we have

$$\begin{aligned}
&\psi_{l+1}(z^{(l+1)}) + E_l \\
&\geq A_{l+1} \min_{y\in\mathcal{Q}}(\langle G^{(l+1)}, y - x^{(l+1)}\rangle + \frac{(1+\gamma)L_2}{2}\|y - x^{(l+1)}\|^2 + F(x^{(l+1)})) - A_l\tilde{\delta}_{l+1} \\
&= A_{l+1}(\langle G^{(l+1)}, y^{(l+1)} - x^{(l+1)}\rangle + \frac{(1+\gamma)L_2}{2}\|y^{(l+1)} - x^{(l+1)}\|^2 + F(x^{(l+1)})) - A_l\tilde{\delta}_{l+1} \\
&\geq A_{l+1}F(y^{(l+1)}) - A_l\tilde{\delta}_{l+1} - A_{l+1}\hat{\delta}_{l+1}
\end{aligned}$$

where

$$\hat{\delta}_{l+1} = \langle \nabla F(x^{(l+1)}) - G^{(l+1)}, y^{(l+1)} - x^{(l+1)}\rangle - \frac{\gamma L_2}{2}\|y^{(l+1)} - x^{(l+1)}\|^2.$$

Taking $\delta_{l+1} = \hat{\delta}_{l+1} + (1-\tau_l)\tilde{\delta}_{l+1}$ proves the result for $l+1$. □

Lemma B.4. *For any $l \geq 0$, we have*

$$\begin{aligned}
F(y^{(l)}) \leq\ & F^* + (1+\gamma)L_2A_l^{-1}V(x^*) + A_l^{-1}E_l \\
&+ A_l^{-1}\sum_{l'=0}^{l}\alpha_{l'}(\langle G^{(l')} - \nabla F(x^{(l')}), x^* - x^{(l')}\rangle - \frac{\theta\mu}{2}\|x^* - x^{(l')}\|^2).
\end{aligned}$$

where $E_l = \sum_{l'=0}^{l} A_{l'}\delta_{l'}$ is as defined in Lemma B.3.

Proof of Lemma B.4. Notice that from the definition of ψ_l and $z^{(l)}$, we have

$$\begin{aligned}
&\psi_l(z^{(l)}) - (1+\gamma)L_2V(x^*) = \min_{x\in\mathcal{Q}}\psi_l(x) - (1+\gamma)L_2V(x^*) \\
&\leq \psi_l(x^*) - (1+\gamma)L_2V(x^*) \\
&= \sum_{l'=0}^{l}\alpha_{l'}(F(x^{(l')}) + \langle G^{(l')}, x^* - x^{(l')}\rangle + \frac{(1-\theta)\mu}{2}\|x^* - x^{(l')}\|^2) \\
&\leq A_lF^* + \sum_{l'=0}^{l}\alpha_{l'}(\langle G^{(l')} - \nabla F(x^{(l')}), x^* - x^{(l')}\rangle - \frac{\theta\mu}{2}\|x^* - x^{(l')}\|^2).
\end{aligned}$$

In addition, from Lemma B.3, we have

$$\begin{aligned}
F(y^{(l)}) &\leq A_l^{-1}(\psi_l(z^{(l)}) + E_l) \\
&= F^* + (1+\gamma)L_2A_l^{-1}V(x^*) + A_l^{-1}E_l \\
&\quad + A_l^{-1}\sum_{l'=0}^{l}\alpha_{l'}(\langle G^{(l')} - \nabla F(x^{(l')}), x^* - x^{(l')}\rangle - \frac{\theta\mu}{2}\|x^* - x^{(l')}\|^2).
\end{aligned}$$

□

Lemma B.5. *If $\mu > 0$, the sequence A_l satisfies that*

$$(1 + \frac{1}{2}\sqrt{\frac{(1-\theta)\mu}{(1+\gamma)L_2}})^2A_l \leq A_{l+1}, \quad l = 0, 1, \ldots.$$

If $\mu = 0$, the sequence α_l satisfies that

$$\frac{1}{2}(l+1) \leq \alpha_l \leq l+1, \quad l = 0, 1, \ldots.$$

Proof of Lemma B.5. The result for the case when $\mu > 0$ is from Lemma 4 in [10]. For $\mu = 0$, notice that $A_l + \alpha_{l+1} = \alpha_{l+1}^2$. The claim is true for $l = 0$ since $\alpha_0 = 1$. Suppose the statement is true for l, thus

$$\frac{(l+1)(l+2)}{4} = \frac{1}{2}\sum_{i=0}^{l}(i+1) \le A_l \le \sum_{i=0}^{l}(i+1) = \frac{(l+1)(l+2)}{2}.$$

Thus, for $l+1$, notice that $\alpha_{l+1} = \frac{1}{2} + \sqrt{A_l + \frac{1}{4}}$, we have

$$\alpha_{l+1} \ge \frac{1}{2} + \sqrt{A_l} \ge \frac{1}{2} + \frac{l+1}{2} = \frac{l+2}{2}.$$

The upper bound follows from the following

$$A_l + \frac{1}{4} \le \frac{(l+1)(l+2)}{2} + \frac{1}{4} = \frac{2l^2+6l+5}{4} \le \frac{4l^2+12l+9}{4} = (l+\frac{3}{2})^2.$$

□

Proof of Lemma 3.4. The proof follows from Lemma B.4. We use $\sum_{t=1}^{T}\tilde{V}_t(X_t)$ where $\tilde{V}_t(X_t) = V_t(X_t) - \langle \nabla v_t(X_t^{(0)}), X_t - X_t^{(0)}\rangle$ and V_t is the distance generating function defined in Section 3.1. In addition, by first order optimality condition, at $X^{(0)}$, $\nabla\tilde{v}_t(X^{(0)})(w) = 0$ for all $w \in \Omega$, thus $X^{(0)}$ is the minimum of V. In addition, $D_{V_t}(\cdot,\cdot) = D_{\tilde{V}_t}(\cdot,\cdot)$ are the same, since adding a linear function does not change the induced Bregman divergence.

Below, following the notation for Lemma B.3, $y^{(l)} = \overline{X}_{1:T}^{(l)}$, $z^{(l)} = \underline{X}_{1:T}^{(l)}$, and $x^{(l)} = X_{1:T}^{(l)}$. For convenience, by $\mathsf{P}_{\mathcal{G}_{1:T}}$, we mean a $KT \times KT$ block diagonal matrix, where the diagonal matrices are $\mathsf{P}_1, \ldots, \mathsf{P}_t$, and defining $\Delta^{(l)} = \left(G^{(l)} - \nabla F(x^{(l)})\right)\mathsf{P}_{\mathcal{G}_{1:T}}$.

With $\gamma = 1$,

$$\delta_0 = \langle G^{(0)} - \nabla F(x^{(0)}), x^{(0)} - y^{(0)}\rangle - \frac{\gamma L_2}{2}\|y^{(0)} - x^{(0)}\|^2 \le \frac{\|\Delta^{(0)}\|^2}{2\gamma L_2} = \frac{\|\Delta^{(0)}\|^2}{2L_2}.$$

Similarly $\hat{\delta}_{l+1} \le \frac{\|\Delta^{(l+1)}\|^2}{2L_2}$. Thus, for any $\mu \ge 0$,

$$\begin{aligned} F(y^{(l)}) \le F^* + 2L_2 A_l^{-1} V(x^*) + A_l^{-1}\sum_{l'=0}^{l} A_{l'}\frac{\|\Delta^{(l')}\|^2}{2L_2} + A_l^{-1}\langle \Delta^{(0)}, x^* - x^{(0)}\rangle \\ + A_l^{-1}\sum_{l'=1}^{l}\langle \Delta^{(l')}, \alpha_{l'}(x^* - x^{(l')}) + A_{l'-1}(y^{(l'-1)} - x^{(l')})\rangle \end{aligned}$$

Thus, the result follows from taking for $l' \ge 0$, $A_{-1} = 0$,

$$\overline{\Delta}^{(l')} = A_{l'}\frac{\|\Delta^{(l')}\|^2}{2L_2} + \langle \Delta^{(l')}, \alpha_{l'}(x^* - x^{(l')}) + A_{l'-1}(y^{(l'-1)} - x^{(l')})\rangle.$$

Further assuming that $\mu > 0$ and taking $\theta = 1/2$, we get

$$\langle G^{(l')} - \nabla F(x^{(l')}), x^* - x^{(l')}\rangle - \frac{\theta\mu}{2}\|x^* - x^{(l')}\|^2 \le \frac{\|\Delta^{(l')}\|^2}{2\theta\mu} = \frac{\|\Delta^{(l')}\|^2}{\mu}.$$

$$\tilde{\delta}_{l+1} = \langle G^{(l+1)} - \nabla F(x^{(l+1)}), y^{(l)} - x^{(l+1)} \rangle - \frac{\mu}{2}\|y^{(l)} - x^{(l+1)}\|^2 \leq \frac{\|\Delta^{(l+1)}\|^2}{2\mu}.$$

Thus, for all $l \geq 0$, $\delta_l \leq \frac{\|\Delta^{(l)}\|^2}{2\mu} + \frac{\|\Delta^{(l)}\|^2}{2L_2}$. Thus,

$$F(y^{(l)}) \leq F^* + 2L_2 A_l^{-1} V(x^*) + A_l^{-1} \sum_{l'=0}^{l} (\frac{\alpha_{l'}}{\mu} + \frac{A_{l'}}{2\mu} + \frac{A_{l'}}{2L_2}) \|\Delta^{(l')}\|^2.$$

The result follows from taking $\overline{\Delta}^{(l')} = (\frac{\alpha_{l'}}{\mu} + \frac{A_{l'}}{2\mu} + \frac{A_{l'}}{2L_2})\|\Delta^{(l')}\|^2$. □

C Discussion on the bound on $\mathcal{Y}_t$

Noticing that for any $Y_t \in \mathcal{G}_t \cap \mathcal{Y}_t$ for all $t \in [T]$, minimizing (27) over $(X_t, \mathbf{b}_t)$, (26) becomes maximizing

$$\sum_{t=1}^{T} \mathbb{E}[-[c_t([w]_t) - \langle Y_t([w]_t), \mathbf{a}_t([w]_t)\rangle]_+ - \langle [Y_{t+1}\mathsf{P}_t - Y_t]_+, \mathbf{b}_0\rangle] - \mathbb{E}[\langle Y_1, \mathbf{b}_0\rangle] \tag{30}$$

where $Y_{T+1} = \mathbf{0}$, $[s]_+ = \max(s, 0)$ for $s \in \mathbb{R}$ and is applied component-wise if applied to a vector.

Now, suppose that $\underline{a}, \overline{c} > 0$, and $\mathbf{a}_t(w) \geq \underline{a}\mathbf{1}$ and $c_t(w) \leq \overline{c}$ for all $w \in \Omega$ and all t. For any $s \in \mathbb{R}$, we use $\Pi(s)$ to denote the projection of s to the interval $[0, \frac{\overline{c}}{\underline{a}} + 1]$ and $\mathcal{Y}'_t = [0, \frac{\overline{c}}{\underline{a}} + 1]^M$. If Π is applied to a (random) vector, we use it component-wise and scenario wise. Then, it's easy to see that for any $Y_t([w]_t) \in [0, \infty)^M$,

$$[c_t([w]_t) - \langle Y_t([w]_t), \mathbf{a}_t([w]_t)\rangle]_+ = [c_t([w]_t) - \langle \Pi(Y_t([w]_t)), \mathbf{a}_t([w]_t)\rangle]_+,$$

since if there is an index $i \in [M]$, such that $Y_t([w]_t)_i > \frac{\overline{c}}{\underline{a}} + 1$, then

$$\langle Y_t([w]_t), \mathbf{a}_t([w]_t)\rangle \geq (\frac{\overline{c}}{\underline{a}} + 1) \cdot \underline{a} > \overline{c},$$

and so the LHS is 0. After projection, the i-th index $\Pi(Y_t([w]_t))_i = \frac{\overline{c}}{\underline{a}} + 1$ and so the RHS is also 0. Suppose $Y_t([w]_t) \in \mathcal{Y}'_t$ then the $Y_t([w]_t) = \Pi(Y_t([w]_t))$ and so the LHS and the RHS are the same. For the second term in (30), notice that for each coordinate $i \in [M]$,

$$\begin{aligned}[Y_{t+1,i}\mathsf{P}_t([w]_t) - Y_{t,i}([w]_t)]_+ &\geq [\Pi(Y_{t+1,i}\mathsf{P}_t([w]_t)) - \Pi(Y_{t,i}([w]_t))]_+ \\ &\geq [\Pi(Y_{t+1,i})\mathsf{P}_t([w]_t) - \Pi(Y_{t,i}([w]_t))]_+\end{aligned}$$

where the first inequality is by $[a - b]_+ \geq [\Pi(a) - \Pi(b)]_+$ for any $a, b \geq 0$. The second inequality is by noticing that $[\cdot]_+$ is non-decreasing, and $\Pi(\mathbb{E}[W]) \geq \mathbb{E}[\Pi(W)]$ for any random variable $W \geq 0$. Thus,

$$\mathbb{E}[-\langle [Y_{t+1}\mathsf{P}_t - Y_t]_+, \mathbf{b}_0\rangle] \leq \mathbb{E}[-\langle [\Pi(Y_{t+1})\mathsf{P}_t - \Pi(Y_t)]_+, \mathbf{b}_0\rangle].$$

For the last term in (30), since projection does not increase the components of Y_1, we have $-\mathbb{E}[\langle Y_1, \mathbf{b}_0\rangle] \leq -\mathbb{E}[\langle \Pi(Y_1), \mathbf{b}_0\rangle]$. Thus, if (27) has an optimal solution, then it has an optimal solution such that $Y_t \in \mathcal{G}_t \cap \mathcal{Y}'_t$ for all t.