Journal of Machine Learning Research 21 (2020) 1-45 Submitted 12/19; Revised 4/20; Published 6/20

A Data Efficient and Feasible Level Set Method for Stochastic Convex Optimization with Expectation Constraints

Qihang Lin QIHANG-LIN@UIOWA.EDU
Tippie College of Business
University of Iowa
Iowa City, Iowa, 52242, USA

Selvaprabu Nadarajah SELVAN@UIC.EDU
College of Business Administration
University of Illinois at Chicago
Chicago, Illinois, 60607, USA

Negar Soheili NAZAD@UIC.EDU
College of Business Administration
University of Illinois at Chicago
Chicago, Illinois, 60607, USA

Tianbao Yang TIANBAO-YANG@UIOWA.EDU
Department of Computer Science
University of Iowa
Iowa City, Iowa, 52242, USA

Editor: Julien Mairal

Abstract

Stochastic convex optimization problems with expectation constraints (SOECs) are encountered in statistics and machine learning, business, and engineering. The SOEC objective and constraints contain expectations defined with respect to complex distributions or large data sets, leading to high computational complexity when solved by the algorithms that use exact functions and their gradients. Recent stochastic first order methods exhibit low computational complexity when handling SOECs but guarantee near-feasibility and near-optimality only at convergence. These methods may thus return highly infeasible solutions when heuristically terminated, as is often the case, due to theoretical convergence criteria being highly conservative. This issue limits the use of first order methods in several applications where the SOEC constraints encode implementation requirements. We design a stochastic feasible level set method (SFLS) for SOECs that has low complexity and emphasizes feasibility before convergence. Specifically, our level-set method solves a root-finding problem by calling a novel first order oracle that computes a stochastic upper bound on the level-set function by extending mirror descent and online validation techniques. We establish that SFLS maintains a high-probability feasible solution at each root-finding iteration and exhibits favorable complexity compared to state-of-the-art deterministic feasible level set and stochastic subgradient methods. Numerical experiments on three diverse applications highlight how SFLS finds feasible solutions with small optimality gaps with lower complexity than the former approaches.

Keywords: constrained stochastic optimization, level set methods, stochastic gradient methods, min-max optimization, online validation

1. Introduction

Consider the stochastic optimization problem with expectation constraints (SOEC)

$$f^* := \min_{\mathbf{x}\in\mathcal{X}} \{f_0(\mathbf{x}) = \mathbb{E}\left[F_0(\mathbf{x},\xi_0)\right]\} \quad \text{s.t.} \quad f_i(\mathbf{x}) := \mathbb{E}\left[F_i(\mathbf{x},\xi_i)\right] \leq r_i,\ i = 1,2,\ldots,m, \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^d$ is a nonempty closed convex set, ξ_i, $i = 0,1,\ldots,m$, is a random vector whose probability distribution is supported on set $\Xi_i \subseteq \mathbb{R}^{q_i}$, and $F_i(\mathbf{x},\xi_i) : \mathcal{X} \times \Xi_i \to \mathbb{R}$ is continuous and convex in $\mathbf{x}$ for each realization of ξ_i for $i = 0,1,2,\ldots,m$. Given $\epsilon > 0$, a solution $\mathbf{x}_\epsilon \in \mathcal{X}$ is called ϵ-*feasible* if $\max_{i=1,\ldots,m}\{f_i(\mathbf{x}_\epsilon) - r_i\} \leq \epsilon$. A solution $\mathbf{x}_\epsilon \in \mathcal{X}$ is referred to as ϵ-*optimal* if $f_0(\mathbf{x}_\epsilon) - f^* \leq \epsilon$. Alternatively, optimality can be measured relative to an initial feasible solution $\mathbf{x}^0 \in \mathcal{X}$. In this case, we say $\mathbf{x}_\epsilon \in \mathcal{X}$ is *relative* ϵ-*optimal* with respect to $\mathbf{x}^0$ if $(f(\mathbf{x}_\epsilon) - f^*)/(f(\mathbf{x}^0) - f^*) \leq \epsilon$.

Problem (1) is pervasive in stochastic optimization and appears as a central challenge in semi-supervised learning (Chapelle et al., 2009), shape-restricted regression (Seijo and Sen, 2011; Sen and Meyer, 2017; Lim, 2014; Cotter et al., 2016; Fard et al., 2016), Neyman-Pearson classification (Tong et al., 2016; Rigollet and Tong, 2011; Tong, 2013; Zhao et al., 2016), approximate linear programming and related relaxations (de Farias and Van Roy, 2003; Adelman and Mersereau, 2013; Nadarajah et al., 2015), portfolio selection (Markowitz, 1952; Abdelaziz et al., 2007), risk management (Rockafellar and Uryasev, 2000), and multi-objective stochastic programming (Marler and Arora, 2004; Abdelaziz, 2012; Mahdavi et al., 2013; Barba-Gonzaléz et al., 2017). In this paper, we focus on overcoming the challenges of applying existing methods for solving SOECs in settings that are both data rich and where expectation constraints capture requirements that cannot be violated during real-world implementation.

In data-rich environments, each expectation appearing in (1) is defined by a data set containing a large number of data points (possibly infinite). The number of data points used when solving SOEC is an important computational bottleneck, which we refer to as the data complexity of an algorithm. Traditional approaches for solving SOECs can lead to large data complexity. For instance, consider the popular strategy of replacing each expectation in (1) by a sample average approximation (SAA; Shapiro, 2013; Oliveira and Thompson, 2017) and solving the resulting model using a deterministic iterative method (see, e.g., Nesterov, 2004; Soheili and Pena, 2012, and references therein). If the number of samples used to construct SAAs is small, the solution from the deterministic approximation may be highly infeasible to the original SOEC, in addition to being suboptimal (Shapiro, 2013; Oliveira and Thompson, 2017). Instead, if a large number of samples are used in each SAA, then the data complexity becomes large because the gradient or objective function evaluation at each iteration requires using a significant portion of each of the data sets.

In contrast, stochastic first order methods for tackling stochastic optimization problems have low per-iteration cost and data complexity and thus play a central role in machine learning packages such as TensorFlow and PyTorch (Robbins and Monro, 1951; Nemirovski et al., 2009; Lan, 2012; Ghadimi and Lan, 2012, 2013; Chen et al., 2012; Lan et al., 2012; Schmidt et al., 2017; Shalev-Shwartz et al., 2017; Lan and Zhou, 2018; Lin et al., 2014;

Duchi and Singer, 2009; Xiao and Zhang, 2014; Xiao, 2010; Hazan and Kale, 2011; Bach and Moulines, 2013; Allen-Zhu, 2017; Goldfarb et al., 2017). These methods update solutions using stochastic gradients that can be computed using a small number of sampled data points. Stochastic first order methods typically ensure feasibility via projections onto a convex set at each iteration, where the convex set is assumed to be simple (e.g. a box or ball) for computational tractability. This assumption limits the applicability of first order methods for solving SOECs with general non-linear constraints. Recently, Lan and Zhou (2016) and Yu et al. (2017) developed stochastic subgradient (SSG) methods devoid of projections for solving (1) with single ($m = 1$) and multiple constraints ($m > 1$), respectively. The SSG methods in these papers guarantee an ϵ-optimal and ϵ-feasible solution only at convergence.

In practice, SSG methods are terminated before their conservative theoretical conditions are met. Premature termination may lead to highly infeasible and sub- or super- optimal solutions. While some deviation from optimality is likely acceptable, a highly infeasible solution may not be implementable. Such situations arise in several data science applications in machine learning, as well as, across business (e.g., operations and finance) and engineering domains. We elaborate on the practical need for feasibility in a few cases below.

- Fairness constraints: Enforcing fairness criteria when learning classifiers across multiple classes (e.g., male and female) has become important in machine learning (Goh et al., 2016). This learning problem can be cast as an SOEC where fairness is modeled via expectation constraints. Constraint violations lead to classifiers that are biased towards one or more classes.

- Risk constraints: Planning problems in supply chain management and portfolio optimization often include bounds on the Conditional Value at Risk (CVaR), which can be cast as expectation constraints (Fábián, 2008; Chen et al., 2010). Such constraints also arise when modeling distributionally robust versions of chance constraints (Wiesemann et al., 2014) and when limiting misclassification risk (i.e., misclassification rates) in multi-class Neyman Pearson classification (Crammer and Singer, 2002). The aforementioned problems can be formulated as SOECs. Solutions violating risk constraints will likely fail stress tests that are performed before implementation.

- Bounding property: Approximate linear programs (ALPs) are well-known models for approximating the value function of high-dimensional Markov decision processes (Schweitzer and Seidmann, 1985; de Farias and Van Roy, 2003), and in particular, are SOECs. A solution satisfying the ALP constraints provides an optimistic bound on the optimal policy value, which is useful to evaluate the suboptimality of heuristic policies. Infeasibility in an ALP setting thus voids this desirable bounding property.

Motivated by the importance of feasibility and the status quo of stochastic first order methods, we design an approach for solving SOECs that has low data complexity and provides high probability feasible solutions before convergence. As a first step, we cast SOEC as a root-finding problem involving a min-max level set function, which is challenging to solve because it is non-smooth and includes high-dimensional expectations in the SOEC objective and constraints. To solve this reformulation, we develop a stochastic feasible level-set method (SFLS) for root finding that requires evaluating a "good" upper bound (we will make this notion of goodness precise in later sections) on the challenging level set function at each iteration. We show that employing the mirror descent method (Nemirovski et al., 2009) for computing such an upper bound requires approximating expectations in SOEC using SAAs at each iteration, which as already discussed above, leads to high data complexity. To overcome this issue, we introduce an SSG method to upper bound the level-set function by combining mirror-decent and online validation techniques, and in particular, extending the latter technique, originally proposed for minimization problems (Lan et al., 2012), to handle saddle point formulations. This method only requires stochastic values and gradients of the objective and constraint functions, respectively, which can be constructed at low cost using a small number of samples of ξ_i in (1), that is, it has low data complexity. Calls to our SSG method return high-probability feasible solutions, which allows it to maintain an implementable solution at each root-finding iteration.

We analyze the iteration complexity of SFLS to find a feasible solution path (i.e., sequence of feasible solutions) that becomes relative ϵ-optimal with high probability. It is encouraging that the dependence of this complexity on ϵ is $1/\epsilon^2$, which is comparable to the method by Yu et al. (2017) (labeled YNW [1]) that also finds an ϵ-optimal solution but only guarantees ϵ-feasibility at convergence. In other words, the intermediate solutions generated by YNW are not necessarily feasible. There is indeed a cost for ensuring feasibility in SFLS, which appears in the form of its iteration complexity depending on a condition measure. Such condition measures do not influence the complexity of YNW.

For deterministic constrained convex optimization problems, the level-set method (DFLS) of Lin et al. (2018b) also guarantees a feasible solution path with its iteration complexity depending on a condition measure. In principle, these DFLS based approaches can be applied to solve SOECs by viewing them as deterministic problems. This perspective is restrictive because it entails computing expectations in f_i for $i = 0, 1, \ldots, m$ exactly or replacing them by SAAs. In either case, the data complexity of DFLS will be high for reasons analogous to the ones already discussed above related to the use of SAAs. Therefore, a fully stochastic approach is required to achieve low data complexity when solving SOECs. Lin et al. (2018a) extend DFLS using variance-reduced sampling, which requires the functions have a finite-sum structure with each summand taking a specific form.[2] Unfortunately, as a result, their method cannot be applied to SOECs with generic expectation while our method does not have such limitation and assumes little structure on the problems. We are not aware of prior efforts to develop a fully stochastic versions of level set methods for SOECs—SFLS in this paper fills this gap.

To assess the performance of SFLS, we provide implementation guidelines with supporting theory and numerically evaluate SFLS on three applications: (i) approximate linear programming for Markov decision processes, (ii) Neyman-Pearson multi-class classification with risk constraints, and (iii) learning a classifier with fairness constraints. Feasibility plays a key role in each of these applications for reasons mentioned earlier in the introduction. Approximate linear programs in the first application are known special cases of SOECs. For the latter two applications, we propose formulations that are SOECs. As algorithmic benchmarks, we consider YNW and DFLS. We find that SFLS delivers feasible solutions quicker than YNW and in several cases also leads to smaller optimality gaps. Moreover,

[1] We abbreviate this method by YNW using the first letters of the last names of the authors.

[2] In particular, Lin et al. (2018a) require each summand has the form of $\phi(\mathbf{x}^\top \xi)$.

when YNW computes infeasible solutions it is challenging to interpret its objective value since it can be superoptimal, an issue that does not arise with SFLS. Both SFLS and DFLS maintain feasible solution paths (with outer iterates) but SFLS produces feasible solutions with much smaller optimality gaps due to its lower data complexity. In other words, DFLS requires significantly more data passes to reduce the suboptimality of its solutions and will thus not be practical for solving SOECs based on large data sets. Our findings underscore two important algorithmic insights: (i) feasible SOEC solutions can be computed well before theoretical convergence criteria are satisfied but doing this hinges on methods being able to emphasize feasibility; and (ii) ensuring that these early feasible solutions have small optimality gaps requires approaches with low data complexity. Both these properties are true for SFLS, while only the first and second properties, respectively, hold for DFLS and YNW.

This paper is organized as follows. In §2, we introduce SFLS, analyze its oracle complexity, and present a saddle-point reformulation of an SOEC. In §3, we discuss how the well-known stochastic mirror descent algorithm provides an idealized stochastic oracle for SFLS and highlight issues that complicate its use. In §4, we propose and analyze a new stochastic oracle to overcome these issues. In §5, we analyze SFLS combined with this oracle and provide implementation guidelines. In §6, we perform a computational study to understand the performance of SFLS across three applications relative to two benchmark methods. We conclude in §7.

2. Stochastic Feasible Level-set Method

Level-set methods tackle a constrained convex optimization problem by transforming it into a one-dimensional root-finding problem that is a function of a scalar level parameter r (Lemaréchal et al., 1995; Nesterov, 2004). We develop in this section a stochastic and feasible level set method that adds to this framework. We make the following standard assumption throughout the paper, which ensures that a strictly feasible and sub-optimal solution exists.

Assumption 1 (Strict Feasibility) *There exists a strictly feasible solution* $\tilde{\mathbf{x}} \in \mathcal{X}$ *such that* $\max_{i=1,\ldots,m}\{f_i(\tilde{\mathbf{x}}) - r_i\} < 0$ *and* $f_0(\tilde{\mathbf{x}}) > f^*$.

The root-finding reformulation of (1) relies on the *level-set function*

$$H(r) := \min_{\mathbf{x}\in\mathcal{X}} \mathcal{P}(r, \mathbf{x}), \tag{2}$$

where $r \in \mathbb{R}$ is a *level parameter* and

$$\mathcal{P}(r, \mathbf{x}) := \max\left\{f_0(\mathbf{x}) - r, f_1(\mathbf{x}) - r_1, \ldots, f_m(\mathbf{x}) - r_m\right\}.$$

Note that the expectation constraints of SOEC are now in the objective function of (2). For a given $(r, \mathbf{x}) \in \mathbb{R} \times \mathcal{X}$, if $\mathcal{P}(r, \mathbf{x}) \leq 0$ then $\mathbf{x}$ is a feasible solution to (1). Formulations (1) and (2) are further linked by known properties of $H(r)$, which are summarized in the following lemma (based on lemmas 2.3.4 and 2.3.6 in Nesterov, 2004 and Lemma 1 in Lin et al., 2018b).

Lemma 1 *It holds that*

(a) $H(r)$ is non-increasing and convex in r;

(b) $H(f^*) = 0$;

(c) $H(r) > 0$, if $r < f^*$ and $H(r) < 0$, if $r > f^*$.

Part (a) of Lemma 1 highlights that $H(r)$ is non-increasing and convex. Moreover, its part (b) implies that $r = f^*$ is the unique root of $H(r) = 0$. Therefore, one can use a root finding procedure to generate both a sequence of level parameters $r^{(1)}, r^{(2)}, \ldots$ that converges to f^* and an associated vector $\mathbf{x}^{(k)} := \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{P}(r^{(k)}, \mathbf{x})$ at each iteration k. Computationally, when a level parameter $r^{(k^*)} \approx f^*$ is found, the solution $\mathbf{x}^{(k^*)} := \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{P}(r^{(k^*)}, \mathbf{x})$ provides an "approximate" solution to (1). From the perspective of feasibility, it is important whether we have $r^{(k^*)} < f^*$ or $r^{(k^*)} > f^*$. To elaborate, if $r^{(k^*)} < f^*$, then $H(r^{(k^*)}) > 0$ by Lemma 1(c) and the corresponding solution $\mathbf{x}^{(k^*)}$ need not be feasible to (1). On the other hand, if $r^{(k^*)} > f^*$, we have $H(r^{(k^*)}) = \mathcal{P}(r^{(k^*)}, \mathbf{x}^{(k^*)}) < 0$ from Lemma 1(c) and the vector $\mathbf{x}^{(k^*)}$ is indeed a feasible solution. A root finding scheme that ensures $r^{(k)} > f^*$ at each iteration k will thus return a sequence of feasible solutions $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k^*)}$, that is a feasible solution path, where k^* is such that $f^* < r^{(k^*)} < f^* + \epsilon$ for a given $\epsilon > 0$ and, in addition, we have $f_0(\mathbf{x}^{(k^*)}) \le r^{(k^*)}$ from $\mathcal{P}(r^{(k^*)}, \mathbf{x}^{(k^*)}) < 0$. These inequalities imply that $f_0(\mathbf{x}^{(k^*)}) - f^* \le \epsilon$. Thus, $\mathbf{x}^{(k^*)}$ is an ϵ-optimal and feasible solution to (1) and it follows that solving SOEC can be cast as a root-finding problem involving $H(r)$.

Applying a root-finding algorithm to solve $H(r) = 0$ requires the exact computation of $H(r)$ at each iteration, which is difficult due to the nontrivial stochastic optimization in (2). Hence, we consider an inexact root-finding method, henceforth *stochastic feasible level set method* (SFLS), extending what is done in Lin et al. (2018b) and Aravkin et al. (2019) in a deterministic setting. Level set methods require an oracle to compute an approximation $U(r)$ of $H(r)$. This approximation is used to update r. A key element that we develop as part of SFLS is the notion of a stochastic oracle, which we introduce next.

Definition 2 (Stochastic Oracle) *Given $r > f^*$, $\epsilon > 0$, and $\delta \in (0, 1)$, a **stochastic oracle** $\mathcal{A}(r, \epsilon, \delta)$ returns a value $U(r)$ and a vector $\hat{\mathbf{x}} \in \mathcal{X}$ that satisfy the inequalities $\mathcal{P}(r, \hat{\mathbf{x}}) - H(r) \le \epsilon$ and $|U(r) - H(r)| \le \epsilon$ with a probability of at least $1 - \delta$.*

Lemma 3 clarifies the importance of the conditions underpinning the above definition to ensure a feasible solution to (1).

Lemma 3 *Given $r > f^*$, $0 < \epsilon \le -\frac{\theta-1}{\theta+1} H(r)$, $\delta \in (0, 1)$, and $\theta > 1$, the vector $\hat{\mathbf{x}} \in \mathcal{X}$ returned by a stochastic oracle $\mathcal{A}(r, \epsilon, \delta)$ defines a feasible solution to (1) with probability of at least $1 - \delta$.*

This lemma states that a stochastic oracle can recover a high probability feasible solution provided the optimality tolerance ϵ is less than $-\frac{\theta-1}{\theta+1} H(r)$.

Algorithm 1 formalizes the steps of SFLS to find an approximate root to $H(r) = 0$. Its inputs include a stochastic oracle $\mathcal{A}$; an initial level parameter value $r^{(0)} > f^*$, which exists because we can set $r^{(0)} = f_0(\tilde{\mathbf{x}})$ by Assumption 1; optimality and error tolerances ϵ_{opt} and $\epsilon_{\mathcal{A}}$, respectively; a probability δ; and a parameter θ that defines a step length as $1/2\theta$. SFLS begins from the level set defined by $r^{(0)}$. At each iteration k it executes lines 3 though 9.

Algorithm 1 Stochastic Feasible Level-Set Method (SFLS)

```
1:  Inputs: A stochastic oracle $\mathcal{A}$, a level parameter $r^{(0)} > f^*$, an optimality tolerance
    $\epsilon_{\text{opt}} > 0$, an oracle error $\epsilon_{\mathcal{A}} > 0$, a probability $\delta \in (0,1)$, and a step length parameter
    $\theta > 1$.
2:  for $k = 0, 1, \dots,$ do
3:    $\delta^{(k)} = \frac{\delta}{2^{k+1}}$.
4:    $\left(U(r^{(k)}), \mathbf{x}^{(k)}\right) = \mathcal{A}\left(r^{(k)}, \epsilon_{\mathcal{A}}, \delta^{(k)}\right)$.
5:    if $U(r^{(k)}) \geq -\epsilon_{\text{opt}}$ then
6:      Halt and return $\mathbf{x}^{(k)}$.
7:    end if
8:    $r^{(k+1)} \leftarrow r^{(k)} + U(r^{(k)})/(2\theta)$.
9:    $k \leftarrow k + 1$.
10: end for
```

In line 3, SFLS computes a probability $\delta^{(k)}$ that is used in the stochastic oracle call of line 4 to obtain an approximation $U(r^{(k)})$ and a high probability feasible solution $x^{(k)}$. The probability $\delta^{(k)}$ decreases with the iteration count k, that is, the probabilistic guarantee required of the stochastic oracle becomes more stringent to ensure the entire solution path is feasible with probability of at least $1 - \delta$. Lines 5-7 model the termination condition, which involves checking whether the approximation $U(r^{(k)})$ is greater than or equal to $-\epsilon_{\text{opt}}$. If this condition holds, then the algorithm halts and returns the incumbent solution $x^{(k)}$. Otherwise, $r^{(k)}$ is updated to $r^{(k+1)}$ in line 8 using $U(r^{(k)})$ and θ. Line 9 increments the iteration counter. While SFLS belongs to the family of level set approaches, it differs from known deterministic level set methods (see, e.g., Lin et al., 2018b and Aravkin et al., 2019) in its update step, termination criterion, and stochastic oracle.

We define the notion of an *input tuple* to ease the exposition of theoretical statements in the rest of the paper.

Definition 4 (Input tuple) *A tuple containing a subset of the elements $r, r^{(0)}, \epsilon, \epsilon_{\mathcal{A}}, \delta, \theta$, and γ_t is an* **input tuple** *if its respective components satisfy $r > f^*$, $r^{(0)} > f^*$, $1 \geq \epsilon > 0$, $\epsilon_{\mathcal{A}} > 0$, $\delta \in (0,1)$, $\theta > 1$, and $\gamma_t = 1/(M\sqrt{t+1})$, where $M > 0$ is a constant that is formally defined in* (9).

Theorem 5 provides the maximum number of calls to the stochastic oracle by Algorithm 1 to obtain a feasible and relative ϵ-optimal solution, which depends on a *condition measure* β of SOEC (1) defined as

$$\beta := -\frac{H(r^{(0)})}{r^{(0)} - f^*} \in (0, 1]. \tag{3}$$

It is easy to see that β provides an assessment of the slope of $H(r)$ at $r = f^*$. Intuitively, for an SOEC instance with a large β (i.e., well conditioned case), a root-finding method will be able to move towards the root of $H(r)$ faster compared to an instance with a small β (i.e., ill-conditioned case). See Figure 2.1 of Lin et al. (2018b) for a graphical illustration of this statement.

Theorem 5 *Given an input tuple $(r^{(0)}, \epsilon, \delta, \theta)$, suppose*

$$\epsilon_{opt} = -\frac{1}{\theta}H(r^{(0)})\epsilon \text{ and } \epsilon_{\mathcal{A}} = -\frac{\theta-1}{2\theta^2(\theta+1)}H(r^{(0)})\epsilon.$$

Algorithm 1 generates a feasible solution at each iteration with a probability of at least $1-\delta$. Moreover, it returns a relative ϵ-optimal and feasible solution with this probability in at most

$$\frac{2\theta^2}{\beta}\ln\left(\frac{\theta^2}{\beta\epsilon}\right)$$

calls to oracle $\mathcal{A}$.

The bound on the number of oracle calls increases with θ because both the step-length $1/2\theta$ and the optimality tolerance ϵ_{opt} decrease with θ. The maximum number of oracle calls is also a decreasing function of both the condition measure β and tolerance ϵ, that is, SFLS requires fewer iterations for problems that are better conditioned and when $\epsilon_{\mathcal{A}}$ and ϵ_{opt} are larger. Here, both $\epsilon_{\mathcal{A}}$ and ϵ_{opt} require knowledge of $H(r^{(0)})$, which is difficult to compute exactly. We want to point out that the dependence of $\epsilon_{\mathcal{A}}$ and ϵ_{opt} on $H(r^{(0)})$ are introduced here only to simplify the theorem and its proof, which helps readers to understand the main idea behind our technique. In §5, we will show that SFLS has a similar complexity even if $H(r^{(0)})$ in $\epsilon_{\mathcal{A}}$ and ϵ_{opt} is replaced by an upper bound $\bar{U}$ with $H(r^{(0)}) \leq \bar{U} < 0$ and $\bar{U}$ can be computed (by Algorithm 4) in a low cost independent of ϵ.

SFLS relies on the availability of a valid stochastic oracle $\mathcal{A}$. Standard subgradient methods cannot be used as oracles to solve (2) since computing a deterministic subgradient of $\mathcal{P}(r, \mathbf{x})$ requires exact evaluations of f_i for $i = 0, 1, \ldots, m$ (see Bertsekas, 1999 or Danskin, 2012, p.737), which is challenging due to the high-dimensional expectations in the definition of these functions. Indeed, the expectation in each f_i can be replaced by a direct SAA to obtain a sampled version $\hat{\mathcal{P}}(r, \mathbf{x})$ of $\mathcal{P}(r, \mathbf{x})$. This replacement is also problematic as subgradients of $\hat{\mathcal{P}}(r, \mathbf{x})$ provide biased subgradients of $\mathcal{P}(r, \mathbf{x})$ due to the maximization in the definition of the latter function.

To avoid this issue, we reformulate (2) into the equivalent min-max (i.e., saddle-point) form

$$H(r) = \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \left\{ \sum_{i=0}^{m} y_i(f_i(\mathbf{x}) - r_i) \right\}, \tag{4}$$

where $r_0 := r$ and $\mathcal{Y} := \left\{\mathbf{y} = (y_0, \ldots, y_m)^\top \in \mathbb{R}^{m+1} | \sum_{i=0}^{m} y_i = 1, y_i \geq 0\right\}$. Given $\mathbf{x} \in \mathcal{X}$, it is easy to check that $\mathbf{y}^* \in \arg\max_{\mathbf{y}\in\mathcal{Y}} \sum_{i=0}^{m} y_i(f_i(\mathbf{x}) - r_i)$ can be chosen as a unit vector with 1 corresponding to an index $i^* \in \arg\max_{i=1,\ldots,m}\{f_i(\mathbf{x}) - r_i\}$ and zeros for the remaining indices. Let $\Xi := \Xi_0 \times \Xi_1 \times \ldots \times \Xi_m$, $\boldsymbol{\xi} = (\xi_0, \xi_1, \ldots, \xi_m)^\top \in \Xi$, $\Phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) := \sum_{i=0}^{m} y_i(F_i(\mathbf{x}, \xi_i) - r_i)$, and $\phi(\mathbf{x}, \mathbf{y}) := \mathbb{E}\left[\Phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi})\right]$, where to ease notation we suppress the dependence of ϕ and Φ on the level parameter r since it is always equal to a fixed value when these functions are invoked. Therefore, (4) can be reformulated as

$$H(r) = \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{x}, \mathbf{y}). \tag{5}$$

Let $\hat{\phi}(\mathbf{x}, \mathbf{y})$ be an SAA of $\phi(\mathbf{x}, \mathbf{y})$. Subgradients of $\hat{\phi}(\mathbf{x}, \mathbf{y})$ provide an unbiased estimate of subgradients of $\phi(\mathbf{x}, \mathbf{y})$ because there is no nonlinear operator (e.g., maximization) acting on the expectation defining ϕ. The oracles that we discuss for SFLS in §§3-4 will thus solve (5).

3. Idealized Stochastic Oracle

In §3.1, we present stochastic mirror descent (SMD) in the form a stochastic oracle. In §3.2, we establish that SMD is indeed a stochastic oracle that can be used in SFLS (i.e., Algorithm 1) and then highlight computational issues that prevent its use. The discussion here serves a dual role. First, it provides practical motivation and sets the stage for developing a tractable stochastic oracle in §4. Second, it provides basic concepts on primal-dual methods needed throughout the paper, also making the paper more accessible to readers potentially unfamiliar with such methods.

3.1. Stochastic Mirror Descent

Stochastic mirror descent (SMD) (Nemirovski et al., 2009) is a well-known primal-dual method for solving saddle-point problems such as (5). SMD updates primal and dual variables $\mathbf{x}$ and $\mathbf{y}$ of (5), respectively, by employing stochastic subgradients of $\phi(\mathbf{x}, \mathbf{y})$ and a projection operator. Let $F_i'(\mathbf{x}, \xi_i) \in \partial F_i(\mathbf{x}, \xi_i)$ for $i = 0, 1, \ldots, m$, where ∂ is the subgradient operator. We denote the stochastic subgradient vector of $\phi(\mathbf{x}, \mathbf{y})$ by

$$G(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) := \begin{bmatrix} G_x(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \\ -G_y(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \end{bmatrix} := \begin{bmatrix} \sum_{i=0}^m y_i F_i'(\mathbf{x}, \xi_i) \\ -(F_0(\mathbf{x}, \xi_0) - r_0, F_1(\mathbf{x}, \xi_1) - r_1, \ldots, F_m(\mathbf{x}, \xi_m) - r_m)^\top \end{bmatrix}.$$

The projection employed by SMD relies on a Bregman divergence defined using a distance generating function $\omega_z(\mathbf{z})$ that has as its argument $\mathbf{z} := (\mathbf{x}, \mathbf{y})$ and operates over $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$. Moreover, $\omega_z(\mathbf{z})$ must be strongly convex with modulus 1 on $\mathcal{Z}$ and continuously differentiable on the set $\mathcal{Z}^o := \{\mathbf{z} \in \mathcal{Z} | \partial\omega_z(\mathbf{z}) \neq \emptyset\}$. The Bregman divergence $V(\mathbf{z}', \mathbf{z}) : \mathcal{Z}^o \times \mathcal{Z} \to \mathbb{R}_+$ expressed using ω_z is

$$V(\mathbf{z}', \mathbf{z}) := \omega_z(\mathbf{z}) - [\omega_z(\mathbf{z}') + \nabla\omega_z(\mathbf{z}')^\top(\mathbf{z} - \mathbf{z}')].$$

The projection operator (or prox-mapping), for any $\boldsymbol{\zeta} \in \mathbb{R}^{d+m+1}$, and $\mathbf{z}' \in \mathcal{Z}^o$, is defined as $P_{\mathbf{z}'}(\boldsymbol{\zeta}) := \arg\min_{\mathbf{z} \in \mathcal{Z}} \left\{\boldsymbol{\zeta}^\top(\mathbf{z} - \mathbf{z}') + V(\mathbf{z}', \mathbf{z})\right\}$.

Algorithm 2 summarizes the steps of SMD presented in the form of a stochastic oracle. The inputs to this algorithm are a level parameter $r \in \mathbb{R}$, an optimality tolerance $\epsilon_{\mathcal{A}} > 0$, a probability $\delta \in (0, 1)$, an iteration limit $W(\delta, \epsilon_{\mathcal{A}})$ (we specify this limit later in Proposition 6), and a step-length rule γ_t for all $t \in \mathbb{Z}_+$. Line 2 sets the initial solution $\mathbf{z}^{(0)} = (\mathbf{x}^{(0)}, \mathbf{y}^{(0)})$. Algorithm 2 executes lines 4 and 5 for $W(\delta, \epsilon_{\mathcal{A}})$ iterations. At iteration t, line 4 constructs a stochastic subgradient $G(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\xi}^{(t)})$ using a sample $\boldsymbol{\xi}^{(t)}$ of the random variables underlying the expectations in the objective and constraints of (1). Line 5 computes a step-length weighted average $\bar{\mathbf{z}}^{(t)}$ of past solutions. It also uses the stochastic subgradient computed in line 4 and a projection operator to find an updated solution $\mathbf{z}^{(t+1)}$. After exiting the for loop, line 7 uses the averaged primal solution $\bar{\mathbf{x}}^{(t)}$ to compute an upper bound $\max_{\mathbf{y} \in \mathcal{Y}} \phi(\bar{\mathbf{x}}^{(t)}, \mathbf{y})$ on $H(r)$. The pair $(U(\bar{\mathbf{x}}^{(t)}), \bar{\mathbf{x}}^{(t)})$ is returned in line 8.

It is worth noting that the update in line 5 relies on subgradients of an SAA $\hat{\phi}(\mathbf{x}, \mathbf{y})$ (with a single sample), which provides unbiased subgradients of $\phi(\mathbf{x}, \mathbf{y})$, unlike the biased subgradients that arise when working with SAAs of $\mathcal{P}(r, \mathbf{x})$ in the primal problem (2). In other words, a key benefit of the primal-dual reformulation (4) is that its objective $\phi(\mathbf{x}, \mathbf{y})$ allows the computation of unbiased subgradients after using SAAs to replace exact expectations.

Algorithm 2 Stochastic Mirror Descent (SMD)

```
Inputs: Level parameter $r \in \mathbb{R}$, optimality tolerance $\epsilon_{\mathcal{A}} > 0$, probability $\delta \in (0,1)$, an iteration limit $W$, and a step length rule $\gamma_t$ for all $t \in \mathbb{Z}_+$.
Set $\mathbf{z}^{(0)} := (\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) \in \arg\min_{\mathbf{z}\in\mathcal{Z}} \omega_z(\mathbf{z})$.
for $t = 0, 1, \ldots, W$ do
    Sample $\boldsymbol{\xi}^{(t)} = (\xi_0^{(t)}, \xi_1^{(t)}, \ldots, \xi_m^{(t)})^\top$ and compute $G(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\xi}^{(t)})$ .
    Execute
        $\bar{\mathbf{z}}^{(t)} := (\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}) := \frac{\sum_{s=0}^{t} \gamma_s \mathbf{z}^{(s)}}{\sum_{s=0}^{t} \gamma_s},$
        $\mathbf{z}^{(t+1)} := (\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)}) := P_{\mathbf{z}^{(t)}}(\gamma_t G(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\xi}^{(t)})).$
end for
Compute $U(\bar{\mathbf{x}}^{(t)}) = \max_{\mathbf{y}\in\mathcal{Y}} \phi(\bar{\mathbf{x}}^{(t)}, \mathbf{y})$.
return $(U(\bar{\mathbf{x}}^{(t)}), \bar{\mathbf{x}}^{(t)})$
```

3.2. Validity of Stochastic Oracle and Computational Issues

We analyze below the validity of SMD as a stochastic oracle and also discuss its computational tractability. Our analysis, based on Nemirovski et al. (2009), requires specifying the distance generating function ω_z introduced in §3.1 and stating a standard assumption.

To define ω_z, we equip $\mathcal{X}$ and $\mathcal{Y}$ with their own distance-generating functions $\omega_x : \mathcal{X} \to \mathbb{R}$ modulus α_x with respect to norm $\|\cdot\|_x$ and $\omega_y : \mathcal{Y} \to \mathbb{R}$ modulus α_y with respect to norm $\|\cdot\|_y$. This means that ω_x is α_x-strongly convex, continuous on $\mathcal{X}$, and continuously differentiable on the set of non-zero subgradients $\mathcal{X}^o := \{\mathbf{x} \in \mathcal{X} | \partial\omega_x(\mathbf{x}) \neq \emptyset\}$. Similarly, ω_y is α_y- strongly convex, continuous on $\mathcal{Y}$, and continuously differentiable on $\mathcal{Y}^o := \{\mathbf{y} \in \mathcal{Y} | \partial\omega_y(\mathbf{y}) \neq \emptyset\}$. Typical choices for $\|\cdot\|_x$ and $\|\cdot\|_y$ are $\|\cdot\|_2$ and $\|\cdot\|_1$, respectively. In addition, it is common to set $w_x(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ and $\omega_y(\mathbf{y}) = \sum_{i=0}^m y_i \ln y_i$. Defining the diameters of the sets $\mathcal{X}$ and $\mathcal{Y}$ as $D_x := \sqrt{\max_{\mathbf{x}\in\mathcal{X}} \omega_x(\mathbf{x}) - \min_{\mathbf{x}\in\mathcal{X}} \omega_x(\mathbf{x})}$ and $D_y := \sqrt{\max_{\mathbf{y}\in\mathcal{Y}} \omega_y(\mathbf{y}) - \min_{\mathbf{y}\in\mathcal{Y}} \omega_y(\mathbf{y})}$, the distance-generating function associated with $\mathcal{Z}$ is

$$\omega_z(\mathbf{z}) := \frac{\omega_x(\mathbf{x})}{2D_x^2} + \frac{\omega_y(\mathbf{y})}{2D_y^2}.$$

Next, the following standard assumption is needed to analyze SMD as well as other methods in the rest of the paper. Denote by $g(\mathbf{x}, \mathbf{y})$ expectation of the $(d + m + 1)$-dimensional vector $G(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi})$, that is, a deterministic subgradient. Moreover, let $\|\cdot\|_{*,x}$ and $\|\cdot\|_{*,y}$ represent the dual norms of $\|\cdot\|_x$ and $\|\cdot\|_y$, respectively.

Assumption 2 *For any $(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \in \mathcal{X} \times \mathcal{Y} \times \Xi$, there exist $F_i'(\mathbf{x}, \xi_i) \in \partial F_i(\mathbf{x}, \xi_i)$ for $i = 0, 1, \ldots, m$ such that $g(\mathbf{x}, \mathbf{y})$ is well defined and satisfies*

$$g(\mathbf{x}, \mathbf{y}) \in \begin{bmatrix} \partial_x \phi(\mathbf{x}, \mathbf{y}) \\ \partial_y[-\phi(\mathbf{x}, \mathbf{y})] \end{bmatrix},$$

where ∂_x and ∂_y represent the sub-differentials with respect to $\mathbf{x}$ and $\mathbf{y}$, respectively. Moreover, there exist positive constants M_x, M_y and Q such that

$$\mathbb{E}\left[\exp(\|G_x(\mathbf{x},\mathbf{y},\boldsymbol{\xi})\|_{*,x}^2/M_x^2)\right] \leq \exp(1), \tag{6}$$

$$\mathbb{E}\left[\exp(\|G_y(\mathbf{x},\mathbf{y},\boldsymbol{\xi})\|_{*,y}^2/M_y^2)\right] \leq \exp(1), \tag{7}$$

$$\mathbb{E}\left[\exp(\left|\Phi(\mathbf{x},\mathbf{y},\boldsymbol{\xi}) - \phi(\mathbf{x},\mathbf{y})\right|^2/Q^2)\right] \leq \exp(1), \tag{8}$$

for any $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$, which indicate that G_x and G_y have a light-tailed distribution and their moments are bounded.

Proposition 6 presents the iteration complexity of SMD, which follows from results in Nemirovski et al. (2009), and in addition, establishes that SMD is a valid stochastic oracle, that is, it satisfies Definition 2. The proof of this proposition relies on establishing that the primal-dual gap $U(\bar{\mathbf{x}}^{(t)}) - L(\bar{\mathbf{y}}^{(t)})$ is guaranteed to be less than a given $\epsilon_{\mathcal{A}} > 0$ with a probability of at least $1-\delta$ for a given $\delta \in (0,1)$, where $L(\bar{\mathbf{y}}^{(t)}) := \min_{\mathbf{x}\in\mathcal{X}} \phi(\mathbf{x},\bar{\mathbf{y}}^{(t)})$ and $U(\bar{\mathbf{x}}^{(t)})$ is computed in Algorithm 2. We also require the following constants:

$$M := \sqrt{\frac{2D_x^2}{\alpha_x}M_x^2 + \frac{2D_y^2}{\alpha_y}M_y^2}; \tag{9}$$

$$\Omega(\delta) := \max\left\{\sqrt{12\ln\left(\frac{24}{\delta}\right)}, \frac{4}{3}\ln\left(\frac{24}{\delta}\right)\right\}. \tag{10}$$

Proposition 6 *Given an input tuple $(r, \epsilon_{\mathcal{A}}, \delta, \gamma_t)$, the SMD solution $(\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)})$ satisfies $U(\bar{\mathbf{x}}^{(t)}) - L(\bar{\mathbf{y}}^{(t)}) \leq \epsilon_{\mathcal{A}}$ with probability at least $1-\delta$ in at most*

$$W(\delta,\epsilon_{\mathcal{A}}) := \max\left\{6, \left(\frac{8\left(10M\Omega(\delta)+4.5M\right)}{\epsilon_{\mathcal{A}}}\ln\left(\frac{4\left(10M\Omega(\delta)+4.5M\right)}{\epsilon_{\mathcal{A}}}\right)\right)^2 - 2\right\}$$

gradient iterations. As a consequence, SMD is a valid stochastic oracle with $W \geq W(\delta,\epsilon_{\mathcal{A}})$.

When solving (5), the dependence of the iteration complexity on $\epsilon_{\mathcal{A}}$ in Proposition 6 has an additional $\ln(1/\epsilon_{\mathcal{A}})$ term compared to the known SMD complexity dependence of $1/\epsilon_{\mathcal{A}}^2$ for solving an unconstrained version of this problem. Moreover, the analogous complexity dependence on δ inside logarithmic terms (see definition of $\Omega(\delta)$) in this proposition is comparable to the unconstrained case.

We note that SMD is a valid stochastic oracle, exhibits a favorable iteration complexity, and is based on unbiased subgradients of $\phi(\mathbf{x},\mathbf{y})$. Nevertheless, SMD is not directly implementable because the upper bound $U(\bar{\mathbf{x}}^{(t)})$ is challenging to compute exactly as the definition of $\phi(\mathbf{x},\mathbf{y})$ embeds expectations. Replacing these expectations by an SAA leads to a biased estimate of the upper bound $U(\bar{\mathbf{x}}^{(t)})$. This bias can be reduced by using a large number of samples but doing this would lead to an approach with high data complexity, which we would like to avoid. In other words, although our saddle-point formulation facilitates the computation of unbiased subgradients needed by SMD to obtain a near optimal and high probability feasible solution, its upper bound $U(\bar{\mathbf{x}}^{(t)})$, which serves as the constant $U(r)$ returned by the oracle (see Definition 2), cannot be computed.

The aforementioned bound computation challenge is further exacerbated if one wishes to change the stopping criterion of Algorithm 2 (i.e., line 3) from a maximum iteration limit to a bound on the primal-dual gap $U(\bar{\mathbf{x}}^{(t)}) - L(\bar{\mathbf{y}}^{(t)})$. In the latter case, implementing SMD would also entail the computation of the lower bound $L(\bar{\mathbf{y}}^{(t)})$, which suffers from analogous bias and data complexity issues when expectations in its definition are replaced by SAAs. In addition, the optimization problem over $\mathbf{x}$ in the definition of $L(\bar{\mathbf{y}}^{(t)})$ is in general a high-dimensional non-smooth convex optimization problem and solving such a problem multiple times is computationally burdensome. Therefore, it is apriori unclear how one should go about designing a computationally tractable oracle to overcome these issues and what the iteration complexity of such an oracle would be.

4. Tractable Stochastic Oracle

In this section, we design a computationally viable stochastic oracle by combining SMD and an online validation technique (Lan et al., 2012), and in particular, extending the latter technique originally proposed for minimization problems to handle min-max saddle point problems. This oracle overcomes the issues highlighted at the end of §3.2 by defining bounds that are (i) tractable to compute with low data complexity and (ii) do not suffer from the bias issue when replacing expectations in their definitions by SAAs, as was the case with the bounds $U(\bar{\mathbf{x}}^{(t)})$ and $L(\bar{\mathbf{y}}^{(t)})$. We present our algorithm in §4.1 and prove that it is a stochastic oracle in §4.2, where we also analyze its complexity.

4.1. Online Validation Based Stochastic Mirror Descent

Algorithm 3 contains the steps of our proposed online validation based stochastic mirror descent (OVSMD) scheme, which differs from Algorithm 2 only in line 7, where the upper bound $U(\bar{\mathbf{x}}^{(t)})$ on $H(r)$ is replaced by $\hat{u}_*^{(t)}$. The quantity $\hat{u}_*^{(t)}$ is an approximation of the following upper bound obtained using the online validation technique:

$$u_*^{(t)} := \max_{\mathbf{y}\in\mathcal{Y}} \left\{ \frac{1}{\sum_{s=0}^{t}\gamma_s} \sum_{s=0}^{t} \gamma_s \left[\phi(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}) + g_y(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})^\top (\mathbf{y} - \mathbf{y}^{(s)}) \right] \right\}.$$

This upper bound holds because

$$\frac{1}{\sum_{s=0}^{t}\gamma_s} \sum_{s=0}^{t} \gamma_s \left[\phi(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}) + g_y(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})^\top (\mathbf{y} - \mathbf{y}^{(s)}) \right] \geq \frac{\sum_{s=0}^{t}\gamma_s \phi(\mathbf{x}^{(s)}, \mathbf{y})}{\sum_{s=0}^{t}\gamma_s} \geq \phi(\bar{\mathbf{x}}^{(t)}, \mathbf{y}),$$

where the first inequality is true because g_y is a subgradient with respect to $\mathbf{y}$ of the function $\phi(\mathbf{x}, \mathbf{y})$, which is concave in $\mathbf{y}$, and the second inequality follows directly from the convexity of $\phi(\mathbf{x}, \mathbf{y})$ in $\mathbf{x}$. Therefore, we have

$$u_*^{(t)} \geq U(\bar{\mathbf{x}}^{(t)}) = \max_{\mathbf{y}\in\mathcal{Y}} \phi(\bar{\mathbf{x}}^{(t)}, \mathbf{y}) \geq H(r), \tag{11}$$

that is, $u_*^{(t)}$ is an upper bound on $H(r)$, albeit potentially weaker than $U(\bar{\mathbf{x}}^{(t)})$. Computing $u_*^{(t)}$ requires the exact evaluations of ϕ, g_x and g_y, which are not in general available because

Algorithm 3 Online Validation based Stochastic Mirror Descent: OVSMD

```
1: Inputs: Level parameter $r \in \mathbb{R}$, probability $\delta \in (0,1)$, optimality tolerance $\epsilon_{\mathcal{A}} > 0$, an
   iteration limit $T$, and a step length rule $\gamma_t$ for all $t \in \mathbb{Z}_+$.
2: Set $\mathbf{z}^{(0)} := (\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) \in \arg\min_{\mathbf{z}\in\mathcal{Z}} \omega_z(\mathbf{z})$.
3: for $t = 0, 1, \ldots, T(\delta, \epsilon_{\mathcal{A}})$ do
4:     Sample $\boldsymbol{\xi}^{(t)} = (\xi_0^{(t)}, \xi_1^{(t)}, \ldots, \xi_m^{(t)})^\top$ and compute $G(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\xi}^{(t)})$ .
5:     Execute
           $\bar{\mathbf{z}}^{(t)} := (\bar{\mathbf{x}}^{(t)}, \bar{\mathbf{y}}^{(t)}) := \frac{\sum_{s=0}^{t} \gamma_s \mathbf{z}^{(s)}}{\sum_{s=0}^{t} \gamma_s}$,
           $\mathbf{z}^{(t+1)} := (\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)}) := P_{\mathbf{z}^{(t)}}(\gamma_t G(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\xi}^{(t)}))$.
6: end for
7: Compute
       $\hat{u}_*^{(t)} := \max_{\mathbf{y}\in\mathcal{Y}} \left\{ \frac{1}{\sum_{s=0}^{t}\gamma_s} \sum_{s=0}^{t} \gamma_s \left[ \Phi(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) + G_y(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})^\top (\mathbf{y} - \mathbf{y}^{(s)}) \right] \right\}$.   (12)
8: return $(\hat{u}_*^{(t)}, \bar{\mathbf{x}}^{(t)})$
```

they involve expectations. In contrast, the term $\hat{u}_*^{(t)}$ computed in line 7 of Algorithm 3, which is stochastic approximation of $u_*^{(t)}$, can be easily computed in an online manner by solving a simple linear optimization problem.

As discussed in §3.2, replacing the iteration limit based stopping criterion by one that approximates an optimality gap requires a lower bound on $H(r)$. Following a similar argument to the upper bounding case above, we define the lower bound

$$l_*^{(t)} := \min_{\mathbf{x}\in\mathcal{X}} \left\{ \frac{1}{\sum_{s=0}^{t}\gamma_s} \sum_{s=0}^{t} \gamma_s \left[\phi(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}) + g_x(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})^\top (\mathbf{x} - \mathbf{x}^{(s)}) \right] \right\}.$$

Since $\phi(\mathbf{x}, \mathbf{y})$ is convex in $\mathbf{x}$, it follows that $l_*^{(t)} \leq L(\bar{\mathbf{y}}^{(t)}) = \min_{\mathbf{x}\in\mathcal{X}} \phi(\mathbf{x}, \bar{\mathbf{y}}^{(t)}) \leq H(r)$. Although $l_*^{(t)}$ is in general a weaker lower bound than $L(\bar{\mathbf{y}}^{(t)})$, the former bound is computed by solving a linear optimization problem as opposed to the potentially challenging non-smooth convex optimization problem defining the latter bound. Finally, we employ an online validation based approximation of $l_*^{(t)}$ to avoid computing expectations and obtain

$$\hat{l}_*^{(t)} := \min_{\mathbf{x}\in\mathcal{X}} \left\{ \frac{1}{\sum_{s=0}^{t}\gamma_s} \sum_{s=0}^{t} \gamma_s \left[\Phi(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) + G_y(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})^\top (\mathbf{x} - \mathbf{x}^{(s)}) \right] \right\}. \quad (13)$$

Despite the computational tractability of $\hat{u}_*^{(t)}$ and $\hat{l}_*^{(t)}$, these are stochastic quantities and subject to noise. Hence they do not always provide valid bounds on $H(r)$. In §4.2, we show that $\hat{l}_*^{(t)}$ and $\hat{u}_*^{(t)}$ are nevertheless sufficiently close to $H(r)$ with high probability after a finite number of iterations (see Theorem 8).

4.2. Validity of Stochastic Oracle and Iteration Complexity

We establish here the validity of OVSMD (i.e., Algorithm 3) as a stochastic oracle and derive its iteration complexity. Proposition 7 contains the two main ingredients underlying the analysis of OVSMD. Part (i) of this proposition shows that for a given $\epsilon_{\mathcal{A}} > 0$ the inequality $u_*^{(t)} - l_*^{(t)} \leq \epsilon_{\mathcal{A}}$ holds with high probability when t is sufficiently large. In other words, the deterministic quantities $u_*^{(t)}$ and $l_*^{(t)}$ computed using the OVSMD solutions provide "good" deterministic estimates of the level set function $H(r)$. This is not directly useful since OVSMD can only compute stochastic approximations of these quantities, as already discussed in §4.1. Part (ii) of Proposition 7 establishes that $\hat{u}_*^{(t)}$ and $\hat{l}_*^{(t)}$ are respectively close stochastic approximations of $u_*^{(t)}$ and $l_*^{(t)}$ at convergence with high probability. It then follows that the quantities $\hat{u}_*^{(t)}$ and $\hat{l}_*^{(t)}$ are "good" stochastic estimates of the level set function, and in particular, allows OVSMD to be used as a stochastic oracle.

Proposition 7 *Given an input tuple $(r, \epsilon_{\mathcal{A}}, \delta, \gamma_t)$, OVSMD computes $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$, $t = 1, 2, \ldots$, such that:*

(i) The inequality Prob$\{u_*^{(t)} - l_*^{(t)} > \epsilon_{\mathcal{A}}\} \leq \frac{\delta}{3}$ holds in at most

$$\max\left\{6, \left(\frac{8\,(10M\Omega(\delta) + 4.5M)}{\epsilon_{\mathcal{A}}}\ln\left(\frac{4\,(10M\Omega(\delta) + 4.5M)}{\epsilon_{\mathcal{A}}}\right)\right)^2 - 2\right\}$$

gradient iterations.

(ii) The inequalities Prob$\{|\hat{l}_*^{(t)} - l_*^{(t)}| > \epsilon_{\mathcal{A}}\} \leq \frac{\delta}{3}$ and Prob$\{|\hat{u}_*^{(t)} - u_*^{(t)}| > \epsilon_{\mathcal{A}}\} \leq \frac{\delta}{3}$ hold in at most

$$\max\left\{6, \left(\frac{8\,(Q\Omega(\delta) + 8M\Omega(\delta) + 2.5M)}{\epsilon_{\mathcal{A}}}\ln\left(\frac{4\,(Q\Omega(\delta) + 8\Omega(\delta)M + 2.5M)}{\epsilon_{\mathcal{A}}}\right)\right)^2 - 2\right\}$$

gradient iterations.

Leveraging Proposition 7, Theorem 8 shows that OVSMD is a valid stochastic oracle and also presents its iteration complexity.

Theorem 8 *Given an input tuple $(r, \epsilon_{\mathcal{A}}, \delta, \gamma_t)$, the OVSMD guarantees $\mathcal{P}(r, \bar{\mathbf{x}}^{(t)}) - H(r) \leq \epsilon_{\mathcal{A}}$ and $|\hat{u}_*^{(t)} - H(r)| \leq \epsilon_{\mathcal{A}}$ with probability at least $1 - \delta$ in at most*

$$T(\delta, \epsilon_{\mathcal{A}}) := \max\left\{6, \left(\frac{16\,(Q\Omega(\delta) + 10M\Omega(\delta) + 4.5M)}{\epsilon_{\mathcal{A}}}\ln\left(\frac{8\,(Q\Omega(\delta) + 10M\Omega(\delta) + 4.5M)}{\epsilon_{\mathcal{A}}}\right)\right)^2 - 2\right\} \tag{14}$$

gradient iterations. As a consequence, OVSMD is a valid stochastic oracle with $T \geq T(\delta, \epsilon_{\mathcal{A}})$.

Despite OVSMD being a tractable oracle, the dependence of its iteration complexity on both $\epsilon_{\mathcal{A}}$ and δ is identical to the analogous dependence seen with the idealized SMD oracle analyzed in Proposition 6. Moreover, in terms of $\epsilon_{\mathcal{A}}$, OVSMD is only a $\ln(1/\epsilon_{\mathcal{A}})$ worse than the known complexity of SMD in the unconstrained case, where feasibility is not a concern.

5. SFLS with OVSMD as its Stochastic Oracle

In this section, we provide theoretical support for the use of OVSMD as SFLS's stochastic oracle in §5.1 and then discuss implementation guidelines in §5.2.

5.1. Theoretical Analysis

Theorems 5 and 8 can be used to derive the (gradient) iteration complexity of SFLS when using OVSMD as the stochastic oracle. We state this complexity in Corollary 9.

Corollary 9 *Given an input tuple $(r^{(0)}, \epsilon, \delta, \gamma_t, \theta)$, let*

$$\epsilon_{opt} = -\frac{1}{\theta} H(r^{(0)})\epsilon \text{ and } \epsilon_{\mathcal{A}} = -\frac{\theta - 1}{2\theta^2(\theta+1)} H(r^{(0)})\epsilon.$$

Moreover, suppose OVSMD with $T = T(\delta, \epsilon_{\mathcal{A}})$ is chosen as the stochastic oracle $\mathcal{A}$. Then SFLS returns a relative ϵ-optimal and feasible solution with probability of at least $1-\delta$ using at most

$$\frac{2\theta^2}{\beta} \ln\left(\frac{\theta^2}{\beta\epsilon}\right)$$

OVSMD calls and

$$\mathcal{O}\left(\frac{1}{\beta\epsilon^2} \cdot \ln^3\left(\frac{1}{\delta}\right) \cdot \ln^2\left(\frac{1}{\epsilon}\right) + \frac{1}{\beta^3\epsilon^2} \cdot \ln^5\left(\frac{1}{\epsilon}\right)\right)$$

gradient iterations.

This complexity result is somewhat idealistic because the inputs to SFLS, namely ϵ_{opt} and $\epsilon_{\mathcal{A}}$, require knowledge of $H(r^{(0)})$, which is difficult to compute exactly. A possible resolution is to compute an upper bound on $H(r^{(0)})$, denoted by $\bar{U}$, such that $H(r^{(0)}) \leq \bar{U} < 0$. If $|\bar{U}|$ is much smaller than $|H(r^{(0)})|$, then the optimality tolerance $\epsilon_{\mathcal{A}}$ will be substantially more stringent and thus lead to a larger complexity than the iteration bound in Corollary 9. Therefore, to obtain a complete theoretical assessment of the computational complexity of SLFS with OVSMD, it is important to incorporate the cost of finding a $\bar{U}$ that is comparable to $H(r^{(0)})$ (i.e., $|\bar{U}| = \Omega(|H(r^{(0)})|)$).

Fortunately, OVSMD can itself be used to compute the desired $\bar{U}$. We discuss the intuition behind its use for this purpose and then formally state the result. Recall that $H(r^{(0)}) < 0$ since $r^{(0)} > f^*$. We consider obtain an upper bound $\bar{U}$ by solving (2) with $r = r^{(0)}$ and a small enough optimality gap. By Theorem 8, OVSMD with $r = r^{(0)}$ can guarantee $H(r^{(0)}) \leq \hat{u}_*^{(t)} + \epsilon_{\mathcal{A}}$ with high probability. This suggests setting $\bar{U} = \hat{u}_*^{(t)} + \epsilon_{\mathcal{A}}$. However, it is a priori unclear how small $\epsilon_{\mathcal{A}}$ should be in order to ensure $\bar{U} < 0$ and $|\bar{U}| = \Omega(|H(r^{(0)})|)$. Therefore, we run OVSMD multiple times, starting from a tolerance $\alpha^{(0)} = \bar{\alpha}$, geometrically reducing this tolerance after each run, and stopping this procedure once $\bar{U} = \hat{u}_*^{(h)} + \alpha^{(h)} < 0$ and $(\hat{u}_*^{(h)} - \alpha^{(h)})/(\hat{u}_*^{(h)} + \alpha^{(h)}) \leq \theta$ hold. We can then use Theorem 8 and the condition $\hat{u}_*^{(h)} + \alpha^{(h)} < 0$ to show that $|H(r^{(0)})|/|\bar{U}| \leq (\hat{u}_*^{(h)} - \alpha^{(h)})/(\hat{u}_*^{(h)} + \alpha^{(h)}) \leq \theta$, which implies $|\bar{U}| = \Omega(|H(r^{(0)})|)$. We formalize the aforementioned approach in Algorithm 4.

Theorem 10 establishes the complexity of employing Algorithm 4 to compute $\bar{U}$ and subsequently running SFLS leveraging this computation.

Algorithm 4 Estimating an upper bound on $H(r^{(0)})$ using OVSMD

```
**Inputs:** Level parameter $r^{(0)} > f^*$, initial approximation tolerance $\bar{\alpha} > 0$, probability $\delta \in (0,1)$, constant $\theta > 1$, and a step length rule $\gamma_h$ for all $h \in \mathbb{Z}_+$.
Set $h = 0$ and $\alpha^{(0)} = \bar{\alpha}$.
**repeat**
    Set $\delta^{(h)} = \frac{\delta}{2^{h+1}}$ and $\alpha^{(h)} = \frac{\alpha^{(0)}}{2^h}$.
    Compute $\hat{u}_*^{(h)} \leftarrow \mathrm{OVSMD}(r^{(0)}, \delta^{(h)}, \alpha^{(h)}, \gamma_h)$.
    Set $h \leftarrow h + 1$.
**until** $\hat{u}_*^{(h)} + \alpha^{(h)} < 0$ and $\frac{\hat{u}_*^{(h)} - \alpha^{(h)}}{\hat{u}_*^{(h)} + \alpha^{(h)}} \leq \theta$.
**return** $\bar{U} = \hat{u}_*^{(h)} + \alpha^{(h)}$.
```

Theorem 10 *Given an input tuple $(r^{(0)}, \epsilon, \delta, \gamma_t, \theta)$, suppose we compute $\bar{U}$ using Algorithm 4 and then execute SFLS to find a relative ϵ-optimal and feasible solution with a probability of at least $1 - \delta$ using $\epsilon_{opt} = -\frac{1}{\theta}\bar{U}\epsilon$, $\epsilon_{\mathcal{A}} = -\frac{\theta-1}{2\theta^2(\theta+1)}\bar{U}\epsilon$, and OVSMD with $T = T(\delta, \epsilon_{\mathcal{A}})$ as the stochastic oracle $\mathcal{A}$. This procedure requires in total at most*

$$\mathcal{O}\left(\frac{1}{\beta}\ln\left(\frac{1}{\beta\epsilon}\right)\right)$$

OVSMD calls and

$$\mathcal{O}\left(\frac{1}{\beta^2}\ln^4\left(\frac{1}{\beta}\right)\ln^2\left(\frac{1}{\delta}\right)\right) + \mathcal{O}\left(\frac{1}{\beta\epsilon^2}\cdot\ln^3\left(\frac{1}{\delta}\right)\cdot\ln^2\left(\frac{1}{\epsilon}\right) + \frac{1}{\beta^3\epsilon^2}\cdot\ln^5\left(\frac{1}{\epsilon}\right)\right)$$

gradient iterations.

Theorem 10 provides a realistic theoretical assessment of the computational burden of solving SOECs using SFLS. Interestingly, it shows that running Algorithm 4 to compute $\bar{U}$ before executing SFLS and replacing the unknown term $H(r^{(0)})$ in the definitions of $\epsilon_{\mathcal{A}}$ and ϵ_{opt} with the computed $\bar{U}$ value does not change the overall big-$\mathcal{O}$ oracle and gradient iteration complexities in Corollary 9, except for logarithmic terms.

The complexity of SFLS (combined with OVSMD) in Theorem 10 is comparable in terms of its dependence on ϵ and δ to the complexity of the algorithm in Yu et al. (2017), which does not ensure feasibility. This suggests that our procedure is efficient at ensuring feasibility. The cost of ensuring feasibility, however, appears in the dependence of the SFLS iteration complexity on the condition measure β. Such dependence is absent in approaches that do not ensure feasibility.

Another relevant comparison is with the deterministic feasible level set approach (DFLS) of Lin et al. (2018b) and its variant in Lin et al. (2018a), which are both applicable to solve deterministic constrained convex optimization problems. The complexity of DFLS based methods depend on the number of data points that define expectations and thus lead to large data complexity, and in particular, have infinite complexity when expectations are defined over continuous random variables. In contrast, the complexity of SFLS in Theorem

10 does not depend on the number of data points. In addition, compared to DFLS, the iteration complexity of SFLS has only additional logarithmic factors involving ϵ and δ, which is encouraging, as the stochastic level set algorithm (i.e., Algorithm 1) and OVSMD oracle need to contend with several challenges that arise due to the presence of expectations in SOECs.

In summary, our theoretical analysis of SFLS and comparison with known complexities of state-of-the-art approaches suggests that SFLS is effective in terms of iteration complexity at computing a high probability feasible solution path for SEOCs, a much broader and challenging class of problems than deterministic constrained convex programs. Moreover, a fully stochastic approach such as SFLS is theoretically necessary to achieve low data complexity in this context.

5.2. Implementation Guidelines

As is common with first-order methods, the implementation of SFLS requires parameter tuning. A direct implementation of SFLS in a manner consistent with Theorem 10 requires selecting $r^{(0)}$, ϵ, δ, θ and γ_t; estimating constants M and Q (needed to define $T = T(\delta, \epsilon_{\mathcal{A}})$ in OVSMD); and then computing $\bar{U}$. While these parameters can be estimated or approximated, we suggest a simpler implementation strategy that largely side-steps such tuning. Firstly, we avoid stopping SFLS by pre-specifying optimality tolerance ϵ_{opt} and instead stop it based on an outer iteration limit. This is possible because the SFLS outer iterations only affect the suboptimality of the incumbent feasible solution, that is, being a feasible level set method, SFLS can return feasible and implementable solutions when terminated after any number of outer iterations. Secondly, instead of choosing the number of inner iteration in OVSMD as $T = T(\delta, \epsilon_{\mathcal{A}})$ based on pre-specified δ and $\epsilon_{\mathcal{A}}$, we directly specify T. According to (14), $T(\delta, \epsilon_{\mathcal{A}})$ is strictly monotonically decreasing in $\epsilon_{\mathcal{A}}$ and thus in ϵ so that a relative ϵ-optimal and feasible solution with $\epsilon = \tilde{\mathcal{O}}(\frac{1}{\sqrt{T}})$ can be guaranteed.[3] Corollary 11 establishes that the convergence of SFLS in this implementation.

Corollary 11 *Suppose we have an input tuple $(r^{(0)}, \gamma_t, \theta)$ and the iteration limit in OVSMD is T. Given $\delta \in (0, 1)$, SFLS finds a relative ϵ-optimal and feasible solution with $\epsilon \leq \mathcal{O}\left(\frac{\ln(1/\delta)\ln(T)\ln(T/\beta)}{\beta\sqrt{T}}\right)$ and with a probability of at least $1 - \delta$ using at most $\mathcal{O}\left(\frac{1}{\beta}\ln\left(\frac{T}{\beta}\right)\right)$ OVSMD calls and $\mathcal{O}\left(\frac{T}{\beta}\ln\left(\frac{T}{\beta}\right)\right)$ gradient iterations.*

Overall, following the aforementioned strategy only requires the choice of T, θ, $r^{(0)}$, and γ_t—a significant reduction in implementation burden.

For choosing θ and T, we consider a discrete set of values and tune the algorithm, that is, we test the performance of SFLS for a few iterations or data passes for each value, and select the one that leads to the largest decrease in suboptimality. Selecting $r^{(0)}$ is easy when an initial feasible solution $\tilde{\mathbf{x}}$ is available because we have $\mathbb{E}\left[F_0(\tilde{\mathbf{x}}, \xi_0)\right] > f^*$. In this case, we estimate $\mathbb{E}\left[F_0(\tilde{\mathbf{x}}, \xi_0)\right]$ using an SAA and then set $r^{(0)}$ to a larger value to account for approximation error and ensure we have $r^{(0)} > f^*$. If a feasible solution is not readily

[3] Here, we use the $\tilde{\mathcal{O}}$ complexity notation, which omits logarithmic terms.

available, we can find one by applying a minor modification of Algorithm 4 to solve

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \left\{ \sum_{i=1}^{m} y_i(f_i(\mathbf{x}) - r_i) \right\},$$

which does not include the term in (4) corresponding to $i = 0$, that is, $f_0 - r$. Finally, the step length can be specified as $\gamma_t = 1/(c\sqrt{t+1})$ for a given constraint $c > 0$, which is tuned. While c is chosen as M in our theoretical analysis to simplify proofs, analogous results hold for a generic constant $c > 0$. We omit these general results for the sake of brevity as they do not change the dependence of our iterations bounds on ϵ, β, and δ.

6. Numerical Experiments

In this section, we evaluate the numerical performance of SFLS on three diverse SOEC applications: (i) approximate linear programs for solving Markov decision processes, (ii) multi-class Neyman-Pearson classification, and (iii) learning with fairness constraints. SOECs in the first application contain expectations of continuous random variables while those in the second and third applications involve discrete random variables. Our first algorithmic benchmark is the stochastic subgradient method YNW of Yu et al. (2017) as it is the only first order approach (we are aware of) that can handle SOECs with multiple constraints. In addition, we also compare against the deterministic feasible level-set method (DFLS) of Lin et al. (2018b) because it ensures a feasible solution path. Specifically, comparing SFLS and DFLS allows us to evaluate the benefits of the reduced data complexity in our stochastic approach. In §6.1, we describe our computational setup and then the performance of algorithms on applications in §§6.2-6.4.

6.1. Computational Setup

We implemented SFLS, DLFS, and YNW in Matlab running on a 64-bit Microsoft Windows 10 machine with a 2.70 Ghz Intel Core i7-6820HQ CPU and 8GB of memory. We set $\omega_x(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ and $\omega_y(\mathbf{y}) = \sum_{i=0}^{m} y_i \ln y_i$ in all three algorithms. We followed the guidelines in §5.2 when implementing SFLS and thus had to choose only $r^{(0)}$, θ, and γ_t. We based $r^{(0)}$ on the solution $\tilde{\mathbf{x}}$. We tuned θ over the discrete set $\{1.1, 2, 5\}$ and T over the discrete set $\{50, 100, 200, 300\}$. We selected $\gamma_t = 1/(c\sqrt{t+1})$ and tuned c over the set of possible values $\{0.05, 0.1, 1, 2, 5\}$. We employed a mini-batch technique to construct the stochastic gradients in SFLS and YNW.

Similar to SFLS, DFLS solves the subproblem $\min_{\mathbf{x}\in\mathcal{X}} \mathcal{P}(r^{(k)}, \mathbf{x})$ approximately in the kth outer iteration and uses the returned solution $\mathbf{x}^{(k)}$ to update $r^{(k)}$ as $r^{(k+1)} \leftarrow r^{(k)} + \mathcal{P}(r^{(k)}, \mathbf{x}^{(k)})/2$. Following Lin et al. (2018b), we use the standard subgradient descent method to solve this subproblem and the parameters $r^{(0)}$ and γ_t and the inner iteration limit T in DLFS are tuned in the same way as in SFLS as described above. To apply DLFS, we constructed a deterministic version of each SOEC using SAAs of expectations. We found, consistent with Lin et al. (2018b), that using SAAs in lieu of expectations over continuous random variables in the perishable control problem (first application) did not sufficiently represent the original problem even when using a large number of samples. We thus omitted DFLS as a benchmark for this application. This was not an issue for the remaining two applications because expectations are defined over discrete random variables. To avoid the quality of SAAs confounding our performance evaluation, we chose instances for these two applications such that expectations can be evaluated exactly, albeit requiring more time.

We followed the guidance in Yu et al. (2017) to setup YNW. Specifically, we chose the control parameters V and α as $V = \sqrt{T}$ and $\alpha = T$, respectively, as a function of the total number of iterations T, where V is the weight of the gradient of the objective function and α is the weight of the proximal term in the updating equation of $\mathbf{x}$ in YNW. Similar to SFLS, we used a mini-batch technique to construct the stochastic gradients and evaluate the objective values.

6.2. Approximate Linear Programming for Markov Decision Processes

Approximate linear programs (ALPs) address the well-known curse of dimensionality associated with directly solving large-scale Markov decision processes (MDPs; Puterman, 1994) by computing a value function approximation. We illustrate how our SFLS method can be applied to tackle ALPs, and thus large-scale MDPs, by considering a challenging perishable inventory control problem with partial backlogging and lead time. We begin by presenting the MDP for this problem and refer the reader to Lin et al. (2020) for its derivation and detailed application context.

Consider the management of orders for a single product with a finite life time of I periods and an order lead time of J periods, that is, the product takes J periods to be delivered from when it is ordered and I periods to perish from receipt. The state space of the MDP is represented by the vector

$$\mathbf{s} = (z_0, z_1, \ldots, z_{I-1}, q_1, q_2, \ldots, q_{J-1}) \in \mathbb{R}^{I+J-1},$$

where q_j, $1 \le j \le J-1$, denotes the order quantities that will be received j periods from now, and z_i, $0 \le i \le I-1$, the on-hand inventory with i periods of lifetime remaining. The order quantity a is at most $\bar{a}$ and belongs to the interval $[0, \bar{a}]$, which implies $z_i \in [0, \bar{a}]$ for $i = 1, \ldots, I-1$ and $q_j \in [0, \bar{a}]$ for $j = 1, \ldots, J-1$. The element z_0 of the state is bounded below by $l_s < 0$ to allow limited or partial backlogging, that is, any units backlogged beyond $|l_s|$ are lost sales. To ease exposition, we write $\mathbf{s} \in \mathcal{S}$ and $a \in \mathcal{A}$ to capture the state and action domains, respectively, and use $\mathbf{s}^0$ to represent the initial state. Assuming orders are served on a first-come-first-serve basis, the MDP state transitions as

$$f(\mathbf{s}, a) = (\max\{z_1 - (G - z_0)_+, l_s - \sum_{i=2}^{I-1} z_i\}, z_2, \ldots, z_{I-1}, q_1, q_2, \ldots, q_{J-1}, a),$$

where G represents stochastic demand with distribution P_G. Moreover, the cost associated with ordering a at state $\mathbf{s}$ is

$$\begin{aligned} c(\mathbf{s}, a) &= \gamma^J c_p a \\ &+\mathbb{E}\left[c_h\left(\sum_{i=1}^{I-1} z_i - (G - z_0)_+\right)_+ + c_b\left(G - \sum_{i=0}^{I-1} z_i\right)_+ + c_d\,(z_0 - G)_+ + c_l\left(l_s + G - \sum_{i=0}^{I-1} z_i\right)_+\right], \end{aligned}$$

where the per unit lost sale, disposal, purchasing, holding, and backlogging costs are c_l, c_d, c_p, c_h, and c_b, respectively; $\mathbb{E}$ is taken over G; and $\gamma \in (0, 1)$ is a discount factor. The infinite horizon (discounted cost) MDP formulated using the aforementioned components can be solved using the fixed point equations

$$V(\mathbf{s}) = \max_{a\in\mathcal{A}} \; c(\mathbf{s},a) + \gamma\mathbb{E}[V(f(\mathbf{s},a))], \quad \forall \mathbf{s}\in\mathcal{S}.$$

ALPs approximate the high-dimensional MDP value function $V(\mathbf{s})$ (Schweitzer and Seidmann, 1985; de Farias and Van Roy, 2003) using a linear combination of basis functions. We construct the ALP value function approximation using an intercept τ and B basis functions $\phi_b : \mathcal{S} \mapsto \mathbb{R}$, $b = 1,\ldots,B$, that is, $V(\mathbf{s}) \approx \tau + \sum_{b=1}^{B}\theta_b\phi_b(\mathbf{s})$, where $\theta := (\theta_1,\ldots,\theta_B)\in\mathbb{R}^B$ is the basis function weight vector. It is common to require that the pair (τ,θ) belongs to a compact set $\mathcal{X}$. The VFA weights are computed by solving

$$\begin{aligned} \max_{(\tau,\theta)\in\mathcal{X}} \quad & \tau + \sum_{b=1}^{B}\theta_b\left[\phi_b(\mathbf{s}^0)\right] \\ \text{s.t.} \quad & (1-\gamma)\tau + \sum_{b=1}^{B}\theta_b\left(\phi_b(\mathbf{s}) - \gamma\mathbb{E}\left[\phi_b(f(\mathbf{s},a))\right]\right) - c(\mathbf{s},a) \le 0, \quad \forall(\mathbf{s},a)\in\mathcal{S}\times\mathcal{A}. \end{aligned}$$

The feasibility of the ALP constraints is important because it ensures that the objective function of a feasible solution provides a lower bound on the optimal policy value, which can be used to assess the suboptimality of heuristic policies (see, e.g., Proposition 4 in Adelman and Mersereau, 2008). Thus, in principle, methods to solve ALP would benefit from emphasizing feasibility as we do in SFLS.

Since the linear program above is semi-infinite, constraint sampling is a popular strategy to approach its solution and obtain a high-probability feasible solution (de Farias and Van Roy, 2004). Specifically, suppose we sample m state-action pairs $(\mathbf{s}_i,a_i), i = 1,\ldots,m$. The ALP with constraints corresponding to these samples takes the form of (1):

$$\begin{aligned} \max_{\mathbf{x}=(\tau,\theta)\in\mathcal{X}} \quad & f_0(\mathbf{x}) := \tau + \sum_{b=1}^{B}\theta_b\left[\phi_b(\mathbf{s}^0)\right] \qquad (15)\\ \text{s.t.} \quad & f_i(\mathbf{x}) := (1-\gamma)\tau + \sum_{b=1}^{B}\theta_b\left(\phi_b(\mathbf{s}_i) - \gamma\mathbb{E}\left[\phi_b(f(\mathbf{s}_i,a_i))\right]\right) - c(\mathbf{s}_i,a_i) \le 0, \quad i = 1,2,\ldots,m. \end{aligned}$$

We solve this linear program in our experiments.

Following Lin et al. (2020), we constructed instances with $I = 2$ and $J = 2$, chose P_G to be a truncated normal in the interval $[0,10]$ with mean 5 and the standard deviation 2, and fixed c_p, c_l, $\bar{a}$, l_s, γ, and $\mathbf{s}^0$ equal to 20, 100, 10, -10, 0.95, and $(5,0,0)$, respectively. We experimented with three instances based on the triple (c_h,c_d,c_b) being equal to $(2,10,10)$, $(5,10,8)$, and $(2,5,10)$. We employed eighteen basis functions ($B = 18$): z_0, z_1, q_1, and $\{(z_0-\nu)_+,(z_0+z_1-2\nu)_+,(z_0+z_1+q_1-3\nu)_+,(2\nu-z_0-z_1-q_1)_+,(\nu-z_1-q_1)_+|\nu\in\{\mathbb{E}[G],G^{0.25},G^{0.5}\}\}$, where $G^{0.25}$ and $G^{0.5}$ are the 25-th and 50-th quartiles of the demand distribution. The domain for the basis function weights $\mathcal{X}$ was taken to be the box $[0,3000]\times[-5,5]^B$. We chose m as 500.

In all methods, we use the initial solution $\tilde{\mathbf{x}} = (\tilde{\tau},\tilde{\theta})$ with $\tilde{\tau} = \min_{i=1,\ldots,m}\frac{c(s_i,a_i)}{1-\gamma}$ and $\tilde{\theta} = \mathbf{0}$ which is feasible for (15). Our SFLS implementation uses $r^{(0)} = f_0(\tilde{\mathbf{x}}) > f^*$, $\theta = 1.1$, and the step length rule $\gamma_t = 5/\sqrt{t+1}$. We do not report results for DFLS because,

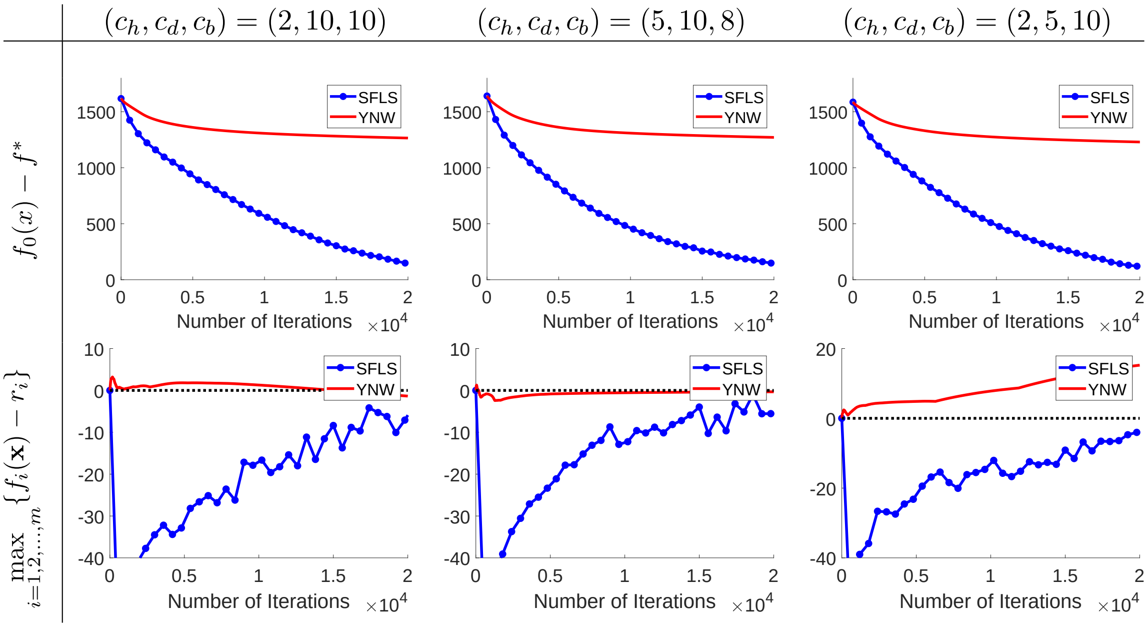

Figure 1: Performance of SFLS and YNW for solving approximate linear programs arising in perishable inventory control.

as alluded to in §6.1, obtaining a good deterministic approximation using SAAs is nontrivial for the perishable inventory control problem. We use a mini-batch technique with a batch size of 100 to construct stochastic estimates of the gradients and function values of f_i, $i = 0, \ldots, m$ in both SFLS and YNW. In SFLS, we choose the numbers of inner (i.e. T) and outer iterations to be 200 and 100, respectively, which leads to $20,000$ stochastic gradient steps (inner iterations) in total. Hence, we choose the total number of iterations in YNW as $T = 20,000$ so that both methods evaluate the same number of stochastic gradients in total which lead to similar runtime (about 1100 seconds).

Figure 1 displays the performance of SFLS and YNW. The y-axes of the top subfigures report the optimality gap $f_0(\mathbf{x}) - f^*$ while these axes in the bottom subfigures show the feasibility of solutions by plotting $\max_{i=1,2,\ldots,m}\{f_i(\mathbf{x}) - r_i\}$. Here, $f_i(\mathbf{x})$ for $i = 1, 2, \ldots, m$ are calculated by approximating the expectations in their definitions in (15) with $10,000$ samples of demand G. The optimal value f^* is approximated by the objective value found by a separated run of SFLS with sufficient iterations (400 outer and 500 inner iterations). We track these measures as a function of the number of iterations performed by each algorithm in the x-axis. To indicate the values of $f_0(x) - f^*$ and $\max_{i=1,2,\ldots,m}\{f_i(\mathbf{x}) - r_i\}$ corresponding to the high probability feasible solutions maintained at each SFLS (outer) iteration we use line markers in Figure 1. The YNW curves have no line markers as there are no outer iterations ensuring feasibility. SFLS finds a feasible solution quickly and maintain a relatively large constraint slack but YNW does not always ensure feasibility. SFLS also reduces the suboptimality of solutions faster, suggesting that SFLS is able to balance optimality and feasibility well on these instances.

6.3. Multi-class Neyman-Pearson classification

Another application that gives rise to (1) is Neyman-Pearson classification. In multi-class classification, there exist m classes of data, where ψ_i, $i = 1, 2, \ldots, m$, denotes a random variable defined using the distribution of data points associated with the i-th class. To classify a data point ψ_i to one of the m classes, we rely on the same number of linear models $\mathbf{x}_i$, $i = 1, 2, \ldots, m$. The predicted class for ψ is $\arg\max_{i=1,2,\ldots,m} \mathbf{x}_i^\top \psi$. High classification accuracy in this scheme requires $\mathbf{x}_i^\top \psi_i - \mathbf{x}_l^\top \psi_i$ with $i \neq l$ to be large and positive (Crammer and Singer, 2002), that is, the classifiers have discriminatory power. Minimizing the expected loss $\mathbb{E}\left[\phi(\mathbf{x}_i^\top \psi_i - \mathbf{x}_l^\top \psi_i)\right]$ is one approach to promote this goal, where ϕ is a non-increasing convex loss function and $\mathbb{E}$ is expectation taken over ψ_i.

Suppose misclassifying ψ_i has a cost that depends on i but not on the predicted class. We propose a model that prioritizes classes with relatively higher misclassification costs using constraints and simultaneously trains the set of m linear models by solving

$$\min_{\substack{\|\mathbf{x}_i\|_2 \leq \lambda, \\ \forall i=1,2,\ldots,m}} \sum_{l \neq 1} \mathbb{E}[\phi(\mathbf{x}_1^\top \psi_1 - \mathbf{x}_l^\top \psi_1)], \text{ s.t. } \sum_{l \neq i} \mathbb{E}[\phi(\mathbf{x}_i^\top \psi_i - \mathbf{x}_l^\top \psi_i)] \leq r_i, \quad i = 2, 3, \ldots, m, \tag{16}$$

where it is assumed (without loss of generality) that class 1 has the highest misclassification cost and the value of r_i is chosen to capture the misclassification cost of class i. Here λ is a regularization parameter. This formulation can be easily extended to handle the case where the mis-classification cost depends on both the true and predicted classes. Indeed, (16) is of the form (1). Infeasible solutions may result in large misclassification costs for some classes, which is undesirable, and creates a need for methods that emphasize feasibility.

We created test instances using the multi-class classification LIBSVM data sets *connect-4*, *covtype*, and *news20* from Chang and Lin (2011). We selected these instances as their size still allows us to run DFLS in the manner discussed in §6.1. We summarize in Table 1 the number of classes, the number of data points in each class, and the number of features in these four data sets. We chose the loss function (16) to be the hinge loss $\phi(z) = (1-z)_+$. Let ψ_i follow the empirical distribution over the data set of class i for $i = 1, 2, \ldots, m$, which implies that all the expectations in (16) become finite-sample averages over data classes. We set the parameters $\lambda = 5$ and $r_i = m - 1$ for $i = 2, \ldots, m$.

In all methods, the solution $\tilde{\mathbf{x}} = \mathbf{0}$ is used as the initial solution and it is feasible for (16). To apply SFLS and DFLS, we chose $T = 100$, $r^{(0)} = m$, and $\theta = 1.1$ across all data sets. Note that $r^{(0)} = m > m - 1 = f_0(\mathbf{0}) \geq f^*$ for (16). In DFLS, we solve subproblems via standard subgradient descent method. In SFLS and DFLS, we choose step size $\gamma_t = 0.05/\sqrt{t+1}$ for *connect-4* and *covtype* and choose $\gamma_t = 1/\sqrt{t+1}$ for *news20*. Both SFLS and YNW employed a mini-batch size of 1000 to construct the stochastic gradients and the objective values. We chose the number of iterations in YNW so that its total number of data passes is 200 for *connect-4* and *news20* and 100 for *covtype*. Then, we also terminated SFLS and DFLS when the total data passes they performed exceed YNW.

Figure 2 displays the performance of each method. The y-axes of the first row reports the term $f_0(\mathbf{x}) - f^*$, that is, it focuses on optimality, while this axis in the second row shows the feasibility of solutions by plotting $\max_{i=1,2,\ldots,m}\{f_i(\mathbf{x}) - r_i\}$. We track these measures as a function of the number of equivalent data passes performed by each algorithm in the x-axis, where a data pass involves going over the number of data points equal to the size of

data set	Number of classes	Number of instances	Number of features
connect-4	3	67557	126
covtype	7	581012	54
news20	20	15935	62061

Table 1: Characteristics of multi-class classification data sets from LIBSVM library

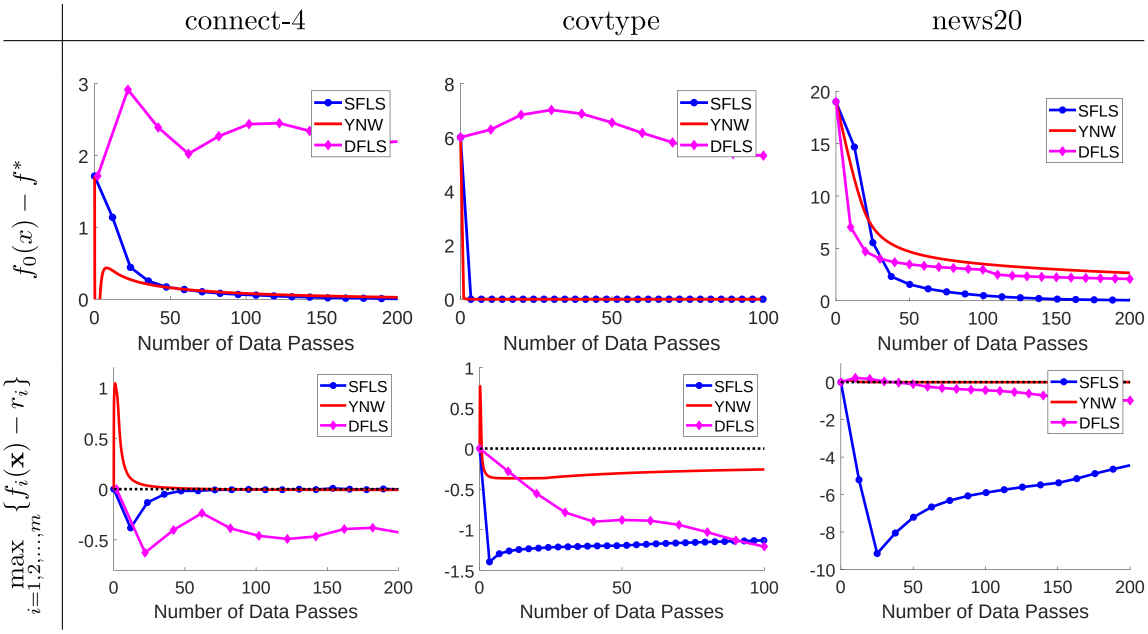

Figure 2: Performance of SFLS, YNW, and DFLS on the multi-class Neyman-Pearson classification problem.

the training data. This is possible since the expectations in our instances are over discrete random variables. Tracking data passes allows us to assess algorithms in terms of data complexity. Similar to Figure 1, we uses line markers to indicate the values of $f_0(x) - f^*$ and $\max_{i=1,2,\ldots,m} \{f_i(\mathbf{x}) - r_i\}$ corresponding to the solutions maintained at each SFLS outer iteration, while YNW has no line marker since it does not maintain feasibility. Since DFLS needs two data passes in each inner iteration, it can only perform one or two outer iterations with the number of data passes in Figure 1. Hence, for a better visualization, we use line markers to also indicate the inner iterations of DFLS instead of only outer iterations. In this figure, f^* is approximated by the objective value returned by DFLS after a sufficient number of data passes (i.e. at least 5000 data passes with $2T$ inner iterations.)

On the connect-4 data set, SFLS maintains feasibility and reduces the optimaliy gap quite rapidly after a few data passes. Interestingly, despite providing an initial feasible solution, YNW decreases the optimality gap at the beginning by moving to a highly infeasible solution. The peformance of both methods on the covtype data are comparable. On the news20 data set, SFLS provides feasible solutions with smaller optimality gaps sooner than the benchmark method. The comparison of SFLS and YNW highlights the advantage of SFLS in terms of feasibility. Specifically, efficient methods that do not emphasize feasibility could lead to highly infeasible solutions if terminated prematurely (e.g., the connect-4 data set).

DFLS also maintains a feasible solution path on all the data sets, as expected. However, its optimality gap reduces at a much slower rate with the number of data passes compared to SFLS because it uses deterministic subgradients based on the entire data set. These results thus underscore the importance of developing methods, such as SFLS, with low data complexity to balance optimality and feasibility.

6.4. Learning with Fairness Constraints

We consider learning a classifier with fairness constraints. Other examples include training predictive models with constraints on coverage rates, churn rates, and stability. Please see Goh et al. (2016) for further motivation and a non-convex formulation. Here we provide a convex formulation for these problems, which can be viewed as a tractable relaxation of the version in Goh et al. (2016) that admits the SOEC structure (1).

Suppose $(\mathbf{a}, b)$ is a data point from a distribution $\mathcal{D}$, where $\mathbf{a}$ is a feature vector and $b \in \{1, -1\}$ is the class label. Let $\mathcal{D}_M$ and $\mathcal{D}_F$ denote two different distributions of features (that are not necessarily labeled), which may represent male and female individuals. The goal is to train a classifier $\mathbf{a}^\top\mathbf{x}$ that minimizes classification loss. The correct classification of data vector $\mathbf{a}$ implies that $b\mathbf{a}^\top\mathbf{x} > 0$. One can train such a classifier subject to fairness constraints by solving

$$\begin{aligned}\min_{\|\mathbf{x}\|_2\leq\lambda} \quad & \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}[\phi(-b\mathbf{a}^\top\mathbf{x})] \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[\sigma(\mathbf{a}^\top\mathbf{x})] \leq \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_F}[\sigma(\mathbf{a}^\top\mathbf{x})]/\kappa, \\ & \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_F}[\sigma(\mathbf{a}^\top\mathbf{x})] \leq \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[\sigma(\mathbf{a}^\top\mathbf{x})]/\kappa,\end{aligned} \tag{17}$$

where λ is a regularization parameter, $\kappa \in (0, 1]$ is a constant, ϕ is a non-increasing loss function,

$$\sigma(z) = \max\{0, \min\{1, \{0.5 + z\}\},$$

and $\sigma(\mathbf{a}^\top\mathbf{x}) \in [0, 1]$ represents the probability of the (random) classifier $\mathbf{x}$ predicting $\mathbf{a}$ as positive. Therefore, $\mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[\sigma(\mathbf{a}^\top\mathbf{x})]$ and $\mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[\sigma(\mathbf{a}^\top\mathbf{x})]$ represent the percentages of instances in $\mathcal{D}_M$ and $\mathcal{D}_F$ predicted as positive, respectively. The first constraint guarantees that the percentage of the positively predicted instances in $\mathcal{D}_F$ is at least a κ fraction of that in $\mathcal{D}_M$. The second constraint has similar interpretation. An analogous model was considered in Goh et al. (2016) but it involves non-convex constraints.

Observing that $\sigma(\mathbf{a}^\top\mathbf{x}) = 1 - \sigma(-\mathbf{a}^\top\mathbf{x})$, we can reformulate the first constraint as $\mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[\sigma(\mathbf{a}^\top\mathbf{x})] + \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_F}[\sigma(-\mathbf{a}^\top\mathbf{x})]/\kappa \leq 1/\kappa$ and approximate σ by $\max\{0, 0.5+z\} = (0.5+z)_+$ so that we obtain a convex constraint $\mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[(\mathbf{a}^\top\mathbf{x}+0.5)_+] + \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_F}[(-\mathbf{a}^\top\mathbf{x}+0.5)_+]/\kappa \leq 1/\kappa$. Applying an analogous convex approximation to the second constraint, we obtain the

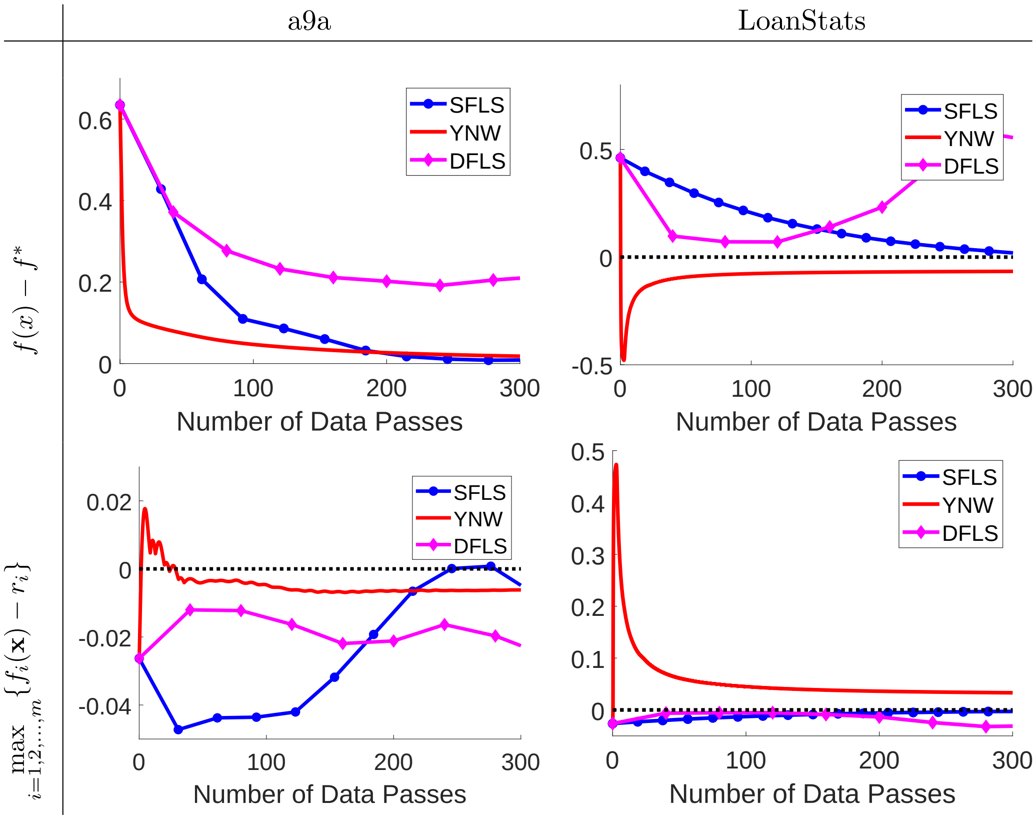

Figure 3: Performance of SFLS, YNW, and DFLS for solving the classification problem with fairness constraints.

following convex formulation for training a classifier subject to fairness constraints:

$$
\begin{aligned}
\min_{\|\mathbf{x}\|_2 \le \lambda} \quad & \mathbb{E}_{(\mathbf{a},b)\sim\mathcal{D}}[\phi(-b\mathbf{a}^\top\mathbf{x})] \\
\text{s.t.} \quad & \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[(\mathbf{a}^\top\mathbf{x}+0.5)_+] + \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_F}[(-\mathbf{a}^\top\mathbf{x}+0.5)_+]/\kappa \le 1/\kappa, \\
& \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_F}[(\mathbf{a}^\top\mathbf{x}+0.5)_+] + \mathbb{E}_{\mathbf{a}\sim\mathcal{D}_M}[(-\mathbf{a}^\top\mathbf{x}+0.5)_+]/\kappa \le 1/\kappa.
\end{aligned}
$$

The left hand side of the first constraint will be large if the classifier $\mathbf{x}$ is not “fair”, that is, it makes $\mathbf{a}^\top\mathbf{x}$ very negative for most of $\mathbf{a}$ from $\mathcal{D}_M$ but very positive for most of $\mathbf{a}$ from $\mathcal{D}_F$. Similarly, the left hand side of the second constraint will be large if the model $\mathbf{x}$ makes $\mathbf{a}^\top\mathbf{x}$ very positive for most of $\mathbf{a}$ from $\mathcal{D}_F$ but very negative for most of $\mathbf{a}$ from $\mathcal{D}_M$. Choosing an appropriate κ ensures that the obtained model is fair to both $\mathcal{D}_M$ and $\mathcal{D}_F$. Indeed, a solution that violates constraints in this formulation translates to a classifier that discriminates against one of the two classes.

For testing, we considered the “a9a” data set, also used by Goh et al. (2016) and another data set dubbed “LoanStats” from LendingClub.[4] We chose $\lambda = 5$, $\kappa = 0.95$, and

[4] https://www.lendingclub.com/info/statistics.action

$\phi(z) = (1-z)_+$ in each case. The distributions $\mathcal{D}$, $\mathcal{D}_M$, and $\mathcal{D}_F$ were defined as empirical distributions based on each data set as described below. The goal in the a9a data set is to predict people making more than 50,000 USD. Following Goh et al. (2016), we used the 32,561 training instances ($\mathcal{D}$) and the 16,281 testing instances in the data set to construct the objective function and constraints, respectively. Since we need male and female subsets to construct constraints, we further split the testing data into 14,720 male instances ($\mathcal{D}_M$) and 1,561 female instances ($\mathcal{D}_F$). The LoanStats data set contains information of $128,375$ loans issued in the fourth quarter of 2018 and the goal is to predict if a loan will be approved or rejected. After creating dummy variables, each loan is represented by a feature vector of 250 dimensions. We randomly partitioned the data set into a set of $63,890$ loans ($\mathcal{D}$) used to construct the objective function and a set of $64,485$ loans used to build the constraints. We further split the second set based on whether the feature "homeOwnership" equals "Mortgage" ($\mathcal{D}_M$) or some other value ($\mathcal{D}_F$) to obtain 31,966 and 32,519 loans in two subsets, respectively.

All methods are initialized at $\tilde{\mathbf{x}} = \mathbf{0}$, which is feasible for (16). In SFLS and DFLS, we chose $r^{(0)} = 1$, $\gamma_t = 0.1/\sqrt{t+1}$ and $\theta = 1.1$ across all data sets. Note that $r^{(0)} = 1 = f_0(\mathbf{0}) \geq f^*$. In SFLS, we chose $T = 300$ and $T = 200$ for a9a and LoanStats data sets, respectively. In DFLS, we chose $T = 100$ and $T = 50$ for a9a and LoanStats data sets, respectively. Both SFLS and YNW employed a mini-batch size of 500 and 1000 for a9a and LoanStats data sets, respectively. Similar to §6.3, we chose the number of iterations in YNW so that its total number of data passes is 300. Then, we also terminated SFLS and DFLS when the total data passes they performed exceed 300.

Figure 3 displays the performance of SFLS, YNW, and DFLS as a function of data passes. The interpretation of the axes and line markers in this figure are analogous to the ones in Figure 2. In this figure, f^* is approximated by the objective value returned by DFLS after a sufficient number of data passes (i.e. at least 5000 data passes with $2T$ inner iterations.) On the a9a data set, SFLS maintains a feasible solution path, as expected, while the YNW solutions are initially infeasible and become feasible with more data passes. The YNW reduces optimality gap more rapidly at the beginning while SFLS catches up quickly. The objective function value of YNW cannot be interpreted as an optimality gap when its solutions are infeasible since the corresponding objective function value can be super optimal. This feature is clearly visible on the LoanStats data. Here most of the YNW solutions are infeasible and superoptimal, that is, $f(x)-f^*$ is non-positive. The SFLS solution path continues to be feasible and suboptimal on this data set, with its suboptimality decreasing consistently after each outer iterations. DFLS also produces a feasible path but does not effectively reduce the optimality gap because its data complexity is high, that is, it requires a large number of data passes to achieve a small optimality gap. Similar to §6.3, we once again find that the low data complexity of SFLS is critical to balance optimality and feasibility when solving an SOEC.

7. Conclusion

We consider constrained optimization models where both the objective function and multiple constraints contain expectations of random convex functions. These models, referred to as stochastic optimization problems with expectation constraints (SOECs), arise in several machine learning, engineering, and business applications. We develop a stochastic feasible level-set method (SFLS) to solve SOECs, propose a tractable oracle to be used with SLFS, and analyze related iteration complexities. SFLS's total iteration complexity is comparable to stochastic subgradient methods in terms of ϵ but depends on a condition number—the cost of requiring feasibility. We evaluate the performance of SFLS across three applications involving approximate linear programming, multi-class classification, and learning classifiers with fairness constraints. We find that SFLS exhibits key advantages over existing methods. First, it ensures a feasible solution path with high probability while an existing state-of-the-art stochastic subgradient method can return highly infeasible solutions when terminated before conservative termination criteria are met. Infeasibilities may void the use of a solution in practice, especially if constraints model implementation requirements. Thus, the ability of SFLS to compute feasible solutions before convergence is practically relevant. Second, SFLS computes feasible solutions with small optimality gaps using only a few data passes owing to its low data-complexity, which is a desirable property when expectations are defined using large data sets that are expensive to scan. In contrast to SFLS, a recent deterministic feasible level set method exhibits high data complexity and large optimality gaps. Our theoretical and numerical findings bode well for the use of SFLS to solve SOECs and motivates further research into stochastic first order methods that emphasize feasibility.

Acknowledgments

We would like to thank the action editor and the two anonymous referees for the time and efforts reviewing our paper. Their suggestions have lead to an improved version of the paper. Tianbao Yang is partially supported by National Science Foundation CAREER Award 1844403.

Appendix A. Proofs of Theoretical Results

In this section, we provide the proofs of all technical results in the paper.

Proof of Lemma 3: Since $r > f^*$, it follows from Lemma 1(c) that $H(r) \leq 0$. Therefore, since $\theta \geq 1$ we have $\epsilon \leq -\frac{\theta-1}{\theta+1}H(r) \leq -H(r)$. Moreover, by Definition 2, we have $\mathcal{P}(r, \hat{\mathbf{x}}) \leq H(r) + \epsilon$ with probability of at least $1 - \delta$, which implies that $\hat{x}$ is a feasible solution to (2) since $\mathcal{P}(r, \hat{\mathbf{x}}) \leq H(r) + \epsilon \leq H(r) - H(r) \leq 0$. ■

Proof of Theorem 5 depends on the following lemma.

Lemma 12 *Given an input tuple $(r, \epsilon, \delta, \theta)$, a stochastic oracle $\mathcal{A}(r, \epsilon, \delta)$ with $0 < \epsilon \leq -\frac{\theta-1}{\theta+1}H(r)$ returns $U(r)$ and $\hat{\mathbf{x}}$ such that $\theta U(r) \leq H(r) \leq \mathcal{P}(r, \hat{\mathbf{x}}) \leq U(r)/\theta$ with probability of at least $1 - \delta$.*

Proof The inequality $H(r) \leq \mathcal{P}(r, \hat{\mathbf{x}})$ holds by definition of $H(r)$. By definition of stochastic oracle (Definition 2) and the property of ϵ, it follows that $\mathcal{P}(r, \hat{\mathbf{x}}) \leq H(r) + \epsilon \leq \frac{2}{\theta+1}H(r)$, $H(r) \leq U(r) + \epsilon \leq U(r) - \frac{\theta-1}{\theta+1}H(r)$, and $U(r) \leq H(r) + \epsilon \leq \frac{2}{\theta+1}H(r)$ hold with probability of at least $1 - \delta$. Since $r > f^*$, Lemma 1(c) implies that $H(r) \leq 0$. Therefore, using the inequality $U(r) \leq \frac{2}{\theta+1}H(r)$ we get $U(r) \leq 0$ and $\theta U(r) \leq \frac{\theta+1}{2}U(r) \leq H(r)$

since $\theta > 1$. Finally, combining the inequalities $\mathcal{P}(r, \hat{\mathbf{x}}) \leq \frac{2}{\theta+1} H(r)$ and $H(r) \leq U(r) - \frac{\theta-1}{\theta+1} H(r)$ (or equivalently $H(r) \leq \frac{\theta+1}{2\theta} U(r)$), we get $\mathcal{P}(r, \hat{x}) \leq \frac{2}{\theta+1} \cdot \frac{\theta+1}{2\theta} U(r) = U(r)/\theta$. ■

In the proof of Theorem 5 we need the following property of the condition measure β. In particular, it can be easily verified from the convexity of $H(r)$ and $H(r) - \delta \leq H(r+\delta) \leq H(r)$ for any $\delta \geq 0$ (Lemma 2.3.5 in Nesterov, 2004) that $\frac{H(r)}{r-f^*}$ is monotonically increasing in r on $(f^*, r^{(0)}]$ and

$$-\beta = \frac{H(r^{(0)})}{r^{(0)} - f^*} \geq \frac{H(r)}{r - f^*} \geq -1, \quad \forall r \in (f^*, r^{(0)}]. \tag{18}$$

Proof Theorem 5: We first show that the Algorithm 1 generates a feasible solution at each iteration with high probability. Let K be the largest value of k such that $r^{(k)} > f^*$ and the following inequality holds:

$$\epsilon_{\mathcal{A}} = -\frac{\theta - 1}{2\theta^2(\theta + 1)} H(r^{(0)})\epsilon \leq -\frac{\theta - 1}{\theta + 1} H(r^{(k)}). \tag{19}$$

Notice that $K \geq 0$ since $0 < \epsilon \leq 1 \leq 2\theta^2$ and $H(r^{(0)}) \leq 0$. It follows from Lemma 12 that with a probability of at least $1 - \delta^{(k)}$ we have,

$$\theta U(r^{(k)}) \leq H(r^{(k)}) \leq \mathcal{P}(r^{(k)}, \mathbf{x}^{(k)}) \leq U(r^{(k)})/\theta, \quad \text{for any } K \geq k \geq 0. \tag{20}$$

Since $r^{(k+1)} = r^{(k)} + U(r^{(k)})/(2\theta)$, we have

$$r^{(k+1)} - f^* = r^{(k)} - f^* + U(r^{(k)})/(2\theta) \geq r^{(k)} - f^* + H(r^{(k)})/2 \geq \frac{1}{2}(r^{(k)} - f^*), \tag{21}$$

and

$$r^{(k+1)} - f^* = r^{(k)} - f^* + U(r^{(k)})/(2\theta) \leq r^{(k)} - f^* + \frac{H(r^{(k)})}{2\theta^2} \leq \left(1 - \frac{\beta}{2\theta^2}\right)(r^{(k)} - f^*) \tag{22}$$

with a probability of at least $1 - \delta^{(k)}$, where the last inequalities in both (21) and (22) follow from (18). Inequality (21) and the condition $r^{(k)} > f^*$ imply that $r^{(k+1)} > f^*$. Applying this argument recurrently and using the fact that $\sum_{k=0}^{\infty} \delta^{(k)} = \delta$, we have (21), (22) and $r^{(k+1)} > f^*$ holds for $k = 0, 1, \ldots, K$. Therefore, since $\epsilon_{\mathcal{A}} \leq -\frac{\theta-1}{\theta+1} H(r^{(k)}) \leq -H(r^{(k)})$ for $k = 0, 1, \ldots, K$, Lemma 3 implies the solution $\mathbf{x}^{(k)}$ generated at iteration $k = 0, 1, \ldots, K$ is feasible to (1) with a probability of at least $1 - \delta$. We next show that (19) holds with a high probability until Algorithm 1 terminates. By the definition of K, we know that (19) is violated when $k = K + 1$, i.e. $-\frac{\theta-1}{2\theta^2(\theta+1)} H(r^{(0)})\epsilon > -\frac{\theta-1}{\theta+1} H(r^{(K+1)})$. Since $r^{(k+1)} \leq r^{(k)}$ and $\frac{H(r)}{r-f^*}$ is monotonically increasing, we can show that

$$-\frac{\theta - 1}{2\theta^2(\theta + 1)} H(r^{(0)})\epsilon > -\frac{\theta - 1}{\theta + 1} H(r^{(K+1)}) \geq -\frac{\theta - 1}{\theta + 1} H(r^{(K)}) \frac{r^{(K+1)} - f^*}{r^{(K)} - f^*} \geq -\frac{\theta - 1}{2(\theta + 1)} H(r^{(K)}), \tag{23}$$

where the last inequality holds by (21). Using the definition of ϵ_{opt}, (23), and (20) for $k = K$ (specifically, $H(r^{(K)}) \leq U(r^{(K)})/\theta$), we have

$$-\frac{\theta-1}{2\theta(\theta+1)}\epsilon_{\text{opt}} = \frac{\theta-1}{2\theta^2(\theta+1)}H(r^{(0)})\epsilon \leq \frac{\theta-1}{2(\theta+1)}H(r^{(K)}) \leq \frac{\theta-1}{2\theta(\theta+1)}U(r^{(K)}),$$

which indicates that Algorithm 1 must stop before $k = K+1$. Therefore, SFLS generates a feasible solution with a probability of at least $1-\delta$ at each iteration before termination.

We now proceed to establish that the terminal solution of SFLS is relative ϵ-optimal solution. By definition of $\mathcal{P}(r^{(k)}, \mathbf{x}^{(k)})$ and (20) it follows that $f_0(\mathbf{x}^{(k)}) - r^{(k)} \leq \mathcal{P}(r^{(k)}, \mathbf{x}^{(k)}) \leq H(r^{(k)})/\theta^2 \leq 0$ for all k. Hence,

$$f_0(\mathbf{x}^{(k)}) - f^* \leq r^{(k)} - f^*, \quad \text{for all } k = 0, 1, 2, \ldots, K. \tag{24}$$

Combining (24) and $r^{(k)} - f^* \leq (r^{(0)} - f^*)H(r^{(k)})/H(r^{(0)})$ derived from (18) stipulates that with a probability of at least $1-\delta$:

$$\frac{f_0(\mathbf{x}^{(k)}) - f^*}{r^{(0)} - f^*} \leq \frac{H(r^{(k)})}{H(r^{(0)})} \leq \frac{\theta U(r^{(k)})}{H(r^{(0)})},$$

where we used (20) in the second inequality. Hence, at termination of Algorithm 1 we get $\frac{f_0(\mathbf{x}^{(k)})-f^*}{r^{(0)}-f^*} \leq \epsilon$ since the algorithm stops when $\theta U(r^{(k)}) \geq H(r^{(0)})\epsilon$.

Finally we show that $K := \dfrac{2\theta^2}{\beta}\ln\left(\dfrac{\theta^2}{\beta\epsilon}\right)$. By recursively applying inequality (22) we get

$$0 \leq r^{(k)} - f^* \leq \left(1 - \frac{\beta}{2\theta^2}\right)^k (r^{(0)} - f^*), \quad \text{for all } k \tag{25}$$

with probability of at least $1-\delta$, which implies $r^{(K)} - f^* \leq -\frac{H(r^{(0)})\epsilon}{\theta^2}$ for the choice of K. Hence, we have $-U(r^{(K)}) \leq -\theta H(r^{(K)}) \leq \theta(r^{(K)} - f^*) \leq -\epsilon H(r^{(0)})/\theta$ where the first inequality follows by (20), the second by (18), and the third by (25). This indicates that the stopping criterion of Algorithm 1 holds with a probability of at least $1-\delta$ when $k = K$ and SFLS requires at most K calls to oracle $\mathcal{A}$. ■

Proof of Proposition 6: The proof of the first part directly follows from Proposition 3.2 in Nemirovski et al., 2009. We only show that SMD is a valid oracle. It is straightforward to see that the inequality $U(\bar{x}^{(t)}) - L(\bar{y}^{(t)}) \leq \epsilon_{\mathcal{A}}$ implies $\mathcal{P}(r, \bar{x}^{(t)}) - H(r) \leq U(\bar{x}^{(t)}) - H(r) \leq U(\bar{x}^{(t)}) - L(\bar{y}^{(t)}) \leq \epsilon_{\mathcal{A}}$, where the first inequality holds since $U(\bar{x}^{(t)})$ is an upper bound on $\mathcal{P}(r, \bar{x}^{(t)})$ and the second since $L(\bar{y}^{(t)})$ is a lower bound on $H(r)$. This indicates that the conditions provided in Definition 2 are satisfied. ■

To show part (i) of Proposition 7, we use known lemmas 13 and 14 as well as prove lemmas 15 and 16. To prove part (ii) of this proposition we need Lemma 17. Before stating these lemmas, we present some required notation and representations, which we present next. We denote the diameter of $\mathcal{Z}$ with respect to ω_z by

$$D_z := \sqrt{\max_{\mathbf{z}\in\mathcal{Z}} \omega_z(\mathbf{z}) - \min_{\mathbf{z}\in\mathcal{Z}} \omega_z(\mathbf{z})} = 1.$$

In addition, for any $\boldsymbol{\zeta}_x \in \mathbb{R}^d$, $\boldsymbol{\zeta}_y \in \mathbb{R}^{m+1}$, $\mathbf{x}' \in \mathcal{X}^o$, $\mathbf{y}' \in \mathcal{Y}^o$, and $\mathbf{z}' = (\mathbf{x}', \mathbf{y}') \in \mathcal{Z}^o$, it is easy to verify for $\boldsymbol{\zeta} = (\boldsymbol{\zeta}_x, \boldsymbol{\zeta}_y)$ that

$$P_{\mathbf{z}'}(\boldsymbol{\zeta}) = \left(P^x_{\mathbf{x}'}(2D_x^2 \boldsymbol{\zeta}_x), P^y_{\mathbf{y}'}(2D_y^2 \boldsymbol{\zeta}_y) \right), \tag{26}$$

where $P^x_{\mathbf{x}'}(\boldsymbol{\zeta}_x) := \arg\min_{\mathbf{x} \in \mathcal{X}} \{\boldsymbol{\zeta}_x^\top (\mathbf{x} - \mathbf{x}') + V_x(\mathbf{x}', \mathbf{x})\}$ and $P^y_{\mathbf{y}'}(\boldsymbol{\zeta}_y) := \arg\min_{\mathbf{y} \in \mathcal{Y}} \{\boldsymbol{\zeta}_y^\top (\mathbf{y} - \mathbf{y}') + V_y(\mathbf{y}', \mathbf{y})\}$.

Lemma 13 (Equation (2.37) and Lemma 6.1 in Nemirovski et al., 2009) *1. Let $\boldsymbol{\zeta}_x^{(t)} \in \mathbb{R}^d$, $t = 0, 1, 2, \ldots$ be a set of random variables, $\mathbf{v}^{(0)} \in \mathcal{X}^o$ and $\mathbf{v}^{(t+1)} = P^x_{\mathbf{v}^{(t)}}(\boldsymbol{\zeta}_x^{(t)})$ for $t = 0, 1, 2, \ldots$. For any $\mathbf{v} \in \mathcal{X}$ and $t \geq 1$, we have*

$$\sum_{s=0}^{t} (\mathbf{v}^{(s)} - \mathbf{v})^\top \boldsymbol{\zeta}_x^{(s)} \leq V_x(\mathbf{v}^{(0)}, \mathbf{v}) + \frac{1}{2\alpha_x} \sum_{s=0}^{t} \left\| \boldsymbol{\zeta}_x^{(s)} \right\|_{*,x}^2 .$$

2. Let $\boldsymbol{\zeta}_y^{(t)} \in \mathbb{R}^{m+1}$, $t = 0, 1, 2, \ldots$ be a set of random variables, $\mathbf{v}^{(0)} \in \mathcal{Y}^o$ and $\mathbf{v}^{(t+1)} = P^y_{\mathbf{v}^{(t)}}(\boldsymbol{\zeta}_y^{(t)})$ for $t = 0, 1, 2, \ldots$. For any $\mathbf{v} \in \mathcal{Y}$ and $t \geq 1$, we have

$$\sum_{s=0}^{t} (\mathbf{v}^{(s)} - \mathbf{v})^\top \boldsymbol{\zeta}_y^{(s)} \leq V_y(\mathbf{v}^{(0)}, \mathbf{v}) + \frac{1}{2\alpha_y} \sum_{s=0}^{t} \left\| \boldsymbol{\zeta}_y^{(s)} \right\|_{*,y}^2 .$$

3. Let $\boldsymbol{\zeta}^{(t)} \in \mathbb{R}^{d+m+1}$, $t = 0, 1, 2, \ldots$ be a set of random variables, $\mathbf{v}^{(0)} \in \mathcal{Z}^o$ and $\mathbf{v}^{(t+1)} = P_{\mathbf{v}^{(t)}}(\boldsymbol{\zeta}^{(t)})$ for $t = 0, 1, 2, \ldots$. For any $\mathbf{v} \in \mathcal{Z}$ and $t \geq 1$, we have

$$\sum_{s=0}^{t} (\mathbf{v}^{(s)} - \mathbf{v})^\top \boldsymbol{\zeta}^{(s)} \leq V(\mathbf{v}^{(0)}, \mathbf{v}) + \frac{1}{2} \sum_{s=0}^{t} \left\| \boldsymbol{\zeta}^{(s)} \right\|_{*,z}^2 .$$

Lemma 14 (Lemma 2 in Lan et al., 2012) *Let $\boldsymbol{\xi}^{(t)}$ and $\sigma_t > 0$ for $t \geq 0$ be respectively a sequence of i.i.d. random variables and deterministic numbers; $\boldsymbol{\xi}^{[t]} = (\boldsymbol{\xi}^{(0)}, \boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(t)})$; $\mathbb{E}_t$ the conditional expectation conditioning on $\boldsymbol{\xi}^{[t-1]}$ for $t \geq 1$; and $\psi_t(\boldsymbol{\xi}^{[t]})$ be a measurable function of $\boldsymbol{\xi}^{[t]}$ such that either*

Case A: $\mathbb{E}_t \left[\psi_t\left(\boldsymbol{\xi}^{[t]}\right)\right] = 0$ *and* $\mathbb{E}_t \left[\exp\left(\psi_t\left(\boldsymbol{\xi}^{[t]}\right)^2 / \sigma_t^2\right)\right] \leq \exp(1)$*, or*

Case B: $\mathbb{E}_t \left[\exp\left(\left|\psi_t\left(\boldsymbol{\xi}^{[t]}\right)\right| / \sigma_t\right)\right] \leq \exp(1)$ *,*

almost surely for all t. Then for any $\Omega > 0$, we have the followings:
In case A:

$$Prob\left\{ \sum_{s=0}^{t} \psi_s > \Omega \sqrt{\sum_{s=0}^{t} \sigma_s^2} \right\} \leq \exp(-\Omega^2/3).$$

In case B:

$$Prob\left\{ \sum_{s=0}^{t} \psi_s > \|\sigma^{[t]}\|_1 + \Omega \|\sigma^{[t]}\|_2 \right\} \leq \exp(-\Omega^2/12) + \exp(-3\Omega/4),$$

where $\sigma^{[t]} = (\sigma_0, \sigma_1, \ldots, \sigma_t)^\top$.

Lemma 15 shows that the stochastic subgradient $G(\cdot,\cdot,\cdot)$ has a light-tailed distribution and bounds the Bregmann distances. Define

$$\Delta_t \;:=\; G(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)}) - g(\mathbf{x}^{(t)},\mathbf{y}^{(t)}) = \begin{bmatrix} \Delta_t^x \\ -\Delta_t^y \end{bmatrix} := \begin{bmatrix} G_x(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)}) - g_x(\mathbf{x}^{(t)},\mathbf{y}^{(t)}) \\ g_y(\mathbf{x}^{(t)},\mathbf{y}^{(t)}) - G_y(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)}) \end{bmatrix}.$$

Lemma 15 *The following inequalities hold:*

$$\mathbb{E}_t\left[\exp\left(\|\Delta_t\|_{*,z}^2/(2M)^2\right)\right] \le \exp(1), \tag{27}$$

$$\mathbb{E}_t\left[\exp\left(\|\Delta_t^x\|_{*,x}^2/(2M_x)^2\right)\right] \le \exp(1), \tag{28}$$

$$\mathbb{E}_t\left[\exp\left(\|\Delta_t^y\|_{*,y}^2/(2M_y)^2\right)\right] \le \exp(1). \tag{29}$$

Moreover, when $\mathbf{z}' = (\mathbf{x}',\mathbf{y}') := \arg\min_{\mathbf{z}\in\mathcal{Z}} \omega_z(\mathbf{z})$, *we have*

$$\frac{\alpha_x}{2}\|\mathbf{x}'-\mathbf{x}\|_x^2 \le V_x(\mathbf{x}',\mathbf{x}) \le D_x^2, \qquad \textit{for all } \mathbf{x}\in\mathcal{X}, \tag{30}$$

$$\frac{\alpha_y}{2}\|\mathbf{y}'-\mathbf{y}\|_y^2 \le V_y(\mathbf{y}',\mathbf{y}) \le D_y^2, \qquad \textit{for all } \mathbf{x}\in\mathcal{Y}, \tag{31}$$

$$\frac{1}{2}\|\mathbf{z}'-\mathbf{z}\|_z^2 \le V(\mathbf{z}',\mathbf{z}) \;\le D_z^2 = 1, \quad \textit{for all } \mathbf{z}\in\mathcal{Z}. \tag{32}$$

Proof Applying Jensen's inequality and using the definitions of $\|\cdot\|_{*,z}$, M, and the inequalities (6) and (7), we have

$$\begin{aligned} & \mathbb{E}\left[\exp\left(\|G(\mathbf{x},\mathbf{y},\boldsymbol{\xi})\|_{*,z}^2/M^2\right)\right] \\ = \;& \mathbb{E}\left[\exp\left(\frac{\frac{2D_x^2}{\alpha_x}\|G_x(\mathbf{x},\mathbf{y},\boldsymbol{\xi})\|_{*,x}^2 + \frac{2D_y^2}{\alpha_y}\|G_y(\mathbf{x},\mathbf{y},\boldsymbol{\xi})\|_{*,y}^2}{\frac{2D_x^2}{\alpha_x}M_x^2 + \frac{2D_y^2}{\alpha_y}M_y^2}\right)\right] \\ \le \;& \frac{\frac{2D_x^2}{\alpha_x}M_x^2\mathbb{E}\left[\exp\left(\|G_x(\mathbf{x},\mathbf{y},\boldsymbol{\xi})\|_{*,x}^2/M_x^2\right)\right] + \frac{2D_y^2}{\alpha_y}M_y^2\mathbb{E}\left[\exp\left(\|G_y(\mathbf{x},\mathbf{y},\boldsymbol{\xi})\|_{*,y}^2/M_y^2\right)\right]}{\frac{2D_x^2}{\alpha_x}M_x^2 + \frac{2D_y^2}{\alpha_y}M_y^2} \\ \le \;& \exp(1). \end{aligned} \tag{33}$$

Using (33) and Jensen's inequality, it follows that

$$\left\|g(\mathbf{x}^{(t)},\mathbf{y}^{(t)})\right\|_{*,z}^2 \le \mathbb{E}_t\left[\left\|G(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)})\right\|_{*,z}^2\right] \le M^2. \tag{34}$$

Hence, we have

$$\begin{aligned} & \mathbb{E}_t\left[\exp\left(\|\Delta_t\|_{*,z}^2/(2M)^2\right)\right] \\ & \le \mathbb{E}_t\left[\exp(2\|G(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)})\|_{*,z}^2/(2M)^2)\exp(2\|g(\mathbf{x}^{(t)},\mathbf{y}^{(t)})\|_{*,z}^2/(2M)^2)\right], \\ & \le \mathbb{E}_t\left[\sqrt{\exp(\|G(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)})\|_{*,z}^2/M^2)}\exp(1/2)\right], \\ & \le \sqrt{\mathbb{E}_t\left[\exp(\|G(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)})\|_{*,z}^2/M^2)\right]}\exp(1/2), \\ & \le \exp(1/2)\exp(1/2) = \exp(1), \end{aligned} \tag{35}$$

where the first inequality follows from the definition of Δ_t and the inequality $\|a+b\|^2 \leq 2a^2+2b^2$ for any $a, b \in \mathbb{R}$, the second from (34), the third from Jensen's inequality for concave functions, and the fourth by inequalities (6) and (7). Following a similar argument, we can also show that $\mathbb{E}_t\left[\exp\left(\|\Delta_t^x\|_{*,x}^2/(2M_x)^2\right)\right] \leq \exp(1)$ and $\mathbb{E}_t\left[\exp\left(\|\Delta_t^y\|_{*,y}^2/(2M_y)^2\right)\right] \leq \exp(1)$. Finally, inequalities (30), (31), and (32) follow because ω_x, ω_y and ω_z are modulus α_x, α_y and 1, respectively. ■

Lemma 16 *Let* $\nu_{s,t} := \dfrac{\gamma_s}{\sum_{s'=0}^t \gamma_{s'}}$. *Given* $\Omega > 0$, *Algorithm 3 computes* $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$, $t = 1, 2, 3, \ldots,$ *such that*

$$Prob\left\{u_*^{(t)} - l_*^{(t)} > 4\sqrt{2}\Omega M\sqrt{\sum_{s=0}^t \nu_{s,t}^2} + \frac{2+2.5M^2\sum_{s=0}^t\gamma_s^2}{\sum_{s=0}^t\gamma_s} + 2.5\Omega M^2\sqrt{\sum_{s=0}^t\gamma_s^2\nu_{s,t}^2}\right\}$$
$$\leq \exp(-\Omega^2/3) + \exp(-\Omega^2/12) + \exp(-3\Omega/4). \tag{36}$$

Proof Since $\mathbf{z}^{(0)} \in \arg\min_{\mathbf{z}\in\mathcal{Z}} \omega_z(\mathbf{z})$ and $\mathbf{z}^{(t+1)} = P_{\mathbf{z}^{(t)}}(\gamma_t G(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\xi}^{(t)}))$ in Algorithm 3, by Lemma 13 we have, for any $\mathbf{z} \in \mathcal{Z}$,

$$\begin{aligned}\sum_{s=0}^t \gamma_s(\mathbf{z}^{(s)} - \mathbf{z})^\top G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) &\leq V(\mathbf{z}^{(0)}, \mathbf{z}) + \frac{1}{2}\sum_{s=0}^t \gamma_s^2 \left\|G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})\right\|_{*,z}^2 \\ &\leq 1 + \frac{1}{2}\sum_{s=0}^t \gamma_s^2 \left\|G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})\right\|_{*,z}^2,\end{aligned} \tag{37}$$

where the second inequality follows by (32). In addition, by definition of Δ_t, for any $\mathbf{z} \in \mathcal{Z}$ we have

$$\begin{aligned}&\frac{1}{\sum_{s=0}^t\gamma_s}\sum_{s=0}^t \gamma_s(\mathbf{z}^{(s)} - \mathbf{z})^\top G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) \\ &= \frac{\sum_{s=0}^t \gamma_s(\mathbf{x}^{(s)} - \mathbf{x})^\top g_x(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})}{\sum_{s=0}^t\gamma_s} - \frac{\sum_{s=0}^t \gamma_s(\mathbf{y}^{(s)} - \mathbf{y})^\top g_y(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})}{\sum_{s=0}^t\gamma_s} + \frac{\sum_{s=0}^t \gamma_s(\mathbf{z}^{(s)} - \mathbf{z})^\top \Delta_s}{\sum_{s=0}^t\gamma_s}\end{aligned} \tag{38}$$

Applying (37) to (38) and reorganizing terms lead to

$$\begin{aligned}&\frac{\sum_{s=0}^t \gamma_s(\mathbf{x}^{(s)} - \mathbf{x})^\top g_x(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})}{\sum_{s=0}^t\gamma_s} - \frac{\sum_{s=0}^t \gamma_s(\mathbf{y}^{(s)} - \mathbf{y})^\top g_y(\mathbf{x}^{(s)}, \mathbf{y}^{(s)})}{\sum_{s=0}^t\gamma_s} \\ &\leq \frac{\sum_{s=0}^t \gamma_s(\mathbf{z} - \mathbf{z}^{(s)})^\top \Delta_s}{\sum_{s=0}^t\gamma_s} + \frac{1 + 0.5\sum_{s=0}^t \gamma_s^2 \left\|G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})\right\|_{*,z}^2}{\sum_{s=0}^t\gamma_s}.\end{aligned}$$

Maximizing both sides of the above inequality over $\mathbf{z} \in \mathcal{Z}$ implies

$$u_*^{(t)} - l_*^{(t)} \leq \frac{\max\limits_{\mathbf{z}\in\mathcal{Z}}\left[\sum_{s=0}^t \gamma_s(\mathbf{z} - \mathbf{z}^{(s)})^\top \Delta_s\right]}{\sum_{s=0}^t\gamma_s} + \frac{1 + 0.5\sum_{s=0}^t \gamma_s^2 \left\|G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})\right\|_{*,z}^2}{\sum_{s=0}^t\gamma_s}. \tag{39}$$

Let $\mathbf{v}^{(0)} = \mathbf{z}^{(0)}$ and $\mathbf{v}^{(t+1)} = P_{\mathbf{v}^{(t)}}(-\gamma_t \Delta_t)$ for $t = 0, 1, 2, \ldots$. From Lemma 13 it follows that for any $\mathbf{z} \in \mathcal{Z}$,

$$-\sum_{s=0}^{t} \gamma_s (\mathbf{v}^{(s)} - \mathbf{z})^\top \Delta_s \leq 1 + 0.5 \sum_{s=0}^{t} \gamma_s^2 \|\Delta_s\|_{*,z}^2, \tag{40}$$

Rewriting $\mathbf{z} - \mathbf{z}^{(s)} = \mathbf{v}^{(s)} - \mathbf{z}^{(s)} + \mathbf{z} - \mathbf{v}^{(s)}$ and applying (40) to (39) yield

$$u_*^{(t)} - l_*^{(t)} \leq \frac{\sum_{s=0}^{t} \gamma_s (\mathbf{v}^{(s)} - \mathbf{z}^{(s)})^\top \Delta_s}{\sum_{s=0}^{t} \gamma_s} + \frac{2 + 0.5 \sum_{s=0}^{t} \gamma_s^2 \left(\left\| G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) \right\|_{*,z}^2 + \|\Delta_s\|_{*,z}^2 \right)}{\sum_{s=0}^{t} \gamma_s}. \tag{41}$$

We next find a probabilistic bound for the right hand side of the above inequality.

Bound on $\frac{\sum_{s=0}^{t} \gamma_s (\mathbf{v}^{(s)} - \mathbf{z}^{(s)})^\top \Delta_s}{\sum_{s=0}^{t} \gamma_s}$*:* By our choice of $\mathbf{z}^{(0)}$, i.e. $\mathbf{z}^{(0)} = \arg\min_{\mathbf{z} \in \mathcal{Z}} \omega_z(\mathbf{z})$ and (32), for any $s = 0, 1, \ldots, t$ we have

$$\|\mathbf{v}^{(s)} - \mathbf{z}^{(s)}\|_z \leq \|\mathbf{z}^{(s)} - \mathbf{z}^{(0)}\|_z + \|\mathbf{v}^{(s)} - \mathbf{z}^{(0)}\|_z \leq \sqrt{2V(\mathbf{z}^{(0)}, \mathbf{z}^{(s)})} + \sqrt{2V(\mathbf{z}^{(0)}, \mathbf{v}^{(s)})} \leq 2\sqrt{2}. \tag{42}$$

Define $\psi_s := \nu_{s,t} (\mathbf{v}^{(s)} - \mathbf{z}^{(s)})^\top \Delta_s$ and $\sigma_s := 4\sqrt{2} M \nu_{s,t}$. Because $\boldsymbol{\xi}^{(s)}$ is independent of $\mathbf{v}^{(s)}$ and $\mathbf{z}^{(s)}$, we have $\mathbb{E}_s[\psi_s] = 0$. In addition, it can be verified that $\psi_s^2 \leq \nu_{s,t}^2 \|\mathbf{v}^{(s)} - \mathbf{z}^{(s)}\|_z^2 \|\Delta_s\|_{*,z}^2 \leq 8\nu_{s,t}^2 \|\Delta_s\|_{*,z}^2$, where the second inequality holds by (42). Using this inequality and (35), we get $\mathbb{E}_s \left[\exp\left(\psi_s^2 / \sigma_s^2 \right) \right] \leq \mathbb{E}_s \left[\exp\left(\|\Delta_s\|_{*,z}^2 / (2M)^2 \right) \right] \leq \exp(1)$. Hence, it follows from Case A in Lemma 14 that

$$\operatorname{Prob}\left\{ \frac{\sum_{s=0}^{t} \gamma_s (\mathbf{v}^{(s)} - \mathbf{z}^{(s)})^\top \Delta_s}{\sum_{s=0}^{t} \gamma_s} > 4\sqrt{2} \Omega M \sqrt{\sum_{s=0}^{t} \nu_{s,t}^2} \right\} \leq \exp(-\Omega^2/3). \tag{43}$$

Bound on $\frac{\sum_{s=0}^{t} \gamma_s^2 \left(\left\| G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) \right\|_{*,z}^2 + \|\Delta_s\|_{*,z}^2 \right)}{\sum_{s=0}^{t} \gamma_s}$*:* Let $\psi_s := \gamma_s \nu_{s,t} \left(\left\| G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) \right\|_{*,z}^2 + \|\Delta_s\|_{*,z}^2 \right)$ and $\sigma_s := 5M^2 \gamma_s \nu_{s,t}$. We then have

$$\begin{aligned} \mathbb{E}\left[\exp\left(|\psi_s| / \sigma_s \right) \right] &= \mathbb{E}\left[\exp\left(\frac{\|G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})\|_{*,z}^2 + \|\Delta_s\|_{*,z}^2}{5M^2} \right) \right] \\ &= \mathbb{E}\left[\exp\left(\frac{\left\| G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) \right\|_{*,z}^2 / M^2 + 4\|\Delta_s\|_{*,z}^2 / (4M^2)}{5} \right) \right] \\ &\leq \frac{1}{5} \mathbb{E}\left[\exp\left(\frac{\left\| G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)}) \right\|_{*,z}^2}{M^2} \right) \right] + \frac{4}{5} \mathbb{E}\left[\exp\left(\frac{\|\Delta_s\|_{*,z}^2}{4M^2} \right) \right] \leq \exp(1), \end{aligned}$$

where the first inequality is from Jensen's inequality and the second inequality is from (33) and (35). Hence, from Case B in Lemma 14 it follows that

$$\begin{aligned} &\operatorname{Prob}\left\{ \frac{\sum_{s=0}^{t} \gamma_s^2 \left(\|G(\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \boldsymbol{\xi}^{(s)})\|_{*,z}^2 + \|\Delta_s\|_{*,z}^2 \right)}{\sum_{s=0}^{t} \gamma_s} > 5M^2 \sum_{s=0}^{t} \gamma_s \nu_{s,t} + 5\Omega M^2 \sqrt{\sum_{s=0}^{t} \gamma_s^2 \nu_{s,t}^2} \right\} \\ &\leq \exp(-\Omega^2/12) + \exp(-3\Omega/4). \end{aligned} \tag{44}$$

The conclusion is hence obtained by upper bounding the right hand size of (41) using the union bound of (43) and (44). ■

Lemma 17 *Let* $\nu_{s,t} := \frac{\gamma_s}{\sum_{s'=0}^{t}\gamma_{s'}}$. *Given* $\Omega > 0$, *Algorithm 3 guarantees that*

$$Prob\left\{\left|\hat{l}_*^{(t)} - l_*^{(t)}\right| > \left(\Omega Q + \frac{4\sqrt{2}\Omega D_x M_x}{\sqrt{\alpha_x}}\right)\sqrt{\sum_{s=0}^{t}\nu_{s,t}^2} + \frac{0.5 + \frac{4D_x^2M_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2}{\sum_{s=0}^{t}\gamma_s} \right.$$
$$\left. + \frac{4\Omega D_x^2 M_x^2}{\alpha_x}\sqrt{\sum_{s=0}^{t}\gamma_s^2\nu_{s,t}^2}\right\} \leq 6\exp(-\Omega^2/3) + \exp(-\Omega^2/12) + \exp\left(-3\Omega/4\right). \tag{45}$$

and

$$Prob\left\{\left|\hat{u}_*^{(t)} - u_*^{(t)}\right| > \left(\Omega Q + \frac{4\sqrt{2}\Omega D_y M_y}{\sqrt{\alpha_y}}\right)\sqrt{\sum_{s=0}^{t}\nu_{s,t}^2} + \frac{0.5 + \frac{4D_y^2M_y^2}{\alpha_y}\sum_{s=0}^{t}\gamma_s^2}{\sum_{s=0}^{t}\gamma_s} \right.$$
$$\left. + \frac{4\Omega D_y^2 M_y^2}{\alpha_y}\sqrt{\sum_{s=0}^{t}\gamma_s^2\nu_{s,t}^2}\right\} \leq 6\exp(-\Omega^2/3) + \exp(-\Omega^2/12) + \exp\left(-3\Omega/4\right). \tag{46}$$

Proof Since the proofs of (45) and (46) are very similar, we will only prove (45). Let

$$l^{(t)}(\mathbf{x}) := \frac{1}{\sum_{s=0}^{t}\gamma_s}\sum_{s=0}^{t}\gamma_s\left[\phi(\mathbf{x}^{(s)},\mathbf{y}^{(s)}) + g_x(\mathbf{x}^{(s)},\mathbf{y}^{(s)})^\top(\mathbf{x}-\mathbf{x}^{(s)})\right],$$

and

$$\hat{l}^t(\mathbf{x}) := \frac{1}{\sum_{s=0}^{t}\gamma_s}\sum_{s=0}^{t}\gamma_s\left[\Phi(\mathbf{x}^{(s)},\mathbf{y}^{(s)},\boldsymbol{\xi}^{(s)}) + G_x(\mathbf{x}^{(s)},\mathbf{y}^{(s)},\boldsymbol{\xi}^{(s)})^\top(\mathbf{x}-\mathbf{x}^{(s)})\right].$$

Define $\delta_t := \Phi(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\xi}^{(t)}) - \phi(\mathbf{x}^{(t)},\mathbf{y}^{(t)})$. Using this definition and those of $l_*^{(t)} = \min_{\mathbf{x}\in\mathcal{X}} l^{(t)}(\mathbf{x})$, $\hat{l}_*^{(t)} = \min_{\mathbf{x}\in\mathcal{X}}\hat{l}^{(t)}(\mathbf{x})$, and Δ_t we have

$$\begin{aligned}\left|\hat{l}_*^{(t)} - l_*^{(t)}\right| &= \left|\min_{\mathbf{x}\in\mathcal{X}}\hat{l}^{(t)}(\mathbf{x}) - \min_{\mathbf{x}\in\mathcal{X}} l^{(t)}(\mathbf{x})\right| \\ &\leq \max_{\mathbf{x}\in\mathcal{X}}\left|\hat{l}^{(t)}(\mathbf{x}) - l^{(t)}(\mathbf{x})\right| \\ &\leq \max_{\mathbf{x}\in\mathcal{X}}\left|\frac{\sum_{s=0}^{t}\gamma_s(\mathbf{x}-\mathbf{x}^{(s)})^\top\Delta_s^x}{\sum_{s=0}^{t}\gamma_s}\right| + \left|\frac{\sum_{s=0}^{t}\gamma_s\delta_s}{\sum_{s=0}^{t}\gamma_s}\right|.\end{aligned} \tag{47}$$

By (26) and line 5 of Algorithm 3, we have $\mathbf{x}^{(t+1)} = P^x_{\mathbf{x}^{(t)}}(2D_x^2\gamma_t G_x(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\xi}^{(t)}))$. Let $\mathbf{w}^{(0)} = \mathbf{v}^{(0)} = \mathbf{x}^{(0)}$, $\mathbf{w}^{(t+1)} := P^x_{\mathbf{w}^{(t)}}(-2D_x^2\gamma_t\Delta_t^x)$ and $\mathbf{v}^{(t+1)} := P^x_{\mathbf{v}^{(t)}}(2D_x^2\gamma_t\Delta_t^x)$ for $t = 0, 1, 2, \ldots$. From Lemma 13 and (30) it follows that

$$-\sum_{s=0}^{t}\gamma_s(\mathbf{w}^{(s)} - \mathbf{x})^\top\Delta_s^x \leq \frac{V_x(\mathbf{w}^{(0)}, \mathbf{x})}{2D_x^2} + \frac{D_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2 \leq \frac{1}{2} + \frac{D_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2,$$
$$\sum_{s=0}^{t}\gamma_s(\mathbf{v}^{(s)} - \mathbf{x})^\top\Delta_s^x \leq \frac{V_x(\mathbf{v}^{(0)}, \mathbf{x})}{2D_x^2} + \frac{D_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2 \leq \frac{1}{2} + \frac{D_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2.$$

Writing $\mathbf{x} - \mathbf{x}^{(s)} = \mathbf{x} - \mathbf{w}^{(s)} + \mathbf{w}^{(s)} - \mathbf{x}^{(s)}$ and $\mathbf{x}^{(s)} - \mathbf{x} = \mathbf{v}^{(s)} - \mathbf{x} + \mathbf{x}^{(s)} - \mathbf{v}^{(s)}$, these two inequalities imply

$$\sum_{s=0}^{t}\gamma_s(\mathbf{x} - \mathbf{x}^{(s)})^\top\Delta_s^x \leq \sum_{s=0}^{t}\gamma_s(\mathbf{w}^{(s)} - \mathbf{x}^{(s)})^\top\Delta_s^x + \frac{1}{2} + \frac{D_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2,$$
$$\sum_{s=0}^{t}\gamma_s(\mathbf{x}^{(s)} - \mathbf{x})^\top\Delta_s^x \leq \sum_{s=0}^{t}\gamma_s(\mathbf{x}^{(s)} - \mathbf{v}^{(s)})^\top\Delta_s^x + \frac{1}{2} + \frac{D_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2.$$

Hence,

$$\left|\sum_{s=0}^{t}\gamma_s(\mathbf{x} - \mathbf{x}^{(s)})^\top\Delta_s^x\right| \leq \max\left\{\left|\sum_{s=0}^{t}\gamma_s(\mathbf{w}^{(s)} - \mathbf{x}^{(s)})^\top\Delta_s^x\right|, \left|\sum_{s=0}^{t}\gamma_s(\mathbf{x}^{(s)} - \mathbf{v}^{(s)})^\top\Delta_s^x\right|\right\} + \frac{1}{2} + \frac{D_x^2}{\alpha_x}\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2. \tag{48}$$

Applying (48) in (47), we get

$$\left|\hat{l}_*^{(t)} - l_*^{(t)}\right| \leq \max\left\{\left|\frac{\sum_{s=0}^{t}\gamma_s(\mathbf{w}^{(s)} - \mathbf{x}^{(s)})^\top\Delta_s^x}{\sum_{s=0}^{t}\gamma_s}\right|, \left|\frac{\sum_{s=0}^{t}\gamma_s(\mathbf{x}^{(s)} - \mathbf{v}^{(s)})^\top\Delta_s^x}{\sum_{s=0}^{t}\gamma_s}\right|\right\} + \frac{0.5 + (D_x^2/\alpha_x)\sum_{s=0}^{t}\gamma_s^2\|\Delta_s^x\|_{*,x}^2}{\sum_{s=0}^{t}\gamma_s} + \left|\frac{\sum_{s=0}^{t}\gamma_s\delta_s}{\sum_{s=0}^{t}\gamma_s}\right|. \tag{49}$$

We next find a probabilistic bound for the right hand side of the above inequality.

Bounds on $\left|\frac{\sum_{s=0}^{t}\gamma_s(\mathbf{w}^{(s)} - \mathbf{x}^{(s)})^\top\Delta_s^x}{\sum_{s=0}^{t}\gamma_s}\right|$ *and* $\left|\frac{\sum_{s=0}^{t}\gamma_s(\mathbf{x}^{(s)} - \mathbf{v}^{(s)})^\top\Delta_s^x}{\sum_{s=0}^{t}\gamma_s}\right|$*:* The inequality (30) indicates that

$$\|\mathbf{w}^{(s)} - \mathbf{x}^{(s)}\|_x \leq \|\mathbf{x}^{(s)} - \mathbf{x}^{(0)}\|_x + \|\mathbf{w}^{(s)} - \mathbf{x}^{(0)}\|_x \leq \sqrt{\frac{2}{\alpha_x}V_x(\mathbf{x}^{(0)}, \mathbf{x}^{(s)})} + \sqrt{\frac{2}{\alpha_x}V_x(\mathbf{x}^{(0)}, \mathbf{w}^{(s)})} \leq \frac{2\sqrt{2}D_x}{\sqrt{\alpha_x}}. \tag{50}$$

Define $\psi_s := \nu_{s,t}(\mathbf{w}^{(s)} - \mathbf{x}^{(s)})^\top \Delta_s^x$ and $\sigma_s := \dfrac{4\sqrt{2} D_x M_x \nu_{s,t}}{\sqrt{\alpha_x}}$. Since $\boldsymbol{\xi}^{(s)}$ is independent of $\mathbf{w}^{(s)}$ and $\mathbf{x}^{(s)}$, we have $\mathbb{E}_s[\psi_s] = 0$. Furthermore,

$$\psi_s^2 \leq \nu_{s,t}^2 \left\| \mathbf{w}^{(s)} - \mathbf{x}^{(s)} \right\|_x^2 \|\Delta_s^x\|_{*,x}^2 \leq 8\nu_{s,t}^2 \|\Delta_s^x\|_{*,x}^2 D_x^2/\alpha_x, \tag{51}$$

where the second inequality follows from (50). Using the definition of σ_s, (51), and (28)-(29), it follows that $\mathbb{E}_s[\exp(\psi_s^2/\sigma_s^2)] \leq \exp(1)$. Hence Case A in Lemma 14 and union bound we get

$$\operatorname{Prob}\left\{ \left| \sum_{s=0}^t \nu_{s,t}(\mathbf{w}^{(s)} - \mathbf{x}^{(s)})^\top \Delta_s^x \right| > \frac{4\sqrt{2}\Omega D_x M_x}{\sqrt{\alpha_x}} \sqrt{\sum_{s=0}^t \nu_{s,t}^2} \right\} \leq 2\exp(-\Omega^2/3). \tag{52}$$

With a similar argument, we can also show

$$\operatorname{Prob}\left\{ \left| \sum_{s=0}^t \nu_{s,t}(\mathbf{x}^{(s)} - \mathbf{v}^{(s)})^\top \Delta_s^x \right| > \frac{4\sqrt{2}\Omega D_x M_x}{\sqrt{\alpha_x}} \sqrt{\sum_{s=0}^t \nu_{s,t}^2} \right\} \leq 2\exp(-\Omega^2/3). \tag{53}$$

Bound on $\dfrac{\sum_{s=0}^t \gamma_s^2 \|\Delta_s^x\|_{*,x}^2}{\sum_{s=0}^t \gamma_s}$*:* Define $\psi_s := \gamma_s \nu_{s,t} \|\Delta_s^x\|_{*,x}^2$ and $\sigma_s := 4M_x^2 \gamma_s \nu_{s,t}$. Using (28), it is easy to verify that $\mathbb{E}_s\left[\exp\left(|\psi_s|/\sigma_s\right)\right] \leq \exp(1)$. Hence, from Case B in Lemma 14 we have

$$\operatorname{Prob}\left\{ \frac{\sum_{s=0}^t \gamma_s^2 \|\Delta_s^x\|_{*,x}^2}{\sum_{s=0}^t \gamma_s} > 4M_x^2 \sum_{s=0}^t \gamma_s \nu_{s,t} + 4\Omega M_x^2 \sqrt{\sum_{s=0}^t \gamma_s^2 \nu_{s,t}^2} \right\} \leq \exp(-\Omega^2/12) + \exp(-3\Omega/4). \tag{54}$$

Bound on $\left| \dfrac{\sum_{s=0}^t \gamma_s \delta_s}{\sum_{s=0}^t \gamma_s} \right|$*:* From definition of δ_s and (8), it follows that

$$\mathbb{E}_s\left[\nu_{s,t}\delta_s\right] = 0 \quad \text{and} \quad \mathbb{E}_s\left[\exp((\nu_{s,t}\delta_s)^2/(\nu_{s,t}Q)^2)\right] \leq \exp(1).$$

Hence by Case A in Lemma 14 and union bound we get

$$\operatorname{Prob}\left\{ \left| \sum_{s=0}^t \nu_{s,t}\delta_s \right| > \Omega Q \sqrt{\sum_{s=0}^t \nu_{s,t}^2} \right\} \leq 2\exp(-\Omega^2/3). \tag{55}$$

The conclusion can be then obtained by upper bounding the right hand side of (49) using the union bound of (52), (53), (54), and (55). ■

Proof of Proposition 7: (i) The definition of $\Omega(\delta)$ in (10) guarantees $\exp(-\Omega(\delta)^2/3) + \exp(-\Omega(\delta)^2/12) + \exp(-3\Omega(\delta)/4) \leq \frac{\delta}{3}$. Recall that $\nu_{s,t} = \dfrac{\gamma_s}{\sum_{s'=0}^t \gamma_{s'}}$. With $\gamma_s = \dfrac{1}{M\sqrt{s+1}}$, it is straightforward to verify the following inequalities:

$$\sum_{s=0}^{t} \nu_{s,t}^2 \quad = \quad \frac{\sum_{s=0}^{t} \frac{1}{s+1}}{\left(\sum_{s=0}^{t} \frac{1}{\sqrt{s+1}}\right)^2} \leq \frac{1+\ln(t+1)}{\left(2\sqrt{t+2}-2\right)^2}, \tag{56}$$

$$\frac{\sum_{s=0}^{t} \gamma_s^2}{\sum_{s=0}^{t} \gamma_s} \quad = \quad \frac{1}{M} \cdot \frac{\sum_{s=0}^{t} \frac{1}{s+1}}{\sum_{s=0}^{t} \frac{1}{\sqrt{s+1}}} \leq \frac{1}{M} \cdot \frac{(1+\ln(t+1))}{2\sqrt{t+2}-2}, \tag{57}$$

$$\frac{1}{\sum_{s=0}^{t} \gamma_s} \quad = \quad \frac{M}{\sum_{s=0}^{t} \frac{1}{\sqrt{s+1}}} \leq \frac{M}{2\sqrt{t+2}-2}, \tag{58}$$

$$\sum_{s=0}^{t} \gamma_s^2 \nu_{s,t}^2 \quad = \quad \left(\frac{1}{M}\right)^2 \cdot \frac{\sum_{s=0}^{t} \frac{1}{(s+1)^2}}{\left(\sum_{s=0}^{t} \frac{1}{\sqrt{s+1}}\right)^2} \leq \frac{2\left(\frac{1}{M}\right)^2}{\left(2\sqrt{t+2}-2\right)^2}. \tag{59}$$

Applying these four inequalities to bound the terms in (36), we get

$$\begin{aligned}
\frac{\delta}{3} &\geq \text{Prob}\left\{ u_*^{(t)} - l_*^{(t)} > 4\sqrt{2}\Omega(\delta)M\sqrt{\sum_{s=0}^{t}\nu_{s,t}^2} + \frac{2+2.5M^2\sum_{s=0}^{t}\gamma_s^2}{\sum_{s=0}^{t}\gamma_s} + 2.5\Omega(\delta)M^2\sqrt{\sum_{s=0}^{t}\gamma_s^2\nu_{s,t}^2} \right\} \\
&\geq \text{Prob}\left\{ u_*^{(t)} - l_*^{(t)} > \left(4\sqrt{2}\Omega(\delta)M + 4.5M + 2.5\sqrt{2}\Omega(\delta)M\right)\frac{(1+\ln(t+1))}{2\sqrt{t+2}-2} \right\} \\
&\geq \text{Prob}\left\{ u_*^{(t)} - l_*^{(t)} > \left(10\Omega(\delta)M + 4.5M\right)\frac{(1+\ln(t+1))}{2\sqrt{t+2}-2} \right\}.
\end{aligned} \tag{60}$$

Given $\epsilon_{\mathcal{A}} > 0$, let $\epsilon' := \epsilon_{\mathcal{A}} / \left(10\Omega(\delta)M + 4.5M\right)$. When $t \geq \max\left\{6, \left(\frac{8\ln(4/\epsilon')}{\epsilon'}\right)^2 - 2\right\}$, we have $\frac{1+\ln(t+1)}{2\sqrt{t+2}-2} \leq \frac{2\ln(t+2)}{\sqrt{t+2}}$ and $\frac{2\ln(t+2)}{\sqrt{t+2}}$ is monotonically decreasing in t. Hence

$$\frac{1+\ln(t+1)}{2\sqrt{t+2}-2} \leq \frac{2\ln(t+2)}{\sqrt{t+2}} \leq \frac{\epsilon'\ln((8/\epsilon')\ln(4/\epsilon'))}{2\ln(4/\epsilon')} \leq \frac{\epsilon'\ln(4/\epsilon') + \epsilon'\ln(2\ln(4/\epsilon'))}{2\ln(4/\epsilon')} \leq \epsilon'. \tag{61}$$

Using the above inequality in (60) we get $\text{Prob}\left\{ u_*^{(t)} - l_*^{(t)} > \left(10\Omega(\delta)M + 4.5M\right)\epsilon' = \epsilon_{\mathcal{A}} \right\} \leq \frac{\delta}{3}$ which completes the proof.

(ii) We only prove this corollary for the lower bounds as the proof of upper bounds is similar. The choice of $\Omega(\delta)$ guarantees $6\exp(-\Omega(\delta)^2/3) + \exp(-\Omega(\delta)^2/12) + \exp(-3\Omega(\delta)/4) \leq \frac{\delta}{3}$. Recall that $\nu_{s,t} = \frac{\gamma_s}{\sum_{s'=0}^{t}\gamma_{s'}}$. Since $\gamma_s = \frac{1}{M\sqrt{s+1}}$, the inequalities (56), (57), (58), and (59) hold. Applying these four inequalities to (45) yields

$$\begin{aligned}
&\text{Prob}\left\{\left|\hat{l}_*^{(t)} - l_*^{(t)}\right| > \left(\Omega(\delta)Q + 8\Omega(\delta)M + 2.5M\right)\frac{(1+\ln(t+1))}{2\sqrt{t+2}-2}\right\} \\
&\leq \text{Prob}\left\{\begin{array}{l}\left|\hat{l}_*^{(t)} - l_*^{(t)}\right| > \\ \left(\Omega(\delta)Q + \frac{4\sqrt{2}\Omega(\delta)D_xM_x}{\sqrt{\alpha_x}} + 0.5M + \frac{4D_x^2M_x^2}{M\alpha_x} + \frac{4\sqrt{2}\Omega(\delta)D_x^2M_x^2}{M\alpha_x}\right)\frac{(1+\ln(t+1))}{2\sqrt{t+2}-2}\end{array}\right\} \\
&\leq \frac{\delta}{3},
\end{aligned} \tag{62}$$

where the first inequality follows from $M^2 \geq \dfrac{2D_x^2M_x^2}{\alpha_x}$ by (9).

Let $\epsilon' := \epsilon_{\mathcal{A}}/\left(\Omega(\delta)Q + 8\Omega(\delta)M + 2.5M\right)$. When $t \geq \max\left\{6, \left(\dfrac{8\ln(4/\epsilon')}{\epsilon'}\right)^2 - 2\right\}$, the inequality (61) holds which can be applied to (62) to show that

$$\text{Prob}\left\{\left|\hat{l}_*^{(t)} - l_*^{(t)}\right| > \left(\Omega(\delta)Q + 8\Omega(\delta)M + 2.5M\right)\epsilon' = \epsilon_{\mathcal{A}}\right\} \leq \frac{\delta}{3}.$$

■

Proof of Theorem 8. We begin by establishing that the following inequalities hold with high probability in at most $T(\delta, \epsilon)$ number of iterations:

$$u_*^{(t)} - l_*^{(t)} \leq \frac{1}{2}\epsilon_{\mathcal{A}}, \quad \left|\hat{l}_*^{(t)} - l_*^{(t)}\right| \leq \frac{1}{2}\epsilon_{\mathcal{A}}, \quad \text{and} \quad \left|\hat{u}_*^{(t)} - u_*^{(t)}\right| \leq \frac{1}{2}\epsilon_{\mathcal{A}}. \tag{63}$$

Given $\Omega(\delta)$, parts (i) and (ii) of Proposition 7 imply that when

$$t \geq \max\left\{6, \left(\frac{16\left(\Omega(\delta)Q + 10\Omega(\delta)M + 4.5M\right)}{\epsilon_{\mathcal{A}}}\ln\left(\frac{8\left(\Omega(\delta)Q + 10\Omega(\delta)M + 4.5M\right)}{\epsilon_{\mathcal{A}}}\right)\right)^2 - 2\right\},$$

we have $\text{Prob}\left\{u_*^{(t)} - l_*^{(t)} > \dfrac{\epsilon_{\mathcal{A}}}{2}\right\} \leq \delta/3$, $\text{Prob}\left\{\left|\hat{l}_*^{(t)} - l_*^{(t)}\right| > \dfrac{\epsilon_{\mathcal{A}}}{2}\right\} \leq \delta/3$, and $\text{Prob}\left\{\left|\hat{u}_*^{(t)} - u_*^{(t)}\right| > \dfrac{\epsilon_{\mathcal{A}}}{2}\right\} \leq \delta/3$. Hence, using union bounds we get

$$\text{Prob}\left\{u_*^{(t)} - l_*^{(t)} \leq \frac{\epsilon_{\mathcal{A}}}{2}, \left|\hat{l}_*^{(t)} - l_*^{(t)}\right| \leq \frac{\epsilon_{\mathcal{A}}}{2}, \left|\hat{u}_*^{(t)} - u_*^{(t)}\right| \leq \frac{\epsilon_{\mathcal{A}}}{2}\right\} \geq 1 - \delta.$$

To complete the proof we show that (63) implies $\mathcal{P}(r, \bar{\mathbf{x}}^{(t)}) - H(r) \leq \epsilon_{\mathcal{A}}$ and $\left|\hat{u}_*^{(t)} - H(r)\right| \leq \epsilon_{\mathcal{A}}$. First note that we have

$$\mathcal{P}(r, \bar{\mathbf{x}}^{(t)}) \leq u_*^{(t)} \leq l_*^{(t)} + \frac{\epsilon_{\mathcal{A}}}{2} \leq H(r) + \frac{\epsilon_{\mathcal{A}}}{2} \leq H(r) + \epsilon_{\mathcal{A}}, \tag{64}$$

where the first inequality follows from (11) as $\max_{\mathbf{y}\in\mathcal{Y}} \phi(\bar{\mathbf{x}}^{(t)}, \mathbf{y}) = \mathcal{P}(r, \bar{\mathbf{x}}^{(t)})$, the second from (63), and the third holds since $l_*^{(t)}$ is a lower bound on $H(r)$. Using (63) and $u_*^{(t)} \leq H(r) + \dfrac{\epsilon_{\mathcal{A}}}{2}$, we get

$$\hat{u}_*^{(t)} \leq u_*^{(t)} + \frac{\epsilon_{\mathcal{A}}}{2} \leq H(r) + \frac{\epsilon_{\mathcal{A}}}{2} + \frac{\epsilon_{\mathcal{A}}}{2} = H(r) + \epsilon_{\mathcal{A}}. \tag{65}$$

In addition,

$$\hat{u}_*^{(t)} \geq u_*^{(t)} - \frac{\epsilon_{\mathcal{A}}}{2} \geq H(r) - \frac{\epsilon_{\mathcal{A}}}{2} \geq H(r) - \epsilon_{\mathcal{A}}, \tag{66}$$

where the first inequality holds by (63) and the second since $u_*^{(t)}$ is an upper bound on $H(r)$. The inequalities (64)-(66) complete the proof. ■

Proof of Corollary 9: By Theorem 8, OVSMD with $T = T(\delta, \epsilon_{\mathcal{A}})$ is a valid stochastic oracle for SFLS. Given the choices of ϵ_{opt} and $\epsilon_{\mathcal{A}}$, Theorem 5 guarantees that SFLS returns a relative ϵ-optimal and feasible solution with probability $1-\delta$ in at most $\frac{2\theta^2}{\beta} \ln\left(\frac{\theta^2}{\beta\epsilon}\right)$ calls to OVSMD, which is the first conclusion of Corollary 9. According to Line 4 in Algorithm 1, OVSMD is called in iteration k of SFLS with input $\delta^{(k)} = \frac{\delta}{2^{k+1}}$, which requires

$$\mathcal{O}(T(\delta^{(k)}, \epsilon_{\mathcal{A}})) = \frac{\theta^6}{(\theta-1)^2\epsilon^2} \cdot \ln^2\left(\frac{1}{\delta^{(k)}}\right) \cdot \ln^2\left(\frac{1}{\epsilon}\right) = \frac{\theta^6}{(\theta-1)^2\epsilon^2} \cdot \left(\ln^2\left(\frac{1}{\delta}\right) + k^2\right) \cdot \ln^2\left(\frac{1}{\epsilon}\right)$$

gradient iterations based on (10), (14), and the fact that $\epsilon_{\mathcal{A}} = \mathcal{O}(\frac{(\theta-1)\epsilon}{\theta^3})$. Note that we only show the dominating terms in ϵ or δ in the complexity above. Summing this complexity for $k = 0, 1, \ldots, \left\lceil \frac{2\theta^2}{\beta} \ln\left(\frac{\theta^2}{\beta\epsilon}\right) \right\rceil$, we obtain the total number of gradient iterations, that is

$$\frac{\theta^8}{(\theta-1)^2\beta\epsilon^2} \cdot \ln\left(\frac{\theta^2}{\beta\epsilon}\right) \cdot \ln^2\left(\frac{1}{\delta}\right) \cdot \ln^2\left(\frac{1}{\epsilon}\right) + \frac{\theta^{12}}{(\theta-1)^2\beta^3\epsilon^2} \cdot \ln^3\left(\frac{\theta^2}{\beta\epsilon}\right) \cdot \ln^2\left(\frac{1}{\epsilon}\right),$$

which is the number claimed in Corollary 9 if only the dominating terms in ϵ or δ are shown and the constant $\theta = \mathcal{O}(1)$ is suppressed. ■

Lemma 18 below shows the number of iterations required by Algorithm 4 to find the upper bound $\bar{U}$ on $H(r^{(0)})$.

Lemma 18 *Given an input tuple $(r^{(0)}, \bar{\alpha}, \delta, \gamma_t, \theta)$, Algorithm 4 terminates with probability of at least $1-\delta$ after at most*

$$\mathcal{O}\left(\log_2\left(\frac{\theta}{(\theta-1)\beta}\right)\right)$$

OVSMD calls and

$$\mathcal{O}\left(\frac{\theta}{(\theta-1)\beta^2} \log_2^4\left(\frac{1}{\beta}\right) \ln^2\left(\frac{1}{\delta}\right)\right)$$

gradient iterations. In addition, $H(r^{(0)}) \leq \bar{U} < 0$ and $|H(r^{(0)})|/|\bar{U}| \leq \theta$ hold at termination.

Proof We first prove that Algorithm 4 terminates with a probability of at least $1-\delta$. Consider the hth iteration of this algorithm. Given $\hat{u}_*^{(h)}$ returned by OVSMD, Theorem 8 guarantees with a probability of at least $1-\delta^{(h)}$ that

$$\hat{u}_*^{(h)} - \alpha^{(h)} \leq H(r^{(0)}) \leq \hat{u}_*^{(h)} + \alpha^{(h)}. \tag{67}$$

Since $\sum_{h=0}^{\infty}\delta^{(h)}=\delta$, using union bound it is clear that (67) holds for $h=0,1,2,\ldots$, with a probability of at least $1-\delta$. In addition, (67) implies that $\hat{u}_*^{(h)}+\alpha^{(h)}\le H(r^{(0)})+2\alpha^{(h)}\le 0$ when $\alpha^{(h)}\le -H(r^{(0)})/2$. Furthermore, when $\alpha^{(h)}\le -\frac{\theta-1}{2\theta}H(r^{(0)})$ (which also indicates that $\alpha^{(h)}\le -\frac{H(r^{(0)})}{2}$ since $\theta>1$ and $H(r^{(0)})\le 0$), we have

$$\frac{\hat{u}_*^{(h)}-\alpha^{(h)}}{\hat{u}_*^{(h)}+\alpha^{(h)}}=\frac{-\hat{u}_*^{(h)}+\alpha^{(h)}}{-\hat{u}_*^{(h)}-\alpha^{(h)}}=\frac{-\hat{u}_*^{(h)}+\alpha^{(h)}}{-\hat{u}_*^{(h)}+\alpha^{(h)}-2\alpha^{(h)}}\le\frac{-H(r^{(0)})}{-H(r^{(0)})-2\alpha^{(h)}}\le\frac{1}{1-\frac{\theta-1}{\theta}}=\theta, \qquad (68)$$

where the first inequality follows from the inequality $-H(r^{(0)})\le-\hat{u}_*^{(h)}+\alpha^{(h)}$ and the fact that the function $x/(x-2\alpha^{(h)})$ is a decreasing function in x. (68) indicates that as soon as $\alpha^{(h)}\le-\frac{\theta-1}{2\theta}H(r^{(0)})$, the stopping criteria of Algorithm 4 hold and the algorithm terminates with a probability of $1-\delta$. Since $\alpha^{(h)}=\alpha^{(0)}/2^h=\bar{\alpha}/2^h$ and $\beta=|H(r^{(0)})|/(r^{(0)}-f^*)$, the inequality $\alpha^{(h)}\le-\frac{\theta-1}{2\theta}H(r^{(0)})$ can be guaranteed in at most $J:=\log_2\left(\frac{2\theta\bar{\alpha}}{(\theta-1)|H(r^{(0)})|}\right)=\mathcal{O}\left(\log_2\left(\frac{\theta}{(\theta-1)\beta}\right)\right)$ iterations. Furthermore, the inequalities (67) and (68) imply that at termination $\bar{U}=\hat{u}_*^{(J)}+\alpha^{(J)}<0$ and $\dfrac{|H(r^{(0)})|}{|\bar{U}|}\le\dfrac{\hat{u}_*^{(J)}-\alpha^{(J)}}{\hat{u}_*^{(J)}+\alpha^{(J)}}\le\theta$.

We next compute the total number of gradient iterations taken by Algorithm 4. We will only keep the key parameters δ, θ, and β in the complexity while suppress others in $\mathcal{O}$. Notice that by Theorem 8, the h-th call of OVSMD requires at most $T(\delta^{(h)},\alpha^{(h)})$ iterations. Therefore, the total number of iterations can be computed as

$$\begin{aligned}
&\sum_{h=0}^{J}T\left(\delta^{(h)},\alpha^{(h)}\right)\\
\le&\sum_{h=0}^{J}\mathcal{O}\left(\frac{16\left(\Omega(\delta^{(h)})Q+10\Omega(\delta^{(h)})M+4.5M\right)}{\alpha^{(h)}}\ln\left(\frac{8\left(\Omega(\delta^{(h)})Q+10\Omega(\delta^{(h)})M+4.5M\right)}{\alpha^{(h)}}\right)\right)^2\\
=&\sum_{h=0}^{J}\mathcal{O}\left(\frac{\Omega(\delta^{(h)})}{\alpha^{(h)}}\ln\left(\frac{\Omega(\delta^{(h)})}{\alpha^{(h)}}\right)\right)^2=\sum_{h=0}^{J}\mathcal{O}\left(\frac{h2^h}{\bar{\alpha}}\ln\left(\frac{1}{\delta}\right)\ln\left(\frac{h2^h}{\bar{\alpha}}\ln\left(\frac{1}{\delta}\right)\right)\right)^2\\
=&\sum_{h=0}^{J}\mathcal{O}\left(h^2 2^h\ln\left(\frac{1}{\delta}\right)\right)^2=\sum_{h=0}^{J}\mathcal{O}\left(J^4 2^{2h}\ln^2\left(\frac{1}{\delta}\right)\right)=\mathcal{O}\left(\frac{J^4\theta\ln^2(1/\delta)}{(\theta-1)\beta^2}\right)\\
=&\mathcal{O}\left(\frac{\theta\ln^2(1/\delta)}{(\theta-1)\beta^2}\log_2^4\left(\frac{\theta}{(\theta-1)\beta}\right)\right)=\mathcal{O}\left(\frac{\theta}{(\theta-1)\beta^2}\log_2^4\left(\frac{1}{\beta}\right)\ln^2\left(\frac{1}{\delta}\right)\right),
\end{aligned}$$

where we used $\Omega(\delta^{(h)})=\mathcal{O}\left(h\log\left(\frac{1}{\delta}\right)\right)$ and $\alpha^{(h)}=\frac{\bar{\alpha}}{2^h}$ in the second inequality, $J=\mathcal{O}\left(\log_2\left(\frac{\theta\bar{\alpha}}{(\theta-1)|H(r^{(0)})|}\right)\right)$ in the third and fourth equations, and $|H(r^{(0)})|=\Theta(\beta)$. ■

Proof of Theorem 10: The proof of this theorem is a direct result of Corollary 9 and Lemma 18. In particular, it is straightforward to see that the total number of OVSMD calls is

$$\mathcal{O}\left(\ln\left(\frac{\theta}{(\theta-1)\beta}\right)\right)+\mathcal{O}\left(\frac{\theta^2}{\beta}\ln\left(\frac{\theta^2}{\beta\epsilon}\right)\right)=\mathcal{O}\left(\frac{\theta^2}{\beta}\ln\left(\frac{\theta^2}{(\theta-1)\beta\epsilon}\right)\right)=\mathcal{O}\left(\frac{1}{\beta}\ln\left(\frac{1}{\beta\epsilon}\right)\right).$$

In addition, combining Corollary 9 and Lemma 18, the total number of gradient iterations can be computed as

$$\mathcal{O}\left(\frac{1}{\beta^2}\ln^4\left(\frac{1}{\beta}\right)\ln^2\left(\frac{1}{\delta}\right)\right)+\mathcal{O}\left(\frac{1}{\beta\epsilon^2}\cdot\ln^3\left(\frac{1}{\delta}\right)\cdot\ln^2\left(\frac{1}{\epsilon}\right)+\frac{1}{\beta^3\epsilon^2}\cdot\ln^5\left(\frac{1}{\epsilon}\right)\right).$$

Here, the universal constant θ is suppressed in $\mathcal{O}$. ■

Proof of Corollary 11: Let $\delta^{(k)}=\dfrac{\delta}{2^k}$ for $k\geq 0$ as defined in SFLS. With a little abuse of notation, we use $\Omega(n)$ to represent a quantity whose order of magnitude is at least n. According to Theorem 8, for any $\delta\in(0,1)$ and $K\geq 0$, there exists $\epsilon_{\mathcal{A}}$ satisfying $\Omega\left(\frac{\ln(1/\delta^K)}{\sqrt{T}}\right)\leq\epsilon_{\mathcal{A}}\leq\mathcal{O}\left(\frac{\ln(1/\delta^K)\ln(T)}{\sqrt{T}}\right)$ such that OVSMD is a valid stochastic oracle $\mathcal{A}\left(r^{(k)},\epsilon_{\mathcal{A}},\delta^{(k)}\right)$ for iteration $k=0,1,\ldots,K$ of SFLS. Let $\epsilon=-\frac{2\theta^2(\theta+1)}{(\theta-1)H(r^{(0)})}\epsilon_{\mathcal{A}}$ such that $\Omega\left(\frac{K\theta^2\ln(1/\delta)}{\sqrt{T}}\right)\leq\epsilon\leq\mathcal{O}\left(\frac{K\theta^2\ln(1/\delta)\ln(T)}{\sqrt{T}}\right)$. Hence, there exists $K=\mathcal{O}\left(\frac{\theta^2}{\beta}\ln\left(\frac{T}{\beta}\right)\right)$ such that $K\geq\frac{2\theta^2}{\beta}\ln\left(\dfrac{\theta^2}{\beta\epsilon}\right)$. With such K and ϵ, according to Theorem 5, SFLS generates a feasible solution at iteration $k=0,1,\ldots,K$ and finds a relative ϵ-optimal and feasible solution with $\epsilon\leq\mathcal{O}\left(\frac{\theta^4\ln(1/\delta)\ln(T)\ln(T/\beta)}{\beta\sqrt{T}}\right)$ with a probability of at least $1-\delta$ in at most K outer iterations (calls of OVSMD), which corresponds to $KT=\mathcal{O}\left(\frac{\theta^2 T}{\beta}\ln\left(\frac{T}{\beta}\right)\right)$ gradient iterations.

■

References

Fouad Ben Abdelaziz. Solution approaches for the multiobjective stochastic programming. *European Journal of Operational Research*, 216(1):1–16, 2012.

Fouad Ben Abdelaziz, Belaid Aouni, and Rimeh El Fayedh. Multi-objective stochastic programming for portfolio selection. *European Journal of Operational Research*, 177(3): 1811–1823, 2007.

Daniel Adelman and Adam Mersereau. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, 56(3):712–727, 2008.

Daniel Adelman and Adam Mersereau. Dynamic capacity allocation to customers who remember past service. *Management Science*, 59(3):592–612, 2013.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Aleksandr Aravkin, James Burke, Dmitriy Drusvyatskiy, Michael Friedlander, and Scott Roy. Level-set methods for convex optimization. *Mathematical Programming*, 174(1-2): 359–390, 2019.

Francis R. Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Cristóbal Barba-Gonzaléz, José García-Nieto, Antonio Nebro, and José Aldana-Montes. Multi-objective big data optimization with jMetal and Spark. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2017.

Dimitri Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massashusetts, 3rd edition, 1999.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Wenqing Chen, Melvyn Sim, Jie Sun, and Chung-Piaw Teo. From CVaR to uncertainty set: Implications in joint chance-constrained optimization. *Operations Research*, 58(2): 470–485, 2010.

Xi Chen, Qihang Lin, and Javier Peña. Optimal regularized dual averaging methods for stochastic optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

Andrew Cotter, Maya Gupta, and Jan Pfeifer. A light touch for heavily constrained SGD. In *Conference on Learning Theory (COLT)*, 2016.

Koby Crammer and Yoram Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2-3):201–233, 2002.

John M Danskin. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer Science & Business Media, 2012.

D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

D. P. de Farias and B. Van Roy. On constraint sampling for the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10(99):2899–2934, 2009.

Csaba Fábián. Handling CVaR objectives and constraints in two-stage stochastic models. *European Journal of Operational Research*, 191(3):888–911, 2008.

Mahdi Milani Fard, Kevin Canini, Andrew Cotter, Jan Pfeifer, and Maya Gupta. Fast and flexible monotonic functions with ensembles of lattices. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: Shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael Friedlander. Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Donald Goldfarb, Garud Iyengar, and Chaoxu Zhou. Linear convergence of stochastic frank wolfe variants. In *Artificial Intelligence and Statistics*, 2017.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Conference on Learning Theory (COLT)*, 2011.

Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical programming*, 171(1-2):167–215, 2018.

Guanghui Lan and Zhiqiang Zhou. Algorithms for stochastic optimization with expectation constraints. *arXiv preprint arXiv:1604.03887*, 2016.

Guanghui Lan, Arkadi Nemirovski, and Alexander Shapiro. Validation analysis of mirror descent stochastic approximation method. *Mathematical programming*, 134(2):425–458, 2012.

Claude Lemaréchal, Arkadii Nemirovskii, and Yurii Nesterov. New variants of bundle methods. *Mathematical programming*, 69(1):111–147, 1995.

Eunji Lim. On convergence rates of convex regression in multiple dimensions. *INFORMS Journal on Computing*, 26(3):616–628, 2014.

Qihang Lin, Xi Chen, and Javier Peña. A smoothing stochastic gradient method for composite optimization. *Optimization Methods and Software*, 29(6):1281–1301, 2014.

Qihang Lin, Runchao Ma, and Tianbao Yang. Level-set methods for finite-sum constrained convex optimization. In *International Conference on Machine Learning (ICML)*, 2018a.

Qihang Lin, Selvaprabu Nadarajah, and Negar Soheili. A level-set method for convex optimization with a feasible solution path. *SIAM Journal on Optimization*, 28(4):3290–3311, 2018b.

Qihang Lin, Selvaprabu Nadarajah, and Negar Soheili. Revisiting approximate linear programming: Constraint-violation learning with applications to inventory control and energy storage. *Management Science*, 66(4):1544–1562, 2020.

Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. Stochastic convex optimization with multiple objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

Timothy Marler and Jasbir Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.

Selvaprabu Nadarajah, François Margot, and Nicola Secomandi. Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science*, 61(12):3054–3076, 2015.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied optimization.* Kluwer Academic Publishers, Norwell, MA, 2004.

Roberto Oliveira and Philip Thompson. Sample average approximation with heavier tails i: non-asymptotic bounds with weak assumptions and stochastic constraints. *arXiv preprint arXiv:1705.00822*, 2017.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, USA, 1994.

Philippe Rigollet and Xin Tong. Neyman-pearson classification, convexity and stochastic constraints. *Journal of Machine Learning Research*, 12(86):2831–2855, 2011.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Paul Schweitzer and Abraham Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

Emilio Seijo and Bodhisattva Sen. Nonparametric least squares estimation of a multivariate convex regression function. *The Annals of Statistics*, 39(3):1633–1657, 2011.

Bodhisattva Sen and Mary Meyer. Testing against a linear regression model using ideas from shape-restricted estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2):423–448, 2017.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 1:3–30, 2017.

Alexander Shapiro. Sample average approximation. In *Encyclopedia of Operations Research and Management Science*, pages 1350–1355. Springer, 2013.

Negar Soheili and Javier Pena. A smooth perceptron algorithm. *SIAM Journal on Optimization*, 22(2):728–737, 2012.

Xin Tong. A plug-in approach to Neyman-Pearson classification. *Journal of Machine Learning Research*, 14(1):3011–3040, 2013.

Xin Tong, Yang Feng, and Anqi Zhao. A survey on neyman-pearson classification and suggestions for future research. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(2):64–81, 2016.

Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(88):2543–2596, 2010.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Anqi Zhao, Yang Feng, Lie Wang, and Xin Tong. Neyman-Pearson classification under high-dimensional settings. *Journal of Machine Learning Research*, 17(1):7469–7507, 2016.