An Adaptive Sequential Sample Average Approximation Framework for Solving Two-stage Stochastic Programs

Raghu Pasupathy*and Yongjia Song†

February 12, 2019

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 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 optimization) sample-path 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 quasi-Monte Carlo. We characterize the almost sure and ℓ_1-norm convergence behavior of the distance of the generated stochastic iterates to the true solution set. We also characterize the corresponding convergence rate, and a sample size schedule that results in the best possible work complexity rate; the latter rate is Monte Carlo canonical and analogous to the $\mathcal{O}(\epsilon^{-2})$ optimal complexity for non-smooth convex optimization. We report extensive numerical tests that clearly indicate favorable performance over existing methods, due primarily to the use of a sequential framework, use of optimal sample sizes, and warm starts. The framework can be stopped in finite-time to return a solution endowed with a probabilistic guarantee on quality.

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 dependent.

2SLPs were originally introduced by [14] and, owing to their usefulness, have been extensively studied over the last few decades [7]. 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 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., [16, 51, 53].

*Department of Statistics, Purdue University, West Lafayette, IN, USA, (pasupath@purdue.edu).

†Department of Industrial Engineering, Clemson University, Clemson, SC, USA, (yongjis@clemson.edu).

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 (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 [48] and surveys [15, 29].

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), an appropriate solver in (ii), and a mechanism for stopping 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 [29, 32, 44], 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 solver 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 study 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 6.

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 compares quite favorably 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 distance between the iterates resulting from the proposed framework and the true solution set converges to zero almost surely and in ℓ_1-norm. We also derive the work complexity of our algorithm framework, that is, we provide an upper bound on the expected total 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. Furthermore, we prove that using sample size schedules that deviate from the proposed schedule will lead to inferior rates.

We emphasize that the framework we propose allows for the use of dependent sampling, e.g., Latin-hypercube sampling (LHS) [33], antithetic variates [36], quasi-Monte Carlo [23] *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 [50, 11] 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 [6] and algorithms to solve 2SLPs can be conveniently classified based on whether or not they can treat the context $|\Xi| = \infty$. As noted in [54], the context $|\Xi| < \infty$ has generated an enormous amount of work resulting in various algorithm classes that directly exploit the finite sum structure stemming from assuming $|\Xi| < \infty$ — see [6] and [10] for entry points into this substantial literature.

When $|\Xi| = \infty$, or for that matter when $|\Xi|$ is "very large," Monte Carlo sampling approaches appear to be the most viable alternative [48, 46, 47]. 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 some time now [13, 19, 25, 27, 28, 46, 52]. For instance, [46] suggests the idea of solving a sequence of sample-path problems with increasing sample sizes as a practical matter, and [27] 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, [41, 43] 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.

Similar to [27], [38, 40] 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 [27], 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 [38, 41, 43] 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, e.g., [8, 9, 24, 39, 49]. However, as of this writing, we know of no example where adaptive sampling has been treated in a non-smooth convex optimization setting such as 2SLPs.

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, [2, 3] 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 [2, 3] 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 5 provides a finite stopping rule for the adaptive sequential SAA framework by incorporating the sequential sampling approaches proposed in [2] and [3]. Section 6 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:

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

where the $r_1 \times n_1$ matrix A, $n_1 \times 1$ vectors b and c are assumed to fixed and known, and the second-stage *value function*

$$q(x) = \mathbb{E}[Q(x,\xi)] = \int_\Xi Q(x,\xi)\,\mathbb{P}(d\xi), \tag{1}$$

and for each $\xi \in \Xi$,

$$Q(x,\xi) = \min\ \left\{d(\xi)^\top y \mid W(\xi)y \geq h(\xi) - T(\xi)x,\, y \in \mathbb{R}^{n_2}_+\right\}. \tag{2}$$

Since $q(x)$ is not directly "observable," the iterative algorithms we propose, during the ℓ-th iteration, construct the following sample-path approximation to problem (P) by "generating scenarios" $\xi^\ell_1, \xi^\ell_2, \ldots, \xi^\ell_{m_\ell} \in \Xi$ that are identically distributed according to some probability measure that we do not make explicit. Formally, the sample-path problem due to scenarios $\xi^\ell_1, \xi^\ell_2, \ldots, \xi^\ell_{m_\ell} \in \Xi$ is given by

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

where the second-stage *sample-path value function*

$$Q^\ell_{m_\ell}(x) = m_\ell^{-1} \sum_{i=1}^{m_\ell} Q(x,\xi^\ell_i), \tag{3}$$

and $Q(x,\xi^\ell_i)$ is given through (2). Notice that problem (P_ℓ) is a sample-path approximation of problem (P), and $Q^\ell_{m_\ell}(\cdot)$ is a sample-path approximation of $q(\cdot)$. The precise sense in which these are approximations will become clear when we state the standing assumptions in Section 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 $Q^\ell_{m_\ell}(\cdot)$ denotes an $\mathcal{F}_\ell$-measurable function estimator of $q(\cdot)$ constructed from $\xi^\ell_i, i = 1,2,\ldots,m_\ell$ identically distributed, $\mathcal{F}_\ell$-measurable random objects. The random objects $\xi^\ell_i, i = 1,2,\ldots,m_\ell; \ell = 1,2,\ldots$ correspond to what have been called "scenarios" in the SP literature. We will use $\xi^\ell \in \Xi$ to denote a generic $\mathcal{F}_\ell$-measurable outcome, and $\xi^\ell_1, \xi^\ell_2, \ldots$ to denote $\mathcal{F}_\ell$-measurable outcomes obtained from Monte Carlo sampling during iteration ℓ.

The framework we propose allows for a variety of dependence structures of $\xi^\ell_i, i = 1,2,\ldots,m_\ell$ 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 $\xi^\ell_i, i = 1,2,\ldots,m_\ell$ are mutually independent and identically distributed for each ℓ; the objects $\xi^\ell_i, i = 1,2,\ldots,m_\ell$ can also be generated so as to satisfy chosen dependency structures that reduce variance, e.g., LHS [33], antithetic variates [35], and quasi-Monte Carlo [23]. Similarly, across iterations $\ell = 1,2,\ldots$, one can arrange for scenarios from previous iterations to be reused in subsequent iterations. Indeed, we will have to make certain assumptions on $Q^\ell_{m_\ell}(\cdot), \ell = 1,2,\ldots$ in Section 2.2 (that will implicitly impose restrictions on the nature of sampling) to ensure that $Q^\ell_{m_\ell}(\cdot)$ approximates $q(\cdot)$ in a precise sense.

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 $\mathcal{S}^*_{m_\ell}(\epsilon) := \{x \in \mathcal{X} : c^\top x + Q^\ell_{m_\ell}(x) - z^*_{m_\ell} \leq \epsilon\}$ the ϵ-optimal solution set associated with 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 $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^n$, $\|x\|_2$ refers to the Euclidean norm $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$. (iii) The distance between a point $x \in \mathbb{R}^n$ and a set $X \subseteq \mathbb{R}^n$ is defined as $\mathrm{dist}(x, X) := \inf\{\|x - z\|_2 : z \in X\}$, and the distance between two sets $X, Y \subseteq \mathbb{R}^n$ is defined as $\mathrm{dist}(X, Y) := \sup_{x \in X}\{\mathrm{dist}(x, Y)\}$. The definition we have used for $\mathrm{dist}(\cdot, \cdot)$ suffices for our purposes even though it is not a metric since $\mathrm{dist}(X, Y) \neq \mathrm{dist}(Y, X)$ in general. (iv) The diameter $\mathrm{diam}(\mathcal{D})$ of a set $\mathcal{D} \subseteq \mathbb{R}^n$ is defined as $\mathrm{diam}(\mathcal{D}) := \sup_{x,y \in \mathcal{D}}\{\|x - y\|_2\}$. (v) The projection of a point $x \in \mathbb{R}^n$ onto a set $X \subseteq \mathbb{R}^n$ is defined as $\Pi(x, X) := \arg\min_{z \in X}\{\|x - z\|_2\}$. (vi) $|X|$ denotes the cardinality of set X. (vii) 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 $n \to \infty$.

2.2 Assumptions

The following is a list of assumptions that we will use to prove various results in the paper. Assumptions 1,2,4 are standing assumptions in that we will assume these to hold always. Assumption 3 and Assumption 5 will be invoked when needed.

Assumption 1. *The first-stage feasible region $\mathcal{X}$ of problem (P) is compact, furthermore, problem (P) has relatively complete recourse, that is, for ξ^ℓ that is $\mathcal{F}_\ell$-measurable,*

$$\mathbb{P}\left[\bigcap_{x \in \mathcal{X}} \left\{y \in \mathbb{R}^{n_2}_+ : W(\xi^\ell)y \geq h(\xi^\ell) - T(\xi^\ell)x\right\} = \emptyset \,|\, \mathcal{F}_{\ell-1}\right] = 0 \ \textit{a.s.}$$

The assumption of relatively complete recourse is made for simplicity of exposition. When this assumption does not hold, the proposed adaptive sequential SAA framework can be augmented with logic that excludes those $x \in \mathcal{X}$ that lead to $\mathbb{P}\left[\left\{y \in \mathbb{R}^{n_2}_+ : W(\xi^\ell)y \geq h(\xi^\ell) - T(\xi^\ell)x\right\} = \emptyset\right] > 0$.

Assumption 2. *The objective function $c^\top x + q(x)$ in problem (P) is continuous and exhibits γ_0-first-order growth on $\mathcal{X}$, that is, there exists $\gamma_0 > 0$ such that for $x \in \mathcal{X} \setminus \mathcal{S}^*$,*

$$c^\top x + q(x) - z^* \geq \gamma_0 \,\mathrm{dist}(x, \mathcal{S}^*).$$

Assumption 3. *Define the expected squared inverse growth rate $\gamma^{-2}(n)$ for the sample-path problem (P_ℓ) having sample size n as*

$$\gamma^{-2}(n) := \limsup_{\epsilon \to 0^+} \frac{1}{\epsilon^2} \mathbb{E}\left[(\mathrm{dist}(\mathcal{S}^*_n(\epsilon), \mathcal{S}^*_n))^2\right].$$

Then, there exists $\gamma^{-2} < \infty$ such that for large enough n, $\gamma^{-2}(n) \leq \gamma^{-2}$.

Assumptions 2 and 3 are regularity conditions having to do with the growth behavior of the true objective function and expected growth behavior of the sample-path objective function, respectively. Assumption 2 is a standard assumption [48] that imposes a minimum growth condition of the objective function $c^\top x + q(x)$ outside the solution set $\mathcal{S}^*$. Assumption 3 defines and imposes an analogous growth rate condition on the sample-path functions. Assumption 3 can be violated but only in pathological sampling situations designed to ensure that the distance between the ϵ-optimal set and the true optimal set of the sample-path problem decays to zero very slowly as $n \to \infty$.

Assumption 4. *The variance $Var\left[Q(x, \xi^\ell)\right]$ is bounded almost surely, that is, there exists $\sigma^2 < \infty$ such that*

$$\sup_{x \in \mathcal{X}} \mathrm{Var}\left[Q(x, \xi^\ell) \,|\, \mathcal{F}_{\ell-1}\right] \leq \sigma^2 < \infty \ \textit{a.s.} \tag{4}$$

Furthermore, the sequence $\xi_1^\ell, \xi_2^\ell, \ldots$ *is such that the sample-mean* $\zeta_m^\ell(x) = m^{-1}\sum_{i=1}^m Q(x, \xi_i^\ell) - q(x)$ *satisfies a large-deviation bound for* $x \in \mathcal{X}$*, that is, there exists a positive-valued convex function* $r_1(\cdot)$ *such that for* t *in some neighborhood of zero,*

$$\lim_{m\to\infty} -\frac{1}{m}\log\left(\mathbb{P}\left[\zeta_m^\ell(x) > t \,|\, \mathcal{F}_{\ell-1}\right]\right) \;\geq\; r_1(t) \;\; a.s. \tag{5}$$

Assumption 5. *Suppose we denote the* $\mathcal{F}_\ell$*-measurable random variable* $L(\xi^\ell)$ *as follows.*

$$L(\xi^\ell) := \inf\{\tilde{L} : |Q(x, \xi^\ell) - Q(y, \xi^\ell)| \leq \tilde{L}\|x - y\| \text{ for all } x, y \in \mathcal{X}\}.$$

The sequence $\xi_1^\ell, \xi_2^\ell, \ldots$ *is such that the sample-mean* $L_m^\ell = m^{-1}\sum_{i=1}^m L(\xi_i^\ell)$ *satisfies a large-deviation bound, that is, there exists a positive-valued convex function* $r_2(\cdot)$ *such that for* t *in some neighborhood of zero,*

$$\lim_{m\to\infty} -\frac{1}{m}\log\left(\mathbb{P}\left[L_m^\ell > t \,|\, \mathcal{F}_{\ell-1}\right]\right) \;\geq\; r_2(t) \;\; a.s. \tag{6}$$

Assumption 4 is a stipulation on the quality of the Monte Carlo estimator that is at hand. Variations of Assumption 4 are usually used as a sufficient condition for the convergence of SAA to the true problem in terms of optimal values and solutions. For example, in Chapter 5 of [48], we see that even for convergence of optimal values of SAA, one needs uniform convergence (across $x \in \mathcal{X}$) of the sample-path functions. Assumption 4 is a sufficient condition for such uniform convergence. Some form of (4) is routinely made in the SP literature [1] and is generally easy to satisfy in 2SLPs when the feasible region $\mathcal{X}$ is compact. Since we have made no assumption about the mutual independence of $\xi_1^\ell, \xi_2^\ell, \ldots$, we are forced to impose a separate stipulation (5) on the sample mean as opposed to invoking Cramér's theorem [18] for iid random variables having a finite moment-generating function in a neighborhood of zero.

Assumption 5 has to do with the smoothness of the estimator; it stipulates that except for a set of measure zero, all second-stage objectives are Lipschitz with a scenario dependent Lipschitz constant $L(\xi^\ell)$. Assumption 4 and Assumption 5 assume that the sample mean constructed from the constituent random variables satisfy a large-deviation type bound. Recall that Cramér's theorem guarantees that the sample means of iid random variables satisfy a large-deviation bound if the random variables constituting the sample mean have a finite moment-generating function in some neighborhood of zero [18]. For example, all bounded random variables, and all random variables that fall within the exponential family, e.g., normal, gamma, have finite moment-generating functions in some neighborhood of zero. Similar large-deviation bounds are common for sample means constructed from dependent random variables, e.g., LHS [20] and quasi-Monte Carlo through the Koksma-Hlawka bound [23].

3 ADAPTIVE SEQUENTIAL SAA

In this section, we present the proposed adaptive sequential SAA framework. The proposed framework, at a high level, is based on the following three 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 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

Algorithm 1 An adaptive sequential SAA framework.

Input: Solver-$\mathcal{A}$, a sample selection policy, a constant $c > 0$, and a constant $\delta > 0$. Set $\ell \leftarrow 0$.
for $\ell = 1, 2, \cdots$ **do**
  Select the sample size m_ℓ for the current outer iteration ℓ and draw a random sample $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell\}$.
  for $t = 1, 2, \cdots$ **do**
    Use Solver-$\mathcal{A}$ to execute the t-th inner iteration for solving problem (P_ℓ), obtain a candidate solution $\hat{x}^{\ell,t}$, and compute $G^{\ell,t}$ and $\hat{\text{se}}_{\ell,t}$ accordingly.
    if $G^{\ell,t} \leq \epsilon_{\ell,t} := c \max\left\{\hat{se}_{\ell,t}, \frac{\delta}{\sqrt{m_\ell}}\right\}$ **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

inexactly up to precision ϵ_ℓ using an iterative optimization algorithm (generically called Solver-$\mathcal{A}$) for non-smooth convex programs, e.g., the subgradient method [34], level bundle method [30]. 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

$$G^{\ell,t} < \epsilon_{\ell,t} := c \max\left\{\hat{\text{se}}_{\ell,t}, \frac{\delta}{\sqrt{m_\ell}}\right\}, \tag{7}$$

where $\delta, c > 0$ are chosen constant parameters, and

$$\hat{\text{se}}_{\ell,t} := m_\ell^{-1} \sqrt{\sum_{i=1}^{m_\ell} \left[Q(\hat{x}^{\ell,t}, \xi_i^\ell) - Q_{m_\ell}^\ell(\hat{x}^{\ell,t})\right]^2}. \tag{8}$$

We informally call $\epsilon_{\ell,t}$ appearing in (7) the *error tolerance*; notice that the condition in (7) 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 constant $\delta > 0$ appearing in (7) has been introduced for practical purposes only, to hedge against the rare event that we generate scenarios such that $\hat{\text{se}}_{\ell,t} = 0$.

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, i.e., 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 (7) 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 1 is satisfied by the candidate solution $\hat{x}^\ell$. We defer our specification of the outer stopping criterion to Section 5.

Algorithm 1 is *adaptive* in that ϵ_ℓ is not pre-specified — it is a function of the 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 the way we have, avoids several mathematical complexities that otherwise manifest.

4 CONSISTENCY AND WORK COMPLEXITY

In this section, we analyze the asymptotic behavior of the proposed algorithm. Specifically, after establishing a result on consistency, we derive results that characterize the work complexity rate of Algorithm 1 as a function of the sample size schedule that is utilized. We start with a basic result on consistency of solution sets in sequential-SAA settings that is stated in generic form.

Lemma 1. *Suppose Assumption 1, Assumption 2 and Assumption 4 hold. Let the deterministic sequence of sample sizes m_ℓ satisfy, for some $\epsilon_0 > 0$,*

$$\lim_{\ell\to\infty} \frac{m_\ell}{(1+\epsilon_0)\log \ell} = \infty. \tag{9}$$

Then, the following assertions are true.

1. $\lim_{\ell\to\infty} \sup_{x\in\mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| = 0$ *a.s.*
2. *Let $\{\tilde{\epsilon}_\ell\}$ be a sequence of positive-valued random variables such that $\tilde{\epsilon}_\ell \to 0$ a.s. as $\ell \to \infty$. Then*
$$\mathrm{dist}(\mathcal{S}_{m_\ell}^*(\tilde{\epsilon}_\ell), \mathcal{S}_{m_\ell}^*) + \mathrm{dist}(\mathcal{S}_{m_\ell}^*, \mathcal{S}_{m_\ell}^*(\tilde{\epsilon}_\ell)) \to 0 \text{ a.s.};$$
3. *If $\mathbb{E}[\tilde{\epsilon}_\ell] \to 0$ as $\ell \to \infty$, then*
$$\mathbb{E}\big[\mathrm{dist}(\mathcal{S}_{m_\ell}^*(\tilde{\epsilon}_\ell), \mathcal{S}_{m_\ell}^*)\big] + \mathbb{E}\big[\mathrm{dist}(\mathcal{S}_{m_\ell}^*, \mathcal{S}_{m_\ell}^*(\tilde{\epsilon}_\ell))\big] \to 0.$$

Proof. Define event $A_\ell(\epsilon) := \{\omega : \sup_{x\in\mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| > \epsilon\}$ for fixed $\epsilon > 0$. From Assumption 4, we see that there exists s_0, ℓ_0 such that for $\ell \geq \ell_0$, $\mathbb{P}[A_\ell(\epsilon)] \leq \exp\{-m_\ell(r_1(\epsilon) + s_0)\}$. Since (9) holds and $r_1(\cdot)$ is positive-valued, we see that $\sum_{\ell=\ell_0}^\infty \mathbb{P}[A_\ell(\epsilon)] \leq \sum_{\ell=\ell_0}^\infty \exp\{-m_\ell(r_1(\epsilon)+s_0)\} < \infty$. Now invoke the Borel-Cantelli lemma [5] to conclude that the first assertion of the theorem holds.

To prove the second assertion, suppose $\mathrm{dist}(\mathcal{S}_{m_\ell}^*(\tilde{\epsilon}_\ell), \mathcal{S}_{m_\ell}^*) \to 0$ a.s. is not satisfied. Then there exists a set $\mathcal{U}$ of positive measure such that for each $\omega \in \mathcal{U}$, $\mathrm{dist}(\mathcal{S}_{m_\ell}^*(\tilde{\epsilon}_\ell), \mathcal{S}_{m_\ell}^*)$ does not converge to zero as $\ell \to \infty$. Hence, for each $\omega \in \mathcal{U}$, we can find a constant $\epsilon > 0$ and a sequence $\tilde{m}_\ell$ such that $\mathrm{dist}(\mathcal{S}_{\tilde{m}_\ell}^*(\tilde{\epsilon}_\ell), \mathcal{S}_{\tilde{m}_\ell}^*) \geq \epsilon$. Since the postulates of Theorem 5.3 in [48] are satisfied, we know that

$$\mathrm{dist}(\mathcal{S}_{\tilde{m}_\ell}^*, \mathcal{S}^*) + \mathrm{dist}(\mathcal{S}^*, \mathcal{S}_{\tilde{m}_\ell}^*) \to 0 \text{ as } \ell \to \infty. \tag{10}$$

Also, since $\mathrm{dist}(\mathcal{S}_{\tilde{m}_\ell}^*(\tilde{\epsilon}_\ell), \mathcal{S}_{\tilde{m}_\ell}^*) \geq \epsilon$ by assumption, and since the sets $\mathcal{S}_{\tilde{m}_\ell}^*(\tilde{\epsilon}_\ell), \mathcal{S}_{\tilde{m}_\ell}^*$ are both closed due to the continuity of $Q_{\tilde{m}_\ell}^\ell(\cdot)$ on $\mathcal{X}$, we see that we can find a sequence $x_\ell \in \mathcal{S}_{\tilde{m}_\ell}^*(\tilde{\epsilon}_\ell) \setminus \mathcal{S}^*$ such that

$$\|x_\ell - \Pi_{\mathcal{S}_{\tilde{m}_\ell}^*}(x_\ell)\| \geq \frac{\epsilon}{2}. \tag{11}$$

On the other hand, we know that since $x_\ell \in \mathcal{S}_{\tilde{m}_\ell}^*(\tilde{\epsilon}_\ell)$,

$$c^\top x_\ell + Q_{\tilde{m}_\ell}^\ell(x_\ell) - \Big(c^\top \Pi_{\mathcal{S}_{\tilde{m}_\ell}^*}(x_\ell) + Q_{\tilde{m}_\ell}^\ell(\Pi_{\mathcal{S}_{\tilde{m}_\ell}^*}(x_\ell))\Big) \leq \tilde{\varepsilon}_\ell. \tag{12}$$

After appropriate addition and subtraction to (12), we get

$$\begin{aligned} &\overbrace{Q_{\tilde{m}_\ell}^\ell(x_\ell) - q(x_\ell)}^{\text{I}} - \overbrace{\Big(Q_{\tilde{m}_\ell}^\ell(\Pi_{\mathcal{S}_{\tilde{m}_\ell}^*}(x_\ell)) - q(\Pi_{\mathcal{S}_{\tilde{m}_\ell}^*}(x_\ell))\Big)}^{\text{II}} + \overbrace{c^\top x_\ell + q(x_\ell) - c^\top \Pi_{\mathcal{S}^*}(x_\ell) - q(\Pi_{\mathcal{S}^*}(x_\ell))}^{\text{III}} \\ &\qquad \leq \underbrace{c^\top \Pi_{\mathcal{S}_{\tilde{m}_\ell}^*}(x_\ell) + q(\Pi_{\mathcal{S}_{\tilde{m}_\ell}^*}(x_\ell)) - c^\top \Pi_{\mathcal{S}^*}(x_\ell) - q(\Pi_{\mathcal{S}^*}(x_\ell))}_{\text{IV}} + \tilde{\epsilon}_\ell. \end{aligned} \tag{13}$$

The first assertion of this lemma ensures that the terms I and II become arbitrarily small for large ℓ, and the term IV also becomes arbitrarily small for large ℓ due to (10) and the continuity of the function $q(\cdot)$. Term III, however, is greater than or equal to $\gamma_0\epsilon/2$ for large ℓ due to the first-order growth (see Assumption 2). We have thus arrived at a contradiction, proving $\text{dist}(\mathcal{S}^*_{m_\ell}(\tilde{\epsilon}_\ell), \mathcal{S}^*_{m_\ell}) \to 0$ a.s.. Along identical lines, we can prove that $\text{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*_{m_\ell}(\tilde{\epsilon}_\ell)) \to 0$ a.s. as well.

We see that the sequence $\text{dist}(\mathcal{S}^*_{m_\ell}(\tilde{\epsilon}_\ell), \mathcal{S}^*_{m_\ell})$ is bounded since $\text{dist}(\mathcal{S}^*_{m_\ell}(\tilde{\epsilon}_\ell), \mathcal{S}^*_{m_\ell}) \leq \text{diam}(\mathcal{X}) < \infty$. Hence, the second assertion of the lemma also assures us that $\mathbb{E}[\text{dist}(\mathcal{S}^*_{m_\ell}(\tilde{\epsilon}_\ell), \mathcal{S}^*_{m_\ell})] \to 0$ as $\ell \to \infty$. By a similar argument, $\mathbb{E}[\text{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*_{m_\ell}(\tilde{\epsilon}_\ell))] \to 0$ as $\ell \to \infty$. The third assertion of the lemma thus holds. □

We now state the main consistency result associated with the iterates $\{\hat{x}^\ell\}$ generated by the sequential SAA Algorithm 1.

Theorem 1 (Consistency)**.** *Suppose Assumption 1, Assumption 2 and Assumption 4 hold. Let the sample size stipulation in* (9) *hold. Then the iterates $\{\hat{x}^\ell\}$ generated by Algorithm 1 satisfy the following.*

1. $\text{dist}(\hat{x}^\ell, \mathcal{S}^*) \to 0$ *a.s. as* $\ell \to \infty$.
2. $\mathbb{E}\big[\text{dist}(\hat{x}^\ell, \mathcal{S}^*)\big] \to 0$ *as* $\ell \to \infty$.

Proof. We write

$$\text{dist}(\hat{x}^\ell, \mathcal{S}^*) \leq \text{dist}(\hat{x}^\ell, \mathcal{S}^*_{m_\ell}(\epsilon_\ell)) + \text{dist}(\mathcal{S}^*_{m_\ell}(\epsilon_\ell), \mathcal{S}^*_{m_\ell}) + \text{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*). \tag{14}$$

Let's analyze the right-hand side of (14) term by term. The first term $\text{dist}(\hat{x}^\ell, \mathcal{S}^*_{m_\ell}(\epsilon_\ell)) = 0$ since $\hat{x}^\ell \in \mathcal{S}^*_{m_\ell}(\epsilon_\ell)$. Next, consider the second term $\text{dist}(\mathcal{S}^*_{m_\ell}(\epsilon_\ell), \mathcal{S}^*_{m_\ell})$. From Algorithm 1, we recall that

$$\epsilon_\ell := c \max\left\{\hat{\text{se}}_\ell, \frac{\delta}{\sqrt{m_\ell}}\right\} \text{ and } \hat{\text{se}}_\ell = \frac{\hat{\sigma}_\ell}{\sqrt{m_\ell}}, \tag{15}$$

where $\delta, c > 0$ are chosen deterministic constants, and

$$\hat{\sigma}_\ell := \sqrt{m_\ell^{-1} \sum_{i=1}^{m_\ell} \left(Q(\hat{x}^\ell, \xi_i^\ell) - m_\ell^{-1} \sum_{i=1}^{m_\ell} Q(\hat{x}^\ell, \xi_i^\ell) \right)^2}. \tag{16}$$

Using (15), (16), and Assumption 4, we see that $\hat{\text{se}}_\ell \to 0$ a.s.; now invoke Lemma 1 to see that $\text{dist}(\mathcal{S}^*_{m_\ell}(\epsilon_\ell), \mathcal{S}^*_{m_\ell}) \to 0$ a.s. and that $\mathbb{E}\big[\text{dist}(\mathcal{S}^*_{m_\ell}(\epsilon_\ell), \mathcal{S}^*_{m_\ell})\big] \to 0$ as $\ell \to \infty$.

Finally, consider the third term $\text{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)$ appearing on the right-hand side of (14). Notice that (i) the objective function $c^\top x + q(x), x \in \mathcal{X}$ is continuous; (ii) the feasible set $\mathcal{X}$ is compact; and (iii) from the first part of Lemma 1, the random function $Q^\ell_{m_\ell}(x)$ converges uniformly to the continuous function $q(x), x \in \mathcal{X}$ a.s. as $\ell \to \infty$. Therefore, the postulates of Theorem 5.3 in [48] hold, implying that $\text{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*) \to 0$ a.s. and that $\mathbb{E}\big[\text{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)\big] \to 0$ as $\ell \to \infty$.

We see from the above arguments that each of the three terms appearing on the right-hand side of (14) converges to zero *a.s.* and in expectation, implying that the assertions of the theorem hold. □

Theorem 1 guarantees that the sequence of iterates $\{\hat{x}^\ell\}$ generated by Algorithm 1 converges "into" the true solution set $\mathcal{S}^*$, that is, their distance from $\mathcal{S}^*$ converges to zero almost surely and in ℓ_1-norm. We will next provide complexity results that characterize the rate at which such convergence occurs, as a function of the total workload incurred through iteration L.

Recall the structure of the proposed sequential SAA algorithm: 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 "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 6. *Consider a deterministic optimization problem*

$$h^* := \min_{x \in \mathcal{X}} h(x), \tag{R}$$

where h is convex and piecewise linear and $\mathcal{X} := \left\{x \in \mathbb{R}^{n_1}_+ \mid Ax = b\right\}$ is a polyhedral constraint set. Let $\mathcal{X}^*$ denote the set of optimal solutions to problem (R). The Solver-$\mathcal{A}$ executed on problem (R) with an initial solution $x_0 \in \mathcal{X}$ exhibits iteration complexity $\left(\frac{\Lambda \operatorname{dist}(x_0, \mathcal{X}^*)}{\epsilon}\right)^2$ to obtain an ϵ-optimal solution, that is,

$$h(x_t) - h^* \leq \frac{\Lambda \operatorname{dist}(x_0, \mathcal{X}^*)}{\sqrt{t}}, t = 1, 2, \ldots,$$

where $\Lambda > 0$ is a universal constant, and x_t is the iterate returned by Solver-$\mathcal{A}$ after t iterations.

Assumption 6 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 6 is met. Furthermore, Assumption 6 has been stated for piecewise linear convex functions h because the objective function of the sample-path problem (P_ℓ) is piecewise linear convex.

A number of well-known subgradient algorithms are endowed with the iteration complexity stipulated through Assumption 6. For example, the standard subgradient descent having the iterative structure $x_{t+1} = x_t - \alpha_t \partial h(x_t), t = 0, 1, 2, \ldots$, when executed with constant step size $\alpha_t = \epsilon/\Lambda^2$ and $\|\partial h(x)\| \leq \Lambda,\ \forall x \in \mathcal{X}$ satisfies Assumption 6. (See, for example, [34]). Another recent example is a variant of the level bundle method [4] under an idealized assumption (we describe this variant in greater detail in Appendix A). In our numerical experiments presented in Section 6, we use an implementable variant of the level bundle method as Solver-$\mathcal{A}$, which is proposed in [51]. The following lemma is an obvious consequence of Assumption 6.

Lemma 2. *Let N_ℓ denote the number of iterations taken by Solver-$\mathcal{A}$ to solve problem (P_ℓ) to within optimality gap $\epsilon_\ell > 0$ starting at $\hat{x}^{\ell-1}$, that is,*

$$N_\ell := \inf\left\{\bar{t} : \left(c^\top \hat{x}^{\ell,t} + Q^\ell_{m_\ell}(\hat{x}^{\ell,t})\right) - z^*_{m_\ell} \leq \epsilon_\ell \text{ for all } t \geq \bar{t}, \quad \hat{x}^{\ell,0} := \hat{x}^{\ell-1}\right\}.$$

Then, there exists $\Lambda < \infty$ such that

$$\mathbb{P}\left[N_\ell > \Lambda \frac{\left(\operatorname{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_\ell}(\epsilon_\ell))\right)^2}{\epsilon_\ell^2} \mid \mathcal{F}_{\ell-1}\right] = 0 \text{ a.s.}$$

We next state a result which will be fundamental to the main result that is to follow.

Lemma 3. *Suppose Assumption 1, Assumption 2, Assumption 4 and Assumption 5 hold. Let $m_\ell \to \infty$ as $\ell \to \infty$. Then, there exists $\eta \in (0, \infty)$ such that for large enough ℓ,*

(i) $\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)\right)^2 \mid \mathcal{F}_{\ell-1}\right] \leq \frac{\eta}{m_\ell}$ *a.s.;*

(ii) $\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)\right)^2\right] \leq \frac{\eta}{m_\ell}$.

Proof. We prove only the first part; the second part follows from the first part trivially.

Let the set $\mathcal{X}^*_\nu := \{x_1, x_2, \ldots, x_{M_\nu}\} \subseteq \mathcal{X} \setminus \mathcal{S}^*(\epsilon), \nu > 0$ be a ν-net associated with $\mathcal{X} \setminus \mathcal{S}^*(\epsilon)$, that is, choose points $x_1, x_2, \ldots, x_{M_\nu}$ such that

$$\sup_{x \in \mathcal{X} \setminus \mathcal{S}^*(\epsilon)} \operatorname{dist}(x, \mathcal{X}^*_\nu) \leq \nu; \quad \nu < \frac{\epsilon}{3\|c\|}. \tag{17}$$

(It is clear that if $\epsilon > 0$ is small enough, then $\mathcal{X} \setminus \mathcal{S}^*(\epsilon)$ is non-empty, and hence there exists a set $\mathcal{X}^*_\nu$ satisfying the condition stipulated in (17). We avoid the triviality $\mathcal{X} = \mathcal{S}^*$.)

Let us first derive a bound on the probability $\mathbb{P}\left[\mathcal{S}^*_{m_\ell} \not\subset \mathcal{S}^*(\epsilon) \,|\, \mathcal{F}_{\ell-1}\right]$. We can write

$$\begin{aligned}
&\mathbb{P}\left[\mathcal{S}^*_{m_\ell} \not\subset \mathcal{S}^*(\epsilon) \,|\, \mathcal{F}_{\ell-1}\right] = \mathbb{P}\left[\mathcal{S}^*_{m_\ell} \bigcap (\mathcal{X} \setminus \mathcal{S}^*(\epsilon)) \neq \emptyset \,|\, \mathcal{F}_{\ell-1}\right] \\
&= \mathbb{P}\left[\left\{x \in \mathcal{X} \setminus \mathcal{S}^*(\epsilon) \mid c^\top x + Q^\ell_{m_\ell}(x) \leq c^\top y + Q^\ell_{m_\ell}(y), \forall y \in \mathcal{X}\right\} \neq \emptyset \,|\, \mathcal{F}_{\ell-1}\right] \\
&\leq \mathbb{P}\left[\left\{x \in \mathcal{X} \setminus \mathcal{S}^*(\epsilon) \mid c^\top x + Q^\ell_{m_\ell}(x) \leq c^\top x^* + Q^\ell_{m_\ell}(x^*)\right\} \neq \emptyset \,|\, \mathcal{F}_{\ell-1}\right].
\end{aligned} \tag{18}$$

Continuing from (18) after recalling from Assumption 5 that $L^\ell_{m_\ell} = m_\ell^{-1}\sum_{i=1}^{m_\ell} L(\xi^\ell_i)$ is the $\mathcal{F}_\ell$-measurable Lipschitz constant associated with the function $Q^\ell_{m_\ell}(x), x \in \mathcal{X}$ and recalling the notation $\zeta^\ell_{m_\ell}(x) := Q^\ell_{m_\ell}(x) - q(x)$, we get for large ℓ that

$$
\begin{aligned}
&\mathbb{P}\left[\mathcal{S}^*_{m_\ell} \not\subset \mathcal{S}^*(\epsilon) \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le \mathbb{P}\left[\left\{x \in \mathcal{X} \setminus \mathcal{S}^*(\epsilon) \mid c^\top \Pi_{\mathcal{X}^*_\nu}(x) + Q^\ell_{m_\ell}(\Pi_{\mathcal{X}^*_\nu}(x)) - (L^\ell_{m_\ell} + \|c\|)\nu \le c^\top x^* + Q^\ell_{m_\ell}(x^*)\right\} \ne \emptyset \,|\, \mathcal{F}_{\ell-1}\right] \\
&= \mathbb{P}\left[\bigcup_{x \in \mathcal{X}^*_\nu} \left\{c^\top x + Q^\ell_{m_\ell}(x) - (L^\ell_{m_\ell} + \|c\|)\nu \le c^\top x^* + Q^\ell_{m_\ell}(x^*)\right\} \ne \emptyset \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le \sum_{x \in \mathcal{X}^*_\nu} \mathbb{P}\left[Q^\ell_{m_\ell}(x) - (L^\ell_{m_\ell} + \|c\|)\nu \le c^\top x^* + Q^\ell_{m_\ell}(x^*) - c^\top x \,|\, \mathcal{F}_{\ell-1}\right] \\
&= \sum_{x \in \mathcal{X}^*_\nu} \mathbb{P}\left[\zeta_{m_\ell}(x) - \zeta_{m_\ell}(x^*) - (L^\ell_{m_\ell} + \|c\|)\nu \le c^\top x^* + q(x^*) - c^\top x - q(x) \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le \sum_{x \in \mathcal{X}^*_\nu} \mathbb{P}\left[\zeta_{m_\ell}(x) - \zeta_{m_\ell}(x^*) - (L^\ell_{m_\ell} + \|c\|)\nu \le -\epsilon \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le \sum_{x \in \mathcal{X}^*_\nu} \mathbb{P}\left[|\zeta_{m_\ell}(x)| \ge \frac{\epsilon}{3} \,|\, \mathcal{F}_{\ell-1}\right] + \mathbb{P}\left[|\zeta_{m_\ell}(x^*)| \ge \frac{\epsilon}{3} \,|\, \mathcal{F}_{\ell-1}\right] + \mathbb{P}\left[(L^\ell_{m_\ell} + \|c\|)\nu \ge \frac{\epsilon}{3} \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le |\mathcal{X}^*_\nu| \cdot \left(3\exp\left\{-m_\ell r_1(\epsilon/3)\right\} + 2\exp\left\{-m_\ell r_2\left(\epsilon/(3\nu) - \|c\|\right)\right\}\right),
\end{aligned}
$$

where the last inequality follows from the large-deviation bounds in Assumption 4 and Assumption 5. Also, since function $c^\top x + q(x)$ exhibits γ_0-first-order growth as assumed in Assumption 2, we can write for ϵ, ν chosen so that $\epsilon/(3\nu) > \|c\|$ and large enough ℓ,

$$
\begin{aligned}
\mathbb{P}\left[\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*) > 2\frac{\epsilon}{\gamma_0} \,|\, \mathcal{F}_{\ell-1}\right] &\le \mathbb{P}\left[\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*(\epsilon)) \ge \frac{\epsilon}{\gamma_0} \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le \mathbb{P}\left[\mathcal{S}^*_{m_\ell} \not\subset \mathcal{S}^*(\epsilon) \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le |\mathcal{X}^*_\nu| \cdot \left(3\exp\left\{-m_\ell r_1(\epsilon/3)\right\} + 2\exp\left\{-m_\ell r_2(\epsilon/(3\nu) - \|c\|)\right\}\right), \qquad (19)
\end{aligned}
$$

where the third inequality in (19) follows from (**??**).

Since $m_\ell \to \infty$ as $\ell \to \infty$, we see from (19) that for large enough ℓ,

$$
\begin{aligned}
\mathbb{P}\left[m_\ell \left(\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)\right)^2 > t \,|\, \mathcal{F}_{\ell-1}\right] &= \mathbb{P}\left[\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*) > \frac{\sqrt{t}}{\sqrt{m_\ell}} \,|\, \mathcal{F}_{\ell-1}\right] \\
&\le |\mathcal{X}^*_\nu| \cdot \left[3\exp\left\{-m_\ell\, r_1\left(\frac{\gamma_0}{6\sqrt{m_\ell}}\sqrt{t}\right)\right\} + 2\exp\left\{-m_\ell\, r_2\left(\frac{\gamma_0}{6\nu\sqrt{m_\ell}}\sqrt{t} - \|c\|\right)\right\}\right]. \qquad (20)
\end{aligned}
$$

Since $\mathbb{E}\left[m_\ell \left(\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)\right)^2 \,|\, \mathcal{F}_{\ell-1}\right] = \int_0^\infty \mathbb{P}\left[m_\ell \left(\operatorname{dist}(S_{m_\ell}, \mathcal{S}^*)\right)^2 > t \,|\, \mathcal{F}_{\ell-1}\right] dt$, and the functions $r_1(\cdot)$ and $r_2(\cdot)$ are convex and positive valued, we conclude from (20) that the assertion of the lemma is true. □

We are now ready to state the main work complexity result associated with the proposed Algorithm 1.

Theorem 2. *Suppose Assumptions 1–6 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. Suppose $\{m_\ell\}$ follows a geometric sequence such that $m_\ell/m_{\ell-1} = c_1,\ \forall \ell = 1, 2, \ldots,$ for $c_1 \in (1, \infty)$. Then, there exists $\kappa_0, \kappa_1 < \infty$ such that for large enough L,*

$$
\mathbb{E}[W_L] \le \kappa_0 + \frac{\kappa_1}{\mathbb{E}\left[\left(\operatorname{dist}(S^*_{m_L}, \mathcal{S}^*)\right)^2\right]}, \qquad (21)
$$

and

$$
\mathbb{E}[W_L] \le \kappa_0 + \frac{2\kappa_1\eta^{-1}(c(\delta^2 + \sigma^2) + \eta)}{\mathbb{E}\left[\left(\operatorname{dist}(\hat{x}^L, \mathcal{S}^*)\right)^2\right]}. \qquad (22)
$$

Proof. According to Lemma 2, for large enough ℓ a.s.,

$$\begin{aligned}
\mathbb{E}\left[\tilde{W}_\ell \,|\, \mathcal{F}_{\ell-1}\right] &\leq \mathbb{E}\left[\Lambda \frac{\left(\operatorname{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_\ell})\right)^2}{\epsilon_\ell^2} m_\ell \,|\, \mathcal{F}_{\ell-1}\right] \\
&\leq \mathbb{E}\left[\Lambda \frac{\left(\operatorname{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_\ell})\right)^2}{c^2\delta^2} m_\ell^2 \,|\, \mathcal{F}_{\ell-1}\right] \\
&\leq \Lambda \frac{m_\ell^2}{c^2\delta^2}\left[\left(\operatorname{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_{\ell-1}})\right)^2 + \left(\operatorname{dist}(\mathcal{S}^*_{m_{\ell-1}}, \mathcal{S}^*)\right)^2 + \mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*, \mathcal{S}^*_{m_\ell})\right)^2 \,|\, \mathcal{F}_{\ell-1}\right]\right] \qquad (23)
\end{aligned}$$

We know from Assumption 3 that for large enough ℓ,

$$\mathbb{E}\left[\left(\operatorname{dist}(\hat{x}^{\ell-1}, \mathcal{S}^*_{m_{\ell-1}})\right)^2\right] \leq 2\gamma^{-2}\epsilon_{\ell-1}^2. \qquad (24)$$

Also, recalling that

$$\hat{\sigma}_\ell^2 = m_\ell^{-1}\left(\sum_{i=1}^{m_\ell}\left(Q(\hat{x}^{\ell-1}, \xi_i^\ell) - Q_{m_\ell}(\hat{x}^{\ell-1})\right)^2\right),$$

we see from Assumption 4 that

$$\mathbb{E}\left[\hat{\sigma}_\ell^2 \,|\mathcal{F}_{\ell-1}\right] \leq \frac{m_\ell - 1}{m_\ell}\sigma^2. \qquad (25)$$

Hence,

$$\begin{aligned}
\mathbb{E}\left[\epsilon_{\ell-1}^2\right] = \mathbb{E}\left[\mathbb{E}\left[\epsilon_{\ell-1}^2 \,|\mathcal{F}_{\ell-2}\right]\right] &= \mathbb{E}\left[\frac{c}{m_{\ell-1}}\left(\int_{\hat{\sigma}^2_{\ell-1}\leq\delta^2}\delta^2 + \int_{\hat{\sigma}^2_{\ell-1}>\delta^2}\hat{\sigma}^2_{\ell-1}\right)\right] \\
&\leq \frac{c}{m_{\ell-1}}\left(\delta^2 + \sigma^2\right). \qquad (26)
\end{aligned}$$

Taking expectations in (23), using (24), (26), Lemma 3, and $m_\ell/m_{\ell-1} = c_1 > 1$, we see that there exists ℓ_0 such that for $\ell \geq \ell_0$, we have

$$\begin{aligned}
\mathbb{E}[\tilde{W}_\ell] &\leq \Lambda\frac{m_\ell^2}{c^2\delta^2}\left(\gamma^{-2}\frac{2cc_1}{m_\ell}(\delta^2+\sigma^2) + \frac{\eta c_1}{m_\ell} + \frac{\eta}{m_\ell}\right) \\
&\leq \Lambda\frac{m_\ell}{c^2\delta^2}\left(2\gamma^{-2}c\,c_1(\delta^2+\sigma^2) + \eta c_1 + \eta\right), \qquad (27)
\end{aligned}$$

and

$$\mathbb{E}[W_L] = \mathbb{E}\left[\sum_{\ell=1}^{L}\tilde{W}_\ell\right] \leq \sum_{\ell=1}^{\ell_0}\mathbb{E}\left[\tilde{W}_\ell\right] + \frac{\Lambda}{c^2\delta^2}\left(2\gamma^{-2}c\,c_1(\delta^2+\sigma^2) + \eta c_1 + \eta\right)\sum_{\ell=\ell_0}^{L} m_\ell. \qquad (28)$$

Since $m_\ell = m_1\, c_1^{\ell-1}$,

$$\sum_{\ell=\ell_0}^{L} m_\ell \leq \sum_{\ell=1}^{L} m_\ell = \sum_{\ell=1}^{L} m_1 c_1^{\ell-1} = m_1\frac{c_1^L - 1}{c_1 - 1} \leq \frac{c_1}{c_1-1}m_L \leq \frac{c_1}{c_1-1}\frac{\eta}{\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_L}, \mathcal{S}^*)\right)^2\right]},$$

where the last inequality follows from part (ii) of Lemma 3. Plugging the above into (28), we get:

$$\mathbb{E}[W_L] \leq \kappa_0 + \frac{\kappa_1}{\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_L}, \mathcal{S}^*)\right)^2\right]},$$

where $\kappa_0 := \sum_{\ell=1}^{\ell_0}\mathbb{E}\left[\tilde{W}_\ell\right]$ and $\kappa_1 := \eta\Lambda\,(c\delta)^{-2}\left(2\gamma^{-2}\,c\,c_1\,(\delta^2+\sigma^2) + \eta(c_1+1)\right)\,c_1\,(c_1-1)^{-1}$. This proves the assertion in (21).

To prove the assertion in (22), use (28) and

$$\begin{aligned}\mathbb{E}\left[\operatorname{dist}(\hat{x}^L, \mathcal{S}^*)^2\right] &\leq 2\left(\mathbb{E}\left[\operatorname{dist}(\hat{x}^L, \mathcal{S}^*_{m_L})^2\right] + \mathbb{E}\left[\operatorname{dist}(\mathcal{S}^*_{m_L}, \mathcal{S}^*)^2\right]\right) \\ &\leq \frac{2c(\delta^2+\sigma^2)}{m_L} + \frac{2\eta}{m_L},\end{aligned}$$

where the last inequality follows from (26) and part (ii) of Lemma 3. □

The following observations on Theorem 2 are noteworthy.

(a) The assertion in (22) of Theorem 2 should be seen as an analogue of the $\mathcal{O}(1/\epsilon^2)$ complexity result in non-smooth convex optimization that is known to be optimal [37].

(b) The complexity result in Theorem 2 has been stated in the general population context. So, the result equally applies for the finite-population scenario $|\Xi| < \infty$ considered routinely in machine learning contexts.

(c) The theorem assumes that the sample size schedule $\{m_\ell\}$ 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 $1 < c_0 \leq c_h < \infty$. As we describe in Section 6, such an algorithm with $c_0 = 1.05$ and $c_h = 3$ achieves the best performance with little to no tuning of required parameters.

The complexity result in Theorem 2 has been obtained assuming that the sample sizes increase geometrically, that is, $m_\ell/m_{\ell-1} = c_1 \in (1, \infty)$. 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 3. *Suppose Assumption 1, Assumption 2, Assumption 4 and Assumption 5 hold. Also, suppose that the inequality in Lemma 3 is tight, that is, there exists $\tilde{\eta}$ such that*

$$\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_\ell}, \mathcal{S}^*)\right)^2 \mid \mathcal{F}_{\ell-1}\right] \geq \frac{\tilde{\eta}}{m_\ell} \ \ a.s. \tag{29}$$

If the sample size schedule is polynomial, that is, $m_\ell = c_0\, \ell^p$ for some $p \in [1, \infty)$, then there exists $\kappa_2 < \infty$ such that

$$\mathbb{E}[W_L] \geq \kappa_2 \left(\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_L}, \mathcal{S}^*)\right)^2\right]\right)^{-1-\frac{1}{p}}.$$

Proof. We notice that the structure of the algorithm is such that each outer iteration consists of at least one inner iteration. So, during the ℓ-th outer iteration, the work $\tilde{W}_\ell$ satisfies $\tilde{W}_\ell \geq m_\ell$, implying that for $L \geq 2$,

$$\begin{aligned}\mathbb{E}[W_L] &\geq \sum_{\ell=1}^{L} m_\ell = \sum_{\ell=1}^{L} c_0\, \ell^p \geq c_0 \int_1^L (\ell-1)^p = \frac{c_0}{p+1}(L-1)^{p+1} \\ &\geq \frac{c_0^{-1/p} 2^{-p-1}}{p+1} m_L^{1+1/p} \\ &\geq \kappa_2 \left(\frac{1}{\mathbb{E}\left[\left(\operatorname{dist}(\mathcal{S}^*_{m_L}, \mathcal{S}^*)\right)^2\right]}\right)^{1+1/p},\end{aligned}$$

where $\kappa_2 := 2^{-p-1}\tilde{\eta}^{1+1/p}\, c_0^{-1/p}\, (p+1)^{-1}$ and the last inequality utilizes (29). □

We observe from Theorem 3 that no matter how large $p \in (0, \infty)$ is chosen when choosing a polynomial sample size schedule, the complexity (22) implied by the geometric sample size schedule is superior. A similar result has been proved recently by [42].

The condition in (29) 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 [45, 21]. In addition to this, if the random variables $\sqrt{m_\ell}\operatorname{dist}(S_{m_\ell}, \mathcal{S}^*)$ also exhibit uniform integrability, then the condition in (29) is guaranteed to hold.

5 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 1. Our intent in this section is to demonstrate that the iterates generated by Algorithm 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 [2]. We first describe a simple stopping procedure which is almost identical to what is called FSP in [2], and then argue that the stipulations laid out in [2] hold here, thereby allowing to invoke the main results of [2].

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 2, 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:

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

$$\tilde{s}_{n_\ell}^2(\hat{x}^\ell) = \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\left[Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell) - \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}[Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell)]\right]^2, \tag{30}$$

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 1.

Algorithm 2 An adaptive sequential SAA framework with a finite stopping criterion.

Input: Solver-$\mathcal{A}$, a sampling policy, a constant $c > 0$, and a constant $\delta > 0$. Set $\ell \leftarrow 0$.
while $\tilde{G}_{n_\ell}^\ell(\hat{x}^\ell) + z_\alpha \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \delta)}{\sqrt{n_\ell}} > \epsilon$ **do**
  Select the sample size m_ℓ for the current outer iteration ℓ and draw a random sample $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \ldots, \xi_{m_\ell}^\ell\}$.
  for $t = 1, 2, \cdots$ **do**
    Use Solver-$\mathcal{A}$, e.g., the adaptive partition-based level decomposition [51], to execute the t-th inner iteration for solving the sample-path problem (P_ℓ), obtain a candidate solution $\hat{x}^{\ell,t}$, and compute $G^{\ell,t}$ and $\hat{\text{se}}_{\ell,t}$ accordingly
    if $G^{\ell,t} \leq \epsilon_{\ell,t} := \max\left\{\hat{se}_{\ell,t}, \frac{\delta}{\sqrt{m_\ell}}\right\} \cdot c$ **then**
      Break the inner loop with a candidate solution $\hat{x}^\ell$.
    end if
  end for
  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 (30), respectively.
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 then

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

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

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

Algorithm 2 lists a terminating version of Algorithm 1 based on the proposed confidence interval.

The factor $\delta\, n_\ell^{-1/2}$ is a thresholding term that is common in sequential settings [12] and plays the same role as the term $h(n_k)$ in [2], ensuring that $L(\epsilon) \to \infty$ as $\epsilon \to 0$.

To analyze the behavior of the coverage probability obtained from Algorithm 2, the following three assumptions are made in [2].

(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) $\lim_{\ell\to\infty} \mathbb{P}\left[\sup_{x\in\mathcal{X}} \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell),\delta)}{\sqrt{n_\ell}} > \beta\right] = 0$ for any $\beta > 0$.

(We have omitted (A2) above to preserve the numbering in [2].) Theorem 2.1 in [47] implies that Assumption (A1) is satisfied if the support Ξ is finite, the true objective function $c^\top x + q(x)$ exhibits linear growth for $x \in \mathcal{X}$, $\sup_{x\in\mathcal{X}}\{c^\top x + q(x)\} < \infty$, and $\mathcal{S}^*$ is a singleton. Also, it is seen that Assumption (A3) and (A4) hold if the standing Assumption 4 holds. The following result characterizes the behavior of the iterates obtained from Algorithm 2, 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 [2].

Theorem 4. *Suppose Assumptions 1–5 hold. Furthermore, let $|\Xi| < \infty$ and let the solution set be a singleton, that is, $\mathcal{S}^* = \{x^*\}$. 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 $\mu(x) := c^\top x + q(x) - z^*$,*

$$\lim_{\epsilon\to 0} \mathbb{P}\left[\mu(\hat{x}^{L(\epsilon)}) \le \epsilon\right] = 1. \tag{31}$$

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 [2], we see that there exists $\epsilon_0 > 0$ such that for all $0 < \epsilon < \epsilon_0$,

$$\tilde{G}^{L(\epsilon)}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}) = 0; \quad \tilde{s}^2_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}) = 0, \tag{32}$$

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 (30) at stopping. According to the stopping criterion of Algorithm 2, we have that:

$$\begin{aligned}
\epsilon^2 n_{L(\epsilon)} &\ge \left(\sqrt{n_{L(\epsilon)}}\tilde{G}^{L(\epsilon)}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}) + z_\alpha \max(\tilde{s}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}),\delta)\right)^2;\\
\epsilon^2 n_{L(\epsilon)-1} &\le \left(\sqrt{n_{L(\epsilon)-1}}\tilde{G}^{L(\epsilon)-1}_{n_{L(\epsilon)-1}}(\hat{x}^{L(\epsilon)-1}) + z_\alpha \max(\tilde{s}_{n_{L(\epsilon)-1}}(\hat{x}^{L(\epsilon)-1}),\delta)\right)^2.
\end{aligned} \tag{33}$$

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

$$n_{L(\epsilon)-1} \ge \tilde{\beta}\, n_{L(\epsilon)} \text{ a.s.} \tag{34}$$

Using (34), (33), and (32), 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 (31) is stronger than classical guarantees in sequential testing such as those in [12]. 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 [12] 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 4 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 $\liminf_{\ell\to\infty} n_{\ell-1}/n_\ell > 0$.

6 COMPUTATIONAL EXPERIMENTS

In this section, we present computational results of the proposed adaptive sequential sampling framework for solving 2SLPs with fixed recourse. 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 [51], 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. We find that performing 20 replications produces standard error estimates small enough that we have are confident in our conclusions from the numerical study.

We run the adaptive sequential SAA framework according to Algorithm 2, 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 2 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 4), we report the coverage probability at stopping in the column titled "cov.", using results obtained from the 20 replications for each test instance.

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 [2] 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.

6.1 Implementation details

The following five algorithms are implemented in our computational study. The procedures described in (iii), (iv), and (v) use Algorithm 2 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 [3].

(i) **PILD-ODA**. This algorithm is the the adaptive partition-based level decomposition algorithm with on-demand accuracy as proposed in [51]. This algorithm is used to solve each instance with the full set of scenarios up to a relative optimality gap of 10^{-4}. Recall that z^* for each instance is obtained by this algorithm as well.

(ii) **Sequential-BP-L(Δ)**. This algorithm follows the sampling schedules in [3] 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 [22]. 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 (30), we solve the corresponding sample-path problem up to a relative optimality gap of 10^{-4}, as suggested by [3]. 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.

(iii) **Adaptive-seq-BP-L(Δ)**. This is Algorithm 2 implemented with the linearly increasing sample size schedule proposed in [3], 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 4 listed in the appendix. We use parameter $c = 0.1$ and safeguard parameter $\delta = 10^{-5}$ in defining the adaptive optimality tolerance ϵ_ℓ according to (7). 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 2 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 2 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 2.

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 $\tilde{G}^\ell_{n_\ell}(\hat{x}^\ell) + z_\alpha \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \delta)}{\sqrt{n_\ell}}$ encountered during all completed outer iterations ℓ, and accordingly 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. 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, Table 3, and Table 4.

6.2 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 the best available approach when solving 2SLPs using the full set of scenarios [51]. Table **??** 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 of algorithm instantiation on five finite-sample instances.

From Table **??**, 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

Instance	First-stage size	Second-stage size
DEAK40×20	(40,20)	(30,20)
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	(2,4)	(7,12)
gbd	(4,17)	(5,10)
ssn	(1,89)	(175,706)

Table 1: Profiles of test instances from [17] and [31]. 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.

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 **??** that the computational time for "Adaptive-seq-BP-L(Δ)" is lower in most cases, while the total number 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 2. Furthermore, a warm start functionality and an adaptive scenario aggregation technique are leveraged in "Adaptive-seq-BP-L(Δ)," by using Algorithm 4 and PILD-ODA [51], 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 2, 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 $c_0 = 1.05, c_h = 3$ (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 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 2), Algorithm 2 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, where utilizing a smaller c_1 ends up with a larger number of outer iterations and slightly more computational time.

Table 2: Computational results of the adaptive partition-based level decomposition approach [51] ("PILD-ODA"), the sequential sampling procedure by [3] ("Sequential-BP-L") on our test instances, and Algorithm 2 with the stopping criterion and sample size schedule proposed in [3] ("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(4,1070)	(0.1,97)	1.5	20(4,1094)	(0.1,97)
40x20	100K	101.8	18	5.1	13(4,1032)	(0.1,99)	1.3	19(4,1014)	(0.1,97)
	50K	74.6	12	4.3	19(2,584)	(0.0,83)	1.2	12(2,630)	(0.1,80)
40x40	100K	134.1	12	5.6	20(2,660)	(0.1,90)	1.3	13(2,676)	(0.1,82)
	50K	206.2	19	4.3	20(1,374)	(0.1,96)	1.7	21(1,396)	(0.1,100)
40x60	100K	413.1	20	4.1	20(1,360)	(0.1,99)	1.6	21(1,366)	(0.1,100)
	50K	114.4	56	86.1	41(12,2540)	(0.1,100)	18.5	64(12,2596)	(0.1,100)
60x20	100K	252.2	60	87.8	42(12,2584)	(0.1,100)	19.1	64(12,2636)	(0.1,100)
	50K	502.0	65	23.2	32(3,824)	(0.1,100)	12.3	70(3,834)	(0.1,100)
60x40	100K	929.4	67	25.1	33(3,864)	(0.1,100)	13.5	70(3,876)	(0.1,100)
	50K	333.8	24	5.9	22(1,414)	(0.1,100)	2.2	25(1,424)	(0.1,100)
60x60	100K	622.3	24	6.5	22(1,436)	(0.1,100)	2.3	25(1,436)	(0.1,100)
40x20	50K	63.9	17	18.6	27(8,1776)	(0.1,96)	4.4	23(7,1698)	(0.1,98)
High	100K	139.2	18	18.2	27(8,1772)	(0.1,96)	5.3	24(8,1854)	(0.1,95)
40x40	50K	58.9	9	4.5	17(2,580)	(0.0,83)	1.3	10(2,640)	(0.0,70)
High	100K	117.0	9	4.0	17(2,556)	(0.1,88)	1.3	10(2,646)	(0.1,80)
40x60	50K	711.5	25	60.8	42(5,1140)	(0.1,99)	22.4	29(5,1132)	(0.1,93)
High	100K	1520.0	24	55.5	41(5,1102)	(0.1,100)	20.8	29(5,1130)	(0.1,93)
60x20	50K	162.6	46	139.4	53(15,3194)	(0.1,99)	78.9	52(15,3280)	(0.1,100)
High	100K	263.2	43	132.9	54(15,3160)	(0.1,100)	73.5	52(15,3230)	(0.1,100)
60x40	50K	432.8	31	112.6	55(8,1824)	(0.1,99)	127.4	42(9,1920)	(0.1,98)
High	100K	958.5	32	124.0	56(8,1834)	(0.1,99)	122.4	42(9,1940)	(0.1,98)
60x60	50K	673.5	23	96.6	48(5,1290)	(0.1,100)	38.2	31(5,1282)	(0.1,90)
High	100K	1591.9	25	107.2	49(6,1316)	(0.1,96)	42.3	31(6,1362)	(0.1,89)

Table 3: Computational results of the adaptive partition-based level decomposition approach [51] - "PILD-ODA", Algorithm 2 with a fixed increasing rate "Adaptive-seq-fixed(1.5)," and Algorithm 2 with a dynamic increase rate "Adaptive-seq-dyn$(1.05, 3)$," 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(6,1377)	(0.1,96)	1.6	19(3,2892)	(0.1,100)
40x20	100K	101.8	18	1.5	21(6,1438)	(0.1,99)	1.6	19(3,2886)	(0.1,100)
	50K	74.6	12	1.2	13(3,568)	(0.1,71)	1.8	13(3,1662)	(0.0,75)
40x40	100K	134.1	12	1.2	14(3,595)	(0.1,72)	1.7	13(2,1489)	(0.0,75)
	50K	206.2	19	1.9	23(2,318)	(0.1,100)	1.9	22(2,454)	(0.1,100)
40x60	100K	413.1	20	1.9	23(2,308)	(0.1,100)	1.9	23(2,458)	(0.1,100)
	50K	114.4	56	10.7	60(8,3675)	(0.1,100)	9.2	56(4,6048)	(0.1,100)
60x20	100K	252.2	60	11.0	60(8,3673)	(0.1,100)	9.5	56(4,6264)	(0.1,100)
	50K	502.0	65	14.1	73(5,921)	(0.1,100)	13.8	69(3,1620)	(0.1,100)
60x40	100K	929.4	67	14.7	73(5,959)	(0.1,100)	13.4	68(3,1566)	(0.1,100)
	50K	333.8	24	2.7	28(3,374)	(0.1,100)	2.8	27(2,617)	(0.1,100)
60x60	100K	622.3	24	2.7	28(3,374)	(0.1,100)	2.7	27(2,580)	(0.1,100)
40x20	50K	63.9	17	4.4	23(8,3034)	(0.1,97)	4.0	19(4,5400)	(0.1,99)
High	100K	139.2	18	4.4	23(8,3013)	(0.1,95)	5.3	20(4,7066)	(0.0,98)
40x40	50K	58.9	9	1.3	11(3,617)	(0.0,69)	1.8	11(3,1485)	(0.0,65)
High	100K	117.0	9	1.3	11(3,601)	(0.0,61)	1.7	10(2,1366)	(0.0,65)
40x60	50K	711.5	25	24.6	31(6,1535)	(0.1,93)	28.1	27(3,3240)	(0.1,96)
High	100K	1520.0	24	22.0	31(6,1427)	(0.1,92)	27.0	27(3,3046)	(0.1,93)
60x20	50K	162.6	46	38.0	46(9,5558)	(0.1,100)	34.3	43(5,9720)	(0.1,100)
High	100K	263.2	43	42.4	46(9,6086)	(0.1,100)	33.1	43(5,9720)	(0.1,100)
60x40	50K	432.8	31	70.4	40(8,2866)	(0.1,99)	78.1	33(4,5706)	(0.1,99)
High	100K	958.5	32	78.6	40(8,2894)	(0.1,98)	75.3	33(4,5688)	(0.1,96)
60x60	50K	673.5	23	42.3	32(6,1878)	(0.1,92)	42.2	27(4,3831)	(0.1,94)
High	100K	1591.9	25	38.4	32(6,1808)	(0.1,85)	50.9	27(4,4078)	(0.1,89)

Table 4: Computational results of Algorithm 2 with the fixed-width stopping criterion and linear sample size schedule proposed in [3] with an increase of 10 scenarios per iteration ("Adaptive-seq-BP-L(10)"), Algorithm 2 with a geometrically increasing sample size schedule with rate $c_1 = 1.1$ ("Adaptive-seq-fixed(1.1)"), and Algorithm 2 with a geometrically increasing sample size schedule having a dynamic rate ("Adaptive-seq-dyn$(1.05, 3)$"), 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(22,551)	2.8	36(21,760)	1.7	21(5,2797)
40x20	100K	3.5	39(23,579)	2.7	35(20,721)	1.7	21(5,2711)
	50K	1.4	17(7,249)	1.4	19(8,250)	1.7	14(4,1319)
40x40	100K	1.3	17(6,239)	1.4	19(8,252)	1.5	14(4,1143)
	50K	2.5	27(5,204)	2.6	29(6,188)	2.3	25(3,421)
40x60	100K	2.1	26(4,186)	2.7	30(6,193)	2.1	25(3,369)
	50K	102.5	144(92,1945)	30.8	87(35,2760)	10.2	58(6,6383)
60x20	100K	103.8	143(92,1936)	31.1	87(35,2768)	10.5	58(6,6435)
	50K	47.9	92(23,560)	38.2	90(20,682)	15.5	73(5,1578)
60x40	100K	51.3	92(23,572)	37.1	88(20,665)	16.7	72(5,1733)
	50K	3.9	35(6,233)	4.4	38(8,235)	3.0	30(3,459)
60x60	100K	3.7	34(6,230)	4.1	37(8,222)	3.3	30(4,539)
40x20	50K	11.6	53(38,875)	7.9	42(27,1410)	4.2	21(6,5371)
High	100K	12.9	54(39,891)	9.5	44(29,1612)	5.1	21(6,6229)
40x40	50K	1.4	14(7,246)	1.4	15(8,231)	1.5	11(4,1030)
High	100K	1.5	14(7,251)	1.5	15(8,238)	1.4	11(4,956)
40x60	50K	263.4	77(29,683)	78.0	65(23,940)	32.8	30(5,3237)
High	100K	200.6	73(27,646)	68.3	63(22,859)	29.3	31(5,2904)
60x20	50K	337.7	128(93,1951)	101.6	73(37,3413)	34.5	45(7,9523)
High	100K	341.5	130(95,1988)	97.6	72(36,3271)	26.8	45(6,7979)
60x40	50K	2283.2	141(58,1271)	268.1	85(30,1758)	97.6	37(6,5817)
High	100K	2075.0	133(54,1196)	261.0	83(29,1710)	78.8	36(6,5363)
60x60	50K	742.6	88(34,793)	134.3	69(25,1106)	53.0	31(6,3987)
High	100K	621.0	82(31,735)	144.6	67(24,1052)	51.0	31(6,3593)

We observe, however, that the performance of option "Adaptive-seq-dyn(c_0, c_h)" is robust to initial value of C_1. This is expected because the sample size increase rate is adjusted depending on how many inner iterations were expended during the previous outer iteration, allowing us to conclude that the option "Adaptive-seq-dyn(c_0, c_h)" is the most preferable due to its efficiency and robustness.

Finally, we present the performance of the best adaptive sequential SAA options (according to the above experiments on DEAK and DEAK-H instances) on three more test sets: LandS, gbd, and ssn taken from [31].

Among the three additional sets of instances, instance ssn is notoriously challenging to solve because of the high variance of the underlying random variables. We see from Table 5 that both options "Adaptive-seq-BP-L(100)" and "Adaptive-seq-dyn$(1.05, 3)$" fail to provide confidence intervals with satisfactory width within the stipulated time limit. The second-stage problems of ssn are rather challenging to solve, making it hard to evaluate candidate solutions. We also see that the final sample size used in the sequential sampling procedures surpasses the total number of scenarios $(5K)$ used to define the instance. This is in contrast to

Table 5: Computational results of the adaptive partition-based level decomposition approach [51] ("PILD-ODA"), Algorithm 2 with the fixed-width stopping criterion and sample size schedule proposed in [3] ("Adaptive-seq-BP-L(100)") and Algorithm 2 with a dynamically chosen increase rate of sample size ("Adaptive-seq-dyn$(1.05, 3)$" on LandS, gbd and ssn instances.

Ins	N	PILD-ODA		Adaptive-seq-BP-L(100)			Adaptive-seq-dyn$(1.05, 3)$		
		Time	M	Time	$M(L, n_L)$	CI (cov.)	Time	$M(L, n_L)$	CI (cov.)
	50K	18.8	12	0.2	10(1,364)	(0.1,100)	0.2	11(2,426)	(0.1,100)
LandS	100K	35.6	12	0.2	10(1,366)	(0.1,100)	0.2	11(2,446)	(0.1,100)
	50K	37.5	32	0.5	24(2,602)	(0.0,89)	0.6	24(2,1104)	(0.0,93)
gbd	100K	75.9	29	0.5	24(2,582)	(0.0,94)	0.6	24(2,1137)	(0.0,93)
ssn	5K	6482.9	804	-	355(10,2044)	(7.3,94)	-	344(5,5897)	(4.6,92)

the DEAK instances. The somewhat negative performance on this challenging instance ssn suggests the use of variance reduction for sampling within the sequential SAA framework.

7 CONCLUDING REMARKS

We propose a sequential SAA framework 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 distance between the stochastic iterates generated by the proposed scheme and the true solution set converges to zero almost surely (and in ℓ_1-norm). 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. We believe that similarly efficient sequential SAA algorithms are possible for large-scale multi-stage 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.

References

[1] G. Bayraksan and D.P. Morton. Assessing solution quality in stochastic programs. *Mathematical Programming*, 108(2-3):495–514, 2006.

[2] G. Bayraksan and D.P. Morton. A sequential sampling procedure for stochastic programming. *Operations Research*, 59(4):898–913, 2011.

[3] G. Bayraksan and P. Pierre-Louis. Fixed-width sequential stopping rules for a class of stochastic programs. *SIAM Journal on Optimization*, 22(4):1518–1548, 2012.

[4] J. Y. Bello-Cruz and W. de Oliveria. Level bundle-like algorithms for convex optimization. *Journal of Global Optimization*, 59(4):787–809, 2014.

[5] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

[6] J. R. Birge. Current trends in stochastic programming computation and applications. Technical Report Report 95-15, Dept of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Ann Arbor, Michigan, 1995.

[7] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[8] Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Adaptive sampling strategies for stochastic optimization. *SIAM Journal on Optimization*, 28(4):3312–3343, 2018.

[9] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[10] 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(4):1667–1687, 2011.

[11] Michael Chen, Sanjay Mehrotra, and Dávid Papp. Scenario generation for stochastic optimization problems via the sparse grid method. *Computational Optimization and applications*, 62(3):669–692, 2015.

[12] Y. S. Chow and H. E. Robbins. On the asymptotic theory of fixed-width confidence intervals for the mean. *Annals of Mathematical Statistics*, 36:457–462, 1965.

[13] K. Cooper, S. R. Hunter, and K. Nagaraj. Bi-objective simulation optimization on integer lattices using the epsilon-constraint method in a retrospective approximation framework. *Under review for INFORMS Journal on Computing*, 2018.

[14] George B Dantzig and Albert Madansky. On the solution of two-stage linear programs under uncertainty. In *Proceedings of the fourth berkeley symposium on mathematical statistics and probability*, volume 1, pages 165–176. University of California Press Berkeley, 1961.

[15] T. Homem de Mello and G. Bayraksan. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85, 2014.

[16] W. de Oliveira and C. Sagastizábal. Level bundle methods for oracles with on demand accuracy. *Optimization Methods and Software*, 29(6):1180–1209, 2014.

[17] I. Deák. Testing successive regression approximations by large-scale two-stage problems. *Annals of Operations Research*, 186(1):83–99, 2011.

[18] Amir Dembo and Ofer Zeitouni. Large deviations techniques and applications. corrected reprint of the second (1998) edition. stochastic modelling and applied probability, 38, 2010.

[19] G. Deng and M. C. Ferris. Variable-number sample-path optimization. *Mathematical Programming*, (117):81–109, 2009.

[20] S. Drew and T. Homem-de-Mello. Some large deviations results for latin hypercube sampling. *Methodol Comput Appl Probab*, 14:203–232, 2012.

[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*, 16:1517–1549, 1988.

[22] Csaba I Fábián and Zoltán Szőke. Solving two-stage stochastic programming problems with level decomposition. *Computational Management Science*, 4(4):313–353, 2007.

[23] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.

[24] F. Hashemi, S. Ghosh, and R. Pasupathy. On adaptive sampling rules for stochastic recursions. In A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, editors, *Proceedings of the 2014 Winter Simulation Conference*, Piscataway, NJ, 2014. Institute of Electrical and Electronics Engineers, Inc.

[25] J. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage stochastic linear programs with recourse. *Mathematics of Operations Research*, 16:650–669, 1991.

[26] J.L. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of operations research*, 16(3):650–669, 1991.

[27] T. Homem-de-Mello. Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):108–133, 2003.

[28] T. Homem-de-Mello and G. Bayraksan. Monte carlo sampling-based methods for stochastic optimization. In *Reviews*, Surveys in Operations Research and Management Science, pages 56–85. Elsevier, 2014.

[29] S. Kim, R. Pasupathy, and S. G. Henderson. A guide to SAA. In M. Fu, editor, *Encyclopedia of Operations Research and Management Science*, Hillier and Lieberman OR Series. Elsevier, 2014.

[30] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Mathematical programming*, 69(1-3):111–147, 1995.

[31] J. Linderoth, A. Shapiro, and S. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142:215–241, 2006.

[32] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19:674–699, 2008.

[33] Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[34] A. Nedić and D. Bertsekas. Convergence rate of incremental subgradient algorithms. In *Stochastic optimization: algorithms and applications*, pages 223–264. Springer, 2001.

[35] B. L. Nelson. *Foundations and Methods of Stochastic Simulation: A First Course*. Springer, New York, NY., 2013.

[36] Barry L Nelson. Antithetic-variate splitting for steady-sate simulations. *European journal of operational research*, 36(3):360–370, 1988.

[37] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[38] R. Pasupathy. On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. *Operations Research*, 58(4-part-1):889–901, 2010.

[39] R. Pasupathy, P. W. Glynn, S. Ghosh, and F. Hashemi. On sampling rates in simulation-based recursions. *SIAM Journal on Optimization*, 28(1):45–73, 2018.

[40] R. Pasupathy and B. W. Schmeiser. Retrospective-approximation algorithms for multidimensional stochastic root-finding problems. *ACM TOMACS*, 19(2):5:1–5:36, 2009.

[41] E. Polak and J. Royset. Efficient sample sizes in stochastic nonlinear programming. *Journal of Computational and Applied Mathematics*, 217(2):301–310, 2008.

[42] J. Royset and R. Szechtman. Optimal budget allocation for sample average approximation. *Operations Research*, 61(3):762–776, 2013. Under Review.

[43] J.O. Royset. On sample size control in sample average approximations for solving smooth stochastic programs. *Computational Optimization and Applications*, 55(2):265–309, 2013.

[44] A. Ruszczynski and A. Shapiro, editors. *Stochastic Programming. Handbook in Operations Research and Management Science*. Elsevier, New York, NY., 2003.

[45] A. Shapiro. Asymptotic behavior of optimal solutions in stochastic programming. *Mathematics of Operations Research*, 18(4):829–845, 1993.

[46] A. Shapiro and T. Homem-de-Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81(3):301–325, 1998.

[47] 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(1):70–86, 2000.

[48] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[49] 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(4):3145–3176, 2018.

[50] Rebecca Stockbridge and Güzin Bayraksan. Variance reduction in monte carlo sampling-based optimality gap estimators for two-stage stochastic linear programming. *Computational Optimization and Applications*, 64(2):407–431, 2016.

[51] 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(1):57–70, 2018.

[52] 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(3), 2013.

[53] C. Wolf, C.I. 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(2):437–448, 2014.

[54] G. Zhao. A log-barrier method with Bender's decomposition for solving two-stage stochastic linear programs. *Mathematical Programming, Series A*, 90:501–536, 2001.

Appendix A A conceptual level method for solving convex non-smooth optimization problem

Consider a generic convex non-smooth optimization problem:

$$h^* := min_{x\in\mathcal{X}} h(x), \tag{35}$$

where $\mathcal{X}$ is a nonempty convex and closed set in $\mathbb{R}^n$, and h is a convex and possibly non-smooth function. We consider the conceptual algorithm proposed in [4][Algorithm 1], which is summarized in Algorithm 3. The key assumption is the availability of finding a level target h_t^{lev} for each iteration t that is lower-bounded by h^*.

Assumption 7. *(See also equations (4) and (5) in [4]) For each iteration t, there exists a level target parameter h_t^{lev} such that*

$$h(x_t) > h_t^{lev} \geq h^*, \ \forall t \tag{36a}$$

$$\lim_{t\to\infty} h_t^{lev} = h^*. \tag{36b}$$

For example, the requirement in Assumption 7 holds, if the optimal objective value h^* is known. In this case h_t^{lev} can be simply selected as: $h_t^{lev} = (1-\lambda)h(x_t) + \lambda h^*$ for some $\lambda \in (0,1)$.

Algorithm 3 A conceptual level bundle algorithm for non-smooth convex optimization [4][Algorithm 1].

Input: Given $x_0 \in \mathcal{X}$, obtain $h(x_0)$ and $g_0 \in \partial h(x_0)$.
Set $\hat{x} = x_0, t = 0$.
while stopping criterion is not met **do**
  Select h_t^{lev} satisfying (36).
  Compute the next iterate:

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \{\|x-\hat{x}\|^2 \mid (x-x_t)^\top(\hat{x}-x_t) \leq 0, g_j^\top(x-x_j) + h(x_j) \leq h_t^{lev}, \ \forall j < t\}$$

  if stopping criterion is met **then**
    Terminate the procedure.
  else
    Compute $h(x_{t+1})$ and $g_{t+1} \in \partial h(x_{t+1})$.
    Set $t \leftarrow t+1$.
  end if
end while

Lemma 4. *Let x_* be an optimal solution to (35), given an initial iterate $\hat{x}$, Algorithm 3 will run at most $\mathcal{O}\left(\frac{\Lambda\|x_*-\hat{x}\|}{\epsilon}\right)^2$ iterations before stopping with a stopping tolerance parameter ϵ, where Λ is the Lipschitz constant on the objective function of (35).*

Proof. Let $j \in \mathbb{Z}^+$ be the iteration index. Based on (36) in Assumption 7, according to Algorithm 3:

$$0 \geq (x_{j+1}-\hat{x})^\top(\hat{x}-x_j) = \frac{1}{2}(\|x_{j+1}-x_j\|^2 - \|x_{j+1}-\hat{x}\|^2 + \|x_j-\hat{x}\|^2), \tag{37}$$

and thus

$$\|x_{j+1}-\hat{x}\|^2 \geq \|x_j-\hat{x}\|^2 + \|x_{j+1}-x_j\|^2. \tag{38}$$

Then according to the definition of the level set (see Algorithm 3), we get:

$$g_j^\top(x_{j+1}-x_j) + h(x_j) \leq h_j^{lev},$$

so that

$$g_j^\top(x_j-x_{j+1}) \geq h(x_j) - h_j^{lev}.$$

Note that $g_j^\top(x_j - x_{j+1}) \leq \|g_j\|\|x_j - x_{j+1}\|$, we have:

$$\|x_{j+1} - x_j\|^2 \geq \frac{(h(x_j) - h_j^{lev})^2}{\|g_j\|^2}. \tag{39}$$

Combining (38) and (39), we get:

$$\|x_{j+1} - \hat{x}\|^2 \geq \|x_j - \hat{x}\|^2 + \frac{(h(x_j) - h_j^{lev})^2}{\|g_j\|^2} \geq \|x_j - \hat{x}\|^2 + \frac{(h(x_j) - h_j^{lev})^2}{\Lambda^2},$$

where Λ is the Lipchitz constant of $f(\cdot)$. Rewriting this inequality as:

$$\|x_{j+1} - \hat{x}\|^2 - \|x_j - \hat{x}\|^2 \geq \frac{(h(x_j) - h_j^{lev})^2}{\Lambda^2},$$

and for any $t \in \mathbb{Z}^+$, summing it over $j = 1, \ldots, t$, we get:

$$\|x_{t+1} - \hat{x}\|^2 \geq \frac{1}{\Lambda^2}\sum_{j=0}^{t}(h(x_j) - h_j^{lev})^2.$$

[4][Theorem 3.4] (Equation (17)) then yields:

$$\|x_* - \hat{x}\|^2 \geq \|x_{t+1} - \hat{x}\|^2 \geq \frac{1}{\Lambda^2}\sum_{j=0}^{t}(h(x_j) - h_j^{lev})^2.$$

Now according to the definition that

$$h_t^{lev} = \alpha\underline{h}_t + (1-\alpha)\bar{h}_t, \text{ with } \alpha \in (0,1] \text{ and } \bar{h}_t = \min_{j\leq t}\{h(x_j)\}.$$

Note that $h_t^{lev} = \bar{h}_t - \alpha\Delta_t$, where $\Delta_t = \bar{h}_t - \underline{h}_t$ is the estimated optimality gap. Suppose the algorithm does not stop at iteration t, i.e., $\Delta_t > \epsilon$, plugging in the definition of h_t^{lev} into the above inequality, and we get:

$$\begin{aligned}
\|x_* - \hat{x}\|^2 &\geq \frac{1}{\Lambda^2}\sum_{j=0}^{t}(h(x_j) - h_j^{lev})^2 \geq \frac{1}{\Lambda^2}\sum_{j=0}^{t}(\bar{h}_j - h_j^{lev})^2 \\
&= \frac{\alpha^2}{\Lambda^2}\sum_{j=0}^{t}\Delta_j^2 > \frac{\alpha^2}{\Lambda^2}\sum_{j=0}^{t}\epsilon^2 \\
&= (t+1)\left(\frac{\alpha\epsilon}{\Lambda}\right)^2,
\end{aligned}$$

i.e.,

$$t + 1 < \left(\frac{\Lambda\|x_* - \hat{x}\|}{\alpha\epsilon}\right)^2.$$

So the algorithm will run at most $\left(\frac{\Lambda\|x_* - \hat{x}\|}{\alpha\epsilon}\right)^2$ iterations before stopping with $\Delta_t \leq \epsilon$. □

Appendix B The adaptive partition-based level decomposition with on-demand accuracy and its warmstarting procedure

Since Algorithm 3 is only a conceptual algorithm (under Assumption 36), for our numerical experiments, we use a variant of level bundle method that is practically implementation, the adaptive partition-based level decomposition method, recently developed by [51]. For details about this algorithm, please refer to [51]. The performance of this algorithm has shown to be more competitive than most of the state-of-the-art algorithms for solving 2SLPs with a finite sample. We use this algorithm as the Solver-$\mathcal{A}$ for solving the

sample-path problem (P_ℓ) in each outer iteration ℓ during our adaptive sequential SAA procedure. We argue that this algorithm, denoted as PILD-ODA, together with the following warmstart procedure, approximates the behavior of the conceptual algorithm (Algorithm 3) quite well.

We next illustrate how information from solving the sample-path problem (P_ℓ) with sample $\mathcal{M}_\ell$ at previous outer iterations $1, 2 \ldots, \ell$ can be leveraged to solve the sample-path problem $(P_{\ell+1})$ with sample $\mathcal{M}_{\ell+1}$. We note that this information is not necessarily only about the solution $\hat{x}^\ell$, and could contain, e.g., dual multipliers collected from solving second-stage subproblems (2) during the inner iterations of Algorithm PILD-ODA, which is possible precisely because of the following key assumption of fixed recourse.

Assumption 8. *The 2SLP problem* (P) *has* fixed recourse*, that is, the second-stage cost vector* $d \in \mathbb{R}^{n_2}$ *and recourse matrix* $W \in \mathbb{R}^{m_2 \times n_2}$ *are not subject to uncertainty. Consequently, the dual polyhedron of the second-stage problem* (2) *is identical for all realization of random variables involved in problem* (P)*:*

$$\Lambda := \{\lambda \mid W^\top \lambda \leq d\}.$$

The fixed recourse property has been exploited in various efficient methods to solve 2SLPs, such as [26, 16, 53, 51].

At the ℓ-th outer iteration, let $\Lambda^\ell \subseteq \Lambda$ be the set of dual multipliers accumulated so far (optimal dual multipliers to the second-stage subproblems (2) encountered during the solution procedure), the master problem to the sample-path problem (P_ℓ) with respect to sample $\mathcal{M}_\ell$ can be initialized as:

$$\min_{x \in \mathcal{X}} c^\top x + \frac{1}{m_\ell}\theta \tag{40a}$$

$$\text{s.t. } \theta \geq \sum_{i=1}^{m_\ell} \lambda_i^\top (h(\xi_i) - T(\xi_i)x), \ \forall (\lambda_i)_{i=1}^{m_\ell} \in \overbrace{\Lambda^\ell \times \Lambda^\ell \times \cdots \Lambda^\ell}^{m_\ell \text{ times}} \tag{40b}$$

When $|\Lambda^\ell|$ gets large, the number of constraints (40b) may get excessively large. Therefore, one could solve (40) using a cutting plane algorithm by adding them on the fly.

Algorithm 4 Initialization of the sample-path problem (P_ℓ) with sample $\mathcal{M}_\ell$ by using a collection of dual multipliers Λ^ℓ.

Input: A starting solution $\hat{x}^0 \in \mathcal{X}$, a collection of dual multipliers Λ^ℓ. Start the master problem (40) with none of constraints (40b).
for each $\xi_i^\ell,\ i = 1, 2, \ldots, m_\ell$ **do**
  Calculate $\lambda_i \in \arg\max_{\lambda \in \Lambda^\ell} \lambda^\top (h(\xi_i^\ell) - T(\xi_i^\ell)\hat{x}^0)$
end for
Add constraint (40b) with $\{\lambda_i\}_{i=1}^{m_\ell}$.
while $loopflag = 1$ **do**
  Solve the master problem, obtain an optimal objective value $\hat{z}$ and the corresponding optimal solution $(\hat{\theta}, \hat{x})$.
  for each $\xi_i^\ell,\ i = 1, 2, \ldots, m_\ell$ **do**
    Compute $\lambda_i \in \arg\max_{\lambda \in \Lambda^\ell} \lambda^\top (h(\xi_i^\ell) - T(\xi_i^\ell)\hat{x})$.
  end for
  Arrange constraint (40b) with $\{\lambda_i\}_{i=1}^{m_\ell}$.
  if the arranged constraint is violated by $(\hat{\theta}, \hat{x})$ **then**
    Add it to the master problem.
  else
    $loopflag = 0$.
  end if
end while
Return $\hat{x}$ and $\hat{z}$.

We implement this warm start procedure (Algorithm 4) prior to executing Algorithm PILD-ODA to solve the sample-path problem (P_ℓ), obtain $\hat{x}$ and $\hat{z}$ and initialize Algorithm PILD-ODA by setting the starting

solution to be $x^0 = \hat{x}$. We keep collecting optimal dual solutions λ's corresponding to the second-stage subproblems solved throughout the process of Algorithm 2 into Λ^ℓ. To avoid $|\Lambda^\ell|$ to get too large, we disgard a dual vector if the euclidean distance between this vector and an existing vector in Λ^ℓ is lower than a given threshold. We note that we could also choose to use the set of samples from the previous iteration $\mathcal{M}_\ell$ as a part of the new set of samples $\mathcal{M}_{\ell+1}$, at least periodically, in which case more warm starts are expected [2, 3].

Remark The effort of generating constraints (40b) is much less than that of a standard Benders cut. Indeed, each iteration of Algorithm 4 only involves m_ℓ vector-matrix multiplications (one for each scenario $\xi_i^\ell, i = 1, 2, \ldots, m_\ell$); while generating a standard Benders cut involves solving m_ℓ second-stage linear programs (2). One could further consider constraints (40b) as a special type of coarse cuts, and integrate Algorithm 4 within Algorithm PILD-ODA. We also remark that the warm start effect is not only captured by the iterates $\hat{x}^\ell$ produced after each outer iteration ℓ. In fact, the majority of the warm start effect comes from "reconstructing" a good cutting-plane approximation to $Q^\ell_{m_\ell}(x)$ using existing collection of dual vectors Λ^ℓ. Unfortunately, the warm start effect brought by keeping track of the dual multipliers is very challenging to characterize theoretically.