

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

ADAPTIVE SEQUENTIAL SAA FOR SOLVING TWO-STAGE STOCHASTIC LINEAR PROGRAMS

RAGHU PASUPATHY* AND YONGJIA SONG†

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

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

21 **AMS subject classifications.** 90C15, 90C06

22 **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 $Q(x, \xi_i)$, $i = 1, 2, \dots, n$ 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” ξ_i , $i = 1, 2, \dots, n$ that may or may not be independent.

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

- 35 (i) generate an implicit approximation of the objective function using a specified number of “scenarios” $\xi_1, \xi_2, \dots, \xi_n$ obtained, e.g., using Monte Carlo sampling;
- 37 (ii) replace the 2SLP by a sample-path optimization problem [35, 60] having the objective function obtained in (i) and having the known constraint set \mathcal{X} , and solve it using one of a variety of decomposition approaches that have been proposed in the literature, e.g., [1, 51, 76].

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

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

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

43 (i) and (ii) has been extensively analyzed in settings that are much more general than just 2SLPs.
44 For example, results on the consistency and rates of convergence of optimal values/solutions, large
45 and small sample properties, and other special properties are now available through standard text-
46 books [67] and surveys [34, 35].

47 It is important to note that SAA is a paradigm and not an algorithm in that important com-
48 ponents within the SAA framework still need to be chosen before implementation can occur. To
49 implement the SAA paradigm as stated in (i) and (ii), a practitioner needs to select a sample size
50 and a Monte Carlo generation mechanism in (i), and an appropriate solver/stopping-mechanism in
51 (ii). For instance, the question of sample size choice for generating the sample-path problem in (i)
52 has sometimes been a vexing issue, with practitioners often making this choice through trial and
53 error, using minimum sample size bounds that have been noted to be conservative [35, 41, 63], and
54 more recently, using multiple sample sizes and solving multiple sample-path problems.

55 A premise of this paper is that SAA’s effective implementation depends crucially on the disci-
56 plined customization (to narrowly defined problem classes, e.g., 2SLPs) of choices internal to SAA.
57 Such customization involves answering specific algorithmic questions that arise during implemen-
58 tation. For instance:

- 59 (a) Is it best to generate and solve (to machine precision) a single sample-path problem with a
60 large Monte Carlo sample size or is it better to progressively and roughly solve a sequence
61 of sample-path problems generated with increasing sample size? If the latter strategy is
62 better, what schedule of sample sizes should be used?
- 63 (b) Recognizing that any generated sample-path problem suffers from sampling error and hence
64 suggests not solving to machine precision, to what extent should a sample-path problem
65 be solved?
- 66 (c) What type of solvers should be used in solving the generated sample-path problems, given
67 that the solution information to previously solved sample-path problem(s) can be fruitfully
68 used as a *warm start* to a subsequent sample-path problem?

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

72 **1.1. Summary and Insight on Main Results.** The essence of our proposed framework is
73 the construction of a sequential SAA framework for solving 2SLPs, where a sequence of approximate
74 2SLPs are generated and solved to progressively increasing precision across iterations. The frame-
75 work is such that the early iterates are obtained with little computational burden since, by design,
76 the generated sample-path problems tend to have small sample sizes and are solved imprecisely; and
77 the later iterates can be expected to be obtained with ease as well since they tend to benefit from
78 the warm starts using solution information obtained in previous iterations. The schedule of sample
79 sizes and the adaptive optimality-tolerance parameters are chosen to be in lock-step, ensuring that
80 no particular sample-path problem is “over-solved.” The framework we provide is an algorithm in
81 the strict sense of the word in that we make specific recommendations for choosing: (i) the schedule
82 of sample sizes to generate the sample-path problems to approximate the 2SLP, (ii) the schedule
83 of error-tolerance parameters to which each of the generated sample-path problems is to be solved,
84 and (iii) the solver to use when solving the sample-path problems. We also demonstrate that our
85 framework can exploit existing results on finite-time stopping to provide solutions with probabilistic
86 guarantees on optimality. Our extensive numerical experience on solving large-scale 2SLPs suggests
87 that the proposed algorithm yields competitive computational performance compared with existing
88 methods.

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

98 We emphasize that the framework we propose is general in that it allows for the use of a wide
99 range of dependent sampling, e.g., Latin-hypercube sampling (LHS) [45], antithetic variates [47],
100 and randomized quasi-Monte Carlo [27, 42] *within* a generated sample-path problem, and the reuse
101 of scenarios *across* generated sample-path problems. While we do not attempt to demonstrate that
102 the use of such variance reduction measures is better than iid sampling, other reports [15, 72] in
103 the literature suggest the fruitfulness of such variance reduction techniques.

104 **1.2. Related Literature.** 2SLPs have been the subject of investigation for a long time [8] and
105 algorithms to solve 2SLPs can be conveniently classified based on whether or not the probability
106 space underlying the 2SLP is endowed with a sample space having a finite number of outcomes.
107 As noted in [77], an enormous amount of work has been generated especially for the context where
108 the sample space is finite, resulting in various algorithm classes that directly exploit the finite sum
109 structure — see [8] and [14] for entry points into this substantial literature.

110 For 2SLPs with sample spaces having countably infinite or an uncountable number of outcomes,
111 or for that matter even sample spaces with large cardinality, Monte Carlo sampling approaches
112 appear to be a viable alternative [67, 68, 69]. In fact, sequential Monte Carlo sampling methods
113 such as what we propose here are not new and have appeared in the stochastic programming (SP)
114 and simulation optimization (SO) literature for several decades now [53, 19, 23, 31, 33, 34, 68, 74].
115 For instance, [23] proposes the stochastic quasi-gradient methods for optimization of discrete event
116 systems, [68] suggests the idea of solving a sequence of sample-path problems with increasing sample
117 sizes as a practical matter, and [33] gives various sufficient conditions on how fast the sample size
118 should grow in order to ensure the consistency of the SAA estimator with varying sample sizes.
119 For SPs where the corresponding sample-path problems are smooth optimization problems, [57, 61]
120 study the sample size selection problem for the sequential sampling procedure. They model the
121 sequential sampling procedure as a stochastic adaptive control problem, by finding the optimal
122 sample size *as well as* the number of iterations that one should apply to solve the sampled problems,
123 so that the total expected computational effort expended in the entire procedure is minimized. A
124 surrogate model is then proposed to approximate this adaptive control model so that the sample
125 size and the number of iterations to be employed at each iteration can be found (relatively) easily
126 according to results from previous iterations, by solving the surrogate model. From an algorithmic
127 perspective, the stochastic decomposition framework initially developed by [31] is perhaps the most
128 well-known practical approach that exploits the connections between statistical inference, sampling,
129 and stochastic LPs. In addition, [28] proposes simulation-based Benders decomposition approach
130 as a variant of the stochastic sub-gradient method specifically for 2SLPs and develops statistical
131 confidence bounds for the optimal values.

132 Similar to [33], [53, 55] suggest *retrospective approximation* (RA) where a smooth stochastic op-
133 timization problem is solved through a sequence of sample-path problems generated with increasing
134 sample sizes. Unlike in [33], RA methods solve the sample-path problems imprecisely, until a gen-

135 erally specified error-tolerance parameter is satisfied. The methods presented here can be thought
 136 to be *adaptive* RA in that the error-tolerance sequence in our current framework is adaptive since it
 137 depends explicitly on a measure of sampling variability. We find that such adaptivity is crucial for
 138 good numerical performance, although it brings additional technical difficulty due to the need to
 139 handle stopping time random variables. Also, whereas the methods in [53, 57, 61] do not apply to
 140 non-smooth problems such as 2SLPs, the methods we present here are tailored (through the choice
 141 of solver) to exploit the structure inherent to 2SLPs. We note in passing that adaptive sampling as
 142 a strategy to enhance efficiency of stochastic optimization algorithms has recently gained popularity
 143 — see, for example, [10, 11, 29, 54, 70].

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

154 **1.3. Organization of the Paper.** The rest of the paper is organized as follows: Section 2
 155 presents important notation, convention, and terminology used throughout the paper, a precise
 156 problem statement of 2SLP, and a listing of key assumptions. Section 3 introduces the proposed
 157 adaptive sequential SAA framework. Section 4 presents various results pertaining to consistency,
 158 work complexity rates, and optimal sample size schedules. Section 6 provides a finite stopping rule
 159 for the adaptive sequential SAA algorithm by incorporating the sequential sampling approaches
 160 proposed in [4] and [5]. Section 7 shows computational performance of the proposed adaptive
 161 sequential SAA framework on a variety of test instances.

162 **2. PROBLEM SETUP.** The 2SLP is formally stated as follows:

$$163 \quad (P) \quad \min c^\top x + q(x)$$

$$164 \quad \text{s.t. } x \in \mathcal{X} := \{x \in \mathbb{R}_+^{n_1} \mid Ax = b\},$$

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

$$168 \quad (2.1) \quad q(x) = \mathbb{E}[Q(x, \xi)] = \int_{\Xi} Q(x, \xi) dP(\xi),$$

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

$$170 \quad (2.2) \quad Q(x, \xi) = \min_{y \in \mathbb{R}_+^{n_2}} \{d(\xi)^\top y \mid W(\xi)y \geq h(\xi) - T(\xi)x\}.$$

171 We assume that the second-stage objective value is finite, i.e., $Q(x, \xi) > -\infty$, $\forall x \in \mathcal{X}$, and
 172 $\xi \in \Xi$. Notice that the function $q(\cdot)$ is not directly “observable” but can be estimated pointwise by
 173 “generating scenarios.” Specifically, we assume that an iterative algorithm, during the ℓ -th iteration,

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

$$176 \quad (P_\ell) \quad \min c^\top x + Q_{m_\ell}^\ell(x)$$

$$177 \quad \text{s.t. } x \in \mathcal{X} := \{x \in \mathbb{R}_+^{n_1} \mid Ax = b\},$$

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

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

192 The notation we use (with the superscript and subscript), while cumbersome, is needed to
 193 reflect the fact that the framework we propose allows for a variety of dependence structures of
 194 $\xi_i^\ell, i = 1, 2, \dots, m_\ell$ within and across iterations $\ell = 1, 2, \dots$. For example, in the simplest and most
 195 prevalent case of independent and identically distributed (iid) sampling, generation is done so that
 196 the random objects $\xi_i^\ell, i = 1, 2, \dots, m_\ell$ are mutually independent and identically distributed for
 197 each ℓ ; the objects $\xi_i^\ell, i = 1, 2, \dots, m_\ell$ can also be generated so as to satisfy chosen dependency
 198 structures that reduce variance, e.g., LHS [45], antithetic variates [48], and randomized quasi-Monte
 199 Carlo [37, 27]. Similarly, across iterations $\ell = 1, 2, \dots$, one can arrange for scenarios from previous
 200 iterations to be reused in subsequent iterations as in *common random numbers* [48]. Indeed, we will
 201 have to make certain assumptions on $Q_{m_\ell}^\ell(\cdot), \ell = 1, 2, \dots$ in Section 2.2 that will implicitly impose
 202 restrictions on the nature of sampling, to ensure that $Q_{m_\ell}^\ell(\cdot)$ approximates $q(\cdot)$ well enough.

203 **2.1. Further Notation and Convention.** We let \mathcal{S}^* denote the optimal solution set, z^*
 204 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
 205 problem (P) . Analogously, $\mathcal{S}_{m_\ell}^*$ denotes the optimal solution set, $z_{m_\ell}^*$ the optimal value, and
 206 $\mathcal{S}_{m_\ell}^*(\epsilon) := \{x \in \mathcal{X} : c^\top x + Q_{m_\ell}^\ell(x) - z_{m_\ell}^* \leq \epsilon\}$ the ϵ -optimal solution set for problem (P_ℓ) .

207 The following definitions are used extensively throughout the paper. (i) \mathbb{R}_+ denotes the set
 208 of non-negative real numbers. (ii) For $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, $\|x\|_2$ refers to the Euclidean
 209 norm $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$. (iii) For a real-valued continuous function $g : \mathcal{X} \rightarrow \mathbb{R}$ defined
 210 on the compact set \mathcal{X} , the sup-norm $\|g\|$ is defined as $\|g\| := \max_{x \in \mathcal{X}} |g(x)|$. (iv) The distance
 211 between a point $x \in \mathbb{R}^n$ and a set $X \subseteq \mathbb{R}^n$ is defined as $\text{dist}(x, X) := \inf\{\|x - z\|_2 : z \in X\}$,
 212 and the distance between two sets $X, Y \subseteq \mathbb{R}^n$ is defined as $\text{dist}(X, Y) := \sup_{x \in X} \{\text{dist}(x, Y)\}$. The
 213 definition we have used for $\text{dist}(\cdot, \cdot)$ suffices for our purposes even though it is not a metric since
 214 $\text{dist}(X, Y) \neq \text{dist}(Y, X)$ in general. (v) The diameter $\text{diam}(X)$ of a set $X \subseteq \mathbb{R}^n$ is defined as
 215 $\text{diam}(X) := \sup_{x, y \in X} \{\|x - y\|_2\}$. (vi) The projection of a point $x \in \mathbb{R}^n$ onto a set $X \subseteq \mathbb{R}^n$
 216 is defined as $\text{proj}(x, X) := \arg \inf_{z \in X} \{\|x - z\|_2\}$. (vii) $|X|$ denotes the cardinality of set X . (viii)
 217 For a sequence of \mathbb{R}^d -valued random variables $\{Z_n\}, Z$, we say $Z_n \rightarrow Z$ a.s. to mean that $\{Z_n\}$
 218 converges to Z almost surely, that is, with probability one. We say that Z_n converges to Z in

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

221 **2.2. Assumptions.** The following is a list of assumptions that we will use to prove various
 222 results in the paper. Assumption 1 and Assumption 2 are standing assumptions in that we will
 223 assume these to hold always. Assumption 3 will be invoked as and when needed.

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

$$226 \quad \mathbb{P}\{(y \in \mathbb{R}_+^{n_2} : W(\xi)y \geq h(\xi) - T(\xi)x) = \emptyset\} = 0, \quad \forall x \in \mathcal{X}.$$

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

$$229 \quad (2.3) \quad \sup_{x \in \mathcal{X}} \text{Var}(Q(x, \xi^\ell) | \mathcal{F}_{\ell-1}) < \infty \text{ a.s.}$$

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

$$232 \quad (2.4) \quad \mathbb{E}[\|\bar{\epsilon}_m\|^2 | \mathcal{F}_{\ell-1}] \leq \frac{\kappa_0}{m} \text{ a.s.,}$$

233 where the sample-mean error function $\bar{\epsilon}_m(x) := Q_m^\ell(x) - q(x) = m^{-1} \sum_{j=1}^m (Q(x, \xi_j^\ell) - q(x))$. (The
 234 $\|\cdot\|$ appearing in (2.4) is the sup-norm defined in Section 2.1).

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

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

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

237 The condition (2.4) in Assumption 2 has been stated for generality, to subsume many contexts
 238 that involve dependent and biased sampling, and needs justification. To get a clear sense of the
 239 conditions under which (2.4) in Assumption 2 holds, let's first observe that in the *iid unbiased*
 240 context, that is, when $\xi_j^\ell, j = 1, 2, \dots$ are iid and $\mathbb{E}[Q(x, \xi_j^\ell) - q(x) | \mathcal{F}_{\ell-1}] = 0$ a.s., the vast body of
 241 recent literature on concentration inequalities [12, 13, 38, 73] guarantees that (2.4) holds under a
 242 variety of moment conditions on $Q(x, \xi^\ell)$. For a general result that can be directly applied in the iid
 243 unbiased context, see [20, Proposition 3.1] established for Banach spaces. (Much of the literature
 244 on concentration inequalities is focused on sharp quantifications of the tail probabilities associated
 245 with $\bar{\epsilon}_m$, and thus characterize the constant κ_0 indirectly; our proposed algorithms do not rely on
 246 knowing κ_0 .)

247 In the *dependent but unbiased* sampling context, that is, when $\mathbb{E}[Q(x, \xi_j^\ell) - q(x) | \mathcal{F}_{\ell-1}] = 0$ a.s.
 248 but $\xi_j^\ell, j = 1, 2, \dots$ are not necessarily independent, Assumption 2 holds in many popular settings
 249 where the estimator can be written as an alternate sum of iid unbiased random variables at each
 250 $x \in \mathcal{X}$. For instance, consider using *antithetic variates* [48], where for even m we set $\xi_j^\ell := U_j^\ell \in$
 251 $(0, 1), \xi_{j+1}^\ell = 1 - U_j^\ell, j = 1, 3, 5, \dots, m - 1$. Then, $Q_m^\ell(x)$ can be written as the sample mean
 252 of $m/2$ (ignoring non-integrality) iid unbiased random variables, each of which is the sum of the

253 two dependent random variables $Q(x, U_j^\ell)$ and $Q(x, 1 - U_j^\ell)$, implying that Assumption 2 again
 254 holds. Similarly, if one chooses stratified sampling [27] as a variance reduction technique, then
 255 $Q_m^\ell(x), x \in \mathcal{X}$ can be written as a finite convex combination of sample means, each of which is
 256 composed of iid random variables that are unbiased with respect to the conditional means.

257 Assumption 2 can be shown to hold in other dependent sampling settings such as LHS [45] as
 258 well. To see this, we “construct” a d -dimensional random variable $\xi_j^\ell := (\xi_{1j}^\ell, \xi_{2j}^\ell, \dots, \xi_{dj}^\ell) \in [0, 1]^d$
 259 where $\xi_{ij}^\ell = m^{-1}(\pi_{ij} + U_{ij})$, $\pi_i = (\pi_{i1}, \pi_{i2}, \dots, \pi_{im}), i = 1, 2, \dots, d$ is each a uniform random
 260 permutation of $(0, 1, 2, \dots, m - 1)$, $U_{ij} \sim [0, 1)$, and U_{ij} ’s and π_i ’s are independent. Under this
 261 setup, we see that $\xi_{ij}^\ell \sim U[0, 1)$, $\xi_i^\ell \sim U[0, 1)^d$, and that $Q_{m_\ell}^\ell(x)$ is an unbiased estimator of
 262 $q(x)$ that is constructed from dependent random variables. Furthermore, under this setup, and as
 263 shown in [45, p. 245] and [52, Section 10.3], $\text{Var}(Q(x, \xi)) < \infty$ guarantees that $\text{Var}(Q_{m_\ell}^\ell(x)) =$
 264 $\sigma_0^2/m_\ell + o(m_\ell^{-1}) = O(m_\ell^{-1})$, where $\sigma_0^2 = \mathbb{E}[(Q(x, \xi) - Q^{\text{add}}(x, \xi))^2]$ and $Q^{\text{add}}(x, \xi)$ is the additive
 265 approximation of $Q(x, \xi)$ obtained using ANOVA. See also [71] for large sample properties in the
 266 LHS context.

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

272 We recognize that we have limited all of the above discussion on dependent sampling by fixing
 273 $x \in \mathcal{X}$. A complete treatment of Assumption 2 that involves dependence across $x \in \mathcal{X}$ will require
 274 us to consider the behavior of the random function $Q_{m_\ell}^\ell(\cdot)$ by directly making assumptions on the
 275 vector $(d(\xi), W(\xi), h(\xi), T(\xi))$ appearing in the second-stage problem (2.2). In general, some sort of
 276 a stipulation on the quality of the Monte Carlo estimator is needed to provide reasonable guarantees
 277 relating to convergence and convergence rates. For example, in Chapter 5 of [67], we see that even
 278 for convergence of sample-path optimal values of SAA to the true optimal value, one needs uniform
 279 convergence (across $x \in \mathcal{X}$) of the sample-path functions.

280 Finally, Assumption 3 is a standard regularity condition [67] having to do with the growth
 281 behavior of the true objective function. Specifically, Assumption 3 imposes a minimum growth
 282 condition on the true objective function $c^\top x + q(x)$.

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

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

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

Algorithm 3.1 An adaptive sequential SAA framework.

```

1: Input: Solver- $\mathcal{A}$ , a sampling scheme, constants  $\nu, \sigma_{\min}, \sigma_{\max} \in (0, \infty)$ , with  $\sigma_{\min} < \sigma_{\max}$ .
2: Set  $\ell \leftarrow 0$ .
3: for  $\ell = 1, 2, \dots$  do
4:   Select sample size  $m_\ell$  for outer iteration  $\ell$  and draw a sample  $\mathcal{M}_\ell := \{\xi_1^\ell, \xi_2^\ell, \dots, \xi_{m_\ell}^\ell\}$ .
5:   for  $t = 1, 2, \dots$  do
6:     Using Solver- $\mathcal{A}$  on  $(P_\ell)$ , execute  $t$ -th inner iteration.
7:     Obtain candidate solution  $\hat{x}^{\ell,t}$ , gap estimate  $G^{\ell,t}$  and variance parameter estimate  $\hat{\sigma}_{\ell,t}$ .
8:     if  $G^{\ell,t} \leq \epsilon_{\ell,t} := \nu m_\ell^{-1/2} \text{proj}(\hat{\sigma}_{\ell,t}, [\sigma_{\min}, \sigma_{\max}])$  then
9:       Break the inner loop with a candidate solution  $\hat{x}^\ell := \hat{x}^{\ell,t}$ .
10:    end if
11:  end for
12:  Set  $\ell \leftarrow \ell + 1$ .
13: end for

```

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

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

$$311 \quad (3.1) \quad G^{\ell,t} < \epsilon_{\ell,t} := \nu m_\ell^{-1/2} \text{proj}(\hat{\sigma}_{\ell,t}, [\sigma_{\min}, \sigma_{\max}]),$$

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

$$313 \quad (3.2) \quad \hat{\sigma}_{\ell,t}^2 := \frac{1}{m_\ell} \sum_{i=1}^{m_\ell} [Q(\hat{x}^{\ell,t}, \xi_i^\ell) - Q_{m_\ell}^\ell(\hat{x}^{\ell,t})]^2.$$

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

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

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

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

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

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

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

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

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

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

343
$$\mathbb{P} \left\{ \sup_{x \in \mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| > t \mid \mathcal{F}_{\ell-1} \right\} \leq t^{-1} \mathbb{E} \left[\sup_{x \in \mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| \mid \mathcal{F}_{\ell-1} \right]$$

344
$$= t^{-1} \mathbb{E} \left[\left(\sup_{x \in \mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)|^2 \right)^{1/2} \mid \mathcal{F}_{\ell-1} \right]$$

345 (3.3)
$$\leq t^{-1} \left(\mathbb{E} \left[\sup_{x \in \mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)|^2 \mid \mathcal{F}_{\ell-1} \right] \right)^{1/2} \leq \frac{\sqrt{\kappa_0} t^{-1}}{\sqrt{m_\ell}}, \quad \square$$

347 where the first line in (3.3) is from Markov's inequality [7], the third from (concave) Jensen's inequality [7], and the last from Assumption 2. Conclude from (3.3), the assumed bound (SS-A), and the filtered version of the Borel-Cantelli lemma [75], that the assertion of the theorem holds.

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

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

360 We begin with Lemma 4.1 — a result on the behavior of approximate minimizers of a sequence
 361 of convex functions that uniformly converge to a limit function. We emphasize that this result
 362 is stated in a *deterministic* setting and will become very useful in explaining the behavior of the
 363 sample paths in the stochastic context in the subsequent section. It also appears to be interesting
 364 in its own right due to applicability in the context of optimization with a deterministic inexact
 365 oracle. See [24, 58, 59] for more on such problems. A complete proof is provided in the appendix
 366 of the online supplementary document [56].

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

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

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

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

371 Then the following assertions hold.

- 372 (a) $f(x_k) \rightarrow v^*$ as $k \rightarrow \infty$;
 373 (b) for each $k \geq 1$, $f(x_k) - v^* \leq 2 \sum_{j=k}^{\infty} \delta_j + 2 \sum_{j=k}^{\infty} \epsilon_j$.
 374 If f obeys a growth rate condition, that is, there exist $\tau > 0, \gamma > 0$ such that for all $x \in \mathcal{X}$,

375 (4.1)
$$f(x) - v^* \geq \tau \text{dist}(x, \mathcal{S}_f^*)^\gamma, \text{ then}$$

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

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

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

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

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

- 394 (a) $c^T \hat{x}^\ell + q(\hat{x}^\ell) \rightarrow z^*$ a.s. as $\ell \rightarrow \infty$;
 395 (b) $\mathbb{E}[c^T \hat{x}^\ell + q(\hat{x}^\ell)] \rightarrow z^*$ as $\ell \rightarrow \infty$.

396 *If Assumption 3 also holds, then the following assertions hold as well.*

- 397 (c) $\text{dist}(\hat{x}^\ell, \mathcal{S}^*) \rightarrow 0$ a.s. as $\ell \rightarrow \infty$;
 398 (d) $\mathbb{E}[\text{dist}(\hat{x}^\ell, \mathcal{S}^*)] \rightarrow 0$ as $\ell \rightarrow \infty$.

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

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

$$404 \quad \delta_\ell \leq \sup_{x \in \mathcal{X}} |Q_{m_{\ell+1}}^{\ell+1}(x) - q(x)| + \sup_{x \in \mathcal{X}} |Q_{m_\ell}^\ell(x) - q(x)| := \zeta_{m_{\ell+1}}^{\ell+1} + \zeta_{m_\ell}^\ell.$$

406 The inequality in (2) and Assumption 2 imply that

$$407 \quad (4.2) \quad \mathbb{E} \left[\sum_{\ell=1}^n \delta_\ell \right] = \sum_{\ell=1}^n \mathbb{E} [\delta_\ell] \leq \sum_{\ell=1}^n \mathbb{E} \left[\left(\zeta_{m_{\ell+1}}^{\ell+1} + \zeta_{m_\ell}^\ell \right) \right] \leq \sqrt{\kappa_0} \left(\sum_{\ell=1}^n \frac{1}{\sqrt{m_{\ell+1}}} + \sum_{\ell=1}^n \frac{1}{\sqrt{m_\ell}} \right),$$

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

$$411 \quad (4.3) \quad \mathbb{E} \left[\sum_{\ell=1}^{\infty} \delta_\ell \right] = \sum_{\ell=1}^{\infty} \mathbb{E} [\delta_\ell] \leq \sqrt{\kappa_0} \left(\sum_{\ell=1}^{\infty} \frac{1}{\sqrt{m_{\ell+1}}} + \sum_{\ell=1}^{\infty} \frac{1}{\sqrt{m_\ell}} \right),$$

412 where the equality is due to the monotone convergence theorem [7, Theorem 16.2] and the inequality
 413 is due to (4.2). The inequality in (4.3) together with the sample size condition (SS-A) implies that
 414 $\mathbb{E}[\sum_{\ell=1}^{\infty} \delta_\ell] < \infty$, and hence that $\sum_{\ell=1}^{\infty} \delta_\ell < \infty$ a.s. Also, recall that the error tolerance sequence
 415 $(\epsilon_\ell)_{\ell \geq 1}$ in Algorithm 3.1 has been chosen as $\epsilon_\ell = \nu \frac{1}{\sqrt{m_\ell}} \text{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$. This choice implies
 416 that $\sum_{\ell=1}^{\infty} \epsilon_\ell < \infty$ a.s. The two inequalities above imply that all postulates leading to assertions
 417 (a) and (b) in Lemma 4.1 are satisfied on a set (of sample-paths) of measure one; we thus conclude
 418 that the assertion (a) of the theorem holds. The assertion in (b) follows from the assertion in (a)
 419 since the function $q(\cdot)$ is continuous on the compact set \mathcal{X} and is hence bounded.

420 If Assumption 3 is satisfied, we know that

$$421 \quad (4.4) \quad \text{dist}(\hat{x}^\ell, \mathcal{S}^*) \leq \gamma_0^{-1} (c^T \hat{x}^\ell + q(\hat{x}^\ell) - z^*). \quad \square$$

422 Use assertion (a) and (4.4) to conclude that assertion (c) holds. Furthermore, since \mathcal{X} is compact,
 423 $\text{dist}(\hat{x}^\ell, \mathcal{S}^*)$ is bounded and hence assertion (d) holds as well.

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

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

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

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

$$443 \text{ (SS-C)} \quad m_\ell = c_1 m_{\ell-1}, \quad c_1 \in (1, \infty).$$

444 *Then,*
445 (5.1)

$$445 \quad \mathbb{E} [c^\top \hat{x}^\ell + q(\hat{x}^\ell) - z^*] \leq 2 \kappa_2 c_1^{-\ell/2}, \quad \text{where } \kappa_2 := \sqrt{\frac{c_1}{m_1}} \frac{1}{\sqrt{c_1} - 1} (\sqrt{\kappa_0}(\sqrt{c_1} + 1) + \sigma_{\max} \sqrt{c_1 \nu}).$$

446 *If Assumption 3 holds as well, then*

$$447 \text{ (5.2)} \quad \mathbb{E} [\text{dist}(\hat{x}^\ell, \mathcal{S}^*)] \leq 2 \tau_0^{-1} \kappa_2 c_1^{-\ell/2}.$$

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

$$452 \text{ (5.3)} \quad c^\top \hat{x}^\ell + q(\hat{x}^\ell) - z^* \leq 2 \sum_{k=\ell}^\infty \delta_k + 2 \sum_{k=\ell}^\infty \epsilon_k \quad \text{a.s.}$$

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

$$455 \text{ (5.4)} \quad \mathbb{E} \left[\sum_{k=\ell}^\infty \delta_k \right] = \sum_{k=\ell}^\infty \mathbb{E} [\delta_k] \leq \sqrt{\kappa_0} \left(\sum_{k=\ell}^\infty \frac{1}{\sqrt{m_{k+1}}} + \sum_{k=\ell}^\infty \frac{1}{\sqrt{m_k}} \right) \leq c_1^{-\ell/2} \sqrt{\frac{\kappa_0 c_1}{m_1}} \frac{\sqrt{c_1} + 1}{\sqrt{c_1} - 1}.$$

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

$$458 \text{ (5.5)} \quad \mathbb{E} [\epsilon_\ell] \leq \sqrt{\frac{\nu}{m_\ell} \sigma_{\max}^2},$$

459 and hence

$$460 \text{ (5.6)} \quad \mathbb{E} \left[\sum_{k=\ell}^\infty \epsilon_k \right] := \lim_{n \rightarrow \infty} \mathbb{E} \left[\sum_{k=\ell}^n \epsilon_k \right] = \lim_{n \rightarrow \infty} \sum_{k=\ell}^n \mathbb{E} [\epsilon_k] \leq c_1^{-\ell/2} \sqrt{\frac{\nu \sigma_{\max}^2 c_1}{m_1}} \frac{\sqrt{c_1}}{\sqrt{c_1} - 1}, \quad \square$$

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

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

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

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

$$484 \quad (5.7) \quad (c^T \hat{x}^{\ell,t} + Q_{m_\ell}(\hat{x}^{\ell,t})) - z_{m_\ell}^* \leq \Lambda_\ell \frac{\text{dist}(\hat{x}^{\ell-1}, S_{m_\ell}^*)}{\sqrt{t}}, \quad t = 1, 2, \dots,$$

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

$$487 \quad (5.8) \quad \gamma_\ell := \sup_s \{s : c^T x + Q_{m_\ell}^\ell(x) - z_{m_\ell}^* \geq s \text{dist}(x, S_{m_\ell}^*) \quad \forall x \in \mathcal{X}\},$$

488 there exists $\lambda < \infty$ such that

$$489 \quad (5.9) \quad \mathbb{E} \left[\left(\frac{\Lambda_\ell}{\gamma_\ell} \right)^2 \mid \mathcal{F}_{\ell-1} \right] \leq \lambda^2 < \infty \text{ a.s.}$$

490 Assumption 4 has been stated in a way that preserves generality of our theory, with the intent of
 491 allowing any choice of Solver- \mathcal{A} as long as the stipulation of Assumption 4 is met. Furthermore,
 492 we emphasize that Assumption 4 has been stated for piecewise linear convex objectives, since the
 493 objective function of the sample-path problem (P_ℓ) is piecewise linear convex. For instance, a
 494 number of well-known subgradient algorithms provide a guaranteed iteration complexity of the
 495 sort stipulated in (5.7) of Assumption 4 even for convex non-smooth objectives. For example, the
 496 standard subgradient descent algorithm having the iterative structure $x_{t+1} = x_t - \alpha_t \partial h(x_t)$, $t =$
 497 $0, 1, 2, \dots$ for solving the convex optimization problem $\min_{x \in \mathcal{X}} \{h(x)\}$, when executed with constant
 498 step size $\alpha_t = \epsilon/M^2$ and $\|\partial h(x)\| \leq M$, $\forall x \in \mathcal{X}$, satisfies the complexity requirement stated in
 499 Assumption 4. Another recent example is a variant of the level bundle method [6] under an idealized

500 assumption. In our numerical experiments presented in Section 7, we use an implementable variant
 501 of the level bundle method as Solver- \mathcal{A} , which is described in greater detail in the appendix of the
 502 online supplementary document [56]

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

$$507 \quad (5.10) \quad \Lambda_\ell \leq \frac{M_\ell}{\sqrt{\alpha(1-\alpha)^2(2-\alpha)}}, \quad \alpha \in (0, 1)$$

508 where $\alpha \in (0, 1)$ is a user-chosen constant within the level method, and $M_\ell := \sup_{x \in \mathcal{X}} \{ \|c +$
 509 $\partial Q_{m_\ell}(x)\| \}$ is the supremum norm (taken over the fixed compact set \mathcal{X}) of the sub-gradient asso-
 510 ciated with the sample-path function. It follows from (5.10) then that

$$511 \quad (5.11) \quad \frac{\Lambda_\ell}{\gamma_\ell} \leq \frac{1}{\sqrt{\alpha(1-\alpha)^2(2-\alpha)}} \frac{M_\ell}{\gamma_\ell},$$

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

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

$$521 \quad \mathbb{P} \left\{ N_\ell > \Lambda_\ell^2 \frac{(\text{dist}(\hat{x}^{\ell-1}, \mathcal{S}_{m_\ell}^*))^2}{\epsilon_\ell^2} \mid \mathcal{F}_{\ell-1} \right\} = 0, \text{ and } \mathbb{E} \left[\left(\frac{\Lambda_\ell}{\gamma_\ell} \right)^2 \mid \mathcal{F}_{\ell-1} \right] < \infty \text{ a.s.}$$

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

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

$$527 \quad (5.12) \quad \mathbb{E} \left[(c^\top \hat{x}^L + q(\hat{x}^L) - z^*) \right] \leq \tau_0 / \mathbb{E}[\sqrt{W_L}], \text{ where } \tau_0 \text{ is a constant independent of } L.$$

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

$$529 \quad (5.13) \quad \mathbb{E} \left[\text{dist}(\hat{x}^L, \mathcal{S}^*) \right] \leq \tau_0 \gamma_0^{-1} / \mathbb{E}[\sqrt{W_L}], \text{ where } \gamma_0 \text{ is the growth-rate constant in Assumption 3.}$$

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

$$532 \quad (5.14) \quad \mathbb{E} \left[\sqrt{\tilde{W}_\ell} \right] = \mathbb{E} \left[\sqrt{N_\ell m_\ell} \right] \leq \mathbb{E} \left[\mathbb{E} \left[\Lambda_\ell \frac{(\text{dist}(\hat{x}^{\ell-1}, \mathcal{S}_{m_\ell}^*))}{\epsilon_\ell} \sqrt{m_\ell} \mid \mathcal{F}_{\ell-1} \right] \right].$$

533

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

$$\begin{aligned}
535 \quad \mathbb{E} \left[\sqrt{\tilde{W}_\ell} \right] &\leq \frac{m_\ell}{\nu \sigma_{\min}} \mathbb{E} \left[\mathbb{E} \left[\Lambda_\ell \text{dist}(\hat{x}^{\ell-1}, \mathcal{S}_{m_\ell}^*) \mid \mathcal{F}_{\ell-1} \right] \right] \\
536 \quad &\leq \frac{m_\ell}{\nu \sigma_{\min}} \mathbb{E} \left[\left(\mathbb{E} \left[\left(\frac{\Lambda_\ell}{\gamma_\ell} \right)^2 \mid \mathcal{F}_{\ell-1} \right] \right)^{\frac{1}{2}} \left(\mathbb{E} \left[(c^T \hat{x}^{\ell-1} + Q_{m_\ell}^\ell(\hat{x}^{\ell-1}) - z_{m_\ell}^*)^2 \mid \mathcal{F}_{\ell-1} \right] \right)^{\frac{1}{2}} \right] \\
537 \quad &\leq \frac{m_\ell \lambda}{\nu \sigma_{\min}} \mathbb{E} \left[\left(\mathbb{E} \left[(c^T \hat{x}^{\ell-1} + Q_{m_\ell}^\ell(\hat{x}^{\ell-1}) - z_{m_\ell}^*)^2 \mid \mathcal{F}_{\ell-1} \right] \right)^{\frac{1}{2}} \right] \\
538 \quad &\leq \frac{m_\ell \lambda}{\nu \sigma_{\min}} \left(\mathbb{E} \left(\mathbb{E} \left[(c^T \hat{x}^{\ell-1} + Q_{m_\ell}^\ell(\hat{x}^{\ell-1}) - z_{m_\ell}^*)^2 \mid \mathcal{F}_{\ell-1} \right] \right) \right)^{\frac{1}{2}} \\
539 \quad (5.15) \quad &\leq \frac{m_\ell \lambda}{\nu \sigma_{\min}} \left(\mathbb{E} \left[\left(\delta_\ell + \epsilon_{\ell-1} + |z_{m_{\ell-1}}^* - z_{m_\ell}^*| \right)^2 \right] \right)^{\frac{1}{2}}, \\
540
\end{aligned}$$

541 where the second inequality in (5.15) uses the Cauchy-Schwarz inequality (conditionally) and the
542 definition in (5.8) of the sample-path growth rate, the third inequality uses the finite second moment
543 assumption in (5.9) of Assumption 4, the fourth inequality uses the concavity of the square root
544 function, and the last inequality uses $\delta_\ell := \sup_{x \in \mathcal{X}} \{ |Q_{m_\ell}^\ell(x) - Q_{m_{\ell-1}}^{\ell-1}(x)| \}$ and the fact that $\hat{x}^{\ell-1} \in$
545 $\mathcal{F}_{\ell-1}$ is $\epsilon_{\ell-1}$ -optimal to $c^T x + Q_{m_{\ell-1}}^{\ell-1}(x)$ over the set \mathcal{X} . Next, let $x_\ell^* \in \mathcal{S}_{m_\ell}^*$, $x_{\ell-1}^* \in \mathcal{S}_{m_{\ell-1}}^*$ and
546 observe that

$$\begin{aligned}
547 \quad |z_{m_{\ell-1}}^* - z_{m_\ell}^*| &\leq (c^T x_\ell^* + Q_{m_{\ell-1}}^{\ell-1}(x_\ell^*) - z_{m_\ell}^*) + (c^T x_{\ell-1}^* + Q_{m_\ell}^\ell(x_{\ell-1}^*) - z_{m_{\ell-1}}^*) \\
548 \quad (5.16) \quad &\leq 2 \sup_{x \in \mathcal{X}} \left\{ |Q_{m_\ell}^\ell(x) - Q_{m_{\ell-1}}^{\ell-1}(x)| \right\} = 2\delta_\ell. \\
549
\end{aligned}$$

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

$$551 \quad (5.17) \quad \mathbb{E} \left[\sqrt{\tilde{W}_\ell} \right] \leq \lambda \frac{m_\ell}{\nu \sigma_{\min}} \left(\mathbb{E} \left[(3\delta_\ell + \epsilon_{\ell-1})^2 \right] \right)^{\frac{1}{2}} \leq \lambda \frac{m_\ell}{\nu \sigma_{\min}} \left(\mathbb{E} \left[18\delta_\ell^2 + 2\nu^2 \frac{\sigma_{\max}^2}{m_{\ell-1}} \right] \right)^{\frac{1}{2}},$$

552 where the second inequality above uses $(a+b)^2 \leq 2a^2 + 2b^2$ and $\epsilon_\ell := \nu m_\ell^{-1/2} \text{proj}(\hat{\sigma}_\ell, [\sigma_{\min}, \sigma_{\max}])$.

553 Observing that $W_L = \sum_{\ell=1}^L \tilde{W}_\ell$, (5.17) implies that

$$\begin{aligned}
554 \quad \mathbb{E}[\sqrt{W_L}] &\leq \sum_{\ell=1}^L \mathbb{E} \left[\sqrt{\tilde{W}_\ell} \right] \leq \sum_{\ell=1}^L \lambda \frac{m_\ell}{\nu \sigma_{\min}} \left(18\mathbb{E}[\delta_\ell^2] + 2\nu^2 \frac{\sigma_{\max}^2}{m_{\ell-1}} \right)^{\frac{1}{2}} \\
555 \quad &\leq \sum_{\ell=1}^L \lambda \frac{m_\ell}{\nu \sigma_{\min}} \left(18 \frac{\kappa_0(1+c_1+2\sqrt{c_1})}{m_\ell} + 2\nu^2 c_1 \frac{\sigma_{\max}^2}{m_\ell} \right)^{\frac{1}{2}} \\
556 \quad &\leq \frac{\lambda}{\nu \sigma_{\min}} \left(18\kappa_0(1+c_1+2\sqrt{c_1}) + 2\nu^2 c_1 \sigma_{\max}^2 \right)^{\frac{1}{2}} \sum_{\ell=1}^L \sqrt{m_\ell} \\
557 \quad (5.18) \quad &= \frac{\lambda}{\nu \sigma_{\min}} \left(18\kappa_0(1+c_1+2\sqrt{c_1}) + 2\nu^2 c_1 \sigma_{\max}^2 \right)^{\frac{1}{2}} \frac{\sqrt{m_1}}{\sqrt{c_1}-1} \left(c_1^{L/2} - 1 \right), \\
558
\end{aligned}$$

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

$$\begin{aligned}
562 \quad \mathbb{E} \left[(c^T \hat{x}^L + q(\hat{x}^L) - z^*)^2 \right] &\leq 4 \mathbb{E} \left[\left(\lim_{n \rightarrow \infty} \sum_{\ell=L}^n (\delta_\ell + \epsilon_\ell) \right)^2 \right] = 4 \lim_{n \rightarrow \infty} \mathbb{E} \left[\left(\sum_{\ell=L}^n (\delta_\ell + \epsilon_\ell) \right)^2 \right] \\
563 \quad (5.19) \quad &\leq 4 \sum_{\ell=L}^{\infty} \mathbb{E} [(\delta_\ell + \epsilon_\ell)^2] + 8 \sum_{\ell=L}^{\infty} (\mathbb{E} [(\delta_\ell + \epsilon_\ell)^2])^{1/2} \sum_{j=\ell+1}^{\infty} (\mathbb{E} [(\delta_j + \epsilon_j)^2])^{1/2}, \\
564
\end{aligned}$$

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

$$\begin{aligned}
568 \quad \sum_{j=\ell+1}^{\infty} (\mathbb{E} [(\delta_j + \epsilon_j)^2])^{1/2} &\leq \sum_{j=\ell+1}^{\infty} (2\mathbb{E}[\delta_j^2] + 2\mathbb{E}[\epsilon_j^2])^{1/2} \\
569 \quad (5.20) \quad &\leq \sum_{j=\ell+1}^{\infty} \frac{1}{\sqrt{m_j}} (2\kappa_0(1 + c_1 + 2\sqrt{c_1}) + 2\nu^2 \sigma_{\max}^2)^{1/2} \leq \tilde{\kappa}_1 c_1^{-\ell/2} \\
570
\end{aligned}$$

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

$$\begin{aligned}
574 \quad (5.21) \quad \sum_{\ell=L}^{\infty} (\mathbb{E} [(\delta_\ell + \epsilon_\ell)^2]) &\leq \tilde{\kappa}_2 c_1^{-L}, \\
575
\end{aligned}$$

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

$$\begin{aligned}
577 \quad \mathbb{E} \left[(c^T \hat{x}^L + q(\hat{x}^L) - z^*)^2 \right] &\leq 4 \sum_{\ell=L}^{\infty} \mathbb{E} [(\delta_\ell + \epsilon_\ell)^2] + 8 \sum_{\ell=L}^{\infty} (\mathbb{E} [(\delta_\ell + \epsilon_\ell)^2])^{1/2} \sum_{j=\ell+1}^{\infty} (\mathbb{E} [(\delta_j + \epsilon_j)^2])^{1/2} \\
578 \quad &\leq 4\tilde{\kappa}_2 c_1^{-L} + 8 \sum_{\ell=L}^{\infty} (\mathbb{E} [(\delta_\ell + \epsilon_\ell)^2])^{1/2} \tilde{\kappa}_1 c_1^{-\ell/2} \\
579 \quad (5.22) \quad &\leq 4\tilde{\kappa}_2 c_1^{-L} + 8(\sqrt{c_1} - 1) \tilde{\kappa}_1^2 \sum_{\ell=L}^{\infty} c_1^{-\ell} = c_1^{-L} \left(4\tilde{\kappa}_2 + \frac{8c_1 \tilde{\kappa}_1^2}{\sqrt{c_1} + 1} \right). \\
580
\end{aligned}$$

581 Finally, we put it all together to get

$$\begin{aligned}
582 \quad & \mathbb{E} \left[\sqrt{W_L} \right] \mathbb{E} \left[(c^T \hat{x}^L + q(\hat{x}^L) - z^*) \right] \\
583 \quad & \leq \frac{\lambda}{\nu \sigma_{\min}} \left(18\kappa_0(1 + c_1 + 2\sqrt{c_1}) + 2\nu^2 c_1 \sigma_{\max}^2 \right)^{\frac{1}{2}} \frac{\sqrt{m_1}}{\sqrt{c_1 - 1}} \left(1 - \frac{1}{c_1^{L/2}} \right) \left(4\tilde{\kappa}_2 + \frac{8c_1 \tilde{\kappa}_1^2}{\sqrt{c_1 + 1}} \right)^{1/2} \\
584 \quad & \leq \frac{\lambda}{\nu \sigma_{\min}} \left(18\kappa_0(1 + c_1 + 2\sqrt{c_1}) + 2\nu^2 c_1 \sigma_{\max}^2 \right)^{\frac{1}{2}} \frac{\sqrt{m_1}}{\sqrt{c_1 - 1}} \left(4\tilde{\kappa}_2 + \frac{8c_1 \tilde{\kappa}_1^2}{\sqrt{c_1 + 1}} \right)^{1/2} =: \tau_0, \\
(5.23) \quad & \hspace{15em} \square
\end{aligned}$$

585

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

591 The following observations on Theorem 5.3 are noteworthy.

- 592 (a) The assertions in Theorem 5.3 should be seen as the analogue of the $\mathcal{O}(1/\epsilon^2)$ complexity
593 result in non-smooth convex optimization that is known to be optimal [50] to within a
594 constant factor.
- 595 (b) The complexity result in Theorem 5.3 has been stated in the general population context.
596 So, the result equally applies for the finite-population scenario $|\Xi| < \infty$, although there is
597 strong evidence that in the finite and the countably infinite populations, the best achievable
598 complexity rates may be much faster due to the existence of sharp minima of the sort
599 discussed in [69].
- 600 (c) The theorem assumes that the sample size schedule $(m_\ell)_{\ell \geq 1}$ increases geometrically with
601 common ratio c_1 . Importantly, the result can be generalized in a straightforward manner to
602 a sample size schedule having a stochastic common ratio C_1 that is allowed to vary between
603 two deterministic bounds c_0 and c_h such that $1 < c_0 \leq c_h < \infty$ (see Section 7).

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

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

$$609 \quad (5.24) \quad \mathbb{E} \left[(\text{dist}(\mathcal{S}_{m_\ell}^*, \mathcal{S}^*)) \right] \geq \frac{\tilde{\eta}}{\sqrt{m_\ell}}.$$

610 *If the sample size schedule is polynomial, that is,*

$$611 \quad (\text{SS-D}) \quad m_\ell = c_0 \ell^p, \quad c_0 \in (0, \infty), p \in [1, \infty).$$

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

$$613 \quad (5.25) \quad \mathbb{E} \left[\text{dist}(\hat{x}^L, \mathcal{S}^*) \right] \geq \frac{\tau_1}{\mathbb{E} \left[W_L^{\frac{1}{2} - \frac{1}{2(1+p)}} \right]}.$$

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

$$616 \quad (5.26) \quad W_L \geq \sum_{\ell=1}^L c_0 \ell^p \geq \int_1^L c_0 (\ell-1)^p d\ell = \frac{c_0}{p+1} ((L-1)^{p+1} - 1).$$

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

$$618 \quad W_L \geq \frac{c_0}{p+1} \left(\frac{m_L}{c_0}\right)^{1+1/p} \left(\left(1 - \left(\frac{c_0}{m_L}\right)^{1/p}\right)^{p+1} - \left(\frac{c_0}{m_L}\right)^{1+1/p} \right)$$

$$619 \quad (5.27) \quad \geq \frac{c_0}{p+1} \left(\frac{m_L}{c_0}\right)^{1+1/p} \left((1-L^{-1})^{p+1} - L^{-(p+1)} \right) \geq \tau_p \frac{c_0}{p+1} \left(\frac{m_L}{c_0}\right)^{1+1/p},$$

$$620$$

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

$$622 \quad (5.28) \quad W_L^{\frac{1}{2} - \frac{1}{2(1+p)}} \geq \left(\tau_p \frac{c_0}{p+1} \right)^{\frac{p}{2(p+1)}} \sqrt{\frac{m_L}{c_0}}.$$

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

625 We observe from Theorem 5.4 that no matter how large $p \in [1, \infty)$ is chosen when choosing a
 626 polynomial sample size schedule, the resulting complexity (5.25) is inferior to the complexity (5.13)
 627 implied by a geometric sample size schedule, with the inferiority characterized by the deviation
 628 $(2(p+1))^{-1}$. A similar result has been proved by [62] in a different context.

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

638 The condition in (5.24) might appear cryptic but we believe that this condition will hold under
 639 mild conditions. General sufficient conditions under which the sequence $\sqrt{m_\ell} \text{dist}(S_{m_\ell}, \mathcal{S}^*)$ will
 640 “stabilize” to a non-degenerate distribution are well-known [66, 21]. Such conditions, along with
 641 assuming the random variables $\sqrt{m_\ell} \text{dist}(S_{m_\ell}, \mathcal{S}^*)$ exhibit uniform integrability, will ensure that the
 642 condition in (5.24) is guaranteed to hold asymptotically.

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

648 identical to what is called FSP in [5], and then argue that the stipulations laid out in [5] hold here,
 649 thereby allowing to invoke the main results of [5]. We note that alternative finite stopping rules
 650 have also been studied in the literature, see, e.g., [65] for a sequential sampling based approach
 651 based on the variance associated with 2SLP solutions rather than their corresponding objective
 652 values.

653 Suppose we wish to stop our procedure with a solution whose optimality gap is within $\epsilon > 0$
 654 with probability exceeding $1 - \alpha$, $\alpha > 0$. Recall that upon terminating the ℓ -th outer iteration of
 655 Algorithm 6.1, we have at our disposal an \mathcal{F}_ℓ -measurable candidate solution \hat{x}^ℓ . To construct a
 656 one-sided $100(1 - \alpha)$ percent confidence interval on the true gap $c^\top \hat{x}^\ell + q(\hat{x}^\ell) - z^*$, we independently
 657 generate an iid sample $\mathcal{N}_\ell = \{\tilde{\xi}_1^\ell, \tilde{\xi}_2^\ell, \dots, \tilde{\xi}_{n_\ell}^\ell\}$. Assume that the sequence $\{n_\ell\}$ of “testing” sample
 658 sizes is non-decreasing; the random objects $\tilde{\xi}_i^\ell, i \geq 1, \ell \geq 1$ can be re-used across iterations, that
 659 is, $\tilde{\xi}_i^\ell$ can be chosen so that if $i < j$ then $\tilde{\xi}_k^i = \tilde{\xi}_k^j$ for $k = 1, 2, \dots, n_i$. We then use the set \mathcal{N}_ℓ to
 660 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:

$$661 \quad \tilde{G}_{n_\ell}^\ell(\hat{x}^\ell) = c^\top(\hat{x}^\ell - \tilde{x}_\ell^*) + \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} [Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell)];$$

$$662 \quad (6.1) \quad \tilde{s}_{n_\ell}^2(\hat{x}^\ell) = \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \left[Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell) - \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} [Q(\hat{x}^\ell, \tilde{\xi}_i^\ell) - Q(\tilde{x}_\ell^*, \tilde{\xi}_i^\ell)] \right]^2,$$

664 where \tilde{x}_ℓ^* is an optimal solution to the sample-path problem (P_ℓ) generated with sample \mathcal{N}_ℓ , and
 665 $\delta > 0$ is the thresholding constant from Algorithm 3.1.

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

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

666 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

$$667 \quad \left[0, \tilde{G}_{n_\ell}^\ell(\hat{x}^\ell) + z_\alpha \frac{\max(\tilde{s}_{n_\ell}(\hat{x}^\ell), \sigma_{\max})}{\sqrt{n_\ell}} \right],$$

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

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

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

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

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

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

676 (We have omitted (A2) above to preserve the numbering in [5].) Theorem 2.3 in [69] implies that
 677 Assumption (A1) is satisfied if the support Ξ is finite, in addition to Assumptions 1–3. Also, it is
 678 seen that Assumption (A3) and (A4) hold if the standing Assumption 2 holds. The following result
 679 characterizes the behavior of the iterates obtained from Algorithm 6.1, along with a probabilistic
 680 guarantee. We provide a proof only for the third part of the theorem since proofs for the rest either
 681 follow trivially or are almost identical to that in [5].

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

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

$$687 \quad (6.2) \quad \lim_{\epsilon \rightarrow 0} \mathbb{P} \left\{ \mu(\hat{x}^{L(\epsilon)}) \leq \epsilon \right\} = 1.$$

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

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

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

$$690 \quad (6.3) \quad \tilde{G}_{n_{L(\epsilon)}}^{L(\epsilon)}(\hat{x}^{L(\epsilon)}) = 0; \quad \tilde{s}_{n_{L(\epsilon)}}^2(\hat{x}^{L(\epsilon)}) = 0,$$

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

$$693 \quad \epsilon^2 n_{L(\epsilon)} \geq \left(\sqrt{n_{L(\epsilon)}} \tilde{G}_{n_{L(\epsilon)}}^{L(\epsilon)}(\hat{x}^{L(\epsilon)}) + z_\alpha \max(\tilde{s}_{n_{L(\epsilon)}}(\hat{x}^{L(\epsilon)}), \delta) \right)^2;$$

$$694 \quad (6.4) \quad \epsilon^2 n_{L(\epsilon)-1} \leq \left(\sqrt{n_{L(\epsilon)-1}} \tilde{G}_{n_{L(\epsilon)-1}}^{L(\epsilon)-1}(\hat{x}^{L(\epsilon)-1}) + z_\alpha \max(\tilde{s}_{n_{L(\epsilon)-1}}(\hat{x}^{L(\epsilon)-1}), \delta) \right)^2.$$

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

$$698 \quad (6.5) \quad n_{L(\epsilon)-1} \geq \tilde{\beta} n_{L(\epsilon)} \text{ a.s.}$$

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

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

705 The condition $\liminf_{\ell \rightarrow \infty} n_{\ell-1}/n_{\ell} > 0$ stipulated by the third assertion of Theorem 6.1 is
 706 satisfied by a wide variety of sequences. For instance, if $q_0, q_1 \in (0, \infty)$, any logarithmic increase
 707 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
 708 geometric increase schedule $n_{\ell}/n_{\ell-1} = q_1$ satisfy the condition $\liminf_{\ell \rightarrow \infty} n_{\ell-1}/n_{\ell} > 0$.

709 **7. COMPUTATIONAL EXPERIMENTS.** In this section, we present computational re-
 710 sults of the proposed adaptive sequential sampling framework for solving 2SLPs with fixed recourse
 711 and fixed second-stage objective coefficients. We chose problems instances of this type to enable
 712 a “warm starting” procedure, where the initial solution and an initial second-stage value function
 713 approximation for every sample-path problem at each outer iteration can be obtained using in-
 714 formation gained from previous iterations. (This procedure is summarized in Algorithm C.1 in
 715 the appendix of the online supplementary document [56].) For the purpose of benchmarking, we
 716 consider finite-sample instances of such problems, that is, problems where $|\Xi| < \infty$, so that we
 717 get access to the true optimal value z^* up to a pre-specified precision by solving these instances
 718 using a deterministic solver. In particular, we apply the adaptive partition-based level decompo-
 719 sition method [1], which has shown to be a competitive state-of-the-art solution approach. Five
 720 finite-sample instances of each problem in a selected problem class are generated; 20 replications
 721 of each competing sequential SAA algorithm are performed on each of the generated problem in-
 722 stances (except for the ssn instances, where only 10 replications are performed due to the extensive
 723 computational effort for solving these instances). We implemented all algorithms in C++ using
 724 the commercial solver CPLEX, version 12.8. All tests are conducted on an iMac desktop with four
 725 4.00GHz processors and 16Gb memory. The number of threads is set to be one.

726 We run the adaptive sequential SAA framework according to Algorithm 6.1, and record the
 727 total number of outer iterations as L , the final candidate solution at the L -th iteration as \hat{x}^L , and
 728 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
 729 of final candidate solution \hat{x}^L . We report in column “CI” the ratio between the width of the reported
 730 confidence interval (at stopping) for the optimality gap and the true objective value corresponding
 731 to \hat{x}^L . The threshold ϵ is chosen to be small enough relative to the objective value corresponding
 732 to the candidate solution obtained from the outer iteration, e.g., $10^{-3} \times (c^\top \hat{x}^1 + Q_{m_1}^1(\hat{x}^1))$. After
 733 Algorithm 6.1 terminates with a final solution \hat{x}^L , we verify whether or not the true optimal
 734 objective value z^* is in the reported confidence interval. Since the confidence interval at stopping is
 735 guaranteed to cover z^* only asymptotically (see Theorem 6.1), we report the coverage probability
 736 at stopping in the column titled “cov.”, using results obtained from the 20 replications for each test
 737 instance except ssn and 20term, where 10 replications are used.

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

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

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

747

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

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

- (i) **PILD-ODA.** This algorithm is the adaptive partition-based level decomposition algorithm with on-demand accuracy as proposed in [1], which is used to solve each instance with the full set of scenarios up to a relative optimality gap of 10^{-4} . Note that z^* for each instance is also obtained by this algorithm using a smaller relative optimality gap threshold of 10^{-6} .
- (ii) **Sequential-BP-L(Δ).** This algorithm follows the sampling schedules in [5] while solving individual sample-path problems to high precision. Specifically, each sample-path problem (with a sample size of m_ℓ) is solved up to a relative optimality gap of 10^{-6} in each outer iteration ℓ , using a standard level decomposition approach for solving 2SLPs [25]. Note that our implementation of this approach does not incorporate the warm starting functionality. The obtained candidate solution \hat{x}^ℓ is then evaluated using a sample of size n_ℓ . To obtain $x_{n_\ell}^*$ that appears in $\tilde{G}_{n_\ell}^\ell$ and $\tilde{s}_{n_\ell}^2$ in (6.1), we solve the corresponding sample-path problem up to a relative optimality gap of 10^{-4} , as suggested by [5]. By default, we use a linear sample size schedule where $\Delta = 100$ additional scenarios are sampled from one iteration to the next, starting with an initial sample size $m_1 = 2 \times n_1 = 100$. We use the same initial sample size for all variants of the sequential sampling approaches that we describe below, although one may tune this parameter for further enhancements.
- (iii) **Adaptive-seq-BP-L(Δ).** This is Algorithm 6.1 implemented with the linearly increasing sample size schedule proposed in [5], that is, $m_{\ell+1} = m_\ell + \Delta$. For “warm starting” the initial solution and an initial second-stage value function approximation for every sample-path problem at each outer iteration, we use Algorithm C.1 in the appendix of the online supplementary document [56]. We use parameter $\alpha = 0.1$ and safeguard parameter $\delta = 10^{-5}$ in defining the adaptive optimality tolerance ϵ_ℓ according to (3.1). PILD-ODA is applied to solve each sample-path problem with the aforementioned warm starting functionality.
- (iv) **Adaptive-seq-fixed (c_1).** This is Algorithm 6.1 implemented with a geometric sample size schedule. The setting is nearly identical to (iii) except that we use a fixed rate c_1 as the geometric increase rate, that is, $m_{\ell+1} = c_1 m_\ell$.
- (v) **Adaptive-seq-dyn(c_0, c_h).** Like in (iv), this is Algorithm 6.1 implemented with a geometric sample size schedule ensuring that $m_{\ell+1} = C_1 m_\ell$. However, unlike in (iv), the rate C_1 is dynamic (and hence, listed in uppercase) within chosen bounds c_0, c_h . Specifically, starting from some initial value of C_1 , if the inner loop finishes after a single iteration, implying that the problem with the current sample size does not deviate much from the one solved in the previous outer iteration, we increase the deviation of C_1 from 1 by a factor of 2 subject to C_1 not exceeding c_h . Formally, we set $C_1 \leftarrow \min(2C_1 - 1, c_h)$. If, on the other hand, the inner loop takes more than four iterations, we shrink the deviation of C_1 from 1 by a factor of 2, subject to C_1 reaching a minimum of c_0 , that is, we set $C_1 \leftarrow \max(c_0, \frac{1}{2}C_1 + \frac{1}{2})$. While our theory does not explicitly cover this “dynamic C_1 ” context, an extension of our theory to this case is straightforward. See comment (c) appearing after Theorem 5.3.

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

TABLE 1

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

Instance	First-stage size	Second-stage size	Reference
DEAK40×20	(40,20)	(30,20)	[18]
DEAK40×40	(40,20)	(60,40)	-
DEAK40×60	(40,20)	(90,60)	-
DEAK60×20	(60,30)	(30,20)	-
DEAK60×40	(60,30)	(60,40)	-
DEAK60×60	(60,30)	(90,60)	-
LandS	(4,2)	(12,7)	[40]
gbd	(17,4)	(10,5)	[26]
4node	(52,14)	(186,74)	[2]
pgp2	(4,2)	(16,7)	[32]
retail	(7,0)	(70,22)	[30]
cep	(8,5)	(15,7)	[32]
baa99-20	(20,0)	(250,40)	[65]
20-term	(63,3)	(764,124)	[43]
ssn	(89,1)	(706,175)	[64]

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

798 **7.2. Numerical results.** We first investigate the empirical performance of “Sequential-BP-
 799 $L(\Delta)$ ”, and its adaptation “Adaptive-seq-BP- $L(\Delta)$ ” into our proposed framework, against “PILD-
 800 ODA” which is arguably a state-of-the-art approach for solving 2SLPs with fixed recourse and fixed
 801 second-stage objective coefficients using the full set of scenarios [1]. Table 2 summarizes the results
 802 on our test instances. We recall that for all the sequential SAA approaches, the numbers shown in
 803 each row are calculated by taking the average of the corresponding values over 20 replications (10
 804 replications for ssn and 20term) of algorithm instantiation on five finite-sample instances.

805 **7.2.1. Computational results on the DEAK instance family.** We first present the per-
 806 formance of aforementioned algorithms on the DEAK instance family. Instances within this family
 807 share the same structure and vary by the problem sizes in terms of the number of variables and
 808 constraints. Experiments on these different instances allow us to see how the algorithms behave as
 809 the problem sizes change given the same underlying problem structure.

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

816 In comparing “Sequential-BP- $L(\Delta)$ ” against “Adaptive-seq-BP- $L(\Delta)$,” notice from Table 2 that
 817 the computational time for “Adaptive-seq-BP- $L(\Delta)$ ” is lower in most cases, while the total number

TABLE 2

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

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

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

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

TABLE 3

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

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

837 reduced, leading to less computational time in general. All options share similar behavior from the
838 standpoint of the width of the confidence interval and its coverage.

839 We next investigate the sensitivity of chosen parameters such as the sample size increase rate
840 for the proposed approaches. We observe from Table 2 and Table 3 that, as opposed to what has
841 been suggested in theory (Theorem 5.3), Algorithm 6.1 with a linear sample size schedule performs
842 competitively with the one with a geometric sample size schedule in our test instances. This may
843 be because the algorithm “Sequential-BP-L(Δ)” in Table 2 with a value $\Delta = 100$ mimics the
844 behavior of a geometric sequence. To validate this suspicion, Table 4 presents the performance of
845 “Adaptive-seq-BP-L(Δ)” implemented with a linear sample size schedule having a smaller increase
846 $\Delta = 10$ and “Adaptive-seq-fixed(c_1)” with a smaller geometric increase rate $c_1 = 1.1$. We also
847 display the performance of “Adaptive-seq-dyn(c_0, c_h)” with $c_0 = 1.05, c_h = 2$ and with C_1 starting
848 at 1.1, alongside these algorithms.

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

TABLE 4

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

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

856 slightly more computational time. On the other hand, the performance of Algorithm 6.1 with a
857 dynamic increase rate (option “Adaptive-seq-dyn(1.05,3)”) does not appear to be impacted much
858 from the choice of the starting increasing rate C_1 .

859 **7.2.2. Computational results on other test instances.** Finally, we present the perfor-
860 mance of the best adaptive sequential SAA options (according to the above experiments on DEAK
861 and DEAK-H instances) on an additional set of test instances that have a background in “real-
862 world” applications. In particular, we consider Algorithm “Adaptive-seq-BP-L(100)” and Algo-
863 rithm “Adaptive-seq-fixed(1.5)”. We consider Algorithm “Adaptive-seq-fixed(1.5)” rather than the
864 one with dynamic rate, “Adaptive-seq-dyn(1.05,3)”, as we find in our experiments that the param-
865 eters c_0 and c_h need to be fine tuned for specific instances in order to yield competitive performance.

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

TABLE 5

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

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

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

882 As noted earlier, the problem instances ssn and 20-term are interesting as negative examples,
883 where the proposed sequential sampling algorithms do not yield gains realized in other problem
884 instances. Instance ssn is challenging most probably due to the high inherent variance of the
885 underlying random variables and the associated computational challenge in solving the second-stage
886 problems while also reporting solution accuracy. For instance, observe from Table 5 that both
887 options Adaptive-seq-BP-L(100) and Adaptive-seq-fixed(1.5) fail to provide confidence intervals
888 with a satisfactory width within the stipulated time limit. We suspect that the variance associated
889 with the second-stage optimal cost, along with the strict nature of the stopping criterion, contributes
890 to ssn being in contrast with other test instances appearing in Table 5. The negative effect of such
891 high variance can be mitigated, at least in principle, by directly using variance reduction techniques,
892 or through alternative stopping ideas such as that proposed in [65].

893 The negative context presented by the instance 20-term appears to be different in spirit than
894 ssn. Specifically, observe that Adaptive-seq-BP-L(100) and Adaptive-seq-fixed(1.5) exhibit longer
895 computational times than the deterministic algorithm PILD-ODA on instances 20-term despite
896 having a small number of outer iterations and small sample sizes used in each outer iteration. In
897 fact, most of the computational effort is expended on solving the master problem, while the second-
898 stage subproblems can be solved efficiently. The increased effort in solving the master problem could
899 be because the “warmstart” feature that worked well for other instances is not as effective here,

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

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

920 Our theoretical results suggest that the optimality gap and the distance from the true solu-
921 tion set (of the generated stochastic iterates) converges to zero almost surely and in expectation.
922 Moreover, when the sample sizes are increased according to a geometric rate, the fastest possible
923 convergence rate under iid Monte Carlo sampling is preserved. This result is analogous to the
924 $\mathcal{O}(\epsilon^{-2})$ optimal complexity rate for deterministic non-smooth convex optimization. Slower sample
925 size increases result in a poorer convergence rate. Interestingly, the proposed framework also fa-
926 cilitates the use of dependent sampling schemes such as LHS, antithetic variates, and quasi-Monte
927 Carlo without affecting convergence or the lower bound on the rate results. The use of such variance
928 reduction ideas have been shown to be effective.

929 Our extensive numerical studies indicate that the proposed adaptive sequential SAA frame-
930 work is able to produce high-quality solutions to 2SLPs significantly more efficiently than existing
931 decomposition approaches that solve a single sample-path problem generated using a large sample
932 size. Such gains are principally due to the sequential framework, the progressive increase in sample
933 sizes in an optimal way, and the use of “warm starts” in solving the sample-path problems. Our
934 numerical experience has also revealed problem instances having certain challenging features that
935 are not directly addressed by the implementations that we have used for illustration. These chal-
936 lenges could be mitigated by using alternative solvers that exploit particular problem structures
937 and/or other termination criteria such as that proposed in [65].

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

942 **Acknowledgments.** We greatly appreciate the comments and suggestions of the associate
943 editor and two anonymous referees.

- 945 [1] W. VAN ACKOOIJ, W. DE OLIVEIRA, AND Y. SONG, *An adaptive partition-based level decomposition for solving*
 946 *two-stage stochastic programs with fixed recourse*, INFORMS Journal on Computing, 30 (2018), pp. 57–70.
- 947 [2] K. A. ARIYAWANSA AND A. J. FELT, *On a new collection of stochastic linear programming test problems*,
 948 INFORMS Journal on Computing, 16 (2004), pp. 291–299.
- 949 [3] G. BAYRAKSAN AND D. MORTON, *Assessing solution quality in stochastic programs*, Mathematical Program-
 950 ming, 108 (2006), pp. 495–514.
- 951 [4] G. BAYRAKSAN AND D. MORTON, *A sequential sampling procedure for stochastic programming*, Operations
 952 Research, 59 (2011), pp. 898–913.
- 953 [5] G. BAYRAKSAN AND P. PIERRE-LOUIS, *Fixed-width sequential stopping rules for a class of stochastic programs*,
 954 SIAM Journal on Optimization, 22 (2012), pp. 1518–1548.
- 955 [6] J. Y. BELLO-CRUZ AND W. DE OLIVEIRA, *Level bundle-like algorithms for convex optimization*, Journal of
 956 Global Optimization, 59 (2014), pp. 787–809, <https://doi.org/10.1007/s10898-013-0096-4>.
- 957 [7] P. BILLINGSLEY, *Probability and Measure*, Wiley, New York, NY., 1995.
- 958 [8] J. BIRGE, *State-of-the-art-survey – stochastic programming: Computation and applications*, INFORMS Journal
 959 on Computing, 9 (1997), pp. 111–133, <https://doi.org/10.1287/ijoc.9.2.111>.
- 960 [9] J. R. BIRGE AND F. LOUVEAUX, *Introduction to stochastic programming*, Springer Science & Business Media,
 961 2011.
- 962 [10] R. BOLLAPRAGADA, R. BYRD, AND J. NOCEDAL, *Adaptive sampling strategies for stochastic optimization*, SIAM
 963 Journal on Optimization, 28 (2018), pp. 3312–3343.
- 964 [11] L. BOTTOU, F. E. CURTIS, AND J. NOCEDAL, *Optimization methods for large-scale machine learning*, SIAM
 965 Review, 60 (2018), pp. 223–311.
- 966 [12] S. BOUCHERON, G. LUGOSI, AND P. MASSART, *Concentration Inequalities and Model Selection*, Springer, Berlin,
 967 Germany, 2007.
- 968 [13] S. BOUCHERON, G. LUGOSI, AND P. MASSART, *Concentration Inequalities: A Nonasymptotic Theory of Inde-
 969 pendence*, Oxford University Press, Cambridge, UK, 2013.
- 970 [14] M. CHEN AND S. MEHROTRA, *Self-concordance and decomposition based interior point methods for the two
 971 stage stochastic convex optimization problem*, SIAM Journal on Optimization, 21 (2011), pp. 1667–1687.
- 972 [15] M. CHEN, S. MEHROTRA, AND D. PAPP, *Scenario generation for stochastic optimization problems via the sparse
 973 grid method*, Computational Optimization and applications, 62 (2015), pp. 669–692.
- 974 [16] Y. S. CHOW AND H. E. ROBBINS, *On the asymptotic theory of fixed-width confidence intervals for the mean*,
 975 Annals of Mathematical Statistics, 36 (1965), pp. 457–462.
- 976 [17] G. B. DANTZIG, *Linear programming under uncertainty*, Management Science (pre-1986), 1 (1955), p. 197.
- 977 [18] I. DEÁK, *Testing successive regression approximations by large-scale two-stage problems*, Annals of Operations
 978 Research, 186 (2011), pp. 83–99.
- 979 [19] G. DENG AND M. C. FERRIS, *Variable-number sample-path optimization*, Mathematical Programming, (2009),
 980 pp. 81–109.
- 981 [20] L. DÜMBGEN, S. A. VAN DE GEER, M. C. VERAAR, AND J. A. WELLNER, *Nemirovski’s inequalities revisited*,
 982 The American Mathematical Monthly, 117 (2010), pp. 138–160.
- 983 [21] J. DUPAČOVÁ AND R. J. B. WETS, *Asymptotic behavior of statistical estimators and of optimal solutions of
 984 stochastic optimization problems*, The Annals of Statistics, 16 (1988), pp. 1517–1549.
- 985 [22] R. DURRETT, *Probability: Theory and Examples*, Cambridge University Press, New York, NY, 2010.
- 986 [23] Y. ERMOLIEV AND A. GAIVORONSKI, *Stochastic quasigradient methods for optimization of discrete event sys-
 987 tems*, Annals of Operations Research, 39 (1992), pp. 1–39.
- 988 [24] Y. M. ERMOLIEV, *Stochastic quasigradient methods and their application to system optimization.*, Stochastics,
 989 9 (1983), pp. 1–36.
- 990 [25] C. I. FÁBIÁN AND Z. SZŐKE, *Solving two-stage stochastic programming problems with level decomposition*,
 991 Computational Management Science, 4 (2007), pp. 313–353.
- 992 [26] A. FERGUSON AND G. DANTZIG, *The allocation of aircraft to routes? an example of linear programming under
 993 uncertain demand*, Management science, 3 (1956), pp. 45–73.
- 994 [27] P. GLASSERMAN, *Monte Carlo methods in financial engineering*, vol. 53, Springer Science & Business Media,
 995 2013.
- 996 [28] P. GLYNN AND G. INFANGER, *Simulation-based confidence bounds for two-stage stochastic programs*, Mathe-
 997 matical Programming, 138 (2013), pp. 15–42.
- 998 [29] F. HASHEMI, S. GHOSH, AND R. PASUPATHY, *On adaptive sampling rules for stochastic recursions*, in Proceed-
 999 ings of the 2014 Winter Simulation Conference, A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley,
 1000 and J. A. Miller, eds., Piscataway, NJ, 2014, Institute of Electrical and Electronics Engineers, Inc.

- 1001 [30] Y. HERER, M. M. TZUR, AND E. YÜCESAN, *The multilocation transshipment problem*, IIE transactions, 38
1002 (2006), pp. 185–200.
- 1003 [31] J. HIGLE AND S. SEN, *Stochastic decomposition: An algorithm for two-stage linear programs with recourse*,
1004 Mathematics of operations research, 16 (1991), pp. 650–669.
- 1005 [32] J. HIGLE AND S. SEN, *Stochastic decomposition: a statistical method for large scale stochastic linear program-*
1006 *ming*, vol. 8, Springer Science & Business Media, 2013.
- 1007 [33] T. HOMEM-DE-MELLO, *Variable-sample methods for stochastic optimization*, ACM Transactions on Modeling
1008 and Computer Simulation (TOMACS), 13 (2003), pp. 108–133.
- 1009 [34] T. HOMEM-DE-MELLO AND G. BAYRAKSAN, *Monte Carlo sampling-based methods for stochastic optimization*,
1010 in Reviews, Surveys in Operations Research and Management Science, Elsevier, 2014, pp. 56–85.
- 1011 [35] S. KIM, R. PASUPATHY, AND S. G. HENDERSON, *A guide to SAA*, in Encyclopedia of Operations Research and
1012 Management Science, M. Fu, ed., Hillier and Lieberman OR Series, Elsevier, 2014.
- 1013 [36] P. L'ECUYER, *Randomized quasi-monte carlo: An introduction for practitioners*, in Monte Carlo and Quasi-
1014 Monte Carlo Methods, Springer, 2018.
- 1015 [37] P. L'ECUYER, C. LÉCOT, AND B. TUFFIN, *A randomized Quasi-Monte Carlo simulation method for Markov*
1016 *Chains*, Operations Research, 56 (2008), pp. 958–975.
- 1017 [38] M. LEDOUX AND M. TALAGRAND, *Probability in Banach Spaces: Isoperimetry and Processes*, Springer, Berlin,
1018 Germany, 2011.
- 1019 [39] C. LEMARÉCHAL, A. NEMIROVSKII, AND Y. NESTEROV, *New variants of bundle methods*, Mathematical pro-
1020 gramming, 69 (1995), pp. 111–147.
- 1021 [40] F. LOUVEAUX AND Y. SMEERS, *Optimal investments for electricity generation: a stochastic model and a test-*
1022 *problem*, in Numerical Techniques for Stochastic Optimization, Springer-Verlag, Berlin, 1988, pp. 33–64.
- 1023 [41] J. LUEDTKE AND S. AHMED, *A sample approximation approach for optimization with probabilistic constraints*,
1024 SIAM Journal on Optimization, 19 (2008), pp. 674–699.
- 1025 [42] P. L'ECUYER, *Randomized quasi-monte carlo: An introduction for practitioners*, in 12th International Confer-
1026 *ence on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (MCQMC2016)*, August
1027 2016.
- 1028 [43] W.-K. MAK, D. P. MORTON, AND R. K. WOOD, *Monte carlo bounding techniques for determining solution*
1029 *quality in stochastic programs*, Operations research letters, 24 (1999), pp. 47–56.
- 1030 [44] V. DE. MATOS, A. PHILPOTT, AND E. FINARDI, *Improving the performance of stochastic dual dynamic pro-*
1031 *gramming*, Journal of Computational and Applied Mathematics, 290 (2015), pp. 196–208.
- 1032 [45] M. D. MCKAY, R. J. BECKMAN, AND W. J. CONOVER, *Comparison of three methods for selecting values of*
1033 *input variables in the analysis of output from a computer code*, Technometrics, 21 (1979), pp. 239–245.
- 1034 [46] A. NEDIĆ AND D. BERTSEKAS, *Convergence rate of incremental subgradient algorithms*, in Stochastic optimiza-
1035 *tion: algorithms and applications*, Springer, 2001, pp. 223–264.
- 1036 [47] B. L. NELSON, *Antithetic-variate splitting for steady-state simulations*, European journal of operational research,
1037 36 (1988), pp. 360–370.
- 1038 [48] B. L. NELSON, *Foundations and Methods of Stochastic Simulation: A First Course*, Springer, New York, NY.,
1039 2013.
- 1040 [49] Y. NESTEROV, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science &
1041 Business Media, 2004.
- 1042 [50] Y. NESTEROV, *Lectures on convex optimization*, Springer, second ed., 2018.
- 1043 [51] W. DE. OLIVEIRA AND C. SAGASTIZÁBAL, *Level bundle methods for oracles with on demand accuracy*, Opti-
1044 mization Methods and Software, 29 (2014), pp. 1180–1209.
- 1045 [52] A. B. OWEN, *Monte Carlo theory, methods and examples*, 2013.
- 1046 [53] R. PASUPATHY, *On choosing parameters in retrospective-approximation algorithms for stochastic root finding*
1047 *and simulation optimization*, Operations Research, 58 (2010), pp. 889–901.
- 1048 [54] R. PASUPATHY, P. W. GLYNN, S. GHOSH, AND F. HASHEMI, *On sampling rates in simulation-based recursions.*,
1049 SIAM Journal on Optimization, 28 (2018), pp. 45–73.
- 1050 [55] R. PASUPATHY AND B. W. SCHMEISER, *Retrospective-approximation algorithms for multidimensional stochastic*
1051 *root-finding problems*, ACM TOMACS, 19 (2009), pp. 5:1–5:36.
- 1052 [56] R. PASUPATHY AND Y. SONG, *Online supplementary document for: Adaptive sequential SAA for solving two-*
1053 *stage stochastic linear programs*, 2020, [http://www.optimization-online.org/DB_HTML/2019/02/7068.](http://www.optimization-online.org/DB_HTML/2019/02/7068.html)
1054 [html](http://www.optimization-online.org/DB_HTML/2019/02/7068.html).
- 1055 [57] E. POLAK AND J. ROYSET, *Efficient sample sizes in stochastic nonlinear programming*, Journal of Computa-
1056 tional and Applied Mathematics, 217 (2008), pp. 301–310.
- 1057 [58] B. POLYAK, *Introduction to Optimization*, Optimization Software In, New York, NY, 1987.
- 1058 [59] B. T. POLYAK, *Nonlinear programming methods in the presence of noise.*, Mathematical Programming, 1

- 1059 (1978), pp. 87–97.
- 1060 [60] S. M. ROBINSON, *Analysis of sample-path optimization*, Mathematics of Operations Research, 21 (1996),
1061 pp. 513–528.
- 1062 [61] J. ROYSET, *On sample size control in sample average approximations for solving smooth stochastic programs*,
1063 Computational Optimization and Applications, 55 (2013), pp. 265–309.
- 1064 [62] J. O. ROYSET AND R. SZECHTMAN, *Optimal budget allocation for sample average approximation*, Operations
1065 Research, 61 (2013), pp. 762–776.
- 1066 [63] A. RUSZCZYŃSKI AND A. SHAPIRO, eds., *Stochastic Programming. Handbook in Operations Research and Man-*
1067 *agement Science*, Elsevier, New York, NY., 2003.
- 1068 [64] S. SEN, R. DOVERSPIKE, AND S. COSARES, *Network planning with random demand*, Telecommunication systems,
1069 3 (1994), pp. 11–30.
- 1070 [65] S. SEN AND Y. LIU, *Mitigating uncertainty via compromise decisions in two-stage stochastic linear program-*
1071 *ming: Variance reduction*, Operations Research, 64 (2016), pp. 1422–1437.
- 1072 [66] A. SHAPIRO, *Asymptotic behavior of optimal solutions in stochastic programming*, Mathematics of Operations
1073 Research, 18 (1993), pp. 829–845.
- 1074 [67] A. SHAPIRO, D. DENTCHEVA, AND A. RUSZCZYŃSKI, *Lectures on stochastic programming: modeling and theory*,
1075 SIAM, 2014.
- 1076 [68] A. SHAPIRO AND T. HOMEM-DE-MELLO, *A simulation-based approach to two-stage stochastic programming with*
1077 *recourse*, Mathematical Programming, 81 (1998), pp. 301–325.
- 1078 [69] A. SHAPIRO AND T. HOMEM-DE-MELLO, *On the rate of convergence of optimal solutions of monte carlo ap-*
1079 *proximations of stochastic programs*, SIAM Journal on Optimization, 11 (2000), pp. 70–86.
- 1080 [70] S. SHASHAANI, F. S. HASHEMI, AND R. PASUPATHY, *ASTRO-DF: A class of adaptive sampling trust-region al-*
1081 *gorithms for derivative-free simulation optimization.*, SIAM Journal on Optimization, 28 (2018), pp. 3145–
1082 3176.
- 1083 [71] M. STEIN, *Large sample properties of simulations using latin hypercube sampling*, Technometrics, 29 (1987),
1084 pp. 143–151.
- 1085 [72] R. STOCKBRIDGE AND G. BAYRAKSAN, *Variance reduction in Monte Carlo sampling-based optimality gap es-*
1086 *timators for two-stage stochastic linear programming*, Computational Optimization and Applications, 64
1087 (2016), pp. 407–431.
- 1088 [73] M. TALAGRAND, *Sharper bounds for gaussian and empirical processes*, Annals of Probability, 22 (1994), pp. 28–
1089 76.
- 1090 [74] H. WANG, R. PASUPATHY, AND B. W. SCHMEISER, *Integer-ordered simulation optimization using R-SPLINE:*
1091 *Retrospective search using piecewise-linear interpolation and neighborhood enumeration*, ACM TOMACS,
1092 23 (2013).
- 1093 [75] D. WILLIAMS, ed., *Probability with Martingales*, Cambridge University Press, Cambridge, UK, 1991.
- 1094 [76] C. WOLF, C. FÁBIÁN, A. KOBERSTEIN, AND L. SUHL, *Applying oracles of on-demand accuracy in two-stage*
1095 *stochastic programming—a computational study*, European Journal of Operational Research, 239 (2014),
1096 pp. 437–448.
- 1097 [77] G. ZHAO, *A log-barrier method with Bender’s decomposition for solving two-stage stochastic linear programs*,
1098 Mathematical Programming, Series A, 90 (2001), pp. 501–536.

1099 Appendix A. Proof of Lemma 4.1.

1100 *Proof.* From the postulates of the lemma, it is seen that the function f has to be real-valued and
1101 convex on \mathcal{X} . Also, real-valued convex functions are known to be continuous on the relative interior
1102 of the effective domain. Assuming that \mathcal{X} is a proper compact subset of this relative interior, the
1103 function f attains its minimum $v^* = \min_{x \in \mathcal{X}} \{f(x)\}$.

1104 We first prove assertion (a). Since \mathcal{X} is bounded, we know that $(x_k)_{k \geq 1}$ has at least one limit
1105 point. Consider any limit point \tilde{x}^* of $(x_k)_{k \geq 1}$, and a sub-sequence $(x_{k_j})_{j \geq 1}$ that satisfies $x_{k_j} \rightarrow \tilde{x}^*$
1106 as $j \rightarrow \infty$. To arrive at a contradiction, let us suppose that \tilde{x}^* is sub-optimal to f over \mathcal{X} , implying
1107 that the directional derivative $f'(\tilde{x}^*, u(\tilde{x}^*, f))$ of f along the unit steepest descent direction $u(\tilde{x}^*, f)$
1108 (of f at \tilde{x}^*) satisfies $f'(\tilde{x}^*, u(\tilde{x}^*, f)) < 0$. Then, since f_{k_j} converges to f uniformly, we are assured
1109 that there exists $\tilde{\delta} > 0$ and $t \leq \tilde{\delta}$ such that for large enough j ,

$$1110 \quad (A.1) \quad f_{k_j}(x_{k_j} + tu(x_{k_j}, f_{k_j})) \leq f_{k_j}(x_{k_j}) + \frac{1}{2}tf'(\tilde{x}^*, u(\tilde{x}^*, f)).$$

1111 Recalling the notation $\delta_{k_j+1} := \sup_{x \in \mathcal{X}} |f_{k_j+1}(x) - f_{k_j}(x)|$ and $v_{k_j+1}^* := \min \{f_{k_j+1}(x), x \in \mathcal{X}\}$, the
 1112 inequality in (A.1) implies that for $j \geq j^*$,

$$1113 \quad (\text{A.2}) \quad v_{k_j+1}^* \leq f_{k_j}(x_{k_j}) + \frac{\tilde{\delta}}{2} f'(\tilde{x}^*, u(\tilde{x}^*, f)) + \delta_{k_j+1}.$$

1114 Since x_{k_j+1} is ϵ_{k_j+1} -optimal to f_{k_j+1} over \mathcal{X} , the inequality in (A.2) implies that

$$1115 \quad (\text{A.3}) \quad f_{k_j+1}(x_{k_j+1}) \leq f_{k_j}(x_{k_j}) + \frac{\tilde{\delta}}{2} f'(\tilde{x}^*, u(\tilde{x}^*, f)) + \delta_{k_j+1} + \epsilon_{k_j+1}$$

1116 Now notice that (A.3) presents a contradiction because the left hand side is tending to $f(\tilde{x}^*)$ since we
 1117 have assumed $(x_{k_j})_{j \geq 1} \rightarrow \tilde{x}^*$, but the right hand side is tending to $f(\tilde{x}^*) + \frac{\tilde{\delta}}{2} f'(\tilde{x}^*, u(\tilde{x}^*, f)) < f(\tilde{x}^*)$.
 1118 Conclude that assertion (a) in Lemma 4.1 holds.

1119 Let us now prove the assertion (b) of Lemma 4.1. Consider selections $x^* \in \arg \min_{x \in \mathcal{X}} \{f(x)\}$ and
 1120 $x_j^* \in \arg \min_{x \in \mathcal{X}} \{f_j(x)\}, j \geq 1$. (The selections $x^*, (x_j^*)_{j \geq 1}$ exist since the functions f and $f_j, j \geq 1$ are
 1121 continuous on the compact set \mathcal{X} .) Now write

$$1122 \quad f(x_k) - f(x^*) = f(x_k) - f_k(x_k) + f_k(x_k) - f(x^*) \\
 1123 \quad (\text{A.4}) \quad = f(x_k) - f_{n+1}(x_k) + T_n + S_n + f_{n+1}(x_{n+1}) - f(x^*),$$

1125 where

$$1126 \quad (\text{A.5}) \quad T_n := \sum_{j=k}^n f_{j+1}(x_k) - f_j(x_k) \leq \sum_{j=k}^n \delta_j$$

1127 and

$$1128 \quad S_n := \sum_{j=k}^n f_j(x_j) - f_{j+1}(x_{j+1}) \\
 1129 \quad (\text{A.6}) \quad = \sum_{j=k}^n (f_j(x_j) - v_j^*) + \sum_{j=k}^n (v_{j+1}^* - f_{j+1}(x_{j+1})) + \sum_{j=k}^n (v_j^* - v_{j+1}^*). \\
 1130$$

1131 Now, notice that since x_j is ϵ_j -optimal to $f_j(\cdot)$ over \mathcal{X} , $f_j(x_j) - v_j^* \leq \epsilon_j$ implying that

$$1132 \quad (\text{A.7}) \quad f_j(x_j) - v_j^* \leq \epsilon_j.$$

1133 Similarly,

$$1134 \quad (\text{A.8}) \quad f_{j+1}(x_{j+1}) - v_{j+1}^* \leq \epsilon_{j+1}.$$

1135 Also,

$$1136 \quad v_j^* - v_{j+1}^* \leq \max\{f_j(x_{j+1}^*) - f_{j+1}(x_{j+1}^*), f_j(x_j^*) - f_{j+1}(x_j^*)\} \\
 1137 \quad (\text{A.9}) \quad \leq \delta_{j+1}.$$

1139 Plugging (A.7), (A.8), and (A.9) in (A.6), we get

$$1140 \quad (\text{A.10}) \quad S_n \leq \sum_{j=k}^n \epsilon_j + \sum_{j=k}^n \epsilon_{j+1} + \sum_{j=k}^n \delta_{j+1}.$$

1141 Now plug (A.5) and (A.10) in (A.4), and send $n \rightarrow \infty$ for fixed k to get

$$1142 \quad f(x_k) - f(x^*) \leq 2 \sum_{j=k}^{\infty} (\epsilon_j + \delta_j), \quad \square$$

1143 proving the assertion (b) of Lemma 4.1.

1144 Part (c) of Lemma 4.1 follows simply from the assertion in part (b) and the assumed growth
1145 rate condition.

1146 **Appendix B. A conceptual level method for solving convex non-smooth optimiza-**
1147 **tion problem.** Consider a generic convex non-smooth optimization problem:

$$1148 \quad (\text{B.1}) \quad r^* := \min_{x \in \mathcal{X}} r(x),$$

1149 where \mathcal{X} is a nonempty convex and closed set in \mathbb{R}^n , and $r(\cdot)$ is a convex and possibly non-smooth
1150 function. We consider the conceptual algorithm proposed in [6][Algorithm 1], which is summarized
1151 in Algorithm B.1. The key assumption is the availability of finding a level target r_t^{lev} for each
1152 iteration t that is lower-bounded by r^* .

1153 **ASSUMPTION 5.** (See also equations (4) and (5) in [6]) For each iteration t , there exists a level
1154 target parameter r_t^{lev} such that

$$1155 \quad (\text{B.2}) \quad r(x_t) > r_t^{lev} \geq r^*, \quad \forall t, \quad \text{and} \quad \lim_{t \rightarrow \infty} r_t^{lev} = r^*.$$

1156 For example, the requirement in Assumption 5 holds, if the optimal objective value r^* is known.
1157 In this case r_t^{lev} can be simply selected as: $r_t^{lev} = (1 - \lambda)r(x_t) + \lambda r^*$ for some $\lambda \in (0, 1)$.

Algorithm B.1 A conceptual level bundle algorithm for non-smooth convex optimization [6][Algorithm 1].

Input: Given $x_0 \in \mathcal{X}$, obtain $r(x_0)$ and $g_0 \in \partial r(x_0)$. Set $\hat{x} = x_0, t = 0$.

while stopping criterion is not met **do**

 Select r_t^{lev} satisfying (B.2). Compute the next iterate:

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

if stopping criterion is met **then**

 Terminate the procedure.

else

 Compute $r(x_{t+1})$ and $g_{t+1} \in \partial r(x_{t+1})$. Set $t \leftarrow t + 1$.

end if

end while

1158 **LEMMA B.1.** Let x_* be an optimal solution to (B.1), given an initial iterate \hat{x} , Algorithm B.1
1159 will run at most $\mathcal{O} \left(\frac{\Lambda \|x_* - \hat{x}\|}{\epsilon} \right)^2$ iterations before stopping with a stopping tolerance parameter ϵ ,
1160 where Λ is the Lipschitz constant on the objective function of (B.1).

1161 *Proof.* The proof of this lemma directly follows the arguments stipulated in [6]. □

1162 **Appendix C. The adaptive partition-based level decomposition with on-demand**
 1163 **accuracy and its warmstarting procedure.**

1164 Since Algorithm B.1 is only a conceptual algorithm (under Assumption B.2), for our numerical
 1165 experiments, we use a variant of level bundle method that is practically implementation, the adap-
 1166 tive partition-based level decomposition method, recently developed by [1]. For details about this
 1167 algorithm, please refer to [1]. The performance of this algorithm has shown to be more competitive
 1168 than most of the state-of-the-art algorithms for solving 2SLPs with a finite sample. We use this
 1169 algorithm as the Solver- \mathcal{A} for solving the sample-path problem (P_ℓ) in each outer iteration ℓ during
 1170 our adaptive sequential SAA procedure. We argue that this algorithm, denoted as PILD-ODA,
 1171 together with the following warmstart procedure, approximates the behavior of the conceptual
 1172 algorithm (Algorithm B.1) quite well.

1173 We next illustrate how information from solving the sample-path problem (P_ℓ) with sample
 1174 \mathcal{M}_ℓ at previous outer iterations $1, 2, \dots, \ell$ can be leveraged to solve the sample-path problem ($P_{\ell+1}$)
 1175 with sample $\mathcal{M}_{\ell+1}$. We note that this information is not necessarily only about the solution \hat{x}^ℓ , and
 1176 could contain, e.g., dual multipliers collected from solving second-stage subproblems (2.2) during
 1177 the inner iterations of Algorithm PILD-ODA, which is possible precisely because of the following
 1178 key assumption of fixed recourse, which has been exploited in various efficient methods to solve
 1179 2SLPs, such as [31, 51, 76, 1].

1180 ASSUMPTION 6. *The 2SLP problem (P) has fixed recourse, that is, the second-stage cost vector*
 1181 *$d \in \mathbb{R}^{n_2}$ and recourse matrix $W \in \mathbb{R}^{m_2 \times n_2}$ are not subject to uncertainty. Consequently, the dual*
 1182 *polyhedron of the second-stage problem (2.2) is identical for all realization of random variables*
 1183 *involved in problem (P):*

$$1184 \mathcal{D} := \{\lambda \mid W^\top \lambda \leq d\}.$$

1185 At the ℓ -th outer iteration, let $\mathcal{D}^\ell \subseteq \mathcal{D}$ be the set of dual multipliers accumulated so far (optimal
 1186 dual multipliers to the second-stage subproblems (2.2) encountered during the solution procedure),
 1187 the master problem to the sample-path problem (P_ℓ) with sample \mathcal{M}_ℓ can be initialized as:

$$1188 \text{(C.1a)} \quad \min_{x \in \mathcal{X}} c^\top x + \frac{1}{m_\ell} \theta$$

$$1189 \text{(C.1b)} \quad \text{s.t. } \theta \geq \sum_{i=1}^{m_\ell} \lambda_i^\top (h(\xi_i) - T(\xi_i)x), \quad \forall (\lambda_i)_{i=1}^{m_\ell} \in \overbrace{\mathcal{D}^\ell \times \mathcal{D}^\ell \times \dots \times \mathcal{D}^\ell}^{m_\ell \text{ times}}$$

1191 When $|\mathcal{D}^\ell|$ gets large, the number of constraints (C.1b) may get excessively large. Therefore,
 1192 one could solve (C.1) using a cutting plane algorithm by adding them on the fly.

1193 We implement this warm start procedure (Algorithm C.1) prior to executing Algorithm PILD-
 1194 ODA to solve the sample-path problem (P_ℓ), obtain \hat{x} and \hat{z} and initialize Algorithm PILD-ODA
 1195 by setting the starting solution to be $x^0 = \hat{x}$. We keep collecting optimal dual solutions λ 's
 1196 corresponding to the second-stage subproblems solved throughout the process of Algorithm 6.1 into
 1197 \mathcal{D}^ℓ . To avoid $|\mathcal{D}^\ell|$ to get too large, we discard a dual vector if the euclidean distance between this
 1198 vector and an existing vector in \mathcal{D}^ℓ is lower than a given threshold. We note that we could also
 1199 choose to use the set of samples from the previous iteration \mathcal{M}_ℓ as a part of the new set of samples
 1200 $\mathcal{M}_{\ell+1}$, at least periodically, in which case more warm starts are expected [4, 5].

Algorithm C.1 Initialization of the sample-path problem (P_ℓ) with a set of dual multipliers \mathcal{D}^ℓ .

Input: A starting solution $\hat{x}^0 \in \mathcal{X}$, a collection of dual multipliers \mathcal{D}^ℓ . Start the master problem (C.1) with none of constraints (C.1b).

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 ξ_i^ℓ , $i = 1, 2, \dots, m_\ell$ **do**

 Compute $\lambda_i \in \arg \max_{\lambda \in \mathcal{D}^\ell} \lambda^\top (h(\xi_i^\ell) - T(\xi_i^\ell)\hat{x})$.

end for

Arrange a constraint of type (C.1b) 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} .
