ADAPTIVE SEQUENTIAL SAA FOR SOLVING TWO-STAGE STOCHASTIC LINEAR PROGRAMS

RAGHU PASUPATHY* AND YONGJIA SONG†

Abstract. We present adaptive sequential SAA (sample average approximation) algorithms to solve large-scale two-stage stochastic linear programs. The iterative algorithm framework we propose is organized into *outer* and *inner* iterations as follows: during each outer iteration, a sample-path problem is implicitly generated using a sample of observations or "scenarios," and solved only *imprecisely*, to within a tolerance that is chosen *adaptively*, by balancing the estimated statistical error against solution error. The solutions from prior iterations serve as *warm starts* to aid efficient solution of the (piecewise linear convex) sample-path optimization problems generated on subsequent iterations. The generated scenarios can be independent and identically distributed (iid), or dependent, as in Monte Carlo generation using Latin-hypercube sampling, antithetic variates, or randomized quasi-Monte Carlo. We first characterize the almost-sure convergence (and convergence in mean) of the optimality gap and the distance of the generated stochastic iterates to the true solution set. We then characterize the corresponding iteration complexity and work complexity rates as a function of the sample size schedule, demonstrating that the best achievable work complexity rate is Monte Carlo canonical and analogous to the generic $\mathcal{O}(\epsilon^{-2})$ optimal complexity for non-smooth convex optimization. We report extensive numerical tests that indicate favorable performance, due primarily to the use of a sequential framework with an optimal sample size schedule, and the use of warm starts. The proposed algorithm can be stopped in finite-time to return a solution endowed with a probabilistic guarantee on quality.

Key words. Two-stage Stochastic Programming, Sample Average Approximation, Retrospective Approximation, Sequential Sampling

AMS subject classifications. 90C15, 90C06

1. INTRODUCTION.

The *two-stage stochastic linear program* (2SLP) is that of minimizing the real-valued function $c^\top x + \mathbb{E}[Q(x,\xi)]$ with respect to decision variables $x \in \mathbb{R}^{n_1}_+$ over a set of linear constraints $\mathcal{X} := \{x \in \mathbb{R}^{n_1}_+ : Ax = b\}$, where $Q(x,\xi)$ is itself the optimal value of a random linear program (LP) parameterized by x. Crucially, in 2SLPs, the term $\mathbb{E}[Q(x,\xi)]$ appearing in the objective function is not observable directly; instead, $\mathbb{E}[Q(x,\xi)]$ can only be *estimated* to requested precision as the sample mean $Q_n(x) := n^{-1}\sum_{i=1}^n Q(x,\xi_i)$ of optimal values $Q(x,\xi_i), i = 1,2,\ldots,n$ from randomly sampled LPs. The generation of the random LPs to estimate $\mathbb{E}[Q(x,\xi)]$ is usually accomplished through Monte Carlo sampling, by generating identically distributed "scenarios" $\xi_i, i = 1,2,\ldots,n$ that may or may not be independent.

It appears that 2SLPs were originally introduced by [17] and, owing to their usefulness, have been extensively studied over the last few decades [9]. The sample average approximation (SAA) method seems to have emerged as a popular approach to solving 2SLPs by constructing a solution estimator as follows:

(i) generate an implicit approximation of the objective function using a specified number of "scenarios" $\xi_1, \xi_2, \ldots, \xi_n$ obtained, e.g., using Monte Carlo sampling;

(ii) replace the 2SLP by a sample-path optimization problem [35, 61] having the objective function obtained in (i) and having the known constraint set $\mathcal{X}$, and solve it using one of a variety of decomposition approaches that have been proposed in the literature, e.g., [1, 51, 77].

SAA's popularity stems from its simplicity and its obvious utility within distributed settings, where its structure lends to easy parallelization. Over the last two decades, SAA as described through

*Department of Statistics, Purdue University, West Lafayette, IN, USA, (pasupath@purdue.edu).
†Department of Industrial Engineering, Clemson University, Clemson, SC, USA, (yongjis@clemson.edu).

(i) and (ii) has been extensively analyzed in settings that are much more general than just 2SLPs. For example, results on the consistency and rates of convergence of optimal values/solutions, large and small sample properties, and other special properties are now available through standard textbooks [68] and surveys [34, 35].

It is important to note that SAA is a paradigm and not an algorithm in that important components within the SAA framework still need to be chosen before implementation can occur. To implement the SAA paradigm as stated in (i) and (ii), a practitioner needs to select a sample size and a Monte Carlo generation mechanism in (i), and an appropriate solver/stopping-mechanism in (ii). For instance, the question of sample size choice for generating the sample-path problem in (i) has sometimes been a vexing issue, with practitioners often making this choice through trial and error, using minimum sample size bounds that have been noted to be conservative [35, 41, 64], and more recently, using multiple sample sizes and solving multiple sample-path problems.

A premise of this paper is that SAA's effective implementation depends crucially on the disciplined customization (to narrowly defined problem classes, e.g., 2SLPs) of choices internal to SAA. Such customization involves answering specific algorithmic questions that arise during implementation. For instance:

(a) Is it best to generate and solve (to machine precision) a single sample-path problem with a large Monte Carlo sample size or is it better to progressively and roughly solve a sequence of sample-path problems generated with increasing sample size? If the latter strategy is better, what schedule of sample sizes should be used?

(b) Recognizing that any generated sample-path problem suffers from sampling error and hence suggests not solving to machine precision, to what extent should a sample-path problem be solved?

(c) What type of solvers should be used in solving the generated sample-path problems, given that the solution information to previously solved sample-path problem(s) can be fruitfully used as a *warm start* to a subsequent sample-path problem?

In this paper, we rigorously investigate questions (a)–(c) for the specific case of 2SLPs. And, consistent with our earlier comments, our answers to (a)–(c) seem to be vital to attaining the encouraging numerical experience we describe in Section 7.

1.1. Summary and Insight on Main Results. The essence of our proposed framework is the construction of a sequential SAA framework for solving 2SLPs, where a sequence of approximate 2SLPs are generated and solved to progressively increasing precision across iterations. The framework is such that the early iterates are obtained with little computational burden since, by design, the generated sample-path problems tend to have small sample sizes and are solved imprecisely; and the later iterates can be expected to be obtained with ease as well since they tend to benefit from the warm starts using solution information obtained in previous iterations. The schedule of sample sizes and the adaptive optimality-tolerance parameters are chosen to be in lock-step, ensuring that no particular sample-path problem is "over-solved." The framework we provide is an algorithm in the strict sense of the word in that we make specific recommendations for choosing: (i) the schedule of sample sizes to generate the sample-path problems to approximate the 2SLP, (ii) the schedule of error-tolerance parameters to which each of the generated sample-path problems is to be solved, and (iii) the solver to use when solving the sample-path problems. We also demonstrate that our framework can exploit existing results on finite-time stopping to provide solutions with probabilistic guarantees on optimality. Our extensive numerical experience on solving large-scale 2SLPs suggests that the proposed algorithm yields competitive computational performance compared with existing methods.

We present a number of results that form the theoretical basis for the proposed algorithm. We present sufficient conditions under which the optimality gap and the distance (from the true solution set) of the algorithm's stochastic iterates converges to zero almost surely and in expectation. We also derive the corresponding iteration complexity and work complexity rates, that is, we provide upper bounds (in expectation) on the number of iterations and the number of Monte Carlo oracle calls to ensure that the solution resulting from the framework is ϵ-optimal. The derived work complexity leads to an optimal sample size schedule which is shown to achieve the fastest possible convergence rate in a Monte Carlo setting. Lastly, we demonstrate that using sample size schedules that deviate from the proposed schedule will lead to inferior convergence rates.

We emphasize that the framework we propose is general in that it allows for the use of a wide range of dependent sampling, e.g., Latin-hypercube sampling (LHS) [45], antithetic variates [47], and randomized quasi-Monte Carlo [27, 42] *within* a generated sample-path problem, and the reuse of scenarios *across* generated sample-path problems. While we do not attempt to demonstrate that the use of such variance reduction measures is better than iid sampling, other reports [15, 73] in the literature suggest the fruitfulness of such variance reduction techniques.

1.2. Related Literature. 2SLPs have been the subject of investigation for a long time [8] and algorithms to solve 2SLPs can be conveniently classified based on whether or not the probability space underlying the 2SLP is endowed with a sample space having a finite number of outcomes. As noted in [78], an enormous amount of work has been generated especially for the context where the sample space is finite, resulting in various algorithm classes that directly exploit the finite sum structure — see [8] and [14] for entry points into this substantial literature.

For 2SLPs with sample spaces having countably infinite or an uncountable number of outcomes, or for that matter even sample spaces with large cardinality, Monte Carlo sampling approaches appear to be a viable alternative [68, 69, 70]. In fact, sequential Monte Carlo sampling methods such as what we propose here are not new and have appeared in the stochastic programming (SP) and simulation optimization (SO) literature for several decades now [54, 19, 23, 31, 33, 34, 69, 75]. For instance, [23] proposes the stochastic quasi-gradient methods for optimization of discrete event systems, [69] suggests the idea of solving a sequence of sample-path problems with increasing sample sizes as a practical matter, and [33] gives various sufficient conditions on how fast the sample size should grow in order to ensure the consistency of the SAA estimator with varying sample sizes. For SPs where the corresponding sample-path problems are smooth optimization problems, [58, 62] study the sample size selection problem for the sequential sampling procedure. They model the sequential sampling procedure as a stochastic adaptive control problem, by finding the optimal sample size *as well as* the number of iterations that one should apply to solve the sampled problems, so that the total expected computational effort expended in the entire procedure is minimized. A surrogate model is then proposed to approximate this adaptive control model so that the sample size and the number of iterations to be employed at each iteration can be found (relatively) easily according to results from previous iterations, by solving the surrogate model. From an algorithmic perspective, the stochastic decomposition framework initially developed by [31] is perhaps the most well-known practical approach that exploits the connections between statistical inference, sampling, and stochastic LPs. In addition, [28] proposes simulation-based Benders decomposition approach as a variant of the stochastic sub-gradient method specifically for 2SLPs and develops statistical confidence bounds for the optimal values.

Similar to [33], [54, 56, 53] suggest *retrospective approximation* (RA) where a smooth stochastic optimization problem is solved through a sequence of sample-path problems generated with increasing sample sizes. Unlike in [33], RA methods solve the sample-path problems imprecisely,

until a generally specified error-tolerance parameter is satisfied. The methods presented here can be thought to be *adaptive* RA in that the error-tolerance sequence in our current framework is adaptive since it depends explicitly on a measure of sampling variability. We find that such adaptivity is crucial for good numerical performance, although it brings additional technical difficulty due to the need to handle stopping time random variables. Also, whereas the methods in [54, 58, 62] do not apply to non-smooth problems such as 2SLPs, the methods we present here are tailored (through the choice of solver) to exploit the structure inherent to 2SLPs. We note in passing that adaptive sampling as a strategy to enhance efficiency of stochastic optimization algorithms has recently gained popularity — see, for example, [10, 11, 29, 55, 71].

There has also been some recent work on the question of assessing solution quality in general SPs that directly applies to the context we consider here. For example, [4, 5] propose sequential sampling methods and study conditions under which their employed optimality gap estimator is asymptotically valid in the sense of lying in a returned confidence interval with a specified probability guarantee. Applying these conditions when stipulating the sample size to be employed in each iteration, one naturally gets a highly reliable stopping criterion for the sequential sampling procedure. As we will demonstrate, the results from [4, 5] can be modified for application within a finite-time version of the proposed framework, notwithstanding the fact that the generated sample-path problems in the proposed framework need only be solved imprecisely, to within a specified error-tolerance parameter.

1.3. Organization of the Paper. The rest of the paper is organized as follows: Section 2 presents important notation, convention, and terminology used throughout the paper, a precise problem statement of 2SLP, and a listing of key assumptions. Section 3 introduces the proposed adaptive sequential SAA framework. Section 4 presents various results pertaining to consistency, work complexity rates, and optimal sample size schedules. Section 6 provides a finite stopping rule for the adaptive sequential SAA algorithm by incorporating the sequential sampling approaches proposed in [4] and [5]. Section 7 shows computational performance of the proposed adaptive sequential SAA framework on a variety of test instances.

2. PROBLEM SETUP.

The 2SLP is formally stated as follows:

$$(P) \qquad \begin{aligned} &\min\ c^\top x + q(x) \\ &\text{s.t. } x \in \mathcal{X} := \left\{x \in \mathbb{R}^{n_1}_+ \mid Ax = b\right\}, \end{aligned}$$

where the $r_1 \times n_1$ matrix A, $r_1 \times 1$ vector b and $n_1 \times 1$ vector c are assumed to be fixed and known. The second-stage *value function* $q(x)$ is defined as:

$$q(x) = \mathbb{E}[Q(x,\xi)] = \int_{\Xi} Q(x,\xi)\, dP(\xi), \tag{2.1}$$

where for each $\xi \in \Xi$, the second-stage objective value

$$Q(x,\xi) = \min_{y \in \mathbb{R}^{n_2}_+} \left\{d(\xi)^\top y \mid W(\xi)y \geq h(\xi) - T(\xi)x\right\}. \tag{2.2}$$

We assume that the second-stage objective value is finite, i.e., $Q(x,\xi) > -\infty$, $\forall x \in \mathcal{X}$, and $\xi \in \Xi$. Notice that the function $q(\cdot)$ is not directly "observable" but can be estimated pointwise by "generating scenarios." Specifically, we assume that an iterative algorithm, during the ℓ-th iteration,

generates scenarios $\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell \in \Xi$ that are identically distributed according to some probability measure. The resulting "sample-path problem" due to scenarios $\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell \in \Xi$ is given by

$$(P_\ell) \qquad \begin{aligned} &\min\ c^\top x + Q_{m_\ell}^\ell(x) \\ &\text{s.t. } x \in \mathcal{X} := \left\{x \in \mathbb{R}_+^{n_1} \mid Ax = b\right\}, \end{aligned}$$

where the second-stage *sample-path value function* $Q_{m_\ell}^\ell(x) := m_\ell^{-1} \sum_{i=1}^{m_\ell} Q(x, \xi_i^\ell)$, and $Q(x, \xi_i^\ell)$ is given through (2.2).

To accommodate the probabilistic analysis of the *adaptive iterative* algorithms we propose, we assume the existence of a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_\ell)_{\ell \geq 1}, \mathbb{P})$ such that the iterates $(\hat{x}^\ell)_{\ell \geq 1}$ generated by the algorithm we propose are adapted to $(\mathcal{F}_\ell)_{\ell \geq 1}$. We note then that $Q_{m_\ell}^\ell(\cdot)$ denotes an $\mathcal{F}_\ell$-measurable function estimator of $q(\cdot)$ constructed from $\xi_i^\ell, i = 1, 2, \ldots, m_\ell$ identically distributed, $\mathcal{F}_\ell$-measurable random objects. The random objects $\xi_i^\ell, i = 1, 2, \ldots, m_\ell; \ell = 1, 2, \ldots$ correspond to what have been called "scenarios" in the SP literature. We will use ξ^ℓ to denote a generic $\mathcal{F}_\ell$-measurable outcome, and $\xi_1^\ell, \xi_2^\ell, \ldots$ to denote $\mathcal{F}_\ell$-measurable outcomes obtained from Monte Carlo sampling during iteration ℓ. Thus, the problem in (P_ℓ) is a "sample-path approximation" of the problem in (P) and the function $Q_{m_\ell}^\ell(\cdot)$ is a "sample-path approximation" of the function $q(\cdot)$. The precise sense in which the function $Q_{m_\ell}^\ell(\cdot)$ approximates $q(\cdot)$ will become clear when we state the standing assumptions in Section 2.2.

The notation we use (with the superscript and subscript), while cumbersome, is needed to reflect the fact that the framework we propose allows for a variety of dependence structures of $\xi_i^\ell, i = 1, 2, \ldots, m_\ell$ within and across iterations $\ell = 1, 2, \ldots$. For example, in the simplest and most prevalent case of independent and identically distributed (iid) sampling, generation is done so that the random objects $\xi_i^\ell, i = 1, 2, \ldots, m_\ell$ are mutually independent and identically distributed for each ℓ; the objects $\xi_i^\ell, i = 1, 2, \ldots, m_\ell$ can also be generated so as to satisfy chosen dependency structures that reduce variance, e.g., LHS [45], antithetic variates [48], and randomized quasi-Monte Carlo [37, 27]. Similarly, across iterations $\ell = 1, 2, \ldots$, one can arrange for scenarios from previous iterations to be reused in subsequent iterations as in *common random numbers* [48]. Indeed, we will have to make certain assumptions on $Q_{m_\ell}^\ell(\cdot), \ell = 1, 2, \ldots$ in Section 2.2 that will implicitly impose restrictions on the nature of sampling, to ensure that $Q_{m_\ell}^\ell(\cdot)$ approximates $q(\cdot)$ well enough.

2.1. Further Notation and Convention. We let $\mathcal{S}^*$ denote the optimal solution set, z^* the optimal value, and $\mathcal{S}^*(\epsilon) := \{x \in \mathcal{X} : c^\top x + q(x) - z^* \leq \epsilon\}$ the ϵ-optimal solution set of problem (P). Analogously, $\mathcal{S}_{m_\ell}^*$ denotes the optimal solution set, $z_{m_\ell}^*$ the optimal value, and $\mathcal{S}_{m_\ell}^*(\epsilon) := \{x \in \mathcal{X} : c^\top x + Q_{m_\ell}^\ell(x) - z_{m_\ell}^* \leq \epsilon\}$ the ϵ-optimal solution set for problem (P_ℓ).

The following definitions are used extensively throughout the paper. (i) $\mathbb{R}_+$ denotes the set of non-negative real numbers. (ii) For $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, $\|x\|_2$ refers to the Euclidean norm $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$. (iii) For a real-valued continuous function $g : \mathcal{X} \to \mathbb{R}$ defined on the compact set $\mathcal{X}$, the sup-norm $\|g\|$ is defined as $\|g\| := \max_{x \in \mathcal{X}} |g(x)|$. (iv) The distance between a point $x \in \mathbb{R}^n$ and a set $X \subseteq \mathbb{R}^n$ is defined as $\text{dist}(x, X) := \inf\{\|x - z\|_2 : z \in X\}$, and the distance between two sets $X, Y \subseteq \mathbb{R}^n$ is defined as $\text{dist}(X, Y) := \sup_{x \in X}\{\text{dist}(x, Y)\}$. The definition we have used for $\text{dist}(\cdot, \cdot)$ suffices for our purposes even though it is not a metric since $\text{dist}(X, Y) \neq \text{dist}(Y, X)$ in general. (v) The diameter $\text{diam}(X)$ of a set $X \subseteq \mathbb{R}^n$ is defined as $\text{diam}(X) := \sup_{x,y \in X}\{\|x - y\|_2\}$. (vi) The projection of a point $x \in \mathbb{R}^n$ onto a set $X \subseteq \mathbb{R}^n$ is defined as $\text{proj}\,(x, X) := \arg\inf_{z \in X}\{\|x - z\|_2\}$. (vii) $|X|$ denotes the cardinality of set X. (viii) For a sequence of $\mathbb{R}^d$-valued random variables $\{Z_n\}, Z$, we say $Z_n \to Z$ a.s. to mean that $\{Z_n\}$ converges to Z almost surely, that is, with probability one. We say that Z_n converges to Z in

L^2-norm if $\mathbb{E}[\|Z_n\|_2] \to \mathbb{E}[\|Z\|_2]$ as $n \to \infty$. (See [7] for modes of convergence of sequences of random variables.)

2.2. Assumptions. The following is a list of assumptions that we will use to prove various results in the paper. Assumption 1 and Assumption 2 are standing assumptions in that we will assume these to hold always. Assumption 3 will be invoked as and when needed.

ASSUMPTION 1 (Condition on Relatively Complete Recourse). *The first-stage feasible region $\mathcal{X}$ of problem (P) is compact; furthermore, Problem (P) has relatively complete recourse, that is,*

$$\mathbb{P}\left\{\left(y \in \mathbb{R}^{n_2}_+ : W(\xi)\, y \geq h(\xi) - T(\xi)\, x\right) = \emptyset\right\} = 0, \ \forall x \in \mathcal{X}.$$

ASSUMPTION 2 (Condition on Estimator Quality). *The individual observations comprising the Monte Carlo estimator have finite variance, that is, for all $\ell \geq 1$,*

$$\sup_{x \in \mathcal{X}} \mathbb{V}ar(Q(x, \xi^\ell) \,|\, \mathcal{F}_{\ell-1}) < \infty \text{ a.s.} \tag{2.3}$$

Moreover, the Monte Carlo estimator error decays at the canonical Monte Carlo rate, that is, there exists a constant $\kappa_0 < \infty$ such that for all $\ell \geq 1$,

$$\mathbb{E}\left[\|\bar{\epsilon}_m\|^2 \,|\, \mathcal{F}_{\ell-1}\right] \leq \frac{\kappa_0}{m} \text{ a.s.}, \tag{2.4}$$

where the sample-mean error function $\bar{\epsilon}_m(x) := Q^\ell_m(x) - q(x) = m^{-1}\sum_{j=1}^m (Q(x, \xi^\ell_j) - q(x))$. (The $\|\cdot\|$ appearing in (2.4) is the sup-norm defined in Section 2.1).

ASSUMPTION 3 (Condition on Growth Rate of Objective Function). *The (true) objective function exhibits γ_0-first-order growth on $\mathcal{X}$, that is,*

$$\gamma_0 := \sup_s\{s : c^\top x + q(x) - z^* \geq s\, \mathrm{dist}(x, \mathcal{S}^*) \quad \forall x \in \mathcal{X}\} > 0.$$

Some form of regularity such as (2.3) in Assumption 2 is routinely made in the SP literature [3] and is generally easy to satisfy in 2SLPs when the feasible region $\mathcal{X}$ is compact.

The condition (2.4) in Assumption 2 has been stated for generality, to subsume many contexts that involve dependent and biased sampling, and needs justification. To get a clear sense of the conditions under which (2.4) in Assumption 2 holds, let's first observe that in the *iid unbiased* context, that is, when $\xi^\ell_j, j = 1, 2, \ldots$ are iid and $\mathbb{E}[Q(x, \xi^\ell_j) - q(x) \,|\, \mathcal{F}_{\ell-1}] = 0$ a.s., the vast body of recent literature on concentration inequalities [12, 13, 38, 74] guarantees that (2.4) holds under a variety of moment conditions on $Q(x, \xi^\ell)$. For a general result that can be directly applied in the iid unbiased context, see [20, Proposition 3.1] established for Banach spaces. (Much of the literature on concentration inequalities is focused on sharp quantifications of the tail probabilities associated with $\bar{\epsilon}_m$, and thus characterize the constant κ_0 indirectly; our proposed algorithms do not rely on knowing κ_0.)

In the *dependent but unbiased* sampling context, that is, when $\mathbb{E}[Q(x, \xi^\ell_j) - q(x) \,|\, \mathcal{F}_{\ell-1}] = 0$ a.s. but $\xi^\ell_j, j = 1, 2, \ldots$ are not necessarily independent, Assumption 2 holds in many popular settings where the estimator can be written as an alternate sum of iid unbiased random variables at each $x \in \mathcal{X}$. For instance, consider using *antithetic variates* [48], where for even m we set $\xi^\ell_j := U^\ell_j \in (0, 1), \xi^\ell_{j+1} = 1 - U^\ell_j, j = 1, 3, 5, \ldots, m-1$. Then, $Q^\ell_m(x)$ can be written as the sample mean of $m/2$ (ignoring non-integrality) iid unbiased random variables, each of which is the sum of the

two dependent random variables $Q(x, U_j^\ell)$ and $Q(x, 1 - U_j^\ell)$, implying that Assumption 2 again holds. Similarly, if one chooses stratified sampling [27] as a variance reduction technique, then $Q_m^\ell(x), x \in \mathcal{X}$ can be written as a finite convex combination of sample means, each of which is composed of iid random variables that are unbiased with respect to the conditional means.

Assumption 2 can be shown to hold in other dependent sampling settings such as LHS [45] as well. To see this, we "construct" a d-dimensional random variable $\xi_j^\ell := (\xi_{1j}^\ell, \xi_{2j}^\ell, \ldots, \xi_{dj}^\ell) \in [0,1)^d$ where $\xi_{ij}^\ell = m^{-1}(\pi_{ij} + U_{ij})$, $\pi_i = (\pi_{i1}, \pi_{i2}, \ldots, \pi_{im}), i = 1, 2, \ldots, d$ is each a uniform random permutation of $(0, 1, 2, \ldots, m-1)$, $U_{ij} \sim [0,1)$, and U_{ij}'s and π_i's are independent. Under this setup, we see that $\xi_{ij}^\ell \sim U[0,1)$, $\xi_i^\ell \sim U[0,1)^d$, and that $Q_{m_\ell}^\ell(x)$ is an unbiased estimator of $q(x)$ that is constructed from dependent random variables. Furthermore, under this setup, and as shown in [45, p. 245] and [52, Section 10.3], $\mathrm{Var}(Q(x,\xi)) < \infty$ guarantees that $\mathrm{Var}(Q_{m_\ell}^\ell(x)) = \sigma_0^2/m_\ell + o(m_\ell^{-1}) = O(m_\ell^{-1})$, where $\sigma_0^2 = \mathbb{E}[(Q(x,\xi) - Q^{\mathrm{add}}(x,\xi))^2]$ and $Q^{\mathrm{add}}(x,\xi))$ is the additive approximation of $Q(x,\xi)$ obtained using ANOVA. See also [72] for large sample properties in the LHS context.

Randomized quasi-Monte Carlo (RQMC) is a broad class of variance reduction methods that subsumes various dependent sampling techniques, and where arguments similar to what we have outlined for LHS apply when considering the variance of the estimator $Q_{m_\ell}^\ell$. See [36, Section 2], and the specific RQMC methods listed there, to see how RQMC yields estimators having variance at least as small as what is obtained using naive Monte Carlo, thus guaranteeing $O(m_\ell^{-1})$ variance.

We recognize that we have limited all of the above discussion on dependent sampling by fixing $x \in \mathcal{X}$. A complete treatment of Assumption 2 that involves dependence across $x \in \mathcal{X}$ will require us to consider the behavior of the random function $Q_{m_\ell}^\ell(\cdot)$ by directly making assumptions on the vector $(d(\xi), W(\xi), h(\xi), T(\xi))$ appearing in the second-stage problem (2.2). In general, some sort of a stipulation on the quality of the Monte Carlo estimator is needed to provide reasonable guarantees relating to convergence and convergence rates. For example, in Chapter 5 of [68], we see that even for convergence of sample-path optimal values of SAA to the true optimal value, one needs uniform convergence (across $x \in \mathcal{X}$) of the sample-path functions.

Finally, Assumption 3 is a standard regularity condition [68] having to do with the growth behavior of the true objective function. Specifically, Assumption 3 imposes a minimum growth condition on the true objective function $c^\top x + q(x)$.

3. ADAPTIVE SEQUENTIAL SAA. In this section, we present the proposed adaptive sequential SAA algorithm. The proposed algorithm is based on the following three high-level ideas.

(1) Instead of solving (to any given precision) a single sample-path problem that is generated with a large pre-specified sample size, solve (using a chosen Solver-$\mathcal{A}$) a *sequence* of sample-path problems generated with increasing sample sizes according to a sample size schedule.
(2) Use the solution information obtained from solving each sample-path problem as a *warm start* for solving the subsequent sample-path problem.
(3) To ensure that no particular sample-path problem is over-solved, solve each generated sample-path problem only *imprecisely* to within an optimality tolerance parameter that is adaptively chosen by explicitly considering the inherent sampling error resulting from the choice of sample size.

As can be seen through the listing for Algorithm 3.1, the iterative framework maintains outer iterations that are indexed by ℓ, each of which is composed of inner iterations indexed by t. During the ℓ-th outer iteration, the ℓ-th sample-path problem (P_ℓ) with sample $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell\}$ is generated and solved inexactly up to precision ϵ_ℓ using an iterative optimization algorithm (gener-

Algorithm 3.1 An adaptive sequential SAA framework.

Input: Solver-$\mathcal{A}$, a sampling scheme, constants $\nu, \sigma_{\min}, \sigma_{\max} \in (0, \infty)$, with $\sigma_{\min} < \sigma_{\max}$.
Set $\ell \leftarrow 0$.
for $\ell = 1, 2, \cdots$ **do**
  Select sample size m_ℓ for outer iteration ℓ and draw a sample $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell\}$.
  for $t = 1, 2, \cdots$ **do**
    Using Solver-$\mathcal{A}$ on (P_ℓ), execute t-th inner iteration.
    Obtain candidate solution $\hat{x}^{\ell,t}$, gap estimate $G^{\ell,t}$ and variance parameter estimate $\hat{\sigma}_{\ell,t}$.
    if $G^{\ell,t} \leq \epsilon_{\ell,t} := \nu\, m_\ell^{-1/2} \operatorname{proj}\left(\hat{\sigma}_{\ell,t}, [\sigma_{\min}, \sigma_{\max}]\right)$ **then**
      Break the inner loop with a candidate solution $\hat{x}^\ell := \hat{x}^{\ell,t}$.
    end if
  end for
  Set $\ell \leftarrow \ell + 1$.
end for

ically called Solver-$\mathcal{A}$) for non-smooth convex programs, e.g., the subgradient method [46], level bundle method [39]. We will see later that any solver that satisfies a certain imposition on convergence rate can be used as Solver-$\mathcal{A}$. The iterations of Solver-$\mathcal{A}$ thus constitute the *inner iterations* generating a sequence of inner solutions $\hat{x}^{\ell,t}, t = 1, 2, \ldots$

During each inner iteration t, an upper bound estimate $G^{\ell,t}$ of the optimality gap associated with $\hat{x}^{\ell,t}$ is readily available for any variant of cutting plane algorithms, where a lower approximation $\check{Q}_{m_\ell}^{\ell,t}(\cdot)$ to $Q_{m_\ell}^\ell(\cdot)$ is maintained and iteratively updated. Specifically, the objective value corresponding to $\hat{x}^{\ell,t}$, $\bar{z}_t^\ell := c^\top \hat{x}^{\ell,t} + Q_{m_\ell}^\ell(\hat{x}^{\ell,t})$, gives an upper bound for $z_{m_\ell}^*$. The true optimality gap associated with $\hat{x}^{\ell,t}$, $\bar{z}_t^\ell - z_{m_\ell}^*$, can then be overestimated if a lower bound $\underline{z}_t^\ell$ for $z_{m_\ell}^*$ is provided. Such a lower bound $\underline{z}_t^\ell$ can be obtained, e.g., by solving $\underline{z}_t^\ell = \min_{x \in \mathcal{X}} \left\{c^\top x + \check{Q}_{m_\ell}^{\ell,t}(x)\right\}$. This optimality gap estimate, $G^{\ell,t} := \bar{z}_t^\ell - \underline{z}_t^\ell$, is then compared against an estimate of the sampling error of the true solution of the ℓ-th sample-path problem calculated using $\hat{x}^{\ell,t}$. Precisely, the inner iterations terminate when

$$G^{\ell,t} < \epsilon_{\ell,t} := \nu\, m_\ell^{-1/2} \operatorname{proj}\left(\hat{\sigma}_{\ell,t}, [\sigma_{\min}, \sigma_{\max}]\right), \tag{3.1}$$

where $\sigma_{\min}, \sigma_{\max}, \nu > 0$ are chosen constant parameters, and, as usual, the sample variance

$$\hat{\sigma}_{\ell,t}^2 := \frac{1}{m_\ell} \sum_{i=1}^{m_\ell} \left[Q(\hat{x}^{\ell,t}, \xi_i^\ell) - Q_{m_\ell}^\ell(\hat{x}^{\ell,t})\right]^2. \tag{3.2}$$

We informally call $\epsilon_{\ell,t}$ appearing in (3.1) the *error tolerance*; notice that the condition in (3.1) is meant to keep the estimate of the solution error (as measured by the optimality gap $G^{\ell,t}$) in balance with the sampling error, as measured by the error tolerance $\epsilon_{\ell,t}$. The constants $\sigma_{\min}, \sigma_{\max}$ appearing in (3.1) have been introduced for practical purposes only, to hedge against the rare event that we generate scenarios resulting in an extremely large or extremely small value of the sample variance. Thus:

- if $G^{\ell,t} \geq \epsilon_{\ell,t}$, that is, the upper bound estimate of the optimality gap for solving the current sample-path problem is no less than a factor of the sampling error estimate, continue to the next *inner iteration* $t + 1$;

– otherwise, stop solving the current sample-path problem, that is, terminate the inner iterations, define $\epsilon_\ell := \epsilon_{\ell,t}$, obtain a new scenario set $\mathcal{M}_{\ell+1} := \{\xi_1^{\ell+1}, \xi_2^{\ell+1}, \ldots, \xi_{m_{\ell+1}}^{\ell+1}\}$ with sample size $m_{\ell+1}$ and continue to the next *outer iteration* $\ell + 1$.

When the inner termination condition (3.1) is achieved, we stop the inner iterations, record the solution $\hat{x}^{\ell,t}$ at termination as the current candidate solution $\hat{x}^\ell$, obtain a new scenario set $\mathcal{M}_{\ell+1}$ and start a new outer iteration $\ell+1$ with $\hat{x}^\ell$ as the initial candidate solution. Additional information such as the optimal dual multipliers collected up to outer iteration ℓ can also be used to warm start the outer iteration $\ell + 1$. The process is then repeated until a stopping criterion for the outer iteration of Algorithm 3.1 is satisfied by the candidate solution $\hat{x}^\ell$. We defer our specification of the outer stopping criterion to Section 6.

Algorithm 3.1 is *adaptive* in that ϵ_ℓ is not pre-specified — it is a function of scenarios $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell\}$ used in the ℓ-th outer iteration. Adaptivity is crucial for practical efficiency and when incorporated in our way, avoids several mathematical complexities that otherwise manifest.

We end this section with a result that quantifies the quality of estimators used within Algorithm 3.1. Specifically, Theorem 3.1 quantifies the quality of $Q_{m_\ell}^\ell(\cdot)$ as an estimator of $q(\cdot)$.

THEOREM 3.1 (Monte Carlo Estimator Quality). *Suppose Assumption 1 and Assumption 2 hold, and the sequence of sample sizes $(m_\ell)_{\ell\geq 1}$ is chosen so that the following condition holds:*

$$\text{(SS-A)} \qquad \sum_{\ell=1}^{\infty} \frac{1}{\sqrt{m_\ell}} < \infty, \ m_\ell \geq 1.$$

Then $\sup_{x\in\mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| = 0$ a.s. as $\ell \to \infty$.

Proof. We can write for $t > 0$, a.s.,

$$\begin{aligned}
\mathbb{P}\left\{\sup_{x\in\mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| > t \,|\, \mathcal{F}_{\ell-1}\right\} &\leq t^{-1}\mathbb{E}\left[\sup_{x\in\mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| \,|\, \mathcal{F}_{\ell-1}\right] \\
&= t^{-1}\,\mathbb{E}\left[\left(\sup_{x\in\mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)|^2\right)^{1/2} \,|\, \mathcal{F}_{\ell-1}\right] \\
&\leq t^{-1}\left(\mathbb{E}\left[(\sup_{x\in\mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)|^2 \,|\, \mathcal{F}_{\ell-1}\right]\right)^{1/2} \leq \frac{\sqrt{\kappa_0}\, t^{-1}}{\sqrt{m_\ell}}, \qquad (3.3)
\end{aligned}$$

where the first line in (3.3) is from Markov's inequality [7], the third from (concave) Jensen's inequality [7], and the last from Assumption 2. Conclude from (3.3), the assumed bound (SS-A), and the filtered version of the Borel-Cantelli lemma [76], that the assertion of the theorem holds. □

We note that the condition in (SS-A) is weak — any sequence $(m_\ell)_{\ell\geq 1}$ that satisfies $m_\ell \geq \ell^{2+\epsilon}$ for large enough ℓ and some $\epsilon > 0$ satisfies the condition in (SS-A). The utility of Theorem 3.1 is that it connects uniform almost sure convergence of the Monte Carlo estimator with the moment assumption specified through Assumption 2. See [33] for analogous results for pointwise convergence. We are now ready to undertake the consistency of the iterates $(\hat{x}^\ell)_{\ell\geq 1}$ generated by Algorithm 3.1.

4. CONSISTENCY.

In this section, we treat the consistency of the stochastic iterates generated by the proposed algorithm. By consistency, we mean convergence guarantees (both almost sure and in expectation) associated with the true function values at the stochastic iterates, and the stochastic iterates themselves. This section also sets up the foundation for work complexity results of the subsequent section.

We begin with Lemma 4.1 — a result on the behavior of approximate minimizers of a sequence of convex functions that uniformly converge to a limit function. We emphasize that this result is stated in a *deterministic* setting and will become very useful in explaining the behavior of the sample paths in the stochastic context in the subsequent section. It also appears to be interesting in its own right due to applicability in the context of optimization with a deterministic inexact oracle. See [24, 59, 60] for more on such problems. A complete proof is provided in the appendix of the online supplementary document [57].

LEMMA 4.1. *Let $(f_k)_{k\geq 1}, f_k : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ be a sequence of real-valued convex functions defined on the compact set $\mathcal{X}$. Let $f : \mathcal{X} \to \mathbb{R}$ be a real-valued function such that f_k uniformly converges to f, that is,*

$$\lim_{k\to\infty} \sup_{x\in\mathcal{X}} |f_k(x) - f(x)| = 0.$$

Denote $\delta_{k+1} := \sup_{x\in\mathcal{X}} |f_k(x) - f_{k+1}(x)|$, $\mathcal{S}_f^* := \arg\min_{x\in\mathcal{X}}\{f(x)\}$ and $v^* := \min_{x\in\mathcal{X}}\{f(x)\}$. The point x_k is said to be ϵ_k-optimal to f_k over $\mathcal{X}$ if x_k satisfies $|f_k(x_k) - v_k^*| \leq \epsilon_k$, where $v_k^* := \min_{x\in\mathcal{X}}\{f_k(x)\}$. Suppose the sequences $(\delta_k)_{k\geq 1}$, $(\epsilon_k)_{k\geq 1}$ satisfy

$$\text{(SS-1)} \qquad \sum_{j=1}^{\infty} \delta_j < \infty; \quad \sum_{j=1}^{\infty} \epsilon_j < \infty.$$

Then the following assertions hold.

(a) *$f(x_k) \to v^*$ as $k \to \infty$;*

(b) *for each $k \geq 1$, $f(x_k) - v^* \leq 2\sum_{j=k}^{\infty} \delta_j + 2\sum_{j=k}^{\infty} \epsilon_j$.*

If f obeys a growth rate condition, that is, there exist $\tau > 0, \gamma > 0$ such that for all $x \in \mathcal{X}$,

$$f(x) - v^* \geq \tau \operatorname{dist}(x, \mathcal{S}_f^*)^{\gamma}, \text{ then} \tag{4.1}$$

(c) *for each $k \geq 1$, $\operatorname{dist}(x_k, \mathcal{S}_f^*) \leq \left(2\tau^{-1}\left(\sum_{j=k}^{\infty} \delta_j + \sum_{j=k}^{\infty} \epsilon_j\right)\right)^{\frac{1}{\gamma}}$.*

We emphasize that the postulates of Lemma 4.1 allow f_k, f to be non-smooth convex functions without a unique minimizer. Moreover, Lemma 4.1 guarantees through assertion (a) that the function values at the iterates converge to the optimal value v^* at a rate characterized in assertion (b). A corresponding rate guarantee on the distance between the k-th approximate solution x_k and the true solution set $\mathcal{S}_f^*$ can be given under a growth rate assumption on the objective function f.

Notice that Lemma 4.1 *does not* assert that the sequence of approximate solutions $(x_k)_{k\geq 1}$ converges to a point in the solution set $\mathcal{S}_f^*$, but only that the distance between the sequence $(x_k)_{k\geq 1}$ and the set $\mathcal{S}_f^*$ converges to zero. A guarantee such as convergence to a point is not possible as is, but may be possible by solving regularized versions of f_k, assuming the regularization parameters are chosen appropriately. This question lies outside the scope of the current paper.

We are now ready to characterize consistency in the stochastic context. The first (Theorem 4.2) of these results asserts that the true function values at the iterates generated by the proposed algorithm converge to the optimal value almost surely and in expectation. Furthermore, if the objective function $q(\cdot)$ satisfies a growth condition on $\mathcal{X}$, then similar guarantees can be provided on the distance between the solutions $(\hat{x}^\ell)_{\ell\geq 1}$ and the solution set $\mathcal{S}^*$.

THEOREM 4.2 (Consistency). *Suppose Assumption 1 and 2, and the sample size condition (SS-A) hold, the following assertions about the iterates $(\hat{x}^\ell)_{\ell\geq 1}$ generated by Algorithm 3.1 are true.*

(a) $c^T\hat{x}^\ell + q(\hat{x}^\ell) \to z^*$ a.s. *as* $\ell \to \infty$;
(b) $\mathbb{E}\left[c^T\hat{x}^\ell + q(\hat{x}^\ell)\right] \to z^*$ *as* $\ell \to \infty$.
If Assumption 3 also holds, then the following assertions hold as well.
(c) $\text{dist}(\hat{x}^\ell, \mathcal{S}^*) \to 0$ a.s. *as* $\ell \to \infty$;
(d) $\mathbb{E}[\text{dist}(\hat{x}^\ell, \mathcal{S}^*)] \to 0$ *as* $\ell \to \infty$.

Proof. We will prove assertion (a) by demonstrating that the postulates for Lemma 4.1 (a) are satisfied except on a set (of sample-paths) of measure zero.

We know that $\hat{x}^\ell$ is ϵ_ℓ-optimal to problem (P_ℓ), that is, $|c^T\hat{x}^\ell + Q^\ell_{m_\ell}(\hat{x}^\ell) - z^*_{m_\ell}| \leq \epsilon_\ell$. We also know that $Q^\ell_{m_\ell}(\cdot)$ is convex on $\mathcal{X}$, and from Theorem 3.1, $Q^\ell_{m_\ell}(\cdot)$ is uniformly convergent to $q(\cdot)$. In preparation to invoke Lemma 4.1, denote $\delta_{\ell+1} := \sup_{x\in\mathcal{X}} |Q^\ell_{m_{\ell+1}}(x) - Q^\ell_{m_\ell}(x)|$ and notice that

$$\delta_\ell \leq \sup_{x\in\mathcal{X}} |Q^{\ell+1}_{m_{\ell+1}}(x) - q(x)| + \sup_{x\in\mathcal{X}} |Q^\ell_{m_\ell}(x) - q(x)| := \zeta^{\ell+1}_{m_{\ell+1}} + \zeta^\ell_{m_\ell}.$$

The inequality in (2) and Assumption 2 imply that

$$\mathbb{E}\left[\sum_{\ell=1}^n \delta_\ell\right] = \sum_{\ell=1}^n \mathbb{E}\left[\delta_\ell\right] \leq \sum_{\ell=1}^n \mathbb{E}\left[\left(\zeta^{\ell+1}_{m_{\ell+1}} + \zeta^\ell_{m_\ell}\right)\right] \leq \sqrt{\kappa_0}\left(\sum_{\ell=1}^n \frac{1}{\sqrt{m_{\ell+1}}} + \sum_{\ell=1}^n \frac{1}{\sqrt{m_\ell}}\right), \tag{4.2}$$

where the last inequality in (4.2) follows from Jensen's inequality [22, Theorem 5.1.3] applied to Assumption 2. Thus,

$$\mathbb{E}\left[\sum_{\ell=1}^\infty \delta_\ell\right] = \sum_{\ell=1}^\infty \mathbb{E}\left[\delta_\ell\right] \leq \sqrt{\kappa_0}\left(\sum_{\ell=1}^\infty \frac{1}{\sqrt{m_{\ell+1}}} + \sum_{\ell=1}^\infty \frac{1}{\sqrt{m_\ell}}\right), \tag{4.3}$$

where the equality is due to the monotone convergence theorem [7, Theorem 16.2] and the inequality is due to (4.2). The inequality in (4.3) together with the sample size condition (SS-A) implies that $\mathbb{E}\left[\sum_{\ell=1}^\infty \delta_\ell\right] < \infty$, and hence that $\sum_{\ell=1}^\infty \delta_\ell < \infty$ a.s. Also, recall that the error tolerance sequence $(\epsilon_\ell)_{\ell\geq 1}$ in Algorithm 3.1 has been chosen as $\epsilon_\ell = \nu\frac{1}{\sqrt{m_\ell}}\text{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$. This choice implies that $\sum_{\ell=1}^\infty \epsilon_\ell < \infty$ a.s. The two inequalities above imply that all postulates leading to assertions (a) and (b) in Lemma 4.1 are satisfied on a set (of sample-paths) of measure one; we thus conclude that the assertion (a) of the theorem holds. The assertion in (b) follows from the assertion in (a) since the function $q(\cdot)$ is continuous on the compact set $\mathcal{X}$ and is hence bounded.

If Assumption 3 is satisfied, we know that

$$\text{dist}(\hat{x}^\ell, \mathcal{S}^*) \leq \gamma_0^{-1}\left(c^T\hat{x}^\ell + q(\hat{x}^\ell) - z^*\right). \tag{4.4}$$

Use assertion (a) and (4.4) to conclude that assertion (c) holds. Furthermore, since $\mathcal{X}$ is compact, $\text{dist}(\hat{x}^\ell, \mathcal{S}^*)$ is bounded and hence assertion (d) holds as well. ∎

Theorem 4.2 gives strong guarantees on the consistency of the objective function value at the iterates generated by Algorithm 3.1. However, as is implied by assertion (c) of Theorem 4.2, the solutions $(\hat{x}^\ell)_{\ell\geq 1}$ can be guaranteed to only "converge into" the true solution set $\mathcal{S}^*$ in the sense that the distance between $\hat{x}^\ell$ and the set $\mathcal{S}^*$ converges to zero almost surely and in expectation, and not that the sequence $(\hat{x}^\ell)_{\ell\geq 1}$ is guaranteed to converge to a point. We are now ready to treat convergence rates in the stochastic context.

5. ITERATION AND WORK COMPLEXITY GUARANTEES. Theorem 4.2 guarantees that the sequence of iterates $(\hat{x}^\ell)_{\ell\geq 1}$ generated by Algorithm 3.1 are such that the corresponding objective function values converge to the optimal value almost surely and in expectation, and the iterates converge "into" the true solution set $\mathcal{S}^*$, that is, their distance from $\mathcal{S}^*$ converges to zero almost surely and in expectation. In this section, we will provide a rigorous sense of how fast such convergence happens. Specifically, we provide complexity results that characterize the rate at which the optimality gap and the distance (from $\mathcal{S}^*$) converge to zero as a function of the iteration number and the total workload incurred through a specific iteration.

The first result characterizes the sample-path *iteration complexity* of the proposed algorithm, that is, the rate at which the convergence (as specified through Theorem 4.2) happens as a function of iteration ℓ.

THEOREM 5.1 (Iteration Complexity). *Suppose that Assumption 1 and Assumption 2 hold, and that the sample size sequence obeys the following geometric increase for $\ell \geq 1$:*

$$m_\ell = c_1\, m_{\ell-1}, \quad c_1 \in (1,\infty). \tag{SS-C}$$

Then,

$$\mathbb{E}\left[c^T\hat{x}^\ell + q(\hat{x}^\ell) - z^*\right] \leq 2\,\kappa_2\, c_1^{-\ell/2}, \text{ where } \kappa_2 := \sqrt{\frac{c_1}{m_1}}\,\frac{1}{\sqrt{c_1}-1}\left(\sqrt{\kappa_0}(\sqrt{c_1}+1) + \sigma_{\max}\sqrt{c_1\nu}\right). \tag{5.1}$$

If Assumption 3 holds as well, then

$$\mathbb{E}\left[\operatorname{dist}(\hat{x}^\ell, \mathcal{S}^*)\right] \leq 2\,\tau_0^{-1}\kappa_2\, c_1^{-\ell/2}. \tag{5.2}$$

Proof. Recall $\delta_{\ell+1} := \sup_{x\in\mathcal{X}} |Q^\ell_{m_{\ell+1}}(x) - Q^\ell_{m_\ell}(x)|$ and that the error tolerance sequence $(\epsilon_\ell)_{\ell\geq 1}$ in Algorithm 3.1 has been chosen as $\epsilon_\ell = \nu\frac{1}{\sqrt{m_\ell}}\operatorname{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$. From arguments in the proof of Theorem 4.2, we know that $\sum_{\ell=1}^\infty \delta_\ell < \infty$ a.s., and that $\sum_{\ell=1}^\infty \epsilon_\ell < \infty$ a.s. This means that we can invoke assertion (b) of Lemma 4.1 on a set of measure one, that is, we have for each $\ell \geq 1$,

$$c^\top\hat{x}^\ell + q(\hat{x}^\ell) - z^* \leq 2\sum_{k=\ell}^\infty \delta_\ell + 2\sum_{k=\ell}^\infty \epsilon_\ell \quad \text{a.s.} \tag{5.3}$$

From the monotone convergence theorem [7, Theorem 16.2], Assumption 2, and the sample size choice (SS-C), we see that

$$\mathbb{E}\left[\sum_{k=\ell}^\infty \delta_k\right] = \sum_{k=\ell}^\infty \mathbb{E}\left[\delta_k\right] \leq \sqrt{\kappa_0}\left(\sum_{k=\ell}^\infty \frac{1}{\sqrt{m_{k+1}}} + \sum_{k=\ell}^\infty \frac{1}{\sqrt{m_k}}\right) \leq c_1^{-\ell/2}\sqrt{\frac{\kappa_0 c_1}{m_1}}\,\frac{\sqrt{c_1}+1}{\sqrt{c_1}-1}. \tag{5.4}$$

Also, since $\epsilon_\ell = \nu\frac{1}{\sqrt{m_\ell}}\operatorname{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$, we see that

$$\mathbb{E}\left[\epsilon_\ell\right] \leq \sqrt{\frac{\nu}{m_\ell}\sigma^2_{\max}}, \tag{5.5}$$

and hence

$$\mathbb{E}\left[\sum_{k=\ell}^\infty \epsilon_k\right] := \lim_{n\to\infty} \mathbb{E}\left[\sum_{k=\ell}^n \epsilon_k\right] = \lim_{n\to\infty}\sum_{k=\ell}^n \mathbb{E}\left[\epsilon_k\right] \leq c_1^{-\ell/2}\sqrt{\frac{\nu\,\sigma^2_{\max}c_1}{m_1}}\,\frac{\sqrt{c_1}}{\sqrt{c_1}-1}, \tag{5.6}$$

where the inequality in (5.6) is due to (5.5) and the sample size choice (SS-C). From (5.4), (5.6) and (5.3), we conclude that the first assertion of the theorem (appearing in (5.1)) holds. The second assertion of the theorem (appearing in (5.2)) follows trivially from the growth condition and the first assertion. □

Iteration complexity results such as that in Theorem 5.1 are generally of limited value (especially by themselves) in sampling contexts because they characterize the convergence rate in terms of the *iteration number*, which is not reflective of the total computational work done. A more useful characterization of the convergence rate is what has been called *work complexity*, which is essentially the error (in function value or distance from solution set) expressed as a function of the total computational work done, which for the current context includes the total number of second stage LPs solved. We take up this question next.

Towards characterizing the work complexity of the proposed algorithm, recall the iterative process: during iteration ℓ, a chosen solver that we generically call Solver-$\mathcal{A}$ uses the solution $\hat{x}^{\ell-1}$ from the previous iteration as well as the dual vector information collected so far (for the special case of fixed recourse [1, 31]) as "warm start," and solves the sample-path problem (P_ℓ) generated with sample $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell\}$ to within tolerance ϵ_ℓ, that is, find $\hat{x}^\ell \in \mathcal{S}^*_{m_\ell}(\epsilon_\ell)$. Given this structure, it makes sense then that the rapidity with which a point $\hat{x}^\ell$ is identified will play a central role in determining the overall work complexity of the proposed algorithm. Accordingly, we now make an assumption on the nature of Solver-$\mathcal{A}$ being used to solve the sample-path problem (P_ℓ).

ASSUMPTION 4. *The Solver-$\mathcal{A}$ executed on the problem (P_ℓ) having a piecewise linear convex objective, and with an initial solution $\hat{x}^{\ell-1} \in \mathcal{X}$, exhibits iteration complexity $\Lambda_\ell^2 \operatorname{dist}^2\left(\hat{x}^{\ell-1}, S^*_{m_\ell}\right) \epsilon^{-2}$ to obtain an ϵ-optimal solution, that is,*

$$\left(c^T \hat{x}^{\ell,t} + Q_{m_\ell}(\hat{x}^{\ell,t})\right) - z^*_{m_\ell} \leq \Lambda_\ell \frac{\operatorname{dist}\left(\hat{x}^{\ell-1}, S^*_{m_\ell}\right)}{\sqrt{t}}, \quad t = 1, 2, \ldots, \tag{5.7}$$

where $\hat{x}^{\ell,t}$ is the t-th iterate returned by Solver-$\mathcal{A}$, and $S^*_{m_\ell}$ is the set of optimal solutions corresponding to problem (P_ℓ). Denote the growth-rate γ_ℓ of the sample-path function

$$\gamma_\ell := \sup_s \left\{ s : c^T x + Q^\ell_{m_\ell}(x) - z^*_{m_\ell} \geq s \operatorname{dist}(x, S^*_{m_\ell}) \quad \forall x \in \mathcal{X} \right\}, \tag{5.8}$$

there exists $\lambda < \infty$ such that

$$\mathbb{E}\left[\left(\frac{\Lambda_\ell}{\gamma_\ell} \right)^2 \mid \mathcal{F}_{\ell-1} \right] \leq \lambda^2 < \infty \ \ a.s. \tag{5.9}$$

Assumption 4 has been stated in a way that preserves generality of our theory, with the intent of allowing any choice of Solver-$\mathcal{A}$ as long as the stipulation of Assumption 4 is met. Furthermore, we emphasize that Assumption 4 has been stated for piecewise linear convex objectives, since the objective function of the sample-path problem (P_ℓ) is piecewise linear convex. For instance, a number of well-known subgradient algorithms provide a guaranteed iteration complexity of the sort stipulated in (5.7) of Assumption 4 even for convex non-smooth objectives. For example, the standard subgradient descent algorithm having the iterative structure $x_{t+1} = x_t - \alpha_t \partial h(x_t), t = 0, 1, 2, \ldots$ for solving the convex optimization problem $\min_{x \in \mathcal{X}}\{h(x)\}$, when executed with constant step size $\alpha_t = \epsilon / M^2$ and $\|\partial h(x)\| \leq M$, $\forall x \in \mathcal{X}$, satisfies the complexity requirement stated in Assumption 4. Another recent example is a variant of the level bundle method [6] under an idealized

assumption. In our numerical experiments presented in Section 7, we use an implementable variant of the level bundle method as Solver-$\mathcal{A}$, which is described in greater detail in the appendix of the online supplementary document [57]

The assumption appearing in (5.9) on the finiteness of the second moment of the ratio Λ_ℓ/γ_ℓ is a stipulation on the extent of the "ill-conditioning" of the sample-path problems. To see this, consider using the level method [49, Chapter 3] as Solver-$\mathcal{A}$ in the proposed algorithm. It follows from a well-known result [49, pp. 163] that Λ_ℓ then satisfies

$$\Lambda_\ell \leq \frac{M_\ell}{\sqrt{\alpha(1-\alpha)^2(2-\alpha)}}, \quad \alpha \in (0,1) \tag{5.10}$$

where $\alpha \in (0,1)$ is a user-chosen constant within the level method, and $M_\ell := \sup_{x\in\mathcal{X}}\{\|c + \partial Q_{m_\ell}(x)\|\}$ is the supremum norm (taken over the fixed compact set $\mathcal{X}$) of the sub-gradient associated with the sample-path function. It follows from (5.10) then that

$$\frac{\Lambda_\ell}{\gamma_\ell} \leq \frac{1}{\sqrt{\alpha(1-\alpha)^2(2-\alpha)}} \frac{M_\ell}{\gamma_\ell}, \tag{5.11}$$

where the ratio M_ℓ/γ_ℓ has the interpretation of the "condition number" of the ℓ-th sample-path problem. It is in this sense that the condition appearing in (5.9) can be violated in pathological settings where, persistently, the sample-path function remains "steep" in certain directions but "flat" in others. Also, notice that from the Cauchy-Schwarz inequality, the condition in (5.9) is satisfied, e.g., if the fourth moments of Λ_ℓ and γ_ℓ^{-1} exist, i.e., $\mathbb{E}[\Lambda_\ell^4 \,|\, \mathcal{F}_{\ell-1}] < \infty$ and $\mathbb{E}[\gamma_\ell^{-4} \,|\, \mathcal{F}_{\ell-1}] < \infty$ a.s. The following lemma is an obvious consequence of Assumption 4.

LEMMA 5.2. *Suppose Assumption 1 and 4 hold. Let N_ℓ denote the number of iterations by Solver-$\mathcal{A}$ to solve problem (P_ℓ) to within optimality gap $\epsilon_\ell > 0$ starting at $\hat{x}^{\ell-1}$, i.e., $N_\ell := \inf\left\{\bar{t} : \left(c^\top \hat{x}^{\ell,t} + Q^\ell_{m_\ell}(\hat{x}^{\ell,t})\right) - z^*_{m_\ell} \leq \epsilon_\ell \text{ for all } t \geq \bar{t}, \quad \hat{x}^{\ell,0} := \hat{x}^{\ell-1}\right\}$. Then, $\exists \Lambda_\ell \in \mathcal{F}_\ell$:*

$$\mathbb{P}\left\{N_\ell > \Lambda_\ell^2 \frac{\left(\text{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_\ell})\right)^2}{\epsilon_\ell^2} \,|\, \mathcal{F}_{\ell-1}\right\} = 0, \text{ and } \mathbb{E}\left[\left(\frac{\Lambda_\ell}{\gamma_\ell}\right)^2 \,|\, \mathcal{F}_{\ell-1}\right] < \infty \text{ a.s.}$$

We will now combine the iteration complexities characterized in Theorem 5.1 and Lemma 5.2 to characterize the work complexity of the proposed Algorithm 3.1.

THEOREM 5.3. *Suppose Assumption 1, 2 and 4 hold. Define $W_L := \sum_{\ell=1}^L \tilde{W}_\ell$, where $\tilde{W}_\ell$ is the number of second-stage LPs solved during the ℓ-th outer iteration of Algorithm 3.1. Suppose $(m_\ell)_{\ell\geq 1}$ satisfies the geometric increase sampling condition in (SS-C). Then, for $L \geq 1$,*

$$\mathbb{E}\left[\left(c^T\hat{x}^L + q(\hat{x}^L) - z^*\right)\right] \leq \tau_0/\mathbb{E}[\sqrt{W_L}], \text{ where } \tau_0 \text{ is a constant independent of } L. \tag{5.12}$$

If Assumption 3 also holds, then for $L \geq 1$,

$$\mathbb{E}\left[\text{dist}(\hat{x}^L, \mathcal{S}^*)\right] \leq \tau_0\gamma_0^{-1}/\mathbb{E}[\sqrt{W_L}], \text{ where } \gamma_0 \text{ is the growth-rate constant in Assumption 3.} \tag{5.13}$$

Proof. According to Lemma 5.2, and recalling that up to m_ℓ second-stage LPs are solved in each iteration (e.g., when one employs a scenario decomposition algorithm), for every $\ell \geq 1$:

$$\mathbb{E}\left[\sqrt{\tilde{W}_\ell}\right] = \mathbb{E}\left[\sqrt{N_\ell\, m_\ell}\right] \leq \mathbb{E}\left[\mathbb{E}\left[\Lambda_\ell \frac{\left(\text{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_\ell})\right)}{\epsilon_\ell}\sqrt{m_\ell}\,\middle|\, \mathcal{F}_{\ell-1}\right]\right]. \tag{5.14}$$

Using (5.14) and $\epsilon_\ell := \nu\, m_\ell^{-1/2}\mathrm{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$, we get for large enough ℓ that

$$\begin{aligned}
\mathbb{E}\left[\sqrt{\tilde{W}_\ell}\right] &\leq \frac{m_\ell}{\nu\sigma_{\min}}\mathbb{E}\left[\mathbb{E}\left[\Lambda_\ell \ \mathrm{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_\ell}) \,|\, \mathcal{F}_{\ell-1}\right]\right] \\
&\leq \frac{m_\ell}{\nu\sigma_{\min}}\mathbb{E}\left[\left(\mathbb{E}\left[\left(\frac{\Lambda_\ell}{\gamma_\ell}\right)^2 \,|\, \mathcal{F}_{\ell-1}\right]\right)^{\frac{1}{2}}\left(\mathbb{E}\left[\left(c^T\hat{x}^{\ell-1} + Q^\ell_{m_\ell}(\hat{x}^{\ell-1}) - z^*_{m_\ell}\right)^2 \,|\, \mathcal{F}_{\ell-1}\right]\right)^{\frac{1}{2}}\right] \\
&\leq \frac{m_\ell\lambda}{\nu\sigma_{\min}}\mathbb{E}\left[\left(\mathbb{E}\left[\left(c^T\hat{x}^{\ell-1} + Q^\ell_{m_\ell}(\hat{x}^{\ell-1}) - z^*_{m_\ell}\right)^2 \,|\, \mathcal{F}_{\ell-1}\right]\right)^{\frac{1}{2}}\right] \\
&\leq \frac{m_\ell\lambda}{\nu\sigma_{\min}}\left(\mathbb{E}\left(\mathbb{E}\left[\left(c^T\hat{x}^{\ell-1} + Q^\ell_{m_\ell}(\hat{x}^{\ell-1}) - z^*_{m_\ell}\right)^2 \,|\, \mathcal{F}_{\ell-1}\right]\right)\right)^{\frac{1}{2}} \\
&\leq \frac{m_\ell\lambda}{\nu\sigma_{\min}}\left(\mathbb{E}\left[\left(\delta_\ell + \epsilon_{\ell-1} + |z^*_{m_{\ell-1}} - z^*_{m_\ell}|\right)^2\right]\right)^{\frac{1}{2}}, \qquad (5.15)
\end{aligned}$$

where the second inequality in (5.15) uses the Cauchy-Schwarz inequality (conditionally) and the definition in (5.8) of the sample-path growth rate, the third inequality uses the finite second moment assumption in (5.9) of Assumption 4, the fourth inequality uses the concavity of the square root function, and the last inequality uses $\delta_\ell := \sup_{x\in\mathcal{X}}\{|Q^\ell_{m_\ell}(x) - Q^{\ell-1}_{m_{\ell-1}}(x)|\}$ and the fact that $\hat{x}^{\ell-1} \in \mathcal{F}_{\ell-1}$ is $\epsilon_{\ell-1}$-optimal to $c^Tx + Q^{\ell-1}_{m_{\ell-1}}(x)$ over the set $\mathcal{X}$. Next, let $x^*_\ell \in S^*_{m_\ell}$, $x^*_{\ell-1} \in S^*_{m_{\ell-1}}$ and observe that

$$\begin{aligned}
|z^*_{m_{\ell-1}} - z^*_{m_\ell}| &\leq (c^Tx^*_\ell + Q^{\ell-1}_{m_{\ell-1}}(x^*_\ell) - z^*_{m_\ell}) + (c^Tx^*_{\ell-1} + Q^\ell_{m_\ell}(x^*_{\ell-1}) - z^*_{m_{\ell-1}}) \\
&\leq 2\sup_{x\in\mathcal{X}}\left\{|Q^\ell_{m_\ell}(x) - Q^{\ell-1}_{m_{\ell-1}}(x)|\right\} = 2\delta_\ell. \qquad (5.16)
\end{aligned}$$

Using (5.16) in (5.15), we get for large enough ℓ,

$$\mathbb{E}\left[\sqrt{\tilde{W}_\ell}\right] \leq \lambda\frac{m_\ell}{\nu\sigma_{\min}}\left(\mathbb{E}\left[(3\delta_\ell + \epsilon_{\ell-1})^2\right]\right)^{\frac{1}{2}} \leq \lambda\frac{m_\ell}{\nu\sigma_{\min}}\left(\mathbb{E}\left[18\delta^2_\ell + 2\nu^2\frac{\sigma^2_{\max}}{m_{\ell-1}}\right]\right)^{\frac{1}{2}}, \qquad (5.17)$$

where the second inequality above uses $(a+b)^2 \leq 2a^2 + 2b^2$ and $\epsilon_\ell := \nu\, m_\ell^{-1/2}\mathrm{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$. Observing that $W_L = \sum_{\ell=1}^L \tilde{W}_\ell$, (5.17) implies that

$$\begin{aligned}
\mathbb{E}[\sqrt{W_L}] &\leq \sum_{\ell=1}^L \mathbb{E}\left[\sqrt{\tilde{W}_\ell}\right] \leq \sum_{\ell=1}^L \lambda\frac{m_\ell}{\nu\sigma_{\min}}\left(18\mathbb{E}\left[\delta^2_\ell\right] + 2\nu^2\frac{\sigma^2_{\max}}{m_{\ell-1}}\right)^{\frac{1}{2}} \\
&\leq \sum_{\ell=1}^L \lambda\frac{m_\ell}{\nu\sigma_{\min}}\left(18\frac{\kappa_0(1+c_1+2\sqrt{c_1})}{m_\ell} + 2\nu^2c_1\frac{\sigma^2_{\max}}{m_\ell}\right)^{\frac{1}{2}} \\
&\leq \frac{\lambda}{\nu\sigma_{\min}}\left(18\kappa_0(1+c_1+2\sqrt{c_1}) + 2\nu^2c_1\sigma^2_{\max}\right)^{\frac{1}{2}}\sum_{\ell=1}^L\sqrt{m_\ell} \\
&= \frac{\lambda}{\nu\sigma_{\min}}\left(18\kappa_0(1+c_1+2\sqrt{c_1}) + 2\nu^2c_1\sigma^2_{\max}\right)^{\frac{1}{2}}\frac{\sqrt{m_1}}{\sqrt{c_1}-1}\left(c_1^{L/2} - 1\right), \qquad (5.18)
\end{aligned}$$

where the third inequality follows since $\mathbb{E}[\delta_\ell^2] \leq \kappa_0 \left(m_\ell^{-1/2} + m_{\ell-1}^{-1/2}\right)^2$ holds from Assumption 2, and from further algebra (also see from the proof of Theorem 4.2). Also, we know from (5.3) that for each $L \geq 1$, $c^T \hat{x}^L + q(\hat{x}^L) - z^* \leq 2 \sum_{\ell=L}^{\infty} (\delta_\ell + \epsilon_\ell)$ a.s., and hence, for $L \geq 1$,

$$\mathbb{E}\left(c^T \hat{x}^L + q(\hat{x}^L) - z^*\right)^2 \leq 4\mathbb{E}\left[\left(\lim_{n\to\infty} \sum_{\ell=L}^{n} (\delta_\ell + \epsilon_\ell)\right)^2\right] = 4 \lim_{n\to\infty} \mathbb{E}\left[\left(\sum_{\ell=L}^{n} (\delta_\ell + \epsilon_\ell)\right)^2\right]$$

$$\leq 4 \sum_{\ell=L}^{\infty} \mathbb{E}\left[(\delta_\ell + \epsilon_\ell)^2\right] + 8 \sum_{\ell=L}^{\infty} \left(\mathbb{E}\left[(\delta_\ell + \epsilon_\ell)^2\right]\right)^{1/2} \sum_{j=\ell+1}^{\infty} \left(\mathbb{E}\left[(\delta_j + \epsilon_j)^2\right]\right)^{1/2}, \tag{5.19}$$

where the equality is from the monotone convergence theorem [7, Theorem 16.2], and the last inequality follows from the repeated application of the Hölder's inequality [7, p. 242]. Let's now bound each term appearing on the right-hand side of (5.19). Notice that

$$\sum_{j=\ell+1}^{\infty} \left(\mathbb{E}\left[(\delta_j + \epsilon_j)^2\right]\right)^{1/2} \leq \sum_{j=\ell+1}^{\infty} \left(2\mathbb{E}[\delta_j^2] + 2\mathbb{E}[\epsilon_j^2]\right)^{1/2}$$

$$\leq \sum_{j=\ell+1}^{\infty} \frac{1}{\sqrt{m_j}} \left(2\kappa_0(1 + c_1 + 2\sqrt{c_1}) + 2\nu^2\sigma_{\max}^2\right)^{1/2} \leq \tilde{\kappa}_1 \, c_1^{-\ell/2}, \tag{5.20}$$

where $\tilde{\kappa}_1 := \left(\frac{1}{\sqrt{m_1}} \frac{\sqrt{c_1}}{\sqrt{c_1}-1}\right) \left(2\kappa_0(1 + c_1 + 2\sqrt{c_1}) + 2\nu^2\sigma_{\max}^2\right)^{1/2}$, the second inequality in (5.20) follows from Assumption 2 and the definition $\epsilon_\ell := \nu \, m_\ell^{-1/2} \mathrm{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$, and the last inequality follows from using the assumed sample size increase (SS-C). Similarly, we also get

$$\sum_{\ell=L}^{\infty} \left(\mathbb{E}\left[(\delta_j + \epsilon_j)^2\right]\right) \leq \tilde{\kappa}_2 \, c_1^{-L}, \tag{5.21}$$

where $\tilde{\kappa}_2 := \left(\frac{1}{m_1} \frac{c_1^2}{c_1 - 1}\right) \left(2\kappa_0(1 + c_1 + 2\sqrt{c_1}) + 2\nu^2\sigma_{\max}^2\right)$. Use (5.20) and (5.21) in (5.19) to get:

$$\mathbb{E}\left[\left(c^T \hat{x}^L + q(\hat{x}^L) - z^*\right)^2\right] \leq 4 \sum_{\ell=L}^{\infty} \mathbb{E}\left[(\delta_\ell + \epsilon_\ell)^2\right] + 8 \sum_{\ell=L}^{\infty} \left(\mathbb{E}\left[(\delta_\ell + \epsilon_\ell)^2\right]\right)^{1/2} \sum_{j=\ell+1}^{\infty} \left(\mathbb{E}\left[(\delta_j + \epsilon_j)^2\right]\right)^{1/2}$$

$$\leq 4\tilde{\kappa}_2 c_1^{-L} + 8 \sum_{\ell=L}^{\infty} \left(\mathbb{E}\left[(\delta_\ell + \epsilon_\ell)^2\right]\right)^{1/2} \tilde{\kappa}_1 c_1^{-\ell/2}$$

$$\leq 4\tilde{\kappa}_2 c_1^{-L} + 8(\sqrt{c_1} - 1) \, \tilde{\kappa}_1^2 \sum_{\ell=L}^{\infty} c_1^{-\ell} = c_1^{-L} \left(4\tilde{\kappa}_2 + \frac{8 c_1 \tilde{\kappa}_1^2}{\sqrt{c_1} + 1}\right). \tag{5.22}$$

Finally, we put it all together to get

$$\begin{aligned}
&\mathbb{E}\left[\sqrt{W_L}\right]\mathbb{E}\left[\left(c^T\hat{x}^L + q(\hat{x}^L) - z^*\right)\right] \\
&\quad\leq \frac{\lambda}{\nu\sigma_{\min}}\left(18\kappa_0(1+c_1+2\sqrt{c_1}) + 2\nu^2 c_1\sigma_{\max}^2\right)^{\frac{1}{2}}\frac{\sqrt{m_1}}{\sqrt{c_1}-1}\left(1-\frac{1}{c_1^{L/2}}\right)\left(4\tilde{\kappa}_2 + \frac{8c_1\tilde{\kappa}_1^2}{\sqrt{c_1}+1}\right)^{1/2} \\
&\quad\leq \frac{\lambda}{\nu\sigma_{\min}}\left(18\kappa_0(1+c_1+2\sqrt{c_1}) + 2\nu^2 c_1\sigma_{\max}^2\right)^{\frac{1}{2}}\frac{\sqrt{m_1}}{\sqrt{c_1}-1}\left(4\tilde{\kappa}_2 + \frac{8c_1\tilde{\kappa}_1^2}{\sqrt{c_1}+1}\right)^{1/2} =: \tau_0,
\end{aligned} \tag{5.23}$$

where the first and second inequalities above follow from applying the bounds in (5.22) and (5.18) and simplifying. This proves the first assertion of the theorem. The second assertion follows simply from the first assertion and the assumed minimum growth rate of the objective function as expressed through Assumption 3. ∎

The following observations on Theorem 5.3 are noteworthy.

(a) The assertions in Theorem 5.3 should be seen as the analogue of the $\mathcal{O}(1/\epsilon^2)$ complexity result in non-smooth convex optimization that is known to be optimal [50] to within a constant factor.

(b) The complexity result in Theorem 5.3 has been stated in the general population context. So, the result equally applies for the finite-population scenario $|\Xi| < \infty$, although there is strong evidence that in the finite and the countably infinite populations, the best achievable complexity rates may be much faster due to the existence of sharp minima of the sort discussed in [70].

(c) The theorem assumes that the sample size schedule $(m_\ell)_{\ell\geq 1}$ increases geometrically with common ratio c_1. Importantly, the result can be generalized in a straightforward manner to a sample size schedule having a stochastic common ratio C_1 that is allowed to vary between two deterministic bounds c_0 and c_h such that $1 < c_0 \leq c_h < \infty$ (see Section 7).

Recall again that the complexity result in Theorem 5.3 has been obtained assuming that the sample sizes increase geometrically, that is, $m_\ell/m_{\ell-1} = c_1 \in (1,\infty)$, ignoring non-integrality. Can a similar complexity be achieved using other sample size schedules? The following negative result explains why using a slower sample size schedule is bound to result in an inferior complexity.

THEOREM 5.4. *Suppose Assumption 1–3 hold. Also, suppose there exists $\tilde{\eta}$ such that*

$$\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)\right)\right] \geq \frac{\tilde{\eta}}{\sqrt{m_\ell}}. \tag{5.24}$$

If the sample size schedule is polynomial, that is,

$$m_\ell = c_0\,\ell^p, \quad c_0 \in (0,\infty), p \in [1,\infty). \tag{SS-D}$$

Then there exists $\tau_1 > 0$ such that for $L \geq 3$,

$$\mathbb{E}\left[\operatorname{dist}(\hat{x}^L, \mathcal{S}^*)\right] \geq \frac{\tau_1}{\mathbb{E}\left[W_L^{\frac{1}{2}-\frac{1}{2(1+p)}}\right]}. \tag{5.25}$$

Proof. The structure of the algorithm is such that each outer iteration consists of at least one inner iteration. Hence $\tilde{W}_\ell \geq m_\ell$, implying that

$$W_L \geq \sum_{\ell=1}^{L} c_0 \ell^p \geq \int_1^L c_0 (\ell - 1)^p d\ell = \frac{c_0}{p+1} \left((L-1)^{p+1} - 1 \right). \tag{5.26}$$

Since (SS-D) has been assumed, $m_L = c_0 L^p$ and (5.26) implies, after some algebra, that for $L \geq 3$,

$$\begin{aligned} W_L &\geq \frac{c_0}{p+1} \left(\frac{m_L}{c_0} \right)^{1+1/p} \left(\left(1 - \left(\frac{c_0}{m_L} \right)^{1/p} \right)^{p+1} - \left(\frac{c_0}{m_L} \right)^{1+1/p} \right) \\ &\geq \frac{c_0}{p+1} \left(\frac{m_L}{c_0} \right)^{1+1/p} \left(\left(1 - L^{-1} \right)^{p+1} - L^{-(p+1)} \right) \geq \tau_p \frac{c_0}{p+1} \left(\frac{m_L}{c_0} \right)^{1+1/p}, \end{aligned} \tag{5.27}$$

where $\tau_p := \left(\frac{2}{3}\right)^{p+1} - \left(\frac{1}{3}\right)^{p+1}$. Continuing from (5.27), we get

$$W_L^{\frac{1}{2} - \frac{1}{2(1+p)}} \geq \left(\tau_p \frac{c_0}{p+1} \right)^{\frac{p}{2(p+1)}} \sqrt{\frac{m_L}{c_0}}. \tag{5.28}$$

Use (5.24) and (5.28) to get, for $L \geq 3$, that $\mathbb{E}\left[W_L^{\frac{1}{2} - \frac{1}{2(1+p)}} \right] \mathbb{E}\left[\operatorname{dist}(\hat{x}^L, \mathcal{S}^*) \right] \geq \left(\tau_p \frac{c_0}{p+1} \right)^{\frac{p}{2(p+1)}} \frac{\tilde{\eta}}{\sqrt{c_0}}$, thus proving the assertion in the theorem. ∎

We observe from Theorem 5.4 that no matter how large $p \in [1, \infty)$ is chosen when choosing a polynomial sample size schedule, the resulting complexity (5.25) is inferior to the complexity (5.13) implied by a geometric sample size schedule, with the inferiority characterized by the deviation $(2(p+1))^{-1}$. A similar result has been proved by [63] in a different context.

While the results of Theorem 5.4 show the superiority of a geometric sequence for the sample size schedule, we emphasize two caveats. First, the lower bound on the (implicit) quality of the sample-path solution set may be violated in, e.g., "non-quantitative," contexts where the underlying probability space generating the random variables naturally consists of only a finite number of outcomes. The question of what is the best sample size schedule in such contexts is open. Second, we make the obvious observation that during implementation, considerations other than those included in our analysis, e.g., storage and wall-clock computation time limits, might influence the sample size choice. The conclusions of Theorem 5.3 and Theorem 5.4 should thus be judged within the purview of the analysis considered here.

The condition in (5.24) might appear cryptic but we believe that this condition will hold under mild conditions. General sufficient conditions under which the sequence $\sqrt{m_\ell} \operatorname{dist}(S_{m_\ell}, \mathcal{S}^*)$ will "stabilize" to a non-degenerate distribution are well-known [67, 21]. Such conditions, along with assuming the random variables $\sqrt{m_\ell} \operatorname{dist}(S_{m_\ell}, \mathcal{S}^*)$ exhibit uniform integrability, will ensure that the condition in (5.24) is guaranteed to hold asymptotically.

6. STOPPING IN FINITE TIME. The results we have presented thus far have implied a non-terminating algorithm, as can be seen in the listing of Algorithm 3.1. Our intent in this section is to demonstrate that the iterates generated by Algorithm 3.1 can be stopped in finite-time while providing a solution with a probabilistic guarantee on the optimality gap. For this, we rely heavily on the finite-stopping results in [5]. We first describe a simple stopping procedure which is almost

identical to what is called FSP in [5], and then argue that the stipulations laid out in [5] hold here, thereby allowing to invoke the main results of [5]. We note that alternative finite stopping rules have also been studied in the literature, see, e.g., [66] for a sequential sampling based approach based on the variance associated with 2SLP solutions rather than their corresponding objective values.

Suppose we wish to stop our procedure with a solution whose optimality gap is within $\epsilon > 0$ with probability exceeding $1-\alpha$, $\alpha > 0$. Recall that upon terminating the ℓ-th outer iteration of Algorithm 6.1, we have at our disposal an $\mathcal{F}_\ell$-measurable candidate solution $\hat{x}^\ell$. To construct a one-sided $100(1-\alpha)$ percent confidence interval on the true gap $c^\top \hat{x}^\ell + q(\hat{x}^\ell) - z^*$, we independently generate an iid sample $\mathcal{N}_\ell = \{\tilde{\xi}_1^\ell, \tilde{\xi}_2^\ell, \ldots, \tilde{\xi}_{n_\ell}^\ell\}$. Assume that the sequence $\{n_\ell\}$ of "testing" sample sizes is non-decreasing; the random objects $\tilde{\xi}_i^\ell, i \geq 1, \ell \geq 1$ can be re-used across iterations, that is, $\tilde{\xi}_i^\ell$ can be chosen so that if $i < j$ then $\tilde{\xi}_k^i = \tilde{\xi}_k^j$ for $k = 1, 2, \ldots, n_i$. We then use the set $\mathcal{N}_\ell$ to calculate a gap estimate $\tilde{G}_{n_\ell}^\ell(\hat{x}^\ell)$ and sample variance $\tilde{s}_{n_\ell}^2(\hat{x}^\ell)$ as follows:

$$\tilde{G}_{n_\ell}^\ell(\hat{x}^\ell) = c^\top(\hat{x}^\ell - \tilde{x}_\ell^*) + \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}[Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell)];$$

$$\tilde{s}_{n_\ell}^2(\hat{x}^\ell) = \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\left[Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell) - \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}[Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell)]\right]^2, \tag{6.1}$$

where $\tilde{x}_\ell^*$ is an optimal solution to the sample-path problem (P_ℓ) generated with sample $\mathcal{N}_\ell$, and $\delta > 0$ is the thresholding constant from Algorithm 3.1.

Algorithm 6.1 An adaptive sequential SAA framework with a finite stopping criterion.

1: **Input:** Solver-$\mathcal{A}$, a sampling policy, a constant $\nu > 0$, and a constant $\sigma_{\max} > 0$. Set $\ell \leftarrow 0$.
2: **while** $\tilde{G}_{n_\ell}^\ell(\hat{x}^\ell) + z_\alpha \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \sigma_{\max})}{\sqrt{n_\ell}} > \epsilon$ **do**
3: Select the sample size m_ℓ and draw a random sample $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell\}$.
4: **for** $t = 1, 2, \cdots$ **do**
5: Use Solver-$\mathcal{A}$, e.g., the adaptive partition-based level decomposition [1], to execute the t-th inner iteration for solving the sample-path problem.
6: If $G^{\ell,t} \leq \epsilon_{\ell,t} := \nu \max\left\{\hat{\text{se}}_{\ell,t}, \frac{\sigma_{\max}}{\sqrt{m_\ell}}\right\}$, break the inner loop with a candidate solution $\hat{x}^\ell$.
7: **end for**
8: Generate a Monte Carlo sample $\mathcal{N}_\ell := \{\tilde{\xi}_1^\ell, \tilde{\xi}_2^\ell, \ldots, \tilde{\xi}_{n_\ell}^\ell\}$ (independent from $\mathcal{M}_\ell$) of sample size n_ℓ, solve the corresponding sample-path problem (P_ℓ), and calculate $\tilde{G}_{n_\ell}^\ell(\hat{x}^\ell)$ and $\tilde{s}_{n_\ell}^2(\hat{x}^\ell)$ according to (6.1), respectively.
9: **end while**

The proposed one-sided $100(1-\alpha)$ percent confidence interval on $\mu(\hat{x}^\ell) = c^\top \hat{x}^\ell + q(\hat{x}^\ell) - z^*$ is

$$\left[0, \tilde{G}_{n_\ell}^\ell(\hat{x}^\ell) + z_\alpha \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \sigma_{\max})}{\sqrt{n_\ell}}\right],$$

where $z_\alpha = \Phi^{-1}(1-\alpha)$ is the $1-\alpha$ quantile of the standard normal distribution, implying that the finite-time procedure stops at iteration

$$L(\epsilon) := \operatorname*{arginf}_{\ell \geq 1}\left\{\ell : \tilde{G}_{n_\ell}^\ell(\hat{x}^\ell) + z_\alpha \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \sigma_{\max})}{\sqrt{n_\ell}} \leq \epsilon\right\}.$$

Algorithm 6.1 lists a terminating version of Algorithm 3.1 (modulo setting $\sigma_{\min} = 0$) based on the proposed confidence interval. The factor $\sigma_{\max} n_\ell^{-1/2}$ is a thresholding term that is common in sequential settings [16] and plays the same role as the term $h(n_k)$ in [5], ensuring that $L(\epsilon) \to \infty$ as $\epsilon \to 0$. To analyze the behavior of the coverage probability obtained from Algorithm 6.1, the following three assumptions are made in [5].

(A1) Event $A_{n_\ell} = \{\mathcal{S}_{n_\ell} \subseteq \mathcal{S}^*\}$ happens with probability 1 as $\ell \to \infty$.

(A3) $\lim_{\ell\to\infty} \mathbb{P}\left\{\sup_{x\in\mathcal{X}} |\tilde{G}^\ell_{n_\ell}(\hat{x}^\ell) - \mu(x)| > \beta\right\} = 0$ for any $\beta > 0$.

(A4) $\lim_{\ell\to\infty} \mathbb{P}\left\{\sup_{x\in\mathcal{X}} n_\ell^{-1/2}\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \sigma_{\max}) > \beta\right\} = 0$ for any $\beta > 0$.

(We have omitted (A2) above to preserve the numbering in [5].) Theorem 2.3 in [70] implies that Assumption (A1) is satisfied if the support Ξ is finite, in addition to Assumptions 1–3. Also, it is seen that Assumption (A3) and (A4) hold if the standing Assumption 2 holds. The following result characterizes the behavior of the iterates obtained from Algorithm 6.1, along with a probabilistic guarantee. We provide a proof only for the third part of the theorem since proofs for the rest either follow trivially or are almost identical to that in [5].

THEOREM 6.1. *Suppose Assumptions 1–3 hold. Furthermore, let $|\Xi| < \infty$. Let m_ℓ and n_ℓ be positive nondecreasing sequences such that $m_\ell \to \infty$ and $n_\ell \to \infty$ as $\ell \to \infty$. Then the following assertions hold.*

1. *$L(\epsilon) < \infty$ a.s. for all $\epsilon > 0$ and $L(\epsilon) \to \infty$ a.s. as $\epsilon \to 0$.*
2. *Recalling the optimality gap $\mu(x) := c^\top x + q(x) - z^*$,*

$$\lim_{\epsilon\to 0} \mathbb{P}\left\{\mu(\hat{x}^{L(\epsilon)}) \le \epsilon\right\} = 1. \tag{6.2}$$

3. *Suppose $\{n_\ell\}$ is chosen so that $\liminf_{\ell\to\infty} n_{\ell-1}/n_\ell > 0$. Then we have that*

$$\lim_{\epsilon\to 0^+} \epsilon^2 n_{L(\epsilon)} = O(1).$$

Proof. (Proof of 3.) Following the proof of Lemma 5 in [5], we see that there exists $\epsilon_0 > 0$ such that for all $0 < \epsilon < \epsilon_0$,

$$\tilde{G}^{L(\epsilon)}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}) = 0; \quad \tilde{s}^2_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}) = 0, \tag{6.3}$$

where $\tilde{G}^{L(\epsilon)}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)})$ and $\tilde{s}^2_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)})$ are from (6.1) at stopping. According to the stopping criterion of Algorithm 6.1, we have that:

$$\begin{aligned} \epsilon^2 n_{L(\epsilon)} &\ge \left(\sqrt{n_{L(\epsilon)}}\tilde{G}^{L(\epsilon)}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}) + z_\alpha \max(\tilde{s}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}), \delta)\right)^2; \\ \epsilon^2 n_{L(\epsilon)-1} &\le \left(\sqrt{n_{L(\epsilon)-1}}\tilde{G}^{L(\epsilon)-1}_{n_{L(\epsilon)-1}}(\hat{x}^{L(\epsilon)-1}) + z_\alpha \max(\tilde{s}_{n_{L(\epsilon)-1}}(\hat{x}^{L(\epsilon)-1}), \delta)\right)^2. \end{aligned} \tag{6.4}$$

Now notice that since $\liminf_{\ell\to\infty} n_{\ell-1}/n_\ell > 0$ and $L(\epsilon) \to \infty$ as $\epsilon \to 0$ a.s., there exists $\tilde{\beta} > 0$ such that for small enough ϵ, we have

$$n_{L(\epsilon)-1} \ge \tilde{\beta}\, n_{L(\epsilon)} \text{ a.s.} \tag{6.5}$$

Using (6.5), (6.4), and (6.3), we get, a.s., $z_\alpha \delta^2 \le \lim_{\epsilon\to 0^+} \frac{n_{L(\epsilon)}}{1/\epsilon^2} \le \frac{z_\alpha}{\tilde{\beta}}\, \delta^2$. ∎

It is worth noting that the main probabilistic guarantee appearing in (6.2) is stronger than classical guarantees in sequential testing such as those in [16]. This deviation from a classical stopping result is primarily because of the fast convergence assured by (A1). It is possible and likely that when (A1) is relaxed, a more classical result such as what one encounters in [16] holds, but we are not aware of the existence of such a result.

The condition $\liminf_{\ell\to\infty} n_{\ell-1}/n_\ell > 0$ stipulated by the third assertion of Theorem 6.1 is satisfied by a wide variety of sequences. For instance, if $q_0, q_1 \in (0,\infty)$, any logarithmic increase schedule $n_\ell = q_0 + q_1 \log \ell$, any polynomial increase schedule $n_\ell = q_0 + q_1 \ell^p, p \in (0,\infty)$, and any geometric increase schedule $n_\ell/n_{\ell-1} = q_1$ satisfy the condition $\liminf_{\ell\to\infty} n_{\ell-1}/n_\ell > 0$.

7. COMPUTATIONAL EXPERIMENTS. In this section, we present computational results of the proposed adaptive sequential sampling framework for solving 2SLPs with fixed recourse and fixed second-stage objective coefficients. We chose problems instances of this type to enable a "warm starting" procedure, where the initial solution and an initial second-stage value function approximation for every sample-path problem at each outer iteration can be obtained using information gained from previous iterations. (This procedure is summarized in Algorithm C.1 in the appendix of the online supplementary document [57].) For the purpose of benchmarking, we consider finite-sample instances of such problems, that is, problems where $|\Xi| < \infty$, so that we get access to the true optimal value z^* up to a pre-specified precision by solving these instances using a deterministic solver. In particular, we apply the adaptive partition-based level decomposition method [1], which has shown to be a competitive state-of-the-art solution approach. Five finite-sample instances of each problem in a selected problem class are generated; 20 replications of each competing sequential SAA algorithm are performed on each of the generated problem instances (except for the ssn instances, where only 10 replications are performed due to the extensive computational effort for solving these instances). We implemented all algorithms in C++ using the commercial solver CPLEX, version 12.8. All tests are conducted on an iMac desktop with four 4.00GHz processors and 16Gb memory. The number of threads is set to be one.

We run the adaptive sequential SAA framework according to Algorithm 6.1, and record the total number of outer iterations as L, the final candidate solution at the L-th iteration as $\hat{x}^L$, and the sample size used in the final iteration L as N_L; $c^\top \hat{x}^L + q(\hat{x}^L)$ then gives the true objective value of final candidate solution $\hat{x}^L$. We report in column "CI" the ratio between the width of the reported confidence interval (at stopping) for the optimality gap and the true objective value corresponding to $\hat{x}^L$. The threshold ϵ is chosen to be small enough relative to the objective value corresponding to the candidate solution obtained from the outer iteration, e.g., $10^{-3} \times \left(c^\top \hat{x}^1 + Q^1_{m_1}(\hat{x}^1)\right)$. After Algorithm 6.1 terminates with a final solution $\hat{x}^L$, we verify whether or not the true optimal objective value z^* is in the reported confidence interval. Since the confidence interval at stopping is guaranteed to cover z^* only asymptotically (see Theorem 6.1), we report the coverage probability at stopping in the column titled "cov.", using results obtained from the 20 replications for each test instance except ssn and 20term, where 10 replications are used.

We set the sample size m_ℓ for the ℓ-th sample-path problem to be twice as large as the sample size n_ℓ for validating the quality of candidate solution $\hat{x}^\ell$, i.e., $m_\ell = 2 \times n_\ell$, $\forall \ell = 1, 2, \ldots$. This choice is motivated by the practical guideline [4] that the computational effort expended to find candidate solutions should be higher than that expended to compare candidate solutions. The following additional notation is used in the tables that follow.

- Time: computational time (recorded in seconds)
- M: total number of inner iterations.
- L: total number of outer iterations.

- n_L: the sample size used in the final outer iteration L.

7.1. Implementation details. The following five algorithms are implemented in our computational study. The procedures described in (iii), (iv), and (v) use Algorithm 6.1 with different sample size schedules. The procedure listed in (i) has been shown to be very competitive recently; the procedure in (ii) is proposed in [5].

(i) **PILD-ODA**. This algorithm is the adaptive partition-based level decomposition algorithm with on-demand accuracy as proposed in [1], which is used to solve each instance with the full set of scenarios up to a relative optimality gap of 10^{-4}. Note that z^* for each instance is also obtained by this algorithm using a smaller relative optimality gap threshold of 10^{-6}.

(ii) **Sequential-BP-L(Δ)**. This algorithm follows the sampling schedules in [5] while solving individual sample-path problems to high precision. Specifically, each sample-path problem (with a sample size of m_ℓ) is solved up to a relative optimality gap of 10^{-6} in each outer iteration ℓ, using a standard level decomposition approach for solving 2SLPs [25]. Note that our implementation of this approach does not incorporate the warm starting functionality. The obtained candidate solution $\hat{x}^\ell$ is then evaluated using a sample of size n_ℓ. To obtain $x^*_{n_\ell}$ that appears in $\tilde{G}^\ell_{n_\ell}$ and $\tilde{s}^2_{n_\ell}$ in (6.1), we solve the corresponding sample-path problem up to a relative optimality gap of 10^{-4}, as suggested by [5]. By default, we use a linear sample size schedule where $\Delta = 100$ additional scenarios are sampled from one iteration to the next, starting with an initial sample size $m_1 = 2 \times n_1 = 100$. We use the same initial sample size for all variants of the sequential sampling approaches that we describe below, although one may tune this parameter for further enhancements.

(iii) **Adaptive-seq-BP-L(Δ)**. This is Algorithm 6.1 implemented with the linearly increasing sample size schedule proposed in [5], that is, $m_{\ell+1} = m_\ell + \Delta$. For "warm starting" the initial solution and an initial second-stage value function approximation for every sample-path problem at each outer iteration, we use Algorithm C.1 in the appendix of the online supplementary document [57]. We use parameter $\alpha = 0.1$ and safeguard parameter $\delta = 10^{-5}$ in defining the adaptive optimality tolerance ϵ_ℓ according to (3.1). PILD-ODA is applied to solve each sample-path problem with the aforementioned warm starting functionality.

(iv) **Adaptive-seq-fixed (c_1)**. This is Algorithm 6.1 implemented with a geometric sample size schedule. The setting is nearly identical to (iii) except that we use a fixed rate c_1 as the geometric increase rate, that is, $m_{\ell+1} = c_1\, m_\ell$.

(v) **Adaptive-seq-dyn(c_0, c_h)**. Like in (iv), this is Algorithm 6.1 implemented with a geometric sample size schedule ensuring that $m_{\ell+1} = C_1\, m_\ell$. However, unlike in (iv), the rate C_1 is dynamic (and hence, listed in uppercase) within chosen bounds c_0, c_h. Specifically, starting from some initial value of C_1, if the inner loop finishes after a single iteration, implying that the problem with the current sample size does not deviate much from the one solved in the previous outer iteration, we increase the deviation of C_1 from 1 by a factor of 2 subject to C_1 not exceeding c_h. Formally, we set $C_1 \leftarrow \min(2C_1 - 1, c_h)$. If, on the other hand, the inner loop takes more than four iterations, we shrink the deviation of C_1 from 1 by a factor of 2, subject to C_1 reaching a minimum of c_0, that is, we set $C_1 \leftarrow \max(c_0, \frac{1}{2}C_1 + \frac{1}{2})$. While our theory does not explicitly cover this "dynamic C_1" context, an extension of our theory to this case is straightforward. See comment (c) appearing after Theorem 5.3.

In all algorithms that we tested except "PILD-ODA," we use a time limit of two hours (7200 seconds). When the stopping criterion is not met by the time limit, we report the smallest value $\tilde{G}^\ell_{n_\ell}(\hat{x}^\ell) + z_\alpha \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \sigma_{\max})}{\sqrt{n_\ell}}$ encountered during all completed outer iterations ℓ, and accordingly

TABLE 1

Profiles of test instances from the literature. Notation (n_a, n_b) means that the number of variables is given by n_a and the number of constraints is given by n_b.

Instance	First-stage size	Second-stage size	Reference
DEAK40×20	(40,20)	(30,20)	[18]
DEAK40×40	(40,20)	(60,40)	-
DEAK40×60	(40,20)	(90,60)	-
DEAK60×20	(60,30)	(30,20)	-
DEAK60×40	(60,30)	(60,40)	-
DEAK60×60	(60,30)	(90,60)	-
LandS	(4,2)	(12,7)	[40]
gbd	(17,4)	(10,5)	[26]
4node	(52,14)	(186,74)	[2]
pgp2	(4,2)	(16,7)	[32]
retail	(7,0)	(70,22)	[30]
cep	(8,5)	(15,7)	[32]
baa99-20	(20,0)	(250,40)	[66]
20-term	(63,3)	(764,124)	[43]
ssn	(89,1)	(706,175)	[65]

consider this quantity the width of the confidence interval on the optimality gap of $\hat{x}^\ell$. The profiles of test instances used in our computational experiments are summarized in Table 1, where the set of DEAK instances are randomly generated test instances from [18], and other instances are taken from existing literature that are linked to certain "real-world" applications. For the purpose of benchmarking, we also create an additional family of instances based on the DEAK instances by increasing the variance of the underlying random variables generating the test instances. We use "High" to label this new set of DEAK instances with higher variance in Table 2, 3, and 4.

7.2. Numerical results. We first investigate the empirical performance of "Sequential-BP-L(Δ)", and its adaptation "Adaptive-seq-BP-L(Δ)" into our proposed framework, against "PILD-ODA" which is arguably a state-of-the-art approach for solving 2SLPs with fixed recourse and fixed second-stage objective coefficients using the full set of scenarios [1]. Table 2 summarizes the results on our test instances. We recall that for all the sequential SAA approaches, the numbers shown in each row are calculated by taking the average of the corresponding values over 20 replications (10 replications for ssn and 20term) of algorithm instantiation on five finite-sample instances.

7.2.1. Computational results on the DEAK instance family. We first present the performance of aforementioned algorithms on the DEAK instance family. Instances within this family share the same structure and vary by the problem sizes in terms of the number of variables and constraints. Experiments on these different instances allow us to see how the algorithms behave as the problem sizes change given the same underlying problem structure.

From Table 2, we see that sequential SAA algorithms "Sequential-BP-L(100)" and "Adaptive-seq-BP-L(100)" are clearly favored over the direct approach "PILD-ODA." The sequential SAA approaches finish in much less computational time at a low price in terms of optimality gap — around 0.1%. The coverage probabilities of these approaches are also satisfactory. The majority of the computational savings come from the fact that sequential SAA approaches expend much less effort in each inner iteration, since only a (small) sample is taken at each early outer iteration ℓ.

In comparing "Sequential-BP-L(Δ)" against "Adaptive-seq-BP-L(Δ)," notice from Table 2 that the computational time for "Adaptive-seq-BP-L(Δ)" is lower in most cases, while the total number

TABLE 2

Computational results of the adaptive partition-based level decomposition approach [1] ("PILD-ODA"), the sequential sampling procedure by [5] ("Sequential-BP-L"), and Algorithm 6.1 with the stopping criterion and sample size schedule proposed in [5] ("Adaptive-seq-BP-L (100)") on our test instances DEAK and DEAK-H.

Ins	N	PILD-ODA		Sequential-BP-L(100)			Adaptive-seq-BP-L(100)		
		Time	M	Time	$M(L, n_L)$	CI (cov.)	Time	$M(L, n_L)$	CI (cov.)
	50K	53.4	19	5.4	14(5,1070)	(0.1,97)	1.5	20(5,1094)	(0.1,97)
40x20	100K	101.8	18	5.1	13(5,1032)	(0.1,99)	1.3	19(5,1014)	(0.1,97)
	50K	74.6	12	4.3	19(3,584)	(0.0,83)	1.2	12(3,630)	(0.1,80)
40x40	100K	134.1	12	5.6	20(3,660)	(0.1,90)	1.3	13(3,676)	(0.1,82)
	50K	206.2	19	4.3	20(2,374)	(0.1,96)	1.7	21(2,396)	(0.1,100)
40x60	100K	413.1	20	4.1	20(2,360)	(0.1,99)	1.6	21(2,366)	(0.1,100)
	50K	114.4	56	86.1	41(13,2540)	(0.1,100)	18.5	64(13,2596)	(0.1,100)
60x20	100K	252.2	60	87.8	42(13,2584)	(0.1,100)	19.1	64(13,2636)	(0.1,100)
	50K	502.0	65	23.2	32(4,824)	(0.1,100)	12.3	70(4,834)	(0.1,100)
60x40	100K	929.4	67	25.1	33(4,864)	(0.1,100)	13.5	70(4,876)	(0.1,100)
	50K	333.8	24	5.9	22(2,414)	(0.1,100)	2.2	25(2,424)	(0.1,100)
60x60	100K	622.3	24	6.5	22(2,436)	(0.1,100)	2.3	25(2,436)	(0.1,100)
40x20	50K	63.9	17	18.6	27(9,1776)	(0.1,96)	4.4	23(8,1698)	(0.1,98)
High	100K	139.2	18	18.2	27(9,1772)	(0.1,96)	5.3	24(9,1854)	(0.1,95)
40x40	50K	58.9	9	4.5	17(3,580)	(0.0,83)	1.3	10(3,640)	(0.0,70)
High	100K	117.0	9	4.0	17(3,556)	(0.1,88)	1.3	10(3,646)	(0.1,80)
40x60	50K	711.5	25	60.8	42(6,1140)	(0.1,99)	22.4	29(6,1132)	(0.1,93)
High	100K	1520.0	24	55.5	41(6,1102)	(0.1,100)	20.8	29(6,1130)	(0.1,93)
60x20	50K	162.6	46	139.4	53(16,3194)	(0.1,99)	78.9	52(16,3280)	(0.1,100)
High	100K	263.2	43	132.9	54(16,3160)	(0.1,100)	73.5	52(16,3230)	(0.1,100)
60x40	50K	432.8	31	112.6	55(9,1824)	(0.1,99)	127.4	42(10,1920)	(0.1,98)
High	100K	958.5	32	124.0	56(9,1834)	(0.1,99)	122.4	42(10,1940)	(0.1,98)
60x60	50K	673.5	23	96.6	48(6,1290)	(0.1,100)	38.2	31(6,1282)	(0.1,90)
High	100K	1591.9	25	107.2	49(7,1316)	(0.1,96)	42.3	31(7,1362)	(0.1,89)

of outer iterations L, inner iterations M, and the final sample size n_L are similar. This is again explainable since in "Sequential-BP-L," the sample-path problems in each outer iteration are solved to a high precision, whereas in "Adaptive-seq-BP-L(Δ)," the sample-path problems are only solved up to a factor of the sampling error as detailed in Algorithm 6.1. Furthermore, a warm start functionality and an adaptive scenario aggregation technique are leveraged in "Adaptive-seq-BP-L(Δ)," by using Algorithm C.1 in the appendix of the online supplementary document [57] and PILD-ODA [1], respectively.

Table 2 provides clear evidence of the effectiveness of the sequential SAA framework and the use of warm starts. In an attempt to investigate the effect of geometric sampling schemes, which assuredly preserve the Monte Carlo canonical rate by Theorem 5.3, we next compare in Table 3 the computational results of the adaptive sequential SAA with a geometric sample size schedule having a fixed increase rate $c_1 = 1.5$ (option "Adaptive-seq-fixed(1.5)") against a dynamically chosen geometric increase rate with $c_0 = 1.05, c_h = 3$ and C_1 starting at 1.5 (option "Adaptive-seq-dyn(1.05, 3)"), when employed with a finite-time stopping criterion. We see that similar results are obtained by the two alternative options in terms of the computational time. "Adaptive-seq-dyn(1.05, 3)" exhibits slightly fewer inner and outer iterations, whereas the sample sizes seem significantly larger. Also, comparing Table 2 against Table 3, it seems clear that a geometrically increasing sample size schedule results in a large sample size at stopping but generally fewer outer iterations than the linear increasing rate employed in "Adaptive-seq-BP-L". In "Adaptive-seq-dyn," the sample size at

TABLE 3

Computational results of the adaptive partition-based level decomposition approach [1] - "PILD-ODA", Algorithm 6.1 with a fixed increasing rate "Adaptive-seq-fixed(1.5)," and Algorithm 6.1 with a dynamic increase rate "Adaptive-seq-dyn(1.05, 3)," and with C_1 starting at 1.5 on our test instances DEAK and DEAK-H.

Ins	N	PILD-ODA		Adaptive-seq-fixed(1.5)			Adaptive-seq-dyn(1.05, 3)		
		Time	M	Time	$M(L, n_L)$	CI(cov.)	Time	$M(L, n_L)$	CI(cov.)
	50K	53.4	19	1.5	21(7,1377)	(0.1,96)	1.6	19(4,2892)	(0.1,100)
40x20	100K	101.8	18	1.5	21(7,1438)	(0.1,99)	1.6	19(4,2886)	(0.1,100)
	50K	74.6	12	1.2	13(4,568)	(0.1,71)	1.8	13(4,1662)	(0.0,75)
40x40	100K	134.1	12	1.2	14(4,595)	(0.1,72)	1.7	13(3,1489)	(0.0,75)
	50K	206.2	19	1.9	23(3,318)	(0.1,100)	1.9	22(3,454)	(0.1,100)
40x60	100K	413.1	20	1.9	23(3,308)	(0.1,100)	1.9	23(3,458)	(0.1,100)
	50K	114.4	56	10.7	60(9,3675)	(0.1,100)	9.2	56(5,6048)	(0.1,100)
60x20	100K	252.2	60	11.0	60(9,3673)	(0.1,100)	9.5	56(5,6264)	(0.1,100)
	50K	502.0	65	14.1	73(6,921)	(0.1,100)	13.8	69(4,1620)	(0.1,100)
60x40	100K	929.4	67	14.7	73(6,959)	(0.1,100)	13.4	68(4,1566)	(0.1,100)
	50K	333.8	24	2.7	28(4,374)	(0.1,100)	2.8	27(3,617)	(0.1,100)
60x60	100K	622.3	24	2.7	28(4,374)	(0.1,100)	2.7	27(3,580)	(0.1,100)
40x20	50K	63.9	17	4.4	23(9,3034)	(0.1,97)	4.0	19(5,5400)	(0.1,99)
High	100K	139.2	18	4.4	23(9,3013)	(0.1,95)	5.3	20(5,7066)	(0.0,98)
40x40	50K	58.9	9	1.3	11(4,617)	(0.0,69)	1.8	11(4,1485)	(0.0,65)
High	100K	117.0	9	1.3	11(4,601)	(0.0,61)	1.7	10(3,1366)	(0.0,65)
40x60	50K	711.5	25	24.6	31(7,1535)	(0.1,93)	28.1	27(4,3240)	(0.1,96)
High	100K	1520.0	24	22.0	31(7,1427)	(0.1,92)	27.0	27(4,3046)	(0.1,93)
60x20	50K	162.6	46	38.0	46(10,5558)	(0.1,100)	34.3	43(6,9720)	(0.1,100)
High	100K	263.2	43	42.4	46(10,6086)	(0.1,100)	33.1	43(6,9720)	(0.1,100)
60x40	50K	432.8	31	70.4	40(9,2866)	(0.1,99)	78.1	33(5,5706)	(0.1,99)
High	100K	958.5	32	78.6	40(9,2894)	(0.1,98)	75.3	33(5,5688)	(0.1,96)
60x60	50K	673.5	23	42.3	32(7,1878)	(0.1,92)	42.2	27(5,3831)	(0.1,94)
High	100K	1591.9	25	38.4	32(7,1808)	(0.1,85)	50.9	27(5,4078)	(0.1,89)

stopping is even larger, but the number of outer iterations and the number of inner iterations are reduced, leading to less computational time in general. All options share similar behavior from the standpoint of the width of the confidence interval and its coverage.

We next investigate the sensitivity of chosen parameters such as the sample size increase rate for the proposed approaches. We observe from Table 2 and Table 3 that, as opposed to what has been suggested in theory (Theorem 5.3), Algorithm 6.1 with a linear sample size schedule performs competitively with the one with a geometric sample size schedule in our test instances. This may be because the algorithm "Sequential-BP-L(Δ)" in Table 2 with a value $\Delta = 100$ mimics the behavior of a geometric sequence. To validate this suspicion, Table 4 presents the performance of "Adaptive-seq-BP-L(Δ)" implemented with a linear sample size schedule having a smaller increase $\Delta = 10$ and "Adaptive-seq-fixed(c_1)" with a smaller geometric increase rate $c_1 = 1.1$. We also display the performance of "Adaptive-seq-dyn(c_0, c_h)" with $c_0 = 1.05, c_h = 2$ and with C_1 starting at 1.1, alongside these algorithms.

Comparing between Table 4 and Table 3, we see that the performance of "Adaptive-seq-BP-L(10)," where the sample size increases by 10 in each iteration, is significantly worse than "Adaptive-seq-BP-L(100)," where the sample size increases by 100 in each iteration. Although the final sample size n_L is lower at stopping when a slower linear sample size schedule is utilized, this comes at the price of a larger number of outer and inner iterations, leading to substantially more computational time. The same effect happens to option "Adaptive-seq-fixed(c_1)" as well, but at a much less

TABLE 4

Computational results of Algorithm 6.1 with the fixed-width stopping criterion and linear sample size schedule proposed in [5] with an increase of 10 scenarios per iteration ("Adaptive-seq-BP-L(10)"), Algorithm 6.1 with a geometrically increasing sample size schedule with rate $c_1 = 1.1$ ("Adaptive-seq-fixed(1.1)"), and Algorithm 6.1 with a geometrically increasing sample size schedule having a dynamic rate ("Adaptive-seq-dyn(1.05, 3)"), with C_1 starting at 1.1, on our test instances DEAK and DEAK-H.

Ins	N	Adaptive-seq-BP-L(10)		Adaptive-seq-fixed(1.1)		Adaptive-seq-dyn(1.05, 3)	
		Time	$M(L, n_L)$	Time	$M(L, n_L)$	Time	$M(L, n_L)$
	50K	3.2	37(23,551)	2.8	36(22,760)	1.7	21(6,2797)
40x20	100K	3.5	39(24,579)	2.7	35(21,721)	1.7	21(6,2711)
	50K	1.4	17(8,249)	1.4	19(9,250)	1.7	14(5,1319)
40x40	100K	1.3	17(7,239)	1.4	19(9,252)	1.5	14(5,1143)
	50K	2.5	27(6,204)	2.6	29(7,188)	2.3	25(4,421)
40x60	100K	2.1	26(5,186)	2.7	30(7,193)	2.1	25(4,369)
	50K	102.5	144(93,1945)	30.8	87(36,2760)	10.2	58(7,6383)
60x20	100K	103.8	143(93,1936)	31.1	87(36,2768)	10.5	58(7,6435)
	50K	47.9	92(24,560)	38.2	90(21,682)	15.5	73(6,1578)
60x40	100K	51.3	92(24,572)	37.1	88(21,665)	16.7	72(6,1733)
	50K	3.9	35(7,233)	4.4	38(9,235)	3.0	30(4,459)
60x60	100K	3.7	34(7,230)	4.1	37(9,222)	3.3	30(5,539)
40x20	50K	11.6	53(39,875)	7.9	42(28,1410)	4.2	21(7,5371)
High	100K	12.9	54(40,891)	9.5	44(30,1612)	5.1	21(7,6229)
40x40	50K	1.4	14(8,246)	1.4	15(9,231)	1.5	11(5,1030)
High	100K	1.5	14(8,251)	1.5	15(9,238)	1.4	11(5,956)
40x60	50K	263.4	77(30,683)	78.0	65(24,940)	32.8	30(6,3237)
High	100K	200.6	73(28,646)	68.3	63(23,859)	29.3	31(6,2904)
60x20	50K	337.7	128(94,1951)	101.6	73(38,3413)	34.5	45(8,9523)
High	100K	341.5	130(96,1988)	97.6	72(37,3271)	26.8	45(7,7979)
60x40	50K	2283.2	141(59,1271)	268.1	85(31,1758)	97.6	37(7,5817)
High	100K	2075.0	133(55,1196)	261.0	83(30,1710)	78.8	36(7,5363)
60x60	50K	742.6	88(35,793)	134.3	69(26,1106)	53.0	31(7,3987)
High	100K	621.0	82(32,735)	144.6	67(25,1052)	51.0	31(7,3593)

significant level, where utilizing a smaller c_1 ends up with a larger number of outer iterations and slightly more computational time. On the other hand, the performance of Algorithm 6.1 with a dynamic increase rate (option "Adaptive-seq-dyn(1.05, 3)") does not appear to be impacted much from the choice of the starting increasing rate C_1.

7.2.2. Computational results on other test instances. Finally, we present the performance of the best adaptive sequential SAA options (according to the above experiments on DEAK and DEAK-H instances) on an additional set of test instances that have a background in "real-world" applications. In particular, we consider Algorithm "Adaptive-seq-BP-L(100)" and Algorithm "Adaptive-seq-fixed(1.5)". We consider Algorithm "Adaptive-seq-fixed(1.5)" rather than the one with dynamic rate, "Adaptive-seq-dyn(1.05, 3)", as we find in our experiments that the parameters c_0 and c_h need to be fine tuned for specific instances in order to yield competitive performance.

From Table 5, we see that our conclusions made based on the results from the DEAK instances also stand for most of this additional set of test instances, except instances ssn and 20-term, which we discuss separately since they serve as interesting negative examples. In particular, we see that both sequential sampling algorithms Adaptive-seq-BP-L(100) and Adaptive-seq-fixed(1.5) yield high-quality solutions and their solution quality validation much more efficiently than PILD-ODA in most cases. Using a geometric sequence for the sample size schedule (Adaptive-seq-fixed(1.5) as opposed

TABLE 5

Computational results of the adaptive partition-based level decomposition approach [1] ("PILD-ODA"), Algorithm 6.1 with the fixed-width stopping criterion and sample size schedule proposed in [5] ("Adaptive-seq-BP-L(100)") and Algorithm 6.1 with a geometrically increasing sample size schedule with rate $c_1 = 1.5$ ("Adaptive-seq-fixed(1.5)") on an additional set of "real-world" test instances.

Ins	N	PILD-ODA		Adaptive-seq-BP-L(100)			Adaptive-seq-fixed(1.5)		
		Time	M	Time	$M(L, n_L)$	CI (cov.)	Time	$M(L, n_L)$	CI (cov.)
	50K	18.8	12	0.2	10(2,364)	(0.1,100)	0.3	11(3,292)	(0.1,100)
LandS	100K	35.6	12	0.2	10(2,366)	(0.1,100)	0.3	11(3,298)	(0.1,100)
	50K	37.5	32	0.5	24(3,602)	(0.0,89)	0.5	25(4,545)	(0.0,94)
gbd	100K	75.9	29	0.5	24(3,582)	(0.0,94)	0.5	25(4,576)	(0.0,94)
	20K	6.9	5	0.1	4(1,280)	(0.0,99)	0.1	4(1,145)	(0.0,99)
cep	50K	17.1	4	0.1	4(1,292)	(0.0,100)	0.1	4(2,151)	(0.0,100)
	20K	13.7	20	2.1	65(4,700)	(0.1,66)	2.7	87(5,892)	(0.1,64)
pgp2	50K	31.0	22	2.3	68(4,732)	(0.1,53)	2.2	77(4,727)	(0.1,49)
	20K	211.5	54	2.2	64(1,146)	(0.0,80)	2.1	65(1,114)	(0.0,75)
4node	50K	487.4	51	2.2	64(1,144)	(0.0,80)	1.9	63(1,111)	(0.0,82)
	20K	82.4	54	140.0	503(16,3136)	(0.1,80)	87.0	305(10,6704)	(0.1,91)
retail	50K	179.3	53	123.9	469(15,3040)	(0.1,78)	91.9	302(10,6998)	(0.1,86)
	20K	735.3	187	593.3	347(12,2346)	(0.1,98)	383.4	354(9,3454)	(0.1,100)
baa99-20	50K	1670.4	184	659.7	366(12,2344)	(0.1,100)	380.4	356(9,3349)	(0.1,100)
	2K	1367.9	616	2451.3	596(2,212)	(0.1,82)	1889.8	657(2,148)	(0.1,82)
20-term	5K	1617.0	726	2571.0	554(2,280)	(0.1,62)	2687.5	696(2,188)	(0.1,82)
ssn	5K	6482.9	804	-	2028(6,1104)	(17.0,100)	-	2477(7,1586)	(16.2,100)

to Adaptive-seq-BP-L(100)), further computational enhancements are obtained. The sequential sampling algorithms usually end up with a larger number of inner iterations than the deterministic algorithm PILD-ODA that employs the full set of samples. However, the computational savings brought by the smaller sample sizes used in the sequential sampling algorithms, which are reflected in the amount of work involved per inner iteration, turn out to offset the increase in the number of inner iterations on these instances. This is consistent with what our theoretical results presented in Section 5. In addition, we can observe some "undercoverage" phenomenon for pgp2 instances (as shown in column "cov."), which is somewhat expected as the variance associated with their solutions is quite large [3]. Procedures that employ more than a single replication, such as A2RP proposed in [3], can be used to address the issue of "undercoverage".

As noted earlier, the problem instances ssn and 20-term are interesting as negative examples, where the proposed sequential sampling algorithms do not yield gains realized in other problem instances. Instance ssn is challenging most probably due to the high inherent variance of the underlying random variables and the associated computational challenge in solving the second-stage problems while also reporting solution accuracy. For instance, observe from Table 5 that both options Adaptive-seq-BP-L(100) and Adaptive-seq-fixed(1.5) fail to provide confidence intervals with a satisfactory width within the stipulated time limit. We suspect that the variance associated with the second-stage optimal cost, along with the strict nature of the stopping criterion, contributes to ssn being in contrast with other test instances appearing in Table 5. The negative effect of such high variance can be mitigated, at least in principle, by directly using variance reduction techniques, or through alternative stopping ideas such as that proposed in [66].

The negative context presented by the instance 20-term appears to be different in spirit than ssn. Specifically, observe that Adaptive-seq-BP-L(100) and Adaptive-seq-fixed(1.5) exhibit longer computational times than the deterministic algorithm PILD-ODA on instances 20-term despite having a small number of outer iterations and small sample sizes used in each outer iteration. In fact, most of the computational effort is expended on solving the master problem, while the second-stage subproblems can be solved efficiently. The increased effort in solving the master problem could

be because the "warmstart" feature that worked well for other instances is not as effective here, since "recovering" a lower cutting-plane approximation using the dual vector information stored from previous iterations, although "generated on the fly," requires the problem to be re-solved with a new right-hand-side at every re-start, and whenever any new first-stage decision vector is generated by the algorithm. This special feature of 20-term — time-consuming master problems alongside easily solved second-stage problems — means that our implementation's premise of the total computational burden being dominated by the task of solving second-stage LPs is not true in the 20-term context. The clear lesson from 20-term is then to adapt the implementation to explicitly account for the cost of solving the master problem alongside the cost of solving the second-stage problems, potentially leading to the use of a larger constant c_1 in such contexts. In addition, alternative "warmstarting" techniques for sequential sampling algorithms, such as those arising in stochastic decomposition [31, 32] and stochastic dual dynamic programming algorithms [44], may be more effective in relieving the computational challenges in repeatedly solving the master problem on these instances.

8. CONCLUDING REMARKS. We propose an adaptive sequential SAA algorithm to solve 2SLPs. During each iteration of the proposed framework, a piecewise linear convex optimization sample-path problem is generated with a scenario set having a specified size, and solved imprecisely to within a tolerance that is chosen to balance statistical and computational errors. We find that (i) the use of an appropriate solver to solve the sample-path problems, (ii) solving each sample-path problem only imprecisely to an appropriately chosen error tolerance, and (iii) the use of warm starts when solving sample-path problems, are crucial for efficiency.

Our theoretical results suggest that the optimality gap and the distance from the true solution set (of the generated stochastic iterates) converges to zero almost surely and in expectation. Moreover, when the sample sizes are increased according to a geometric rate, the fastest possible convergence rate under iid Monte Carlo sampling is preserved. This result is analogous to the $\mathcal{O}(\epsilon^{-2})$ optimal complexity rate for deterministic non-smooth convex optimization. Slower sample size increases result in a poorer convergence rate. Interestingly, the proposed framework also facilitates the use of dependent sampling schemes such as LHS, antithetic variates, and quasi-Monte Carlo without affecting convergence or the lower bound on the rate results. The use of such variance reduction ideas have been shown to be effective.

Our extensive numerical studies indicate that the proposed adaptive sequential SAA framework is able to produce high-quality solutions to 2SLPs significantly more efficiently than existing decomposition approaches that solve a single sample-path problem generated using a large sample size. Such gains are principally due to the sequential framework, the progressive increase in sample sizes in an optimal way, and the use of "warm starts" in solving the sample-path problems. Our numerical experience has also revealed problem instances having certain challenging features that are not directly addressed by the implementations that we have used for illustration. These challenges could be mitigated by using alternative solvers that exploit particular problem structures and/or other termination criteria such as that proposed in [66].

We believe that similarly efficient sequential SAA algorithms are possible for large-scale multistage convex stochastic programs, and possibly even stochastic integer programs. The key appears to be principled choices for adaptive sample sizes, solver for the sample-path problems, and adaptive optimality tolerance parameters. Ongoing research efforts are accordingly directed.

Acknowledgments. We greatly appreciate the comments and suggestions of the associate editor and two anonymous referees. The first author acknowledges support provided by the Of-

fice of Naval Research (ONR) through ONR Grant N000141712295, and by the National Science Foundation through the grant CMMI 1538050. The second author acknowledges partial support by the National Science Foundation (NSF) under grant CMMI 1854960. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ONR or NSF.

REFERENCES

[1] W. van. Ackooij, W. de. Oliveira, and Y. Song, *An adaptive partition-based level decomposition for solving two-stage stochastic programs with fixed recourse*, INFORMS Journal on Computing, 30 (2018), pp. 57–70.
[2] K. A. Ariyawansa and A. J. Felt, *On a new collection of stochastic linear programming test problems*, INFORMS Journal on Computing, 16 (2004), pp. 291–299.
[3] G. Bayraksan and D. Morton, *Assessing solution quality in stochastic programs*, Mathematical Programming, 108 (2006), pp. 495–514.
[4] G. Bayraksan and D. Morton, *A sequential sampling procedure for stochastic programming*, Operations Research, 59 (2011), pp. 898–913.
[5] G. Bayraksan and P. Pierre-Louis, *Fixed-width sequential stopping rules for a class of stochastic programs*, SIAM Journal on Optimization, 22 (2012), pp. 1518–1548.
[6] J. Y. Bello-Cruz and W. de. Oliveira, *Level bundle-like algorithms for convex optimization*, Journal of Global Optimization, 59 (2014), pp. 787–809, https://doi.org/10.1007/s10898-013-0096-4.
[7] P. Billingsley, *Probability and Measure*, Wiley, New York, NY., 1995.
[8] J. Birge, *State-of-the-art-survey – stochastic programming: Computation and applications*, INFORMS Journal on Computing, 9 (1997), pp. 111–133, https://doi.org/10.1287/ijoc.9.2.111.
[9] J. R. Birge and F. Louveaux, *Introduction to stochastic programming*, Springer Science & Business Media, 2011.
[10] R. Bollapragada, R. Byrd, and J. Nocedal, *Adaptive sampling strategies for stochastic optimization*, SIAM Journal on Optimization, 28 (2018), pp. 3312–3343.
[11] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Review, 60 (2018), pp. 223–311.
[12] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities and Model Selection*, Springer, Berlin, Germany, 2007.
[13] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities:A Nonasymptotic Theory of Independence*, Oxford University Press, Cambridge, UK, 2013.
[14] M. Chen and S. Mehrotra, *Self-concordance and decomposition based interior point methods for the two stage stochastic convex optimization problem*, SIAM Journal on Optimization, 21 (2011), pp. 1667–1687.
[15] M. Chen, S. Mehrotra, and D. Papp, *Scenario generation for stochastic optimization problems via the sparse grid method*, Computational Optimization and applications, 62 (2015), pp. 669–692.
[16] Y. S. Chow and H. E. Robbins, *On the asymptotic theory of fixed-width confidence intervals for the mean*, Annals of Mathematical Statistics, 36 (1965), pp. 457–462.
[17] G. B. Dantzig, *Linear programming under uncertainty*, Management Science (pre-1986), 1 (1955), p. 197.
[18] I. Deák, *Testing successive regression approximations by large-scale two-stage problems*, Annals of Operations Research, 186 (2011), pp. 83–99.
[19] G. Deng and M. C. Ferris, *Variable-number sample-path optimization*, Mathematical Programming, (2009), pp. 81–109.
[20] L. Dümbgen, S. A. van de Geer, M. C. Veraar, and J. A. Wellner, *Nemirovski's inequalities revisited*, The American Mathematical Monthly, 117 (2010), pp. 138–160.
[21] J. Dupačová and R. J. B. Wets, *Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems*, The Annals of Statistics, 16 (1988), pp. 1517–1549.
[22] R. Durrett, *Probability: Theory and Examples*, Cambridge University Press, New York, NY, 2010.
[23] Y. Ermoliev and A. Gaivoronski, *Stochastic quasigradient methods for optimization of discrete event systems*, Annals of Operations Research, 39 (1992), pp. 1–39.
[24] Y. M. Ermoliev, *Stochastic quasigradient methods and their application to system optimization.*, Stochastics, 9 (1983), pp. 1–36.
[25] C. I. Fábián and Z. Szőke, *Solving two-stage stochastic programming problems with level decomposition*, Computational Management Science, 4 (2007), pp. 313–353.
[26] A. Ferguson and G. Dantzig, *The allocation of aircraft to routes?an example of linear programming under*

uncertain demand, Management science, 3 (1956), pp. 45–73.
[27] P. Glasserman, *Monte Carlo methods in financial engineering*, vol. 53, Springer Science & Business Media, 2013.
[28] P. Glynn and G. Infanger, *Simulation-based confidence bounds for two-stage stochastic programs*, Mathematical Programming, 138 (2013), pp. 15–42.
[29] F. Hashemi, S. Ghosh, and R. Pasupathy, *On adaptive sampling rules for stochastic recursions*, in Proceedings of the 2014 Winter Simulation Conference, A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds., Piscataway, NJ, 2014, Institute of Electrical and Electronics Engineers, Inc.
[30] Y. Herer, M. M. Tzur, and E. Yücesan, *The multilocation transshipment problem*, IIE transactions, 38 (2006), pp. 185–200.
[31] J. Higle and S. Sen, *Stochastic decomposition: An algorithm for two-stage linear programs with recourse*, Mathematics of operations research, 16 (1991), pp. 650–669.
[32] J. Higle and S. Sen, *Stochastic decomposition: a statistical method for large scale stochastic linear programming*, vol. 8, Springer Science & Business Media, 2013.
[33] T. Homem-de-Mello, *Variable-sample methods for stochastic optimization*, ACM Transactions on Modeling and Computer Simulation (TOMACS), 13 (2003), pp. 108–133.
[34] T. Homem-de-Mello and G. Bayraksan, *Monte Carlo sampling-based methods for stochastic optimization*, in Reviews, Surveys in Operations Research and Management Science, Elsevier, 2014, pp. 56–85.
[35] S. Kim, R. Pasupathy, and S. G. Henderson, *A guide to SAA*, in Encyclopedia of Operations Research and Management Science, M. Fu, ed., Hillier and Lieberman OR Series, Elsevier, 2014.
[36] P. L'Ecuyer, *Randomized quasi-monte carlo: An introduction for practitioners*, in Monte Carlo and Quasi-Monte Carlo Methods, Springer, 2018.
[37] P. l'Ecuyer, C. Lécot, and B. Tuffin, *A randomized Quasi-Monte Carlo simulation method for Markov Chains*, Operations Research, 56 (2008), pp. 958–975.
[38] M. Ledoux and M. Talagrand, *Probability in Banach Spaces: Isoperimetry and Processes*, Springer, Berlin, Germany, 2011.
[39] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, *New variants of bundle methods*, Mathematical programming, 69 (1995), pp. 111–147.
[40] F. Louveaux and Y. Smeers, *Optimal investments for electricity generation: a stochastic model and a test-problem*, in Numerical Techniques for Stochastic Optimization, Springer-Verlag, Berlin, 1988, pp. 33–64.
[41] J. Luedtke and S. Ahmed, *A sample approximation approach for optimization with probabilistic constraints*, SIAM Journal on Optimization, 19 (2008), pp. 674–699.
[42] P. lEcuyer, *Randomized quasi-monte carlo: An introduction for practitioners*, in 12th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (MCQMC2016), August 2016.
[43] W.-K. Mak, D. P. Morton, and R. K. Wood, *Monte carlo bounding techniques for determining solution quality in stochastic programs*, Operations research letters, 24 (1999), pp. 47–56.
[44] V. de. Matos, A. Philpott, and E. Finardi, *Improving the performance of stochastic dual dynamic programming*, Journal of Computational and Applied Mathematics, 290 (2015), pp. 196–208.
[45] M. D. McKay, R. J. Beckman, and W. J. Conover, *Comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, 21 (1979), pp. 239–245.
[46] A. Nedić and D. Bertsekas, *Convergence rate of incremental subgradient algorithms*, in Stochastic optimization: algorithms and applications, Springer, 2001, pp. 223–264.
[47] B. L. Nelson, *Antithetic-variate splitting for steady-sate simulations*, European journal of operational research, 36 (1988), pp. 360–370.
[48] B. L. Nelson, *Foundations and Methods of Stochastic Simulation: A First Course*, Springer, New York, NY., 2013.
[49] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2004.
[50] Y. Nesterov, *Lectures on convex optimization*, Springer, second ed., 2018.
[51] W. de. Oliveira and C. Sagastizábal, *Level bundle methods for oracles with on demand accuracy*, Optimization Methods and Software, 29 (2014), pp. 1180–1209.
[52] A. B. Owen, *Monte Carlo theory, methods and examples*, 2013.
[53] R. Pasupathy, *On choosing parameters in retrospective-approximation algorithms for simulation-optimization*, in Proceedings of the 2006 Winter Simulation Conference, L. Perrone, F. Wieland, J. Liu, B. Lawson, D. Nicol, and R. Fujimoto, eds., Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 2006.
[54] R. Pasupathy, *On choosing parameters in retrospective-approximation algorithms for stochastic root finding*

and simulation optimization, Operations Research, 58 (2010), pp. 889–901.
[55] R. Pasupathy, P. W. Glynn, S. Ghosh, and F. Hashemi, *On sampling rates in simulation-based recursions.*, SIAM Journal on Optimization, 28 (2018), pp. 45–73.
[56] R. Pasupathy and B. W. Schmeiser, *Retrospective-approximation algorithms for multidimensional stochastic root-finding problems*, ACM TOMACS, 19 (2009), pp. 5:1–5:36.
[57] R. Pasupathy and Y. Song, *Online supplementary document for: Adaptive sequential SAA for solving two-stage stochastic linear programs*, 2020, http://www.optimization-online.org/DB_HTML/2019/02/7068.html.
[58] E. Polak and J. Royset, *Efficient sample sizes in stochastic nonlinear programming*, Journal of Computational and Applied Mathematics, 217 (2008), pp. 301–310.
[59] B. Polyak, *Introduction to Optimization*, Optimization Software In, New York, NY, 1987.
[60] B. T. Polyak, *Nonlinear programming methods in the presence of noise.*, Mathematical Programming, 1 (1978), pp. 87–97.
[61] S. M. Robinson, *Analysis of sample-path optimization*, Mathematics of Operations Research, 21 (1996), pp. 513–528.
[62] J. Royset, *On sample size control in sample average approximations for solving smooth stochastic programs*, Computational Optimization and Applications, 55 (2013), pp. 265–309.
[63] J. O. Royset and R. Szechtman, *Optimal budget allocation for sample average approximation*, Operations Research, 61 (2013), pp. 762–776.
[64] A. Ruszczynski and A. Shapiro, eds., *Stochastic Programming. Handbook in Operations Research and Management Science*, Elsevier, New York, NY., 2003.
[65] S. Sen, R. Doverspike, and S. Cosares, *Network planning with random demand*, Telecommunication systems, 3 (1994), pp. 11–30.
[66] S. Sen and Y. Liu, *Mitigating uncertainty via compromise decisions in two-stage stochastic linear programming: Variance reduction*, Operations Research, 64 (2016), pp. 1422–1437.
[67] A. Shapiro, *Asymptotic behavior of optimal solutions in stochastic programming*, Mathematics of Operations Research, 18 (1993), pp. 829–845.
[68] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*, SIAM, 2014.
[69] A. Shapiro and T. Homem-de-Mello, *A simulation-based approach to two-stage stochastic programming with recourse*, Mathematical Programming, 81 (1998), pp. 301–325.
[70] A. Shapiro and T. Homem-de-Mello, *On the rate of convergence of optimal solutions of monte carlo approximations of stochastic programs*, SIAM Journal on Optimization, 11 (2000), pp. 70–86.
[71] S. Shashaani, F. S. Hashemi, and R. Pasupathy, *ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization.*, SIAM Journal on Optimization, 28 (2018), pp. 3145–3176.
[72] M. Stein, *Large sample properties of simulations using latin hypercube sampling*, Technometrics, 29 (1987), pp. 143–151.
[73] R. Stockbridge and G. Bayraksan, *Variance reduction in Monte Carlo sampling-based optimality gap estimators for two-stage stochastic linear programming*, Computational Optimization and Applications, 64 (2016), pp. 407–431.
[74] M. Talagrand, *Sharper bounds for gaussian and empirical processes*, Annals of Probability, 22 (1994), pp. 28–76.
[75] H. Wang, R. Pasupathy, and B. W. Schmeiser, *Integer-ordered simulation optimization using R-SPLINE: Retrospective search using piecewise-linear interpolation and neighborhood enumeration*, ACM TOMACS, 23 (2013).
[76] D. Williams, ed., *Probability with Martingales*, Cambridge University Press, Cambridge, UK, 1991.
[77] C. Wolf, C. Fábián, A. Koberstein, and L. Suhl, *Applying oracles of on-demand accuracy in two-stage stochastic programming–a computational study*, European Journal of Operational Research, 239 (2014), pp. 437–448.
[78] G. Zhao, *A log-barrier method with Bender's decomposition for solving two-stage stochastic linear programs*, Mathematical Programming, Series A, 90 (2001), pp. 501–536.