IMA Journal of Numerical Analysis (2023) **00**, 1–37
https://doi.org/10.1093/imanum/drac083

Adaptive sampling strategies for risk-averse stochastic optimization with constraints

Florian Beiser
Mathematics and Cybernetics, SINTEF Digital, Forskningsveien 1, 0373 Oslo, Norway

Brendan Keith
Division of Applied Mathematics, Brown University, Providence, RI 02912, United States

Simon Urbainczyk*
Maxwell Institute for Mathematical Sciences and Department of Actuarial Mathematics and Statistics, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom
*Corresponding author: su2004@hw.ac.uk

and

Barbara Wohlmuth
Department of Mathematics, Technical University of Munich, Boltzmannstraße 3, 80333 Munich, Germany

[Received on 28 August 2021; revised on 28 September 2022]

We introduce adaptive sampling methods for stochastic programs with deterministic constraints. First, we propose and analyze a variant of the stochastic projected gradient method, where the sample size used to approximate the reduced gradient is determined on-the-fly and updated adaptively. This method is applicable to a broad class of expectation-based risk measures, and leads to a significant reduction in the individual gradient evaluations used to estimate the objective function gradient. Numerical experiments with expected risk minimization and conditional value-at-risk minimization support this conclusion, and demonstrate practical performance and efficacy for both risk-neutral and risk-averse problems. Second, we propose an SQP-type method based on similar adaptive sampling principles. The benefits of this method are demonstrated in a simplified engineering design application, featuring risk-averse shape optimization of a steel shell structure subject to uncertain loading conditions and model uncertainty.

Keywords: stochastic optimization; sample size selection; constrained optimization; portfolio optimization; shape optimization.

1. Introduction

In this article, we consider the following general class of stochastic programs:

$$\min_{x\in C} \Big\{ F(x) = \mathcal{R}[f(x;\xi)] \Big\}. \tag{1.1}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function, ξ is a random variable on a probability space denoted $(\Xi, \mathscr{B}, \mathbb{P})$, $C \subseteq \mathbb{R}^n$ is a closed subdomain and $\mathcal{R} : L^1(\Xi, \mathscr{B}, \mathbb{P}) \to \mathbb{R}$ is a *coherent risk measure* (Artzner *et al.*, 1999). A canonical example of the objective function $F(x)$ is the expected value of f at x, namely

$$\mathbb{E}[f(x;\xi)] = \int_{\Xi} f(x;\xi)\,\mathrm{d}\mathbb{P}(\xi). \tag{1.2}$$

In this situation, $\mathcal{R} = \mathbb{E}$ and we say that $F(x) = \mathbb{E}[f(x;\xi)]$ defines the expected risk at x.

With the expected value risk measure, $\mathcal{R} = \mathbb{E}$, the stochastic program (1.1) only seeks out the point $x = x^*$ that minimizes $f(x)$ on average. More general risk measures are often used when it is desirable to optimize for low probability events. The other risk measure considered in this work is the conditional value-at-risk (CVaR) (Rockafellar & Uryasev, 2000, 2002). The CVaR at confidence level $\beta \in (0,1)$, denoted CVaR_β, is a well-established decision-making tool in finance (Krokhmal *et al.*, 2002; Shapiro *et al.*, 2009), and is becoming increasingly prominent in engineering (Rockafellar & Royset, 2015; Kouri & Surowiec, 2016; Yang & Gunzburger, 2017; Kouri & Surowiec, 2018; Chaudhuri *et al.*, 2020a,b). In this work, stochastic programs featuring the risk measure $\mathcal{R} = \mathbb{E}$ are referred to as *risk-neutral*; meanwhile, those involving the risk measure $\mathcal{R} = \mathrm{CVaR}_\beta$ are referred to as *risk-averse*.

In many practical problems, the integral in (1.2) cannot be computed directly because the space Ξ is high-dimensional (Kouri & Shapiro, 2018; Bottou *et al.*, 2018). One common approach to approximate the integral is to draw a set of i.i.d. samples $S = \{\xi_i\}$, $i = 1, \dots, N$, of the random variable ξ and substitute $F(x)$ with the following empirical estimate of the expected risk:

$$F_S(x) = \frac{1}{N}\sum_{i=1}^{N} f(x;\xi_i). \tag{1.3}$$

When the sample set S is fixed, stochastic optimization methods that employ this type of approximation of the objective function are commonly referred to as sample average approximation methods (Shapiro *et al.*, 2009; Royset & Szechtman, 2013; Kouri & Shapiro, 2018). In this paper, we design algorithms that *adaptively* determine the size of the set S in each iteration.

Since in high-dimensional settings the computational cost of optimizing (1.1) typically hinges on the total number of gradient evaluations of f, we aim at keeping this number as small as possible. To this end, we propose a sampling strategy that adaptively balances the algorithm's sampling error and optimization error throughout the entire optimization process. The strategy works by updating the size of the sample set $S = S_k$ together with the point $x = x_k$. More precisely, at each iteration k, we generate a sample set S_k based on gradient evaluations at x_k to compute an updated point x_{k+1}. This leads to robust and practical methods that can treat many stochastic optimization problems efficiently.

1.1 *Literature review and motivation*

There are many articles on stochastic optimization methods with dynamic sample sizes (Homem-De-Mello, 2003; Friedlander & Schmidt, 2012; Byrd *et al.*, 2012; Royset & Szechtman, 2013; Kouri *et al.*, 2013; De *et al.*, 2017; Cartis & Scheinberg, 2018; Pasupathy *et al.*, 2018; Roosta-Khorasani & Mahoney, 2019; Bollapragada *et al.*, 2018a,b; Bottou *et al.*, 2018; Bollapragada *et al.*, 2019; Paquette & Scheinberg, 2020). Nevertheless, very few of these works consider *constrained* optimization problems or risk-averse settings in detail (Royset & Szechtman, 2013); the majority of the present literature focuses on *unconstrained* stochastic programs, such as those commonly found in machine learning. One notable exception is the recent contributions by Xie *et al.* (Xie *et al.*, 2020; Xie, 2021), which appeared online shortly after an earlier version of this work (Beiser *et al.*, 2020) and complements our contribution by, among other novelties, introducing alternative adaptive sampling strategies with separate convergence proofs, as well as analyzing composite optimization problems. A thorough comparison of this work, and Xie *et al.* (2020) and Xie (2021), is given at multiple points later in the text; see Remarks 2.9, 2.14 and 3.3.

The difficulty in generalizing previous work on adaptive sampling to constrained optimization problems lies in developing new criteria to quantify and balance the statistical and optimization errors, while accounting for the influence of the constraint set. We refer the interested reader to Xie *et al.* (2020, Section 1) for an overview of the pitfalls of applying well-established adaptive sampling strategies designed for unconstrained problems to the constrained setting. For constrained stochastic programs, most contemporary methods rely on *a priori* error analysis that results in a prescribed growth in the sample size (Friedlander & Schmidt, 2012; Royset & Szechtman, 2013; Bollapragada *et al.*, 2019). In this work, as was also done in Xie *et al.* (2020) and Xie (2021), we choose to estimate the correct sample size *a posteriori* and update it adaptively.

Our work has a great deal in common with the adaptive sampling approaches taken in Byrd *et al.* (2012), Bollapragada *et al.* (2018a) and Bollapragada *et al.* (2018b). In order to highlight the primary similarities, we note that in our approach to (stochastic) projected gradient descent (Subsections 2.2 and 2.3 and Section 3), we arrive at a condition similar to the ‘norm test’ introduced for unconstrained optimization in Carter (1991) and later used in Byrd *et al.* (2012).[1] A similar test also appears in our sequential quadratic programming (SQP) algorithm (cf. Section 6).

Another approach to deal with stochastic programs with deterministic constraints appeared online a few months after the initial version of this article (Na *et al.*, 2021a,b). While Xie *et al.* (2020) present an adaptive sampling algorithm that shows similar properties to our methods, Na *et al.* (2021a) use an independent approach to develop a novel stochastic line search procedure and associated stochastic SQP algorithm that can also be extended to work for inequality-constrained stochastic programs based on active-set strategies (Na *et al.*, 2021b).

The present work arose from a need to develop efficient stochastic programming methods for large-scale decision-making problems; especially in engineering design, where each individual sample computation is extremely costly (Kouri *et al.*, 2013; Ion *et al.*, 2018; Shi *et al.*, 2018; Zou *et al.*, 2019; Geiersbach *et al.*, 2020). In these high-cost scenarios, one wishes to evaluate as few samples as possible. It is well established that the expected risk (1.2) is often unsuitable to predict immediate and long-term performance, manufacturing and maintenance costs, system response, levels of damage and numerous other quantities of interest (Rockafellar & Royset, 2010; Kouri & Surowiec, 2016; Kouri & Shapiro, 2018; Kouri & Surowiec, 2018). Therefore, today’s industrial problems are made even more challenging because they typically require a risk-averse formulation (Rockafellar & Royset, 2015; Kodakkal *et al.*, 2022).

1.2 *Layout*

Apart from the expected value operator $\mathbb{E}$, the CVaR is the only risk measure we consider in detail. It is well known that this risk measure can be reformulated as a separate optimization problem involving $\mathbb{E}$; cf. Rockafellar & Uryasev (2000, 2002) and Section 4. This observation informs the layout of the paper by allowing us to first focus on the case $\mathcal{R} = \mathbb{E}$ and then deal with the treatment of risk-averse problems in the later sections. A large family of other important risk measures, including the entropic risk and the conditional entropic risk (Kouri & Surowiec, 2018), have a similar reformulation involving $\mathbb{E}$ (Rockafellar & Uryasev, 2013; Kouri & Surowiec, 2018), which leads us to conclude that there is little loss of generality in treating (1.1) in this incremental and case-specific way. Likewise, in order to develop our sample size conditions and then analyze the corresponding algorithms, we begin with a *convex* constraint set $C \subseteq \mathbb{R}^n$. Later on, we give an outlook on how the algorithm can be generalized to treat nonconvex equality constraints.

[1] The words norm test are not actually used in either Carter (1991) or Byrd *et al.* (2012), but recent works by the authors of Byrd *et al.* (2012) have promoted this terminology; see, e.g., Bollapragada *et al.* (2018a) and Xie *et al.* (2020).

In Section 2, we use the expected risk problem to introduce basic adaptive sampling principles for stochastic projected gradient descent (SPGD) and theoretical conditions that imply convergence. We then use these conditions in Section 3 to propose a simple SPGD algorithm that solves the expected risk problem with convex constraints. In Section 4, we present two ways in which these algorithms may be extended to handle risk-averse problems. Here, we focus on the CVaR risk measure and state consequences for other important risk measures only in passing. Section 5 is dedicated to in-depth numerical studies that test the efficacy of our adaptive sampling method in its various forms. Afterwards, in Section 6, we generalize the algorithm using SQP principles, and arrive at a new method appropriate for an important class of problems with nonconvex constraints. This is complemented by a complex numerical example from engineering design. The paper then closes with a short summary of results. Finally, additional numerical experiments are documented in Appendix A.

2. Adaptive sampling with convex constraints

In this and the following section, we only consider $\mathcal{R} = \mathbb{E}$. This setting allows (1.1) to be rewritten as

$$\min_{x\in C} \Big\{ F(x) = \mathbb{E}[f(x;\xi)] \Big\}. \tag{2.1}$$

For the time being, we also assume that $C \subseteq \mathbb{R}^n$ is convex. To treat this problem, we propose a projected gradient descent algorithm and sufficient conditions on the sample sets S_k, which guarantee that it is a descent method in expectation.

2.1 *Preliminaries and notation*

Let $\nabla F(x)$ denote the gradient of F at x, and let $\langle\cdot,\cdot\rangle$ denote the ℓ^2 inner product on vectors in $\mathbb{R}^n$. It is well known (see, e.g., Nesterov, 2018) that if F is both convex and continuously differentiable then x^* is a solution of (2.1) if and only if

$$\langle \nabla F(x^*), x - x^* \rangle \geq 0 \tag{2.2}$$

for all $x \in C$.

When $C = \mathbb{R}^n$, one may use the stochastic gradient descent algorithm, $x_{k+1} = x_k - \alpha \nabla F_{S_k}(x_k)$, to uncover locally optimal solutions of (2.1); cf. Byrd *et al.* (2012) and Bollapragada *et al.* (2018a). Here, $\alpha > 0$ is a step-length parameter and $\nabla F_{S_k}(x_k)$ denotes the gradient of the sample average defined in (1.3) with an iteration-dependent sample set $S = S_k$. When the convex set $C \neq \mathbb{R}^n$, the analogue of this approach is the SPGD algorithm; $y_{k+1} = x_k - \alpha \nabla F_{S_k}(x_k)$, $x_{k+1} = \arg\min_{x\in C} \|y_{k+1} - x\|^2$, where $\|\cdot\|$ denotes the Euclidean norm. Equivalently (Nesterov, 2018), we write

$$x_{k+1} = \arg\min_{x\in C} \Big\{ F_{S_k}(x_k) + \langle \nabla F_{S_k}(x_k), x - x_k \rangle + \frac{1}{2\alpha}\|x - x_k\|^2 \Big\}. \tag{2.3}$$

We will write $\mathbb{E}_k[\cdot]$ to denote the expected value operator (1.2), given x_k. With this notation, quantities such as $\mathbb{E}_k[F(x_{k+1})]$ are well defined because x_{k+1} depends only on the random variable ξ through (1.3) and (2.3). We will also assume that S_k, for each k, is a set of i.i.d. samples, independent of each previous set $S_0, S_1, \ldots, S_{k-1}$. With this assumption, $\nabla F_{S_k}(x_k)$ forms an unbiased estimator for the gradient at x_k, namely

$$\mathbb{E}_k[\nabla F_{S_k}(x_k)] = \nabla F(x_k). \tag{2.4}$$

When we wish to analyze the *total expectation* of an iteration-dependent quantity, say $\mathbb{E}[F(x_k)]$, we note that it is completely determined by the joint distribution of samples in $S_0, S_1, \ldots, S_{k-1}$. For this reason, we have the identity (Bottou *et al.*, 2018)

$$\mathbb{E}[F(x_k)] = \mathbb{E}_0\mathbb{E}_1 \cdots \mathbb{E}_{k-1}[F(x_k)]. \tag{2.5}$$

In the sequel, it will also be convenient to assign symbols to certain terms in the equations above. First, we define the orthogonal projection onto C,

$$P(y) = \arg\min_{x\in C} \|y - x\|^2. \tag{2.6}$$

Note that, because C is convex, $P(y)$ is unique and nonexpansive (Nesterov, 2018, Corollary 2.2.3), namely

$$\|P(x) - P(y)\|^2 \leq \langle x - y, P(x) - P(y)\rangle \leq \|x - y\|^2, \tag{2.7}$$

and generally it is *nonlinear*. Next, we denote the projected gradient mapping, $Q : \mathbb{R}^n \to C$, as $Q(x) = P(x - \alpha\nabla F(x))$. Equivalently, one may write

$$Q(x) = \arg\min_{y\in C} \left\{F(x) + \langle\nabla F(x), y - x\rangle + \frac{1}{2\alpha}\|y - x\|^2\right\}. \tag{2.8}$$

The subsampled gradient map is then defined analogously to (2.8), namely

$$Q_{S_k}(x) = \arg\min_{y\in C} \left\{F_{S_k}(x) + \langle\nabla F_{S_k}(x), y - x\rangle + \frac{1}{2\alpha}\|y - x\|^2\right\}. \tag{2.9}$$

With this notation in hand, one may note that $x_{k+1} = Q_{S_k}(x_k)$ by (2.3).

The reduced gradient, defined by

$$R(x) = \alpha^{-1}(x - Q(x)), \tag{2.10}$$

is another important operator we will make judicious use of. The subsampled reduced gradient is likewise defined

$$R_{S_k}(x) = \alpha^{-1}(x - Q_{S_k}(x)).$$

Clearly, $x_{k+1} = x_k - \alpha R_{S_k}(x_k)$. Moreover, when $C = \mathbb{R}^n$, one may note that $R(x_k) = \nabla F(x_k)$ and $R_{S_k}(x_k) = \nabla F_{S_k}(x_k)$.

We may now formulate the first-order optimality condition for (2.1) as follows (Nesterov, 2018):

$$\text{if } x^* \text{ is a minimizer of (2.1) then } Q(x^*) = x^* \text{ and } R(x^*) = 0.$$

We may also state two lemmas based on Nesterov (2018), which will be useful later on. For reference, we say that F is L-smooth if

$$\|\nabla F(x) - \nabla F(y)\| \le L\|x - y\|, \tag{2.11}$$

for all $x, y \in C$, and we say that F is μ-strongly convex if

$$F(y) \ge F(x) + \langle \nabla F(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2. \tag{2.12}$$

The proof of Lemma 2.1 can be found in Nesterov (2018, Corollary 2.3.2). For the reader's convenience, we include the proof of Lemma 2.2.

Lemma 2.1 Assume that F is L-smooth, let $0 < \alpha \le 1/L$ and let C be convex. If F is convex then the following inequality holds for all $x \in C$:

$$F(Q(x)) - F(x) \le -\frac{\alpha}{2}\|R(x)\|^2. \tag{2.13}$$

If, moreover, F is μ-strongly convex then it also holds that

$$\frac{\mu}{2}\|x - x^*\|^2 + \frac{\alpha}{2}\|R(x)\|^2 \le \langle R(x), x - x^*\rangle. \tag{2.14}$$

Lemma 2.2 Let F be L-smooth, and let C be closed and convex. For all $y \in C$ and $z \in \mathbb{R}^n$, it holds that

$$\langle R(z), y - Q(z)\rangle \le \langle \nabla F(z), y - Q(z)\rangle. \tag{2.15}$$

Proof. Fix $z \in \mathbb{R}^n$, and consider $\phi(y) = F(z) + \langle \nabla F(z), y - z\rangle + \frac{1}{2\alpha}\|y - z\|^2$. Note that ϕ is both convex and continuously differentiable and that $\nabla\phi(y) = \nabla F(z) + \frac{1}{\alpha}(y - z)$. Therefore, following from the optimality condition (2.2) and the definition of $Q(\cdot)$ in (2.8), it holds that

$$\langle \nabla\phi(Q(z)), y - Q(z)\rangle = \langle \nabla F(z) - R(z), y - Q(z)\rangle \ge 0$$

for all $y \in C$. □

2.2 *Descent conditions*

The goal of our adaptive sampling scheme is to balance sampling and optimization error. One way to strike this balance is through state-dependent conditions that ensure that $\mathbb{E}_k\big[F(x_{k+1})\big] \le F(x_k)$. In Theorem 2.3, we show that it is sufficient that each sample set S_k satisfies only two idealized conditions. However, we note that these conditions require foreknowledge of the exact gradient at each iterate, x_k. (This detail is dealt with in Section 3.) The descent conditions are the following:

Condition 1 Control of the norm of the reduced gradient:

$$\mathbb{E}_k\left[\left\|R_{S_k}(x_k)\right\|^2\right] \le (1+\nu^2)\|R(x_k)\|^2, \tag{2.16}$$

for some fixed $\nu > 0$.

Condition 2 Control of the bias in the projected gradient mapping:

$$\|\mathbb{E}_k[Q_{S_k}(x_k) - Q(x_k)]\| \le \frac{\gamma^2}{2}\|Q(x_k) - x_k\|^2, \tag{2.17}$$

for some fixed $\gamma > 0$.

Theorem 2.3 Assume that F is L-smooth (2.11), and that the sequence of iterates $\{x_k\}_{k=0}^{\infty}$ is contained in an open set over which $\|\nabla F(x)\|$ is bounded above by some constant $M > 0$. If S_k satisfies Conditions 1 and 2 and $\alpha \le \frac{2}{M\gamma^2 + L(1+\nu^2)}$ then

$$\mathbb{E}_k\big[F(x_{k+1})\big] \le F(x_k).$$

Proof. By standard arguments following from the L-smoothness of F (Bertsekas, 2015), we have that

$$\begin{aligned} F(x_{k+1}) &\le F(x_k) + \langle \nabla F(x_k), x_{k+1} - x_k\rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &= F(x_k) + \langle \nabla F(x_k), Q(x_k) - x_k\rangle + \langle \nabla F(x_k), E_{S_k}(x_k)\rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2, \end{aligned} \tag{2.18}$$

where $E_{S_k}(x_k) = x_{k+1} - Q(x_k)$. Now, substituting $y = z = x_k$ into (2.15), we arrive at the identity

$$\langle \nabla F(x_k), Q(x_k) - x_k\rangle \le -\langle R(x_k), x_k - Q(x_k)\rangle = -\frac{1}{\alpha}\|x_k - Q(x_k)\|^2. \tag{2.19}$$

Next, Condition 2 implies that

$$\mathbb{E}_k\big[\langle \nabla F(x_k), E_{S_k}(x_k)\rangle\big] \le \|\nabla F(x_k)\|\|\mathbb{E}_k[E_{S_k}(x_k)]\| \le \frac{M}{2}\gamma^2\|Q(x_k) - x_k\|^2. \tag{2.20}$$

Moreover, Condition 1 implies that

$$\mathbb{E}_k\big[\|x_{k+1} - x_k\|^2\big] = \alpha^2\mathbb{E}_k\big[\|R_{S_k}(x_k)\|^2\big] \le \alpha^2(1+\nu^2)\|R(x_k)\|^2 = (1+\nu^2)\|Q(x_k) - x_k\|^2. \tag{2.21}$$

Combining (2.18), (2.19), (2.20) and (2.21), we arrive at

$$\mathbb{E}_k[F(x_{k+1})] - F(x_k) \le -c\|Q(x_k) - x_k\|^2, \tag{2.22}$$

where $c = \frac{1}{\alpha} - \frac{L}{2}(1+\nu^2) - \frac{M}{2}\gamma^2$. Note that if $\alpha \le \frac{2}{M\gamma^2 + L(1+\nu^2)}$, then the right-hand side of (2.22) is nonpositive. □

Although Conditions 1 and 2 are simple to write out, it is unfortunately difficult to design practical algorithms that guarantee them strictly. This difficulty is due in part to the presence of the projection operator $P\colon \mathbb{R}^n \to C$ inside the expected values on the left-hand sides of (2.16) and (2.17). Therefore, checking these conditions would require numerous applications of P, which may be prohibitively expensive. To avoid this difficulty, we turn to an alternative condition in the next subsection.

Remark 2.4 It is interesting to relate Conditions 1 and 2 to the analysis of stochastic gradient descent with adaptive sampling for unconstrained problems. In doing so, Condition 1 can be viewed as a generalization of Bollapragada *et al.* (2018a, Equation 3.5), which is one of the key inequalities in that work. Moreover, we note that Condition 2 is trivially satisfied for all $\gamma \geq 0$, whenever C is an affine subspace of $\mathbb{R}^n$.

Remark 2.5 The parameters in Conditions 1 and 2 are defined so that, if $f\colon \mathbb{R}^n \to \mathbb{R}$ is a deterministic function, then both conditions hold for all non-negative parameter values $\nu, \gamma \geq 0$.

Moreover, by setting $\nu = \gamma = 0$, we can recover Theorem 2.1. To observe this fact, we must inspect the proof of Theorem 2.3 and, in particular, inequality (2.22). Here, we see that if $\alpha \leq \frac{1}{M\gamma^2+L(1+\nu^2)}$ then $c \geq \frac{1}{2\alpha}$. With this stronger condition, we may write

$$\mathbb{E}_k[F(x_{k+1})] - F(x_k) \leq -\frac{\alpha}{2}\|R(x_k)\|^2,$$

which is analogous to (2.13) and equivalent to (2.13) $f\colon \mathbb{R}^n \to \mathbb{R}$ is a deterministic function and $\nu = \gamma = 0$. Similar bounds on the step size α will appear again. In anticipation of these expressions, we adopt the notation

$$\widetilde{L} = M\gamma^2 + L(1+\nu^2). \tag{2.23}$$

Remark 2.6 The reader may notice that the normed quantity on the right-hand side of Condition 2 may be rewritten by definition (2.10) as

$$\|Q(x_k) - x_k\|^2 = \alpha^2\|R(x_k)\|^2. \tag{2.24}$$

This is a useful identity that we will rely on in the sequel.

2.3 *Alternative condition*

Let us focus on the bias condition given by (2.17). It may appear odd that its left-hand side involves a norm and its right-hand side involves a norm squared. However, the bias term on the left-hand side is not absolutely homogeneous with respect to $\nabla F(x)$. This is easily seen in the specific case where the boundary of the constraint set C is smooth and, therefore, $P\colon \mathbb{R}^n \to C$ is also smooth. In this setting, we may write out a first-order Taylor expansion for $Q_{S_k}(x) = P(x - \alpha\nabla F(x) - \alpha(\nabla F_{S_k}(x) - \nabla F(x)))$ as follows:

$$\begin{aligned} Q_{S_k}(x) = Q(x) - \alpha\langle \nabla P(x - \alpha\nabla F(x)), \nabla F_{S_k}(x) - \nabla F(x)\rangle \\ + \mathcal{O}(\alpha^2\|\nabla F_{S_k}(x) - \nabla F(x)\|^2). \end{aligned} \tag{2.25}$$

Therefore, because $\mathbb{E}_k[\nabla F_{S_k}(x) - \nabla F(x)] = 0$, by (2.4), we arrive at the second-order relationship

$$\|\mathbb{E}_k[Q_{S_k}(x_k) - Q(x_k)]\| = \mathcal{O}(\alpha^2\mathbb{E}_k[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2]). \tag{2.26}$$

If we recall (2.24), it now seems appealing to replace Condition 2 by an alternative condition that delivers a probabilistic threshold on $\nabla F_{S_k}(x_k)$ lying within a ball around $\nabla F(x_k)$.

CONDITION 3 Control of the error in the full gradient by the norm of the reduced gradient:

$$\mathbb{E}_k[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2] \leq \theta^2 \|R(x_k)\|^2, \tag{2.27}$$

for some fixed $\theta > 0$.

Condition 3 is a direct generalization of the so-called norm test for stochastic gradient descent proposed in Byrd *et al.* (2012). Although Condition 3 also requires unattainable foreknowledge of the exact gradient, it is possible to design a practical algorithm around it. This aspect is discussed in the next section. Before then, however, we establish a number of theoretical properties related to the conditions above.

We finish this subsection by showing that, under certain assumptions, Condition 3 implies Conditions 1 and 2. This observation is encapsulated in Theorem 2.7. The remainder of this section is devoting to analyzing the convergence of the SPGD algorithm (2.3) when either Conditions 1, 2 or 3 is enforced.

THEOREM 2.7 Condition 3 implies Condition 1 with $\nu = \sqrt{2\theta + \theta^2}$. If, in addition,

$$\mathbb{E}[\|\nabla f(x;\xi) - \nabla F(x)\|^2] < \infty \tag{2.28}$$

for all $x \in C$, $P(\cdot) : \mathbb{R}^n \to C$ is twice differentiable and $|S_k|$ is sufficiently large, then Condition 3 implies Condition 2 for some $\gamma \propto \theta$.

Proof. To prove the first statement, it is important that we recall that $P(\cdot)$ is nonexpansive (2.7). Due to this property, we have

$$\begin{aligned}\|R_{S_k}(x_k) - R(x_k)\| &= \frac{1}{\alpha}\|P(x_k - \alpha\nabla F_{S_k}(x_k)) - P(x_k - \alpha\nabla F(x_k))\| \\ &\leq \|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|.\end{aligned}$$

Therefore, by (2.27),

$$\|\mathbb{E}_k[R_{S_k}(x_k) - R(x_k)]\| \leq \left(\mathbb{E}_k[\|R_{S_k}(x_k) - R(x_k)\|^2]\right)^{1/2} \leq \theta\|R(x_k)\|. \tag{2.29}$$

Likewise,

$$\begin{aligned}\mathbb{E}_k[\|R_{S_k}(x_k)\|^2] &= \|R(x_k)\|^2 + 2\langle R(x_k), \mathbb{E}_k[R_{S_k}(x_k) - R(x_k)]\rangle + \mathbb{E}_k[\|R_{S_k}(x_k) - R(x_k)\|^2] \\ &\leq \|R(x_k)\|^2 + 2\|R(x_k)\|\|\mathbb{E}_k[R_{S_k}(x_k) - R(x_k)]\| + \mathbb{E}_k[\|R_{S_k}(x_k) - R(x_k)\|^2] \\ &\leq \|R(x_k)\|^2 + 2\theta\|R(x_k)\|^2 + \theta^2\|R(x_k)\|^2 \\ &\leq (1+\theta)^2\|R(x_k)\|^2.\end{aligned}$$

In other words, Condition 1 holds with $\nu = \sqrt{2\theta + \theta^2}$.

To prove the second statement, we must argue that $\mathbb{E}_k[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2] \to 0$ as $|S_k| \to \infty$. Indeed, notice that

$$\mathbb{E}_k[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2] = \frac{\mathbb{E}_k[\|\nabla f(x_k;\xi) - \nabla F(x_k)\|^2]}{|S_k|} \to 0, \tag{2.30}$$

since the numerator is independent of $|S_k|$ by (2.28). Now, immediately following from (2.26), there exists some α- and k-independent constant, say $c \geq 0$, such that

$$\|\mathbb{E}_k[Q_{S_k}(x_k) - Q(x_k)]\| \leq c\alpha^2 \mathbb{E}_k[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2],$$

for all sufficiently large $|S_k|$. Invoking Condition 3, we have

$$\|\mathbb{E}_k[Q_{S_k}(x_k) - Q(x_k)]\| \leq c\theta^2\alpha^2\|R(x_k)\|^2 = c\theta^2\|Q(x_k) - x_k\|^2,$$

and thus Condition 2 holds with $\gamma = \theta\sqrt{2c}$. This completes the proof. □

Remark 2.8 One may notice that if C is an affine subspace then the second-order term in (2.25) actually disappears and Condition 2 is satisfied trivially. We will argue in Subsection 6.1 that this special setting permits us to propose other alternative conditions that are weaker than Condition 3.

Remark 2.9 (Comparison to Xie *et al.*, 2020). An alternative to Condition 3 that leads to similar convergence results is proposed in Xie *et al.* (2020, Equation 1.4). In our notation, this condition would be written

$$\mathbb{E}_k[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2] \leq \theta^2\|\mathbb{E}_k[R_{S_k}(x_k)]\|^2. \tag{2.31}$$

It may be argued that the upper bound in (2.31) is more expensive to estimate than $\|R(x_k)\|^2$, because a Monte Carlo estimate of $\mathbb{E}_k[R_{S_k}(x_k)]$ would require repeated application of the projection operator $P\colon \mathbb{R}^n \to C$. Meanwhile, estimating $\|R(x_k)\|^2$ requires only a careful estimate of $\nabla F(x_k)$ and a single application of P. Remarks 2.14 and 3.3 further compare our conditions to those in Xie *et al.* (2020).

2.4 *Convergence*

Convergence of SPGD can be shown under a variety of assumptions involving Conditions 1– 3. We begin this subsection by showing that Condition 3 implies q-linear convergence when F is strongly convex.

Theorem 2.10 (Strongly convex objective). Let F be both L-smooth (2.11) and μ-strongly convex (2.12), and let C be both convex and closed. Moreover, let the infinite sequence $\{x_k\}_{k=0}^{\infty}$ be generated by (2.3), with

$$\alpha < \frac{1}{L} \tag{2.32}$$

and each S_k satisfying Condition 3. Then, for all sufficiently small $\theta > 0$, x_k converges q-linearly in expectation, i.e.,

$$\mathbb{E}[\|x_{k+1} - x^*\|] \leq \rho^k \|x_0 - x^*\|,$$

for some $\rho \in [0, 1)$, where $x^* = \arg\min_{x \in C} F(x)$.

Proof. By (2.5), it is sufficient to show that $\mathbb{E}_k[\|x_{k+1} - x^*\|] \leq \rho \|x_k - x^*\|$, for every k. To this end, denote $E_{S_k}(x_k) = Q_{S_k}(x_k) - Q(x_k)$, and observe that

$$\begin{aligned}\left(\mathbb{E}_k[\|x_{k+1} - x^*\|]\right)^2 &\leq \mathbb{E}_k[\|x_{k+1} - x^*\|^2] = \mathbb{E}_k[\|x_k - \alpha\, R_{S_k}(x_k) - x^*\|^2] \\ &= \|x_k - x^*\|^2 + \alpha^2\, \mathbb{E}_k[\|R_{S_k}(x_k)\|^2] - 2\alpha\, \mathbb{E}_k[\langle R_{S_k}(x_k), x_k - x^*\rangle] \\ &= \|x_k - x^*\|^2 + \alpha^2\, \mathbb{E}_k[\|R_{S_k}(x_k)\|^2] - 2\alpha\, \langle R(x_k), x_k - x^*\rangle \\ &\quad + 2\, \langle \mathbb{E}_k[E_{S_k}(x_k)], x_k - x^*\rangle.\end{aligned}$$

Now, by Theorem 2.7, we have

$$\mathbb{E}_k[\|R_{S_k}(x_k)\|^2] \leq (1 + \nu^2)\, \|R(x_k)\|^2,$$

with $\nu^2 = 2\theta + \theta^2$. Furthermore, by (2.14), we have

$$-\, 2\alpha \langle R(x_k), x_k - x^*\rangle \leq -\mu\alpha \|x_k - x^*\|^2 - \alpha^2 \|R(x_k)\|^2,$$

and, by (2.29), we have

$$2\, \langle \mathbb{E}_k[E_{S_k}(x_k)], x_k - x^*\rangle \leq 2\|\mathbb{E}_k[E_{S_k}(x_k)]\|\, \|x_k - x^*\| \leq 2\alpha\theta \|R(x_k)\|\, \|x_k - x^*\|.$$

Combining each of these bounds, we find that

$$\mathbb{E}_k[\|x_{k+1} - x^*\|]^2 \leq (1 - \mu\alpha)\, \|x_k - x^*\|^2 + 2\alpha\theta \|R(x_k)\|\, \|x_k - x^*\| + \alpha^2 (2\theta + \theta^2)\, \|R(x_k)\|^2. \quad (2.33)$$

Invoking (2.14) a second time, along with the Cauchy–Schwarz inequality, yields

$$\frac{\mu}{2} \|x_k - x^*\|^2 + \frac{\alpha}{2} \|R(x_k)\|^2 \leq \|R(x_k)\|\, \|x_k - x^*\|.$$

Note that $\mu \leq L$ and so $\alpha\mu < \mu/L \leq 1$. Moreover, the two roots of the equation $\mu a^2 + \alpha b^2 = 2ab$ are $b = (1 \pm \sqrt{1 - \alpha\mu})a/\alpha$. Thus, it follows that

$$(1 - \sqrt{1 - \alpha\mu}) \|x_k - x^*\| \leq \alpha \|R(x_k)\| \leq (1 + \sqrt{1 - \alpha\mu}) \|x_k - x^*\|. \quad (2.34)$$

We may now replace every $\alpha\|R(x_k)\|$ factor in (2.33) by the upper bound given in (2.34). A straightforward simplification of the resulting inequality yields

$$\mathbb{E}_k[\|x_{k+1}-x^*\|]^2 \le \Big(1+2\big(1+\sqrt{1-\alpha\mu}\big)(3\theta+\theta^2)-(1+\theta)^2\mu\alpha\Big)\,\|x_k-x^*\|^2.$$

Finally, note that if θ is chosen sufficiently small, then

$$\rho^2 := 1+2\big(1+\sqrt{1-\alpha\mu}\big)(3\theta+\theta^2)-(1+\theta)^2\mu\alpha \le 1+4(3\theta+\theta^2)-\mu\alpha < 1,$$

as necessary. □

Conditions 1 and 2 can also be shown to imply convergence. In the following theorem, we show that it is possible to arrive at a sublinear convergence rate with a general convex objective function F.

THEOREM 2.11 (General convex objective). Assume that F is L-smooth (2.11) and C is convex and closed, and that the sequence of iterates $\{x_k\}_{k=0}^{\infty}$ is contained in a bounded open set D over which $\|\nabla F(x)\|$ is bounded above by some constant $M > 0$. Moreover, assume that

$$\widetilde{L}\alpha + \nu^2 + \gamma^2 \,\mathrm{diam}(D) \le 1,$$

where each S_k satisfies Conditions 1 and 2. Then, for every any positive integer T,

$$\mathbb{E}[F(x_T)] - F^* \le \frac{1}{2\alpha T}\|x_0 - x^*\|^2,$$

where F^* is the optimal objective function value and $x^* \in \{x : x = \arg\min_{x\in C} F(x)\}$.

Proof. Notice that

$$\|x_{k+1}-x_k\|^2 + \|x_k - x^*\|^2 - \|x_{k+1}-x^*\|^2 = 2\langle x_k - x_{k+1}, x_k - x^*\rangle.$$

Using the identity $x_k - x_{k+1} = \alpha R_{S_k}(x_k)$ and rearranging terms, we arrive at

$$\|x_{k+1}-x^*\|^2 - \|x_k-x^*\|^2 = 2\alpha\langle R(x_k), x^*-x_k\rangle - 2\langle E_{S_k}(x_k), x^*-x_k\rangle + \alpha^2\|R_{S_k}(x_k)\|^2,$$

where $E_{S_k}(x_k) = x_{k+1} - Q(x_k)$. Taking the expected value of both sides, we find

$$\begin{aligned}
&\mathbb{E}_k[\|x_{k+1}-x^*\|^2] - \|x_k - x^*\|^2\\
&\qquad\le 2\alpha\langle R(x_k), x^*-x_k\rangle + 2\|\mathbb{E}_k[E_{S_k}(x_k)]\|\,\|x^*-x_k\| + \alpha^2\mathbb{E}_k[\|R_{S_k}(x_k)\|^2]\\
&\qquad\le 2\alpha\langle R(x_k), x^*-x_k\rangle + \alpha^2(1+\nu^2+\gamma^2\|x^*-x_k\|)\|R(x_k)\|^2.
\end{aligned}\tag{2.35}$$

By setting $y = x^*$ and $z = x_k$ in Theorem 2.2, we may write

$$\langle R(x_k), x^*-x_k\rangle + \langle R(x_k), x_k - Q(x_k)\rangle \le \langle \nabla F(x_k), x^*-x_k\rangle + \langle \nabla F(x_k), x_k - Q(x_k)\rangle.$$

Note that $\langle R(x_k), x_k - Q(x_k)\rangle = \alpha\|R(x_k)\|^2$, by definition, and $\langle \nabla F(x_k), x^* - x_k\rangle \leq F^* - F(x_k)$, by convexity. By standard arguments following from the L-smoothness of F (Bertsekas, 2015), we have that

$$\begin{aligned}\langle \nabla F(x_k), x_k - Q(x_k)\rangle &= \langle \nabla F(x_k), x_k - x_{k+1}\rangle + \langle \nabla F(x_k), x_{k+1} - Q(x_k)\rangle \\ &\leq F(x_k) - F(x_{k+1}) + \frac{L}{2}\|x_{k+1} - x_k\|^2 + \langle \nabla F(x_k), x_{k+1} - Q(x_k)\rangle.\end{aligned}$$

Taking the conditional expectation of both sides yields

$$\begin{aligned}&\langle \nabla F(x_k), x_k - Q(x_k)\rangle \\ &\qquad \leq F(x_k) - \mathbb{E}_k[F(x_{k+1})] + \frac{L}{2}\mathbb{E}_k[\|x_{k+1} - x_k\|^2] + \langle \nabla F(x_k), \mathbb{E}_k[x_{k+1} - Q(x_k)]\rangle \\ &\qquad \leq F(x_k) - \mathbb{E}_k[F(x_{k+1})] + \frac{\alpha^2}{2}\big(L(1+\nu^2) + M\gamma^2\big)\|R(x_k)\|^2.\end{aligned}$$

Therefore,

$$\begin{aligned}\langle R(x_k), x^* - x_k\rangle &\leq \langle \nabla F(x_k), x^* - x_k\rangle + \langle \nabla F(x_k), x_k - Q(x_k)\rangle - \langle R(x_k), x_k - Q(x_k)\rangle \\ &\leq (F^* - F(x_k)) + \Big(F(x_k) - \mathbb{E}_k[F(x_{k+1})] + \frac{\widetilde{L}}{2}\alpha^2\|R(x_k)\|^2\Big) - \alpha\|R(x_k)\|^2 \\ &= F^* - \mathbb{E}_k[F(x_{k+1})] + \alpha\Big(\frac{\widetilde{L}\alpha}{2} - 1\Big)\|R(x_k)\|^2. \qquad (2.36)\end{aligned}$$

Finally, collecting together (2.35) and (2.36), we find

$$\begin{aligned}&\mathbb{E}_k[\|x_{k+1} - x^*\|^2] - \|x_k - x^*\|^2 \\ &\qquad \leq 2\alpha(F^* - \mathbb{E}_k[F(x_{k+1})]) + \alpha^2(\widetilde{L}\alpha + \nu^2 + \gamma^2\|x^* - x_k\| - 1)\|R(x_k)\|^2 \\ &\qquad \leq 2\alpha(F^* - \mathbb{E}_k[F(x_{k+1})]),\end{aligned}$$

where the second inequality follows from the bounds on α, ν and γ made in the theorem statement. We can now write

$$\mathbb{E}\big[F(x_{k+1})\big] - F^* \leq \frac{1}{2\alpha}\Big(\mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2]\Big),$$

which, after invoking Theorem 2.3, delivers the bound

$$\begin{aligned}\mathbb{E}\big[F(x_T)\big] - F^* &\leq \sum_{k=0}^{T-1}\frac{1}{T}\big(\mathbb{E}\big[F(x_{k+1})\big] - F^*\big) \\ &\leq \frac{1}{2\alpha T}\Big(\mathbb{E}[\|x_0 - x^*\|^2] - \mathbb{E}[\|x_T - x^*\|^2]\Big) \\ &\leq \frac{1}{2\alpha T}\mathbb{E}[\|x_0 - x^*\|^2].\end{aligned}$$

□

REMARK 2.12 The two previous theorems show q-linear and sublinear convergence rates, respectively. An important difference in the assumptions is that the sequence of iterates is assumed to be bounded in Theorem 2.11. Although this assumption trivially holds for any optimization problem posed over a bounded set, this does appear to be a strong assumption to make. At the same time, if one analyzes the proof in detail, it can be seen that $\widetilde{L}\alpha + \nu^2 + \gamma^2\|x^* - x_k\| \leq 1$ for each x_k is sufficient. This fact suggests the possible benefit of evolving $\gamma > 0$ with x_k. It also demonstrates the deteriorating role of the bias (cf. (2.17)) on the expected accuracy as x_k approaches x^*.

The following theorem shows an even weaker version of convergence that requires only the same mild assumptions as were made in Theorem 2.3. In particular, it shows that the sequence of reduced gradients $\{R(x_k)\}$ converges to zero in expectation. Therefore, every limit point x^* of the sequence x_k is stationary, i.e., $Q(x^*) = x^*$. This theorem also establishes a global sublinear rate of convergence of the smallest reduced gradients.

THEOREM 2.13 (Nonconvex objective). Under the assumptions of Theorem 2.3, if $\alpha < 2/\widetilde{L}$, then it holds that

$$\lim_{k\to\infty} \mathbb{E}[\|R(x_k)\|^2] = \lim_{k\to\infty} \mathbb{E}[\|Q(x_k) - x_k\|^2] = 0.$$

Moreover, for any positive integer T,

$$\min_{0\leq k\leq T-1} \mathbb{E}[\|R(x_k)\|^2] \leq \frac{1}{c\alpha^2 T}\big(F(x_0) - F_{\min}\big),$$

where $c = \frac{1}{\alpha} - \frac{\widetilde{L}}{2} > 0$ and $F_{\min}$ is a finite lower bound on F in C.

Proof. Begin by taking the total expected value of both sides of (2.22) and rewriting the result as

$$\mathbb{E}[\|R(x_k)\|^2] = \frac{1}{\alpha^2}\mathbb{E}[\|Q(x_k) - x_k\|^2] \leq \frac{1}{c\alpha^2}\big(\mathbb{E}[F(x_k)] - \mathbb{E}[F(x_{k+1})]\big). \tag{2.37}$$

It follows from the step size assumption in Theorem 2.3 that $c > 0$. Therefore, summing both sides of (2.37) delivers

$$\sum_{k=0}^{T-1} \mathbb{E}[\|R(x_k)\|^2] \leq \frac{1}{c\alpha^2}\big(\mathbb{E}[F(x_0)] - \mathbb{E}[F(x_T)]\big) \leq \frac{1}{c\alpha^2}\big(F(x_0) - F_{\min}\big).$$

Since this sum of T positive terms is bounded from above by a constant independent of T, the first statement follows. Moreover, notice that

$$\min_{0\leq k\leq T-1} \mathbb{E}[\|R(x_k)\|^2] \leq \frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}[\|R(x_k)\|^2] \leq \frac{1}{c\alpha^2 T}\big(F(x_0) - F_{\min}\big).$$

This completes the proof. □

REMARK 2.14 (Comparison to Xie *et al.*, 2020). In Xie *et al.* (2020, Theorem 3.3), it is shown that (2.31) also leads to q-linearly convergence in expectation, when F is strongly convex and sublinear convergence when F is convex, but not strongly convex. No theorem similar to Theorem 2.13, for convergence in the case of nonconvex F, appears in Xie *et al.* (2020).

3. A practical algorithm

In this section, we develop a practical SPGD algorithm based on Condition 3. In order to test whether this condition is satisfied, we introduce an approximation of the true gradient $\nabla F(x_k)$ and the risk measure $\mathbb{E}[\cdot]$. We begin by recalling (2.30), which allows us to we rewrite (2.27) as

$$\frac{\mathbb{E}_k[\|\nabla f(x_k;\xi) - \nabla F(x_k)\|^2]}{|S_k|} \leq \theta^2 \|R(x_k)\|^2. \tag{3.1}$$

We then approximate the true gradient $\nabla F(x_k)$ by the sample average gradient $\nabla F_{S_k}(x_k)$, as done in similar work on adaptive sampling; cf. Bollapragada *et al.* (2019). Likewise, we approximate the conditional expected value $\mathbb{E}_k[\cdot]$ by a sample average. Altogether, we propose the following practical test to check Condition 3.

TEST 1 (Approximation of Condition 3). Approximate control of the error in the full gradient by the norm of the reduced gradient:

$$\frac{1}{|S_k|-1} \frac{\sum_{\xi\in S_k} \|\nabla f(x_k;\xi) - \nabla F_{S_k}(x_k)\|^2}{|S_k|} \leq \theta^2 \|R_{S_k}(x_k)\|^2, \tag{3.2}$$

for some fixed $\theta > 0$.

In (3.2), we have used the factor $\frac{1}{|S_k|-1}$ instead of $\frac{1}{|S_k|}$ so that the left-hand side becomes an unbiased estimator for $\mathbb{E}_k[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2]$.

In order to construct a set S_k satisfying (3.2), one may envision starting with a sample set S_k of a minimal size, say $|S_k| = |S_0|$, and simply adding samples until (3.2) holds. This strategy, however, would be too expensive to be practical, as it would require recomputing $R_{S_k}(x_k)$ each time the set S_k is updated. Because of the expense of applying $P(\cdot)$, we choose to only consider strategies that involve computing $R_{S_k}(x_k)$ once each iteration.

One natural thing to consider is to use (3.2) to predict the correct size of the *upcoming* sample set S_{k+1}. The prediction of an *a posteriori* sample size for the next iteration is also presented in Bollapragada *et al.* (2018a), where unconstrained problems are considered. For the constrained optimization problems at hand, such a strategy may work as follows. Begin by dividing the left-hand side of (3.2) by $\theta^2\|R_{S_k}(x_k)\|^2$ and, in turn, define the new quantity

$$\rho = \frac{\sum_{\xi\in S_k} \|\nabla f(x_k;\xi) - \nabla F_{S_k}(x_k)\|^2}{\theta^2(|S_k|-1)|S_k|\|R_{S_k}(x_k)\|^2}.$$

When (3.2) is satisfied, we clearly have $\rho \leq 1$, and we simply keep the sample size fixed, that is, $|S_{k+1}| = |S_k|$. On the other hand, if the test fails, $\rho > 1$ is used to increase the sample size via the update rule

$$|S_{k+1}| = \lceil \rho\, |S_k| \rceil. \tag{3.3}$$

The procedure above leads to the following algorithm.

Algorithm 1: SPGD adaptive sampling algorithm for convex stochastic programs

```
input: $x_0$, step size $\alpha > 0$, initial sample set $S_0$, sampling rate parameter $\theta > 0$
Set $k \leftarrow 0$.
repeat
    Update $x_{k+1} = Q_{S_k}(x_k)$.
    if Test 1 is not satisfied then
        Construct $S_{k+1}$ obeying (3.3).
    else
        Construct $S_{k+1}$ satisfying $|S_{k+1}| = |S_k|$.
    Set $k \leftarrow k + 1$.
until a convergence test is satisfied
```

Remark 3.1 Upon rewriting $R_{S_k}(x_k) = (x_k - x_{k+1})/\alpha$, it is clear that Test 1 can be checked without performing any additional projections. The fact that Algorithm 1 requires only one projection per iteration makes the algorithm particular appealing, when the most expensive ingredient is the evaluation of the projection operator $P\colon \mathbb{R}^n \to C$.

Remark 3.2 It is important to note that Algorithm 1 is only an approximation of the algorithm analyzed in Theorem 2.10, since Test 1 differs in two important ways from Condition 3. In particular, the reduced gradient $R(x_k)$ on the right-hand side of Condition 3 is replaced by the estimate $R_{S_k}(x_k)$, and the variance on the left-hand side of (3.1) is replaced by the sample variance $\frac{1}{|S_k|-1}\sum_{\xi \in S_k} \|\nabla f(x_k;\xi) - \nabla F_{S_k}(x_k)\|^2$. Because of these differences, we cannot guarantee the same convergence rates predicted by Theorem 2.10. Nevertheless, as we will see in Section 5, our experiments with Algorithm 1 demonstrate extremely good agreement with the theoretical results of Theorem 2.10. Previous authors have made similar observations for their own practical adaptive sampling strategies (Bollapragada *et al.*, 2018a; Xie *et al.*, 2020). These repeated observations hint at a promising robustness in the adaptive sampling technique used here.

Remark 3.3 (Comparison to (Xie *et al.*, 2020)). Although the original conditions and analysis differ in numerous ways, the practical adaptive sampling algorithm proposed in Xie *et al.* (2020, Section 3.5) differs only marginally from Algorithm 1. Indeed, the only minor difference is that the practical algorithm in Xie *et al.* (2020) requires computing a second search direction before advancing to the next iteration when Test 1 is not satisfied. We propose a variant of this approach in Section 6.

4. Risk-averse problems

We now turn toward extending the algorithm proposed above to stochastic programs involving the CVaR. We present two different approaches to achieving this goal; both involve a regularization technique proposed in Kouri & Surowiec (2016) and rewriting $\mathrm{CVaR}_\beta(X)$ as the solution of an auxiliary optimization problem. Our first method follows a well-established course of action in risk-averse stochastic programming (Shapiro *et al.*, 2009; Kouri & Surowiec, 2016, 2018) and conforms to the assumptions used in the previous section. Our second method involves solving an additional one-dimensional optimization problem at each iteration.

4.1 *Conditional value-at-risk*

Let $\Psi_X(x) := \mathbb{P}(X \leq x)$ denote the cumulative distribution function (CDF) of a random variable X. The VaR of X, at confidence level $0 < \beta < 1$, also known as the β-quantile, is defined by

$$\mathrm{VaR}_\beta(X) := \inf\{t \in \mathbb{R} : \Psi_X(t) \geq \beta\}.$$

The CVaR of X, at confidence level β, is essentially the expected value of X beyond $\mathrm{VaR}_\beta(X)$. Indeed, if $\Psi_X(x)$ is right-continuous then $\mathrm{CVaR}_\beta(X)$ is precisely the conditional expectation $\mathbb{E}[X|X > \mathrm{VaR}_\beta(X)]$. This implies that $\mathrm{CVaR}_\beta(X) \geq \mathbb{E}[X]$. In order to accommodate more general CDFs, one may alternatively define $\mathrm{CVaR}_\beta(X)$ as the weighted integral of the VaR over the interval $(\beta, 1)$,

$$\mathrm{CVaR}_\beta(X) := \frac{1}{1-\beta}\int_\beta^1 \mathrm{VaR}_\alpha(X)\,\mathrm{d}\alpha.$$

Since $\mathrm{VaR}_\alpha(X)$ is a nondecreasing function of α, note that

$$\mathrm{CVaR}_\beta(X) \geq \frac{1}{1-\beta}\mathrm{VaR}_\beta(X)\int_\beta^1 \mathrm{d}\alpha = \mathrm{VaR}_\beta(X).$$

In many applications, $\mathrm{CVaR}_\beta(X)$ is a more useful measure of risk than $\mathrm{VaR}_\beta(X)$, because controlling expected failure states can be more important than controlling the most optimistic failure state only. For instance, consider the case where X can be identified with a stress acting on/within a physical system. In such scenarios, lower values of X are generally preferable to higher values of X. Thus, $\mathrm{VaR}_\beta(X)$ represents the most optimistic value that X can achieve in the worst $(1-\beta)\cdot 100\%$ of possible events. Alternatively, $\mathrm{CVaR}_\beta(X)$ represents the expected value of X in the worst $(1-\beta)\cdot 100$ percent of possible events.

The properties above make $\mathrm{CVaR}_\beta(X)$ a suitable risk measure for industrial optimization problems (Rockafellar & Royset, 2015). There are a variety of ways to treat stochastic programs that incorporate the CVaR (Kouri & Surowiec, 2016; Curi *et al.*, 2019). However, in this work, we find that the following ‘dual formulation’ is particularly useful.

In Rockafellar & Uryasev (2000), it is shown that $\mathrm{CVaR}_\beta(X)$ can be interpreted as the solution of a scalar optimization problem, namely

$$\mathrm{CVaR}_\beta(X) = \inf_{t\in\mathbb{R}}\left\{t + \frac{1}{1-\beta}\mathbb{E}[(X-t)_+]\right\}, \tag{4.1}$$

where $(x)_+ := \max\{0, x\}$. Therefore, the stochastic program

$$\min_{x\in C} F(x) = \mathrm{CVaR}_\beta[f(x;\xi)] \tag{4.2}$$

can be conveniently reformulated as

$$\min_{(x,t)\in C\times\mathbb{R}} F(x,t) = \mathbb{E}\Big[t + \frac{1}{1-\beta}(f(x;\xi)-t)_+\Big]. \tag{4.3}$$

It is well known that nonsmoothness of the operator $(\,\cdot\,)_+$, implies nonsmoothness of the objective function $F(x,t)$ (Rockafellar & Uryasev, 2000). Therefore, (4.3) is often solved with subgradient types methods; see, e.g., Royset & Szechtman (2013). An alternative option is to replace CVaR_β by a smooth approximation, which maintains many of its essential properties. In this work, we choose to use a smoothing technique proposed by Kouri & Surowiec (2016).

4.2 *Smoothing*

The nondifferentiability of $F(x,t)$ can be circumvented by regularizing the $(\cdot)_+$ function. In Kouri & Surowiec (2016, Section 4.1.1), several strategies are proposed. We choose the smooth approximation $(\cdot)_+^\varepsilon$ defined as follows:

$$(y)_+^\varepsilon = y + \varepsilon \ln\left(1+\exp\left(\frac{-y}{\varepsilon}\right)\right).$$

Likewise, we replace the nonsmooth CVaR_β risk measure by the smoothed risk measure

$$\mathrm{CVaR}_\beta^\varepsilon(X) = \inf_{t\in\mathbb{R}}\Big\{t + \frac{1}{1-\beta}\mathbb{E}[(X-t)_+^\varepsilon]\Big\} \tag{4.4}$$

and replace (4.3) by

$$\min_{(x,t)\in C\times\mathbb{R}} F^\varepsilon(x,t) = \mathbb{E}\Big[t + \frac{1}{1-\beta}(f(x;\xi)-t)_+^\varepsilon\Big]. \tag{4.5}$$

All of the conclusions in the previous sections carry over to the regularized CVaR problem because the objective function $F^\varepsilon(x,t)$ is now smooth. This means that Algorithm 1 can be used to solve (4.5). It is also important to point out that this smooth CVaR formulation enjoys the advantage that many of the original CVaR properties are preserved, including convexity and monotonicity (Kouri & Surowiec, 2016). Accordingly, if $f(x;\xi)$ is convex for almost every ξ then $F^\varepsilon(x,t)$ is also convex.

REMARK 4.1 The regularization constant ε is a problem-dependent parameter, which must be tuned. To guide the tuning process, one may use Lemma 4.3 in Kouri & Surowiec (2016), which shows that

$$|\mathrm{CVaR}_\beta^\varepsilon(X) - \mathrm{CVaR}_\beta(X)| \le \frac{\log 2}{1-\beta}\,\varepsilon.$$

Thus, the value of ε necessary to achieve an intended relative error will depend on both the magnitude of $\mathrm{CVaR}_\beta(X)$ and the confidence level β. A short study on the influence of ε is carried out in Urbainczyk (2020, Chapter 5.1.3).

REMARK 4.2 The function $(\cdot)_+$ falls into the special class of the so-called scalar regret functions (Rockafellar & Uryasev, 2013). Specifically, such functions $v:\mathbb{R}\to\overline{\mathbb{R}}$ are closed, convex, increasing and satisfy $v(0) = 0$ and $v(x) > x$ for all $x \neq 0$. If one replaces $\frac{1}{1-\beta}(\cdot)_+^{\varepsilon}$ in (4.5), with any scalar regret function $v(\cdot)$, then one arrives at an important class of risk-averse stochastic programs, which has also received a great deal of attention (Ben-Tal & Teboulle, 1986; Rockafellar & Uryasev, 2013; Kouri & Surowiec, 2018):

$$\min_{x\in C} F(x) = \mathcal{R}[f(x;\xi)] \quad \text{where} \quad \mathcal{R}(X) = \inf_{t\in\mathbb{R}}\Big\{t + \mathbb{E}[v(X-t)]\Big\}.$$

If v is also smooth then Algorithm 1 may also be used without further modification to solve this entire family of risk-averse stochastic programs.

4.3 *Nested quantile estimation*

Although Algorithm 1 can be used to solve (4.5), when there are only a small number of samples, the initial error may be quite large; cf. Subsection 5.2.2. For this reason, we introduce an alternative algorithm. We begin with two observations.

It is well known that the unique minimizer of (4.1), t^*, is simply the VaR, namely

$$\mathrm{CVaR}_\beta(X) = t^* + \frac{1}{1-\beta}\mathbb{E}[(X-t^*)_+], \quad \text{where} \quad t^* = \mathrm{VaR}_\beta(X).$$

Accordingly, if we assume that $\mathrm{VaR}_\beta(f(x;\xi))$ was somehow determined *a priori*, it would be possible to rewrite (4.2) as

$$\min_{x\in C} \widetilde{F}(x) = \mathbb{E}[(f(x;\xi) - \mathrm{VaR}_\beta(f(x;\xi)))_+]. \tag{4.6}$$

This technique of rewriting (4.2) is analogous to the scalar regret function reformulation of stochastic programs involving the entropic risk measure; see, e.g., Kouri & Surowiec (2018, Section 2.4.2).

It turns out that there are a large number of methods to estimate quantiles that are widely available in scientific software such as R (R Core Team, 2017), Python (specifically, SciPy Virtanen *et al.* (2020)) and Julia (Bezanson *et al.*, 2017). Any of these approximations could be substituted for $\mathrm{VaR}_\beta(f(x;\xi))$ in (4.6), once a set of samples of $f(x;\xi)$ is collected. Nevertheless, we choose to approximate the VaR by estimating t^* at each iteration and then solving the regularized form of (4.6). That is, we first compute

$$t_{S_k} = \operatorname*{arg\,min}_{t\in\mathbb{R}} \left\{ t + \frac{1}{1-\beta}\frac{1}{|S_k|}\sum_{\xi_i\in S_k}(f(x;\xi_i) - t)_+^{\varepsilon} \right\} \tag{4.7}$$

with a root finding algorithm. This is no more expensive that a standard line search and generally cheaper than applying $P(\cdot)$. Furthermore, one may argue that $t_{S_k} \to t^*$ as $|S_k| \to \infty$. We then compute the new iterate $x_{k+1} = Q_{S_k}(x_k)$ via the subsampled gradient map of

$$\widetilde{F}^{\varepsilon}_{S_k}(x) := \frac{1}{|S_k|}\sum_{\xi_i\in S_k}(f(x;\xi) - t_{S_k})_+^{\varepsilon}.$$

The entire adaptive sampling process is described in Algorithm 2, below.

Algorithm 2: Nested quantile estimation and adaptive sampling with CVaR

```
input: $x_0$, step size $\alpha > 0$, initial sample set $S_0$, constant $\theta > 0$
Set $k \leftarrow 0$.
repeat
    Compute $t_{S_k} = arg\,min_{t\in\mathbb{R}} \left\{t + \frac{1}{1-\beta} \frac{1}{|S_k|} \sum_{\xi_i\in S_k} (f(x_k;\xi_i) - t)^{\varepsilon}_{+}\right\}$.
    Update $x_{k+1} = arg\,min_{y\in C}\left\{\widetilde{F}_{S_k}(x_k) + \langle\nabla\widetilde{F}_{S_k}(x_k), y - x_k\rangle + \frac{1}{2\alpha}\|y - x_k\|^2\right\}$.
    if Test 1 is not satisfied then
        Construct $S_{k+1}$ obeying (3.3).
    else
        Construct $S_{k+1}$ satisfying $|S_{k+1}| = |S_k|$.
    Set $k \leftarrow k + 1$.
until a convergence test is satisfied
```

5. Numerical examples

In this section, we conduct two sets of numerical experiments to illustrate Algorithms 1 and 2. We begin with a simple example problem that allows us to test the theory presented in Subsection 2.4. Subsequently, we assess the practicality and robustness of the adaptive sampling algorithms with a risk-averse portfolio optimization application. In order to discuss the performance of Algorithms 1 and 2, we include plots showing the objective function values at each iteration. These function values were estimated to a high accuracy independent of the algorithms' approximation of the objective function value.

5.1 *Basic example*

Our first stochastic programming example is inspired by Royset & Szechtman (2013, Section 6.2). Consider a function

$$f(x;\xi) = \sum_{l=1}^{20} a_l(x^l - b_l\xi^l)^2, \qquad x = (x^1, \ldots, x^{20}), \quad \xi = (\xi^1, \ldots, \xi^{20}), \tag{5.1}$$

where the coefficients $a_l \sim \mathsf{Unif}(1,2)$ and $b_l \sim \mathsf{Unif}(-1,1)$ have been randomly sampled once for the sake of simulation and, thereafter, left fixed. Next, assume that ξ is a random vector where each coefficient $\xi^l \sim \mathsf{Unif}(0,1)$. Finally, define the admissible set $C = [0,\infty)^{20}$, which is closed, convex and unbounded.

With the definitions given above, we consider the (risk-neutral) stochastic program

$$\min_{x\in C} \left\{F(x) = \mathbb{E}[f(x;\xi)]\right\}. \tag{5.2}$$

Note that this program is strongly convex and that $f(x;\xi)$ is differentiable for every $\xi \in \varXi = [0,1]^{20}$. Therefore, there exists a unique global minimizer and Theorem 2.10 applies. In fact, the unique global

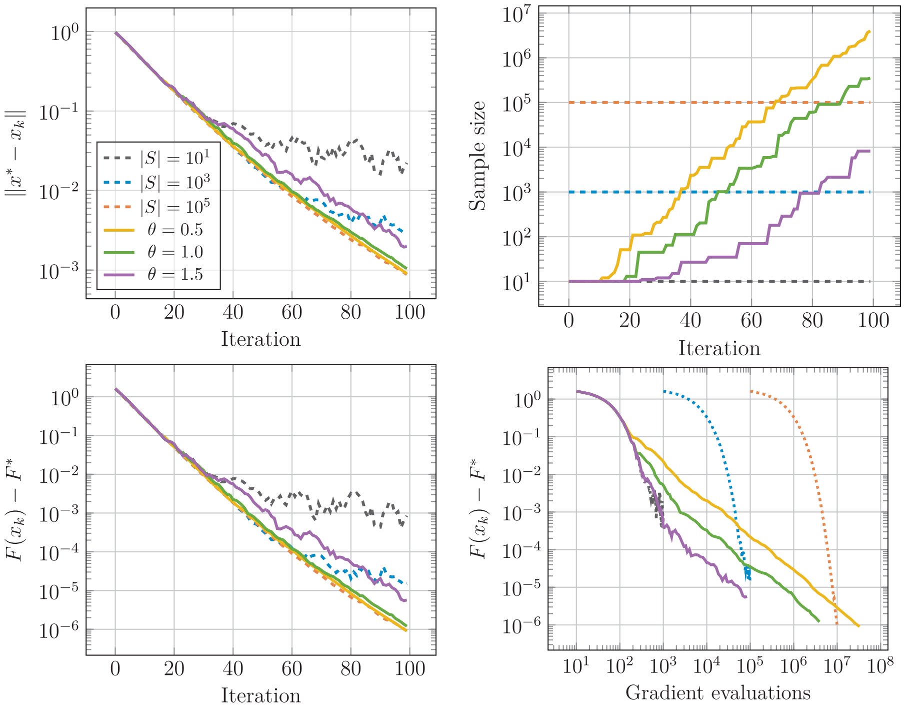


FIG. 1. Comparison of the stochastic approximation with fixed sample sizes and Algorithm 1 applied to the stochastic program (5.2). The top-left and bottom-left plots show the error in the solution vs. the iteration number and the error in the objective function vs. the iteration number, respectively. The bottom-right plot shows the error in the objective function vs. the cumulative number of gradient evaluations.

minimizer $x^* = (x^{*,1}, \ldots, x^{*,20})$ of (5.2) can be written out explicitly; i.e., $x^{*,l} = \max\{0, b_l/2\}$, for each $l = 1, \ldots, 20$.

This example has two purposes: first, to suggest that the theory presented in Subsection 2.4 also holds when the practical Test 1 is used and, second, to compare the performance of Algorithm 1 with different values of θ. In Fig. 1, we see the results from six representative optimization runs. The first three runs use fixed sample sizes of $|S_k| = 10,\ 10^3$ and 10^5, respectively, for all iterations k; these runs imitate naive approaches to compare against. The subsequent three runs each begin with the common initial sample size $|S_0| = 10$, and are executed using Algorithm 1 with the parameter values $\theta = 0.5,\ 1.0$ and 1.5, respectively. All of the runs use a fixed step size of $\alpha = 0.025$. Due to Theorem 2.10, similar results are expected for all step sizes $\alpha < 1/L$ and sufficiently small $\theta > 0$. We present further experiments on the influence of the step size in Appendix A.1, whereas in the current section we focus on the effects of the adaptive sampling.

The leftmost plots in Fig. 1 illustrate q-linear convergence for each of the adaptive sampling runs, albeit, at different levels of efficiency. Recalling Theorem 2.10, this is the best outcome one could hope for. For all smaller values of $\theta > 0$, the algorithm continues to converge linearly; however, for larger values of θ, the convergence eventually breaks down. The value of θ where linear convergence fails depends on the step size α, as one would expect from Theorem 2.10. In contrast, the fixed sample size examples with $|S| = 10^1,\ 10^3$ eventually stop converging. The same would happen in the case of the

$|S| = 10^5$ example, given enough iterations. Since we use a fixed step size here, this is the expected behavior.

The plots on the right in Fig. 1 provide the sample sizes and resulting gradient evaluations used to obtain the results shown on the left. In the top-right, we observe that the adaptive algorithm increases the sample size roughly exponentially. This behavior can be interpreted positively from Bottou *et al.* (2018, Section 5). Indeed, assuming a uniform bound on the individual gradient samples' variance, an exponentially increasing sample size leads to the variance of the resulting gradient estimate decreasing exponentially. This allows Algorithm 1 to converge linearly and, in this regard, outperform the fixed-sample size algorithm. The bottom-right plot shows the number of gradient evaluations required for each fixed sample size or value of θ. When using fixed sample and step sizes, the error in the objective function will eventually stop decreasing. This is avoided when using our adaptive sampling strategy. Moreover, the number of computed gradient samples can be significantly reduced by adopting an adaptive sample size rule, especially in the early stages of the optimization, when the objective function error is still large.

As a rule of thumb in choosing the adaptive sampling parameter θ, we suggest that one starts with a value around 1.0, and then track the adaptive algorithm until the first significant growth in the sample size plateaus. If there has already been a meaningful decrease in the objective value by this point, keep θ fixed; otherwise, θ should probably be decreased moderately. In all cases we have looked at, a reasonable value for θ can be chosen based on the behavior of the algorithm in its first 10 to 20 iterations.

5.2 *Portfolio optimization*

With this set of optimization problems, we continue to illustrate the practicality of the adaptive sampling algorithm proposed above. Specifically, we choose to focus on a class of archetypal operations research problems taken from Royset & Szechtman (2013, Section 6.1). In this example, we incorporate the paradigm of risk-averse stochastic optimization; cf. Section 4.

5.2.1 *Problem description.* Let us consider a random cost model with $n = 100$ financial instruments whose outputs are each given as $\xi = A + Bu$. In this model, A is an n-dimensional vector representing the expected rate of return of a single instrument, and B is an $n \times n$-dimensional matrix that correlates the uncertainty in this return. Each component of A is defined through an independent sample of a uniform distribution over $[0.9, 1.2]$ and, likewise, each entry in B is defined by an independent sample of a uniform distribution over $[0, 0.1]$. As with the model parameters a_l and b_l appearing in (5.1), both A and B only specify parameters in the model. Therefore, A and B are randomly generated and then held fixed throughout the entire optimization process. Finally, each component of the n-dimensional random vector u, which itself acts to introduce uncertainty in the model, is taken to be independent and obey a standard normal distribution.

Given the financial instrument model described above, we now consider the investment of one share of wealth distributed over the $n = 100$ independent random financial instruments. We choose to denote the amount of investment into the l-th asset by $x^l \geq 0$, whereby $\sum_{l=1}^{100} x^l = 1$. Accordingly, we arrive at the following (stochastic) loss function:

$$f(x; \xi) = -\sum_{l=1}^{100} \xi^l x^l, \tag{5.3}$$

where $x = (x^1, \dots, x^{100})$ is our given portfolio allocation strategy.

Let us say that we would like to minimize the loss over all portfolio strategies that have an expected return no smaller than 1.05. We therefore define the following admissible set of normalized portfolios:

$$C = \left\{ x \in \mathbb{R}^{100} : x^l \geq 0, \quad \sum_{l=1}^{100} x^l = 1, \quad \sum_{l=1}^{100} A_l x^l \geq 1.05 \quad l = 1, \ldots, 100 \right\}.$$

In a risk-neutral paradigm, we seek only to minimize the expected value of (5.3) over C. The corresponding stochastic program is simply

$$\min_{x \in C} \left\{ F(x) = \mathbb{E}[f(x;\xi)] \right\}. \tag{5.4}$$

With this definition of $F(x)$, the strong convexity assumption made in Theorem 2.10 is not satisfied.

It turns out that the expected loss problem above tends not to serve well for most practical investment decisions. Alternatively, one can minimize the loss with CVaR_β as a risk measure; this is a common choice in financial applications (Dowd, 2007; Föllmer & Schied, 2011). Accordingly, we focus on the following class of risk-averse stochastic programs:

$$\min_{x \in C} \left\{ F_\beta(x) = \mathrm{CVaR}_\beta[f(x;\xi)] \right\}, \tag{5.5}$$

where $\beta \in [0, 1)$ is the risk-averseness parameter. Note that $F(x) = F_0(x)$ in the notation of (5.4) and (5.5), and so the risk-neutral program (5.4) has not actually been ignored (Rockafellar & Uryasev, 2000, 2002).

Remark 5.1 As already pointed out in Section 4, the CVaR risk measure introduces nonsmoothness into the objective functional that commonly breaks the convergence of traditional gradient descent algorithms. Therefore, we follow Subsection 4.2 in our experiments and replace the CVaR in (5.5) by the risk measure $\mathrm{CVaR}_\beta^\varepsilon$, defined in (4.4), with some small regularization parameter $\varepsilon > 0$.

5.2.2 *Risk-averse portfolio optimization.* In our first set of portfolio optimization experiments, we compare the performance of Algorithm 1 on the stochastic program (5.5), for a variety of risk-averseness parameters $\beta > 0$. Recall (4.4), and note that each of these problems may be written as

$$\min_{(x,t) \in C \times \mathbb{R}} \left\{ F_\beta(x, t) = t + \frac{1}{1-\beta}\, \mathbb{E}\left[(f(x;\xi) - t)_+^\varepsilon \right] \right\}, \tag{5.6}$$

after regularization. Because our experiments in Subsection 5.1 already indicated a robustness with respect to the algorithm parameter θ, we choose to focus our attention here on its sensitivity to the risk-averseness parameter β. In this example, we consider $\beta = 0,\ 0.5,\ 0.9$ and 0.95. For all $\beta > 0$, we set $\varepsilon = 0.01$. The value for ε is chosen as a compromise between a small error w.r.t. the true CVaR and a well-behaved objective function. A bound for this error was introduced in Kouri & Surowiec (2016); see also Remark 4.1. The step size is kept constant as before, this time set to $\alpha = 0.5$. The value was manually selected such that the simulations give reasonable results, but was not extensively tuned.

Note that β actually changes the optimization problem. Hence, for each β, we choose a different sampling rate parameter θ. For $\beta = 0$ and $\beta = 0.5$, we set $\theta = 2.0$; for $\beta = 0.9$, we take $\theta = 1.5$; and for $\beta = 0.95$, we specify $\theta = 0.125$. These parameters were chosen to promote comparable growth of the sample size across the different cases. Generally, as $\beta < 1$ grows, θ should shrink. The effect of the sampling rate θ on the sample size is illustrated further in Appendix A.2.

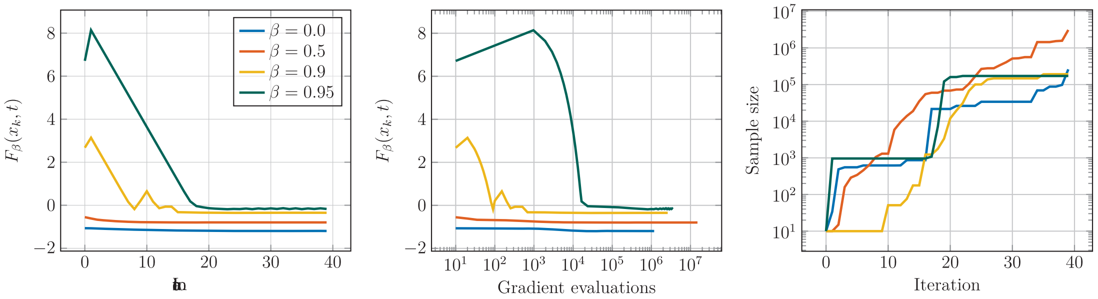


FIG. 2. Numerical results for the portfolio optimization problem (5.5) for the risk-averseness parameters $\beta = 0.0$ (which also corresponds to the risk-neutral problem (5.4)), 0.5, 0.9 and 0.95. On the left, we see the value of the objective function converge with respect to the iteration number. In the middle, we see the value of the objective function converge with respect to the cumulative number of gradient evaluations. On the right, we see the sample size grow with respect to the iteration number.

Figure 2 presents the results of our numerical experiments. When the risk averseness parameter β is increased, we expect that the achieved objective value increases too. This feature is clearly observed in Fig. 2. It is also evident from Fig. 2 that the initial value of the objective function, $F_\beta(x_0, t_0)$, moves progressively further from its optimal value as β grows. Additionally, for $\beta = 0.9$ and $\beta = 0.95$, the algorithm fails to improve the objective during the first few iterations. Both effects are largely due to the fact that the same initial value of the auxiliary variable, $t = t_0$, has been used in each experiment. In turn, the algorithm takes longer to converge as β increases. It should be noted that the auxiliary variable is optimized using the same step size as used with the spatial variable x without accounting for possibly different scales.

This experiment shows that the performance of Algorithm 1 is very sensitive to the choice of the initial value t_0. In light of Subsection 4.3, one could set t_0 to be the solution of (4.7) to obtain a first estimate of the optimal value of t, and thus circumvent this issue. Of course, however, optimizing for t independently of x is essentially what is done at each iteration of Algorithm 2.

5.2.3 *Risk-averse portfolio optimization with Algorithm 2.* Here, we briefly compare Algorithm 2 to Algorithm 1. Since the setting $\beta = 0$ simply amounts to problem (5.4), our comparison only involves $\beta = 0.5,\ 0.9$ and 0.95. It may seem natural to also choose the same values of θ used in Subsection 5.2.2; however, we found that the auxiliary variable t appearing in (5.6) has a strong effect on variance of the objective function. Because the gradient with respect to t does not appear in Algorithm 2, we were able to use larger values of θ than in Subsection 5.2.2, and this tended to result in better sample size efficiency. To be specific, for $\beta = 0.5$, we set $\theta = 4.0$; for $\beta = 0.9$, we set $\theta = 4.5$; and for $\beta = 0.95$, we set $\theta = 4.5$.

The results of our comparison are presented in Fig. 3. Evidently, both algorithms converge to the same optimal objective value. Although Algorithm 2 involves solving a one-dimensional optimization problem at each iteration, it also appears to generate fewer samples that could make it more efficient overall in some applications.

6. Adaptive sampling with nonconvex constraints

Up to this point, the convexity of the constraint set C has been critical. In general, it is even required to uniquely define the orthogonal projection (2.6). Nevertheless, many optimization problems involve

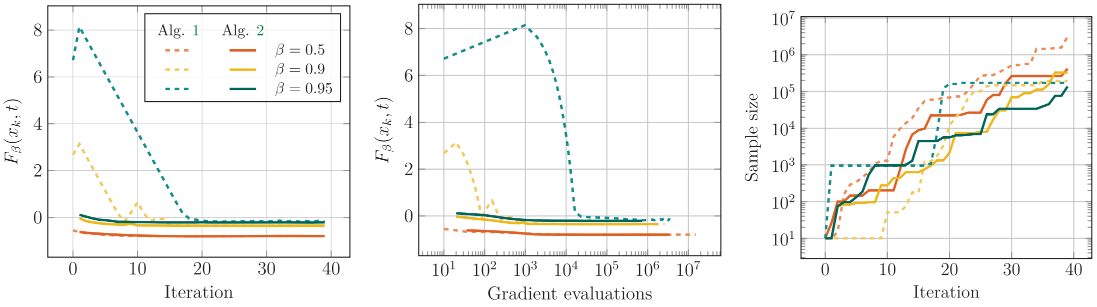


FIG. 3. Numerical results for problem (5.5), with $\beta = 0.5,\ 0.9$ and 0.95, comparing Algorithms 1 and 2.

nonconvex constraints, and we seek to show that some of the ideas introduced above can still be used in that setting. Our treatment is not intended to be comprehensive; we give one practical example and leave its generalizations for future study.

6.1 *Treatment of nonconvex constraints*

Assume that we are required to optimize over the level set of a smooth function $G : \mathbb{R}^n \to \mathbb{R}$, and so we rewrite (2.1) as

$$\min_{x\in\mathbb{R}^n} \Big\{F(x) = \mathbb{E}[f(x;\xi)] \quad \text{subject to } G(x) = 0\Big\}.$$

Let us also assume that the gradient of G never vanishes on the feasible set. In this case, the linear independence constraint qualification is trivially satisfied, because the set of constraint gradients is always one-dimensional and the above problem can be solved with SQP principles (Nocedal & Wright, 2006; Hinze *et al.*, 2008; Ulbrich & Ulbrich, 2012).

For simplicity, let B_k be a symmetric positive definite (SPD) matrix,[2] and define $\|d\|_{B_k} = \sqrt{\langle d, B_k d\rangle}$. Typically, we seek the optimum

$$d_{S_k} = \operatorname*{arg\,min}_{d\in\mathbb{R}^n} \langle \nabla F_{S_k}(x_k), d\rangle + \frac{1}{2}\|d\|^2_{B_k} \quad \text{subject to } \langle \nabla G(x_k), d\rangle + G(x_k) = 0 \tag{6.1}$$

and update the solution $x_k \mapsto x_k + \alpha d_{S_k}$. In its most basic form (Powell, 1983), we may assume that each $B_k = I$ and so $\|d\|_{B_k} = \|d\|$. In this setting, a straightforward computation shows that $d_{S_k} = -R_{S_k}(x_k)$ when the affine subspace

$$C_k = \{y \in \mathbb{R}^n : \langle \nabla G(x_k), y - x_k\rangle + G(x_k) = 0\}$$

is substituted for C in definition (2.9). This observation establishes a well known connection between projected gradient algorithms and SQP (Powell, 1983). As such, it also provides a connection between the preceding analysis and a treatment of nonconvex constraints, where a linearized constraint space is updated at each iteration k. Throughout the rest of this section, when we refer to $R_{S_k}(x_k)$ or $R(x_k)$, we assume that $C = C_k$.

[2] It is well known that we may relax this requirement to being that B_k is only SPD on the tangent space of the constraint set; cf. Powell (1983) and Nocedal & Wright (2006).

As stated in Remark 2.8, Condition 2 is trivially satisfied when C is an affine subspace. Of course, this happens to be the case in (6.1) because every C_k is an affine subspace. It is interesting to note that an affine subspace constraint makes it possible to propose alternatives to Condition 3 that are less restrictive on the size of the sample set S_k. One possibility is to propose a threshold on the expected value of $R_{S_k}(x_k)$ lying within a ball around $R(x_k)$. This may be written as follows.

CONDITION 4 Control of the error in the reduced gradient by the norm of the reduced gradient:

$$\mathbb{E}_k\left[\|R_{S_k}(x_k) - R(x_k)\|^2\right] \le \theta^2 \|R(x_k)\|^2,$$

for some fixed $\theta > 0$.

We note that $\mathbb{E}_k[R_{S_k}(x_k)] = R(x_k)$ by the affine nature of $P : \mathbb{R}^n \to C_k$. Therefore, in this specific setting, we have the identity

$$\mathbb{E}_k\left[\|R_{S_k}(x_k) - R(x_k)\|^2\right] = \mathbb{E}_k\left[\|R_{S_k}(x_k)\|^2\right] - \mathbb{E}_k\left[\|R(x_k)\|^2\right],$$

and so Condition 4 is actually equivalent to Condition 1 with $\nu = \theta$. Regardless, to avoid confusing with the general setting where Conditions 1 and 4 are not equivalent, we choose to denote Conditions 1 and 4 differently.

6.2 *Practical algorithm*

In order to derive an SQP-type projected gradient algorithm of practical relevance with adaptive sampling relying on Subsection 6.1, let us define

$$d(x_k;\xi) = \underset{d\in\mathbb{R}^n}{\arg\min}\ \langle \nabla f(x_k;\xi), d\rangle + \frac{1}{2}\|d\|^2 \quad \text{subject to } \langle \nabla G(x_k), d\rangle + G(x_k) = 0 \tag{6.2}$$

and, accordingly, $R(x_k;\xi) = -d(x_k;\xi)$. We then propose the following test that may be used to check Condition 4.

TEST 2 (Approximation of Condition 4). Approximate control of the error in the reduced gradient by the norm of the reduced gradient:

$$\frac{1}{|S_k|-1}\,\frac{\sum_{\xi\in S_k}\|R(x_k;\xi) - R_{S_k}(x_k)\|^2}{|S_k|} \le \theta^2\|R_{S_k}(x_k)\|^2,$$

for some fixed $\theta > 0$.

REMARK 6.1 Test 2 may appear undesirable because it involves computing individual reduced gradients $R(x_k;\xi)$ and, thus, repeated applications of the projection operator $P : \mathbb{R}^n \to C_k$. This, however, is not a deep concern, since a projection onto an affine subspace is usually very cheap to evaluate and can often be performed via sparse matrix operations (Trefethen & Bau III, 1997).

REMARK 6.2 One could also propose other alternatives to Condition 4. For instance, one could follow Bollapragada *et al.* (2018a) and derive conditions that lead to a probabilistic threshold on $R_{S_k}(x_k)$ pointing in the same direction as $R(x_k)$. For sake of space, we do not include any algorithms based on this approach. The interested reader is referred to Bollapragada *et al.* (2018a) and Urbainczyk (2020) for further details on how such algorithms could be constructed.

Feasibility Algorithms for constrained optimization problems need to verify both optimality and feasibility conditions. Since G is generally not an affine function, the proposed update steps d_{S_k} alone do not necessarily result in feasible iterates x_k. Moreover, we assume that we do not have the ability to perform an exact projection onto the feasible set.

One way to proceed is to add a correction term g_k to each step proposal, with the aim of approximating the constraint rather than satisfying it exactly (Rosen, 1961). This is the approach chosen in the KratosMultiphysics Shape Optimization Application (Antonau *et al.*, 2022) and is reproduced here as part of Algorithm 3. The algorithm introduces a scaling parameter ψ_k for each iteration k that determines the magnitude of the correction term g_k. Then, in an **if** statement, we determine whether the value of G has changed its sign following the last update step. If this is true, we are close to the level set C. As a consequence, the parameter ψ_k, and thus the magnitude of the correction term, are reduced. The subsequent **else if** statement checks the opposite case, that is, whether we have drifted further from the constraint manifold during the two last step updates. In that case, the correction scaling is increased to counteract this drifting. As a last step, the correction term is calculated as the scaled gradient of the constraint function $\nabla G(x_k)$. This correction is then deducted from the proposed update step d_{S_k} to steer the iterates closer to the feasible set.

Algorithm 3 below adopts the SQP-inspired, projected gradient optimization step with an adaptive sample size relying on Test 2 and the correction scheme described above in a practical manner.

Algorithm 3: Adaptive sampling algorithm for stochastic programs with functional equality constraints

```
input: Feasible $x_0 \in C$, step size $\alpha > 0$, initial sample set $S_0$, constant $\theta > 0$, scaling parameter
       $\psi_0 > 0$.
Set $k \leftarrow 0$, $g_0 \leftarrow 0^n \in \mathbb{R}^n$ and compute $d_{S_0}$.
repeat
    if Condition 2 is satisfied then
        Update $x_{k+1} = x_k + \alpha d_{S_k} - \alpha \|d_{S_k}\| g_k$.
        Set $k \leftarrow k + 1$.
        Construct $S_k$ satisfying $|S_k| = |S_{k-1}|$.
        if $G(x_k)\ G(x_{k-1}) < 0$ then
            Set $\psi_{k+1} \leftarrow \frac{1}{2}\psi_k$ .
        else if $G(x_k)\ G(x_{k-1}) > 0$ and $\|G(x_k)\| > \|G(x_{k-1})\|$ then
            Set $\psi_{k+1} \leftarrow \min(2\psi_k, 1)$ .
        Compute correction term $g_k = \operatorname{sign}(G(x_k))\ \psi_{k+1} \frac{\nabla G(x_k)}{\|\nabla G(x_k)\|}$ .
    else
        Set $|S_k| \leftarrow \lceil \rho' \, |S_k| \rceil$, where $\rho' = \frac{\sum_{\xi \in S_k} \|R(x_k;\xi) - R_{S_k}(x_k)\|^2}{\theta^2 \, (|S_k| - 1) \, |S_k| \, \|R_{S_k}(x_k)\|^2}$ .
        Obtain additional i.i.d. samples for $S_k$.
    (Re-)compute $d_{S_k}$.
until a given convergence test is satisfied
```

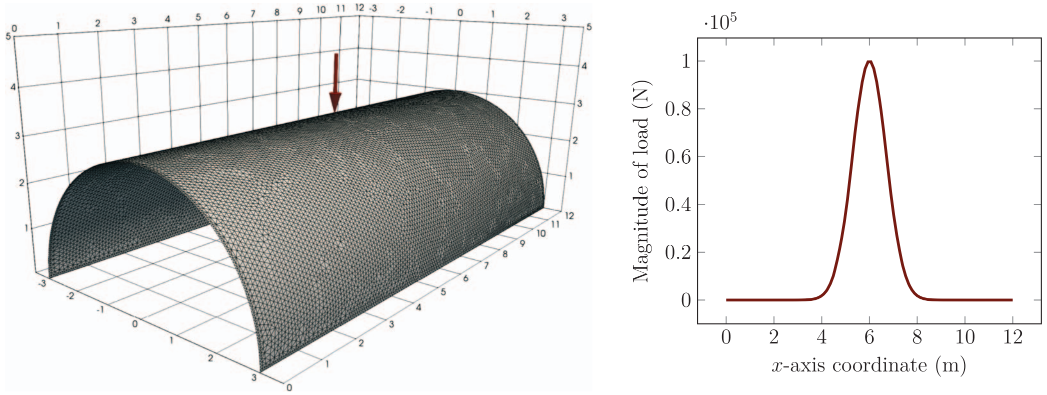


FIG. 4. Initial design of shell structure and applied force, **f**, when $a = 0$.

Note that in Algorithm 3, the sample set S_k is updated *before* progressing to the next iteration. This is a desirable choice when collecting samples is more expensive than solving (6.1) or (6.2), which happens to be the case in the following engineering application.

6.3 *Shape optimization of shell structures*

Finally, we turn our attention toward a problem in engineering shape design. Specifically, we consider the design of a thin steel shell structure with physical model uncertainties.

It is well accepted that shape optimization problems are difficult to characterize, as well as solve, and often involve significant engineering oversight (Bletzinger, 2017). The intention in such problems is usually not to seek a globally optimal design, but instead to begin with an initial 'good' design and find a nearby local optimum, which improves on a specified quantity of interest. This is the setting in which we test the performance of Algorithm 3.

Problem description Our chosen example centers on the question of how to find the shape of a steel shell that minimizes some measurement of the internal strains resulting from a specified distribution of external loads. For the initial shape, we choose a half-cylinder on its side, as depicted in Fig. 4. We assume that an uncertain load **f** will be applied to the shell structure from above, and that the final manufactured thickness t of the shell is also uncertain. To simplify our implementation, we assume that every cross-section of the applied load follows a simple bell shape profile along the major axis of the shell, and that the uncertainty in the load lies only in the position where it achieves its maximum. More specifically, we model the applied load (measured in Newtons) by the vector field

$$\mathbf{f}(x, y, z) = -10^5 \exp\big(-(x - 6 + 4a/3)^2\big)\mathbf{e}_z, \quad \text{with } a \sim \mathsf{N}(0, 1), \quad \mathbf{e}_z = (0, 0, 1),$$

and all distances measured in units of meters (m); cf. Fig. 4. Furthermore, we model the uncertain shell thickness by the uniformly distributed random variable

$$t \sim \mathsf{Unif}(4.05\ \text{cm}, 5.05\ \text{cm}).$$

It is of course possible to consider other uncertain model parameters, in addition to the thickness. In this example, however, we choose to fix the mass density ($7.85{\cdot}10^3$ kg m^{-3}), Young's modulus ($2.069{\cdot}10^{11}$ Pa) and Poisson's ratio ($2.9{\cdot}10^{11}$ Pa) of the steel shell structure, judging them to be far less sensitive sources of uncertainty.

Our goal here is to find a geometry parameterization x that optimizes the shell's internal energy $\varPi(x) = \varPi(x; \mathbf{f}, t)$, subject to the stochastic load $\mathbf{f}$ and thickness t, given above. In order to arrive at a realistic and practical optimum, we only look at a set of similarly expensive geometries—namely, those having (i) equal surface area—and physically reasonable geometries, wherein (ii) the supporting sides of the shell structure remain on the ground and (iii) the open ends of the structure stay perpendicular to the ground. These three sets of constraints lead to an abstract design space C and an associated stochastic optimization problem, which may be compactly written as

$$\min_{x\in C} \Big\{ F(x) = \mathcal{R}[\varPi(x)] \Big\}, \tag{6.3}$$

where $\mathcal{R}$ is a given risk measure. We only consider $\mathcal{R} = \mathrm{CVaR}^{\varepsilon}_{\beta}$, where $\beta \in [0, 1)$ and $\varepsilon \geq 0$.

Experiment setup It is common practice to represent the design geometry by the position of the nodes in its finite element representation (Bletzinger, 2014). These nodes, in turn, serve as control variables $x \in \mathbb{R}^n$, which may be updated along their physical normals at each step in the optimization algorithm (Bletzinger, 2017). In this example, we follow the semi-analytical adjoint-based procedure outlined in Bletzinger (2017, Subsection 5.5.4), and implemented in the KratosMultiphysics Structural Mechanics and Shape Optimization Application (Dadvand *et al.*, 2010). The geometry update rule we employ at the end of each optimization step k uses sophisticated filtering techniques and mesh movement algorithms outlined in Bletzinger (2014).

Using node positions as our design variable allows us to implement constraints (ii) and (iii) quite simply. By excluding the appropriate coordinates of the side and end nodes from design variable updates, we effectively fix them in the desired planes. Constraint (i) is encoded via the computation of the geometry's surface area $A(x)$ by requiring that $A(x) = A(x_0)$, with x_0 denoting the initial design. The starting design is thus feasible by definition. Gradients of A, which are required for computing the update step d_{S_k} in (6.1), were evaluated using an internal KratosMultiphysics (Dadvand *et al.*, 2010) routine. In practice, we add a correction term to each update step to account for the nonlinearity of this constraint, as described in Algorithm 3.

Following Bletzinger (2017), for each independent pair of load and thickness realizations, $\mathbf{f}_i$ and t_i, the calculation of the corresponding strain energy realization, $\varPi_i(x_k) = \varPi(x_k; \mathbf{f}_i, t_i)$, together with its gradient, $\nabla\varPi_i(x_k)$, involves the discrete solution of two partial differential equations (PDEs). In this study, we choose to represent the shell using the classical three-parameter Kirchhoff–Love PDE model and form a discretization of it with lowest-order C^0-continuous finite elements; cf. Bischoff *et al.* (2018). In particular, we use three-node ANDES elements (Felippa, 2003) on the ($n = 13783$)-node simplicial mesh depicted in Fig. 4.

We perform five numerical shape design optimization experiments with the discretization just described. The first experiment uses $\beta = \varepsilon = 0$. Since for $\beta = 0$ the CVaR is identical with the expectation, we require no smoothing in this case. The second, third, fourth and fifth experiments use $\beta = 0.5$, 0.75, 0.8 and 0.9, respectively, each with $\varepsilon = 0.1$. In the risk-neutral case (i.e., $\beta = 0$, $\mathcal{R} = \mathbb{E}$), we begin with an initial sample size of $|S_0| = 5$. In each of the risk-averse cases (i.e., $\beta > 0$, $\mathcal{R} = \mathrm{CVaR}^{\varepsilon}_{\beta}$), we begin with $|S_0| = 10$ in order to be able to produce meaningful estimates of t^* from the start. In the expectation case, the initial sample size is chosen smaller to exhibit a more notable

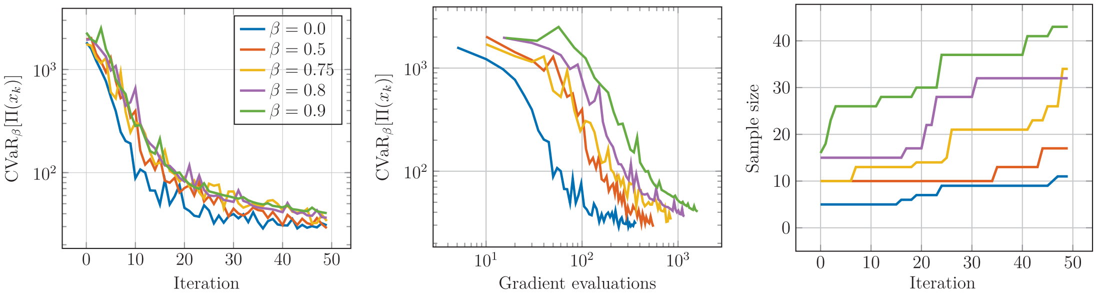


FIG. 5. Optimization logs for the design optimization problem of shell structures (6.3) with $\beta = 0,\ 0.5,\ 0.75,\ 0.8$ and 0.9. In all cases, the algorithm parameters $\alpha = 0.1$ and $\theta = 0.8$ were used. On the left, we see the convergence in the value of the objective function vs. the iteration number. In the middle, we see the convergence in the value of the objective function vs. the cumulative number of gradient evaluations. On the right, we see the growth in the sample size vs. iteration number.

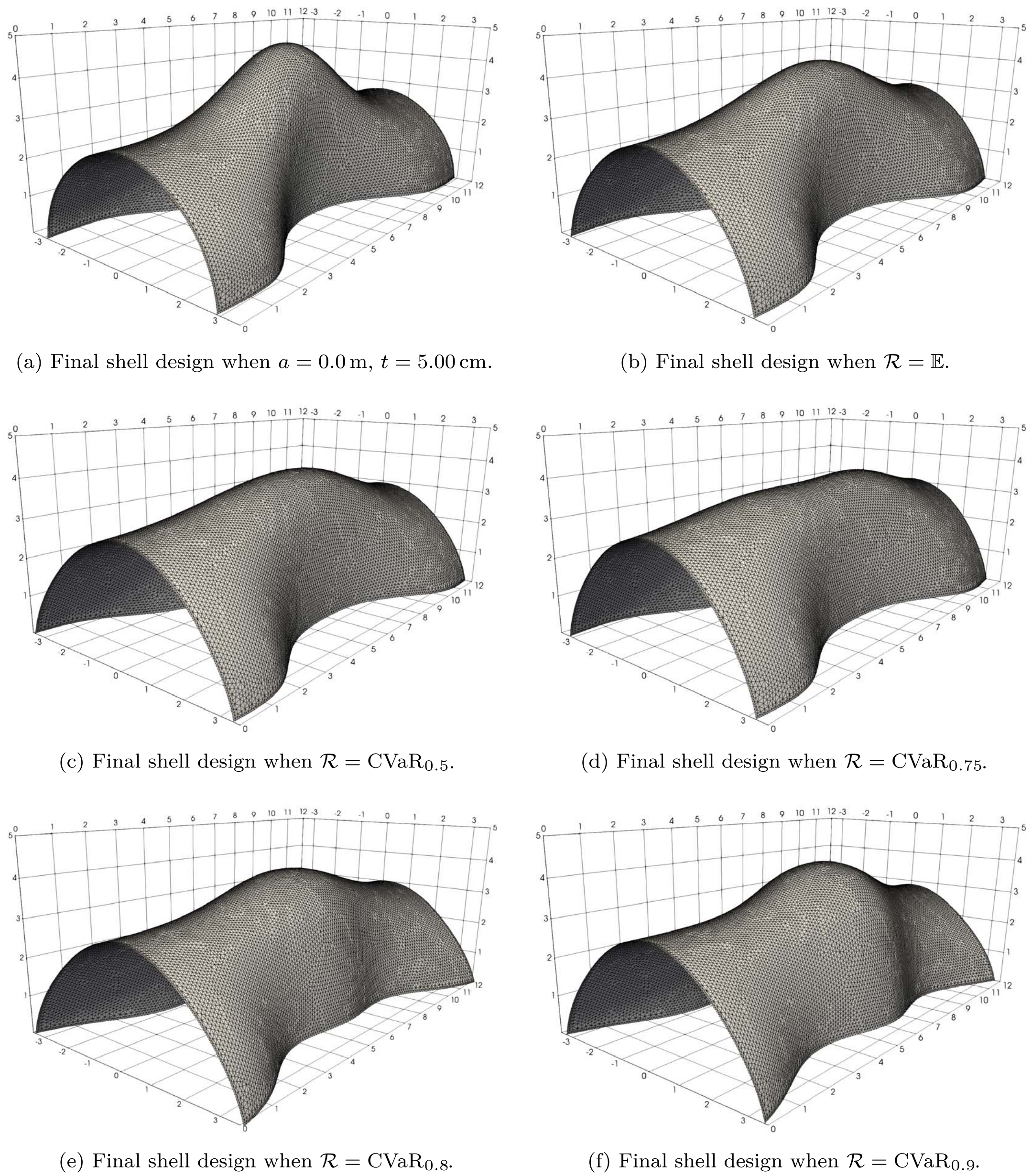
increase, as the sample size barely surpasses 10 toward the last iterations. In every experiment, we use the step size $\alpha = 0.1$ and the sampling rate parameter $\theta = 0.8$.

Each of the optimization problems is solved using the SQP approach introduced in Subsection 6.1. For the risk-neutral problem, $\beta = 0$, we use Algorithm 3. However, for the CVaR problems, $\beta > 0$, we modify the algorithm with the nested quantile estimation strategy described in Subsection 4.3. For further details, see Urbainczyk (2020, Chapter 4.5). In each experiment, the stochastic optimization algorithm is stopped after 50 iterations. Plots of the optimization logs are given in Fig. 5 and the final geometries are shown in Fig. 6. For visual comparison, we also present the final geometry one would find by optimizing the shell if it had exactly the expected thickness $t = 5.00$ cm and exactly the expected stress $\mathbf{f}(x, y, z) = -10^5 \exp\big(-(x-6)^2\big)\mathbf{e}_z$ was being applied (i.e., $a = 0.0$ m). The interested reader may also consult Urbainczyk (2020) for additional shape optimization experiments.

Results In Fig. 5, we see that the objective value in each stochastic setting decreases significantly throughout the course of optimization. As usual, the more risk-averse the problem, the more samples are required. In fact, in the risk-neutral setting, $\beta = 0$, just 10 samples is easily enough to fulfill Test 2 throughout nearly the entire course of the optimization. This is the reason why the results we present for this experiment begin with fewer than 10 samples. However, using fewer than 10 samples for the risk-averse experiments did not lead to predictable growth in the initial sample sizes. This is likely because of a large error in estimating the corresponding quantiles using (4.7) with very few samples.

It remains to investigate the performance of the correction term g_k introduced in Algorithm 3 to enforce the equality constraint in the numerical examples presented above. In Fig. 7, the geometry's surface area is shown for the expectation case, that is, $\mathcal{R} = \mathbb{E}$. The values of $A(x_k)$ are normalized by the initial surface area $A(x_0)$ and scaled by 100% so that, in an ideal scenario, we would observe a constant value of 100% as is indicated by the dotted line. In practice, one notes that we do not satisfy the constraint well after the first step. This is due to the fact that no correction term is computed for this step, as our correction procedure relies on two previous values. However, as the optimization progresses, the normalized surface area stabilizes around 100% as desired. Similar results were achieved for every other choice of risk measure considered in the numerical examples of this section.

It is interesting to note that the various optimization problems deliver visually distinct optimal shapes. The risk-neutral design in Fig. 6(b) is better-suited to the more likely, but less damaging, loads centered on the middle of the structure. This is most evident when it is compared with the optimal design in Fig. 6(a), which comes from the deterministic scenario where the load is centered on middle of the



(a) Final shell design when $a = 0.0\,\mathrm{m}$, $t = 5.00\,\mathrm{cm}$.

(b) Final shell design when $\mathcal{R} = \mathbb{E}$.

(c) Final shell design when $\mathcal{R} = \mathrm{CVaR}_{0.5}$.

(d) Final shell design when $\mathcal{R} = \mathrm{CVaR}_{0.75}$.

(e) Final shell design when $\mathcal{R} = \mathrm{CVaR}_{0.8}$.

(f) Final shell design when $\mathcal{R} = \mathrm{CVaR}_{0.9}$.

FIG. 6. Final shell designs for deterministic, risk-neutral and risk-averse optimization problems.

structure with absolute certainty. On the other hand, the risk-averse designs Figs 6(c–f) are progressively better suited to the less likely, but more damaging, loads centered near the ends of the structure. Initially, as β grows, we witness progressively flatter bumps in the center of the structure until a new type of shape appears somewhere around $\beta = 0.75$. This indicates a change in the local minima landscape. Likely

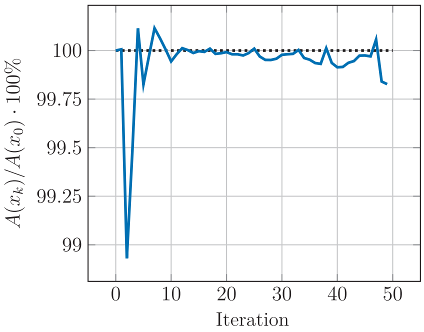


FIG. 7. Normalized surface area in the shape optimization experiment for $\mathcal{R} = \mathbb{E}$. The values were normalized by the initial geometry's surface area and are recorded as percentages.

there are two nearby local minima in this region, one that is a continuation of the risk-neutral minimum and a second local minimum influenced by the extreme values of the stress $\Pi(x)$. Additional numerical experiments (not included here) indicate to us that the basin of attraction of the new state exhibited in Figs 6(e and f) grows with $\beta > 0.75$. At the same time, the basin of attraction of the original risk-neutral category of designs appears to shrink and eventually vanish.

The experiments presented in this subsection highlight one important practical advantage of our adaptive sampling approach over fixed sample size SPGD. For many engineering applications, gradient evaluations form the majority of the total optimization cost. This is why we aim to reduce the amount of gradient computations. In the most risk-averse setting we considered, i.e., $\beta = 0.9$, we observe that the adaptive sample size reaches $|S_k| = 43$ at the final iteration, with 1600 gradient evaluations in total. Had we used this sample size from the start, we would have had to evaluate the gradient 2150 times. Therefore, in comparison, our adaptive sampling strategy decreased the total number of evaluations by $\sim 25\%$. In other examples, the savings were even greater. Indeed, when $\beta = 0.75$, adaptive sampling required $\sim 50\%$ fewer gradient computations than the fixed sample size approach.

7. Conclusion

This paper deals with stochastic optimization algorithms with dynamic sample sizes. We focus on a large class of stochastic programs with deterministic constraints. In doing so, we pose sufficient conditions on the sample sizes that guarantee in the case of convex constrains that a class of adaptive sampling methods converge. We then introduce practical tests to check these conditions and demonstrate their efficiency in a set of numerical examples. Our methods not only apply to risk-neutral optimization problems, that is, when the objective function is the expected value of some stochastic quantity. Indeed, using the CVaR as a working example, we show how a large family of risk-averse problems can be treated with the strategies developed here. Lastly, we propose an extension for more general nonconvex constraints, and illustrate its robustness in a contemporary application in engineering design with loading and model uncertainties.

Funding

European Union's Horizon 2020 research and innovation programme (800898); German Research Foundation (WO671/11-1); National Science Foundation (DMS-1439786); International Research Training Group IGDK1754, funded by the German Science Foundation and the Austrian Science Fund.

Acknowledgements

The authors are grateful for advice and suggestions from the anonymous referees, as well as Chandrajit Bajaj and Thomas Surowiec. The authors acknowledge the computer resources on MareNostrum and Salomon provided by the Barcelona Supercomputing Center (IM-2019-2-0008) and the Czech Republic Ministry of Education, Youth and Sports (from the Large Infrastructures for Research, Experimental Development, and Innovations project 'e-INFRA CZ – LM2018140'), respectively. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and the LLNL-LDRD Program under Project tracking No. 22-ERD-009. Release number LLNL-JRNL-827097.

References

Antonau, I., Warnakulasuriya, S., Bletzinger, K.-U., Bluhm, F. M., Hojjat, M. & Wüchner, R. (2022) Latest developments in node-based shape optimization using vertex morphing parameterization. *Struct. Multidiscip. Optim.*, **65**, 1–19.

Artzner, P., Delbaen, F., Eber, J.-M. & Heath, D. (1999) Coherent measures of risk. *Math. Finance*, **9**, 203–228.

Beiser, F., Keith, B., Urbainczyk, S. & Wohlmuth, B. (2020) Adaptive sampling strategies for risk-averse stochastic optimization with constraints. Preprint arXiv:2012.03844v1.

Ben-Tal, A. & Teboulle, M. (1986) Expected utility, penalty functions, and duality in stochastic nonlinear programming. *Manag. Sci.*, **32**, 1445–1466.

Bertsekas, D. P. (2015) *Convex Optimization Algorithms*. Nashua, New Hampshire, USA: Athena Scientific Belmont.

Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. (2017) Julia: a fresh approach to numerical computing. *SIAM Rev.*, **59**, 65–98.

Bischoff, M., Ramm, E. & Irslinger, J. (2018) Models and finite elements for thin-walled structures. *Encyclopedia of Computational Mechanics*, 2nd edn. Chichester, UK: Wiley, pp. 1–86.

Bletzinger, K.-U. (2014) A consistent frame for sensitivity filtering and the vertex assigned morphing of optimal shape. *Struct. Multidiscip. Optim.*, **49**, 873–895.

Bletzinger, K.-U. (2017) *Shape Optimization*. Chichester, UK: Wiley.

Bollapragada, R., Byrd, R. H. & Nocedal, J. (2018a) Adaptive sampling strategies for stochastic optimization. *SIAM J. Optim.*, **28**, 3312–3343.

Bollapragada, R., Mudigere, D., Nocedal, J., Shi, H. J. M. & Tang, P. T. P. (2018b) A progressive batching L-BFGS method for machine learning. *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 2, pp. 989–1013.

Bollapragada, R., Byrd, R. H. & Nocedal, J. (2019) Exact and inexact subsampled Newton methods for optimization. *IMA J. Numer. Anal.*, **39**, 545–578.

Bottou, L., Curtis, F. E. & Nocedal, J. (2018) Optimization methods for large-scale machine learning. *SIAM Rev.*, **60**, 223–311.

Byrd, R. H., Chin, G. M., Nocedal, J. & Wu, Y. (2012) Sample size selection in optimization methods for machine learning. *Math. Programming*, **134**, 127–155.

Carter, R. G. (1991) On the global convergence of trust region algorithms using inexact gradient information. *SIAM J. Numer. Anal.*, **28**, 251–265.

Cartis, C. & Scheinberg, K. (2018) Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Math. Programming*, **169**, 337–375.

Chaudhuri, A., Norton, M. & Kramer, B. (2020a) Risk-based design optimization via probability of failure, conditional value-at-risk, and buffered probability of failure. *AIAA Scitech 2020 Forum*. Reston, Virginia, USA: AIAA, p. 2130.

Chaudhuri, A., Peherstorfer, B. & Willcox, K. (2020b) Multifidelity cross-entropy estimation of conditional value-at-risk for risk-averse design optimization. *AIAA Scitech 2020 Forum*. Reston, Virginia, USA: AIAA, p. 2129.

Curi, S., Levy, K. Y., Jegelka, S. & Krause, A. (2019) Adaptive Sampling for stochastic risk-averse learning. *Adv. Neural Inf. Process. Syst.*, **33**, 1036–1047.

Dadvand, P., Rossi, R. & Onate, E. (2010) An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Arch. Comput. Methods Eng.*, **17**, 253–297.

De, S., Yadav, A., Jacobs, D. & Goldstein, T. (2017) Automated inference with adaptive batches. *Artificial Intelligence and Statistics*. Sheffield, UK: PMLR, pp. 1504–1513.

Dowd, K. (2007) *Measuring Market Risk*. Hoboken, New Jersey, USA: Wiley.

Felippa, C. A. (2003) A study of optimal membrane triangles with drilling freedoms. *Comput. Methods Appl. Mech. Engrg.*, **192**, 2125–2168.

Föllmer, H. & Schied, A. (2011) *Stochastic Finance: An Introduction in Discrete Time*. Berlin, Germany: Walter de Gruyter.

Friedlander, M. P. & Schmidt, M. (2012) Hybrid deterministic-stochastic methods for data fitting. *SIAM J. Sci. Comput.*, **34**, A1380–A1405.

Geiersbach, C., Loayza-Romero, E. & Welker, K. (2020) Stochastic approximation for optimization in shape spaces. *SIAM J. Optim.*, **31**, 348–376.

Hinze, M., Pinnau, R., Ulbrich, M. & Ulbrich, S. (2008) *Optimization with PDE Constraints*, vol. 23. Dordrecht, Netherlands: Springer Science & Business Media.

Homem-De-Mello, T. (2003) Variable-sample methods for stochastic optimization. *ACM Trans. Model. Comput. Simul.*, **13**, 108–133.

Ion, I. G., Bontinck, Z., Loukrezis, D., Römer, U., Ulbrich, S., Schöps, S. & Gersem, H. D. (2018) Robust shape optimization of electric devices based on deterministic optimization methods and finite-element analysis with affine parametrization and design elements. *Electr. Eng.*, **100**, 2635–2647.

Kodakkal, A., Keith, B., Khristenko, U., Apostolatos, A., Bletzinger, K.-U., Wohlmuth, B. & Wuechner, R. (2022) Risk-averse design of tall buildings for uncertain wind conditions. Preprint arXiv:2203.12060.

Kouri, D. P., Heinkenschloss, M., Ridzal, D. & van Bloemen Waanders, B. G. (2013) A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty. *SIAM J. Sci. Comput.*, **35**, A1847–A1879.

Kouri, D. P. & Shapiro, A. (2018) Optimization of PDEs with uncertain inputs. *Frontiers in PDE-Constrained Optimization*. SIAM, pp. 41–81.

Kouri, D. P. & Surowiec, T. M. (2016) Risk-averse PDE-constrained optimization using the conditional value-at-risk. *SIAM J. Optim.*, **26**, 365–396.

Kouri, D. P. & Surowiec, T. M. (2018) Existence and optimality conditions for risk-averse PDE-constrained optimization. *SIAM-ASA J. Uncertain. Quantif.*, **6**, 787–815.

Krokhmal, P., Palmquist, J. & Uryasev, S. (2002) Portfolio optimization with conditional value-at-risk objective and constraints. *J. Risk*, **4**, 43–68.

Na, S., Anitescu, M. & Kolar, M. (2021a) An adaptive stochastic sequential quadratic programming with differentiable exact augmented Lagrangians. Preprint arXiv:2102.05320.

Na, S., Anitescu, M. & Kolar, M. (2021b) Inequality constrained stochastic nonlinear optimization via active-set sequential quadratic programming. Preprint arXiv:2109.11502.

Nesterov, Y. (2018) *Lectures on Convex Optimization*, vol. 137, 2nd edn. Cham, Switzerland: Springer.

Nocedal, J. & Wright, S. (2006) *Numerical Optimization*. New York, New York, USA: Springer Science & Business Media.

Paquette, C. & Scheinberg, K. (2020) A stochastic line search method with expected complexity analysis. *SIAM J. Optim.*, **30**, 349–376.

Pasupathy, R., Glynn, P., Ghosh, S. & Hashemi, F. S. (2018) On sampling rates in simulation-based recursions. *SIAM J. Optim.*, **28**, 45–73.

Powell, M. J. (1983) Variable metric methods for constrained optimization. *Mathematical Programming the State of the Art*. Berlin, Germany: Springer, pp. 288–311.

R Core Team (2017) *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rockafellar, R. T. & Royset, J. O. (2010) On buffered failure probability in design and optimization of structures. *Reliab. Eng. Syst. Saf.*, **95**, 499–510.

Rockafellar, R. T. & Uryasev, S. (2000) Optimization of conditional value-at-risk. *J. Risk*, **2**, 21–41.

Rockafellar, R. T. & Uryasev, S. (2002) Conditional value-at-risk for general loss distributions. *J. Banking Finance*, **26**, 1443–1471.

Rockafellar, R. T. & Uryasev, S. (2013) The fundamental risk quadrangle in risk management, optimization and statistical estimation. *Surv. Operations Res. Manag. Sci.*, **18**, 33–53.

Rockafellar, R. T. & Royset, J. O. (2015) Engineering decisions under risk averseness. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part A Civ. Eng.*, **1**, 1–12.

Roosta-Khorasani, F. & Mahoney, M. W. (2019) Sub-sampled Newton methods. *Math. Programming*, **174**, 293–326.

Rosen, J. B. (1961) The gradient projection method for nonlinear programming. Part II. Nonlinear constraints. *J. Soc. Indust. Appl. Math.*, **9**, 514–532.

Royset, J. O. & Szechtman, R. (2013) Optimal budget allocation for sample average approximation. *Oper. Res.*, **61**, 762–776.

Shapiro, A., Dentcheva, D. & Ruszczyński, A. (2009) *Lectures on Stochastic Programming*. Philadelphia, Pennsylvania, USA: SIAM.

Shi, R., Liu, L., Long, T. & Tang, Y. (2018) Filter-based adaptive Kriging method for black-box optimization problems with expensive objective and constraints. *Comput. Methods Appl. Mech. Engrg.*, **347**, 782–805.

Trefethen, L. N. & Bau III, D. (1997) *Numerical Linear Algebra*, vol. 50. Philadelphia, Pennsylvania, USA: SIAM.

Ulbrich, M. & Ulbrich, S. (2012) *Nichtlineare Optimierung*. Basel, Switzerland: Springer.

Urbainczyk, S. (2020) Adaptive sampling for stochastic optimization with applications in risk-averse engineering design and machine learning. *Master's Thesis*, Germany: Technische Universität München.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., de Miranda Cardoso, J. V., Reimer, J., Harrington, J., RodrÍguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, Perry, McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O. & Vázquez-Baeza, Y. (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, **17**, 261–272.

Xie, Y. (2021) Methods for nonlinear and noisy optimization. *Ph.D. Thesis*. Northwestern University, Evanston, Illinois, USA.

Xie, Y., Bollapragada, R., Byrd, R. & Nocedal, J. (2020) Constrained and composite optimization via adaptive sampling methods. Preprint arXiv:2012.15411.

Yang, H. & Gunzburger, M. (2017) Algorithms and analyses for stochastic optimization for turbofan noise reduction using parallel reduced-order modeling. *Comput. Methods Appl. Mech. Engrg.*, **319**, 217–239.

Zou, Z., Kouri, D. & Aquino, W. (2019) An adaptive local reduced basis method for solving PDEs with uncertain inputs and evaluating risk. *Comput. Methods Appl. Mech. Engrg.*, **345**, 302–322.

A. More numerical examples

To complement the numerical experiments in Section 5, we present two short parameter studies. These illustrate the influence of the choice of step size and sampling rate on the optimization results presented earlier.

A.1 *Basic example: the step size*

The analysis in Section 2 suggests that we choose a fixed step size α that satisfies (2.32). For the example set-up in Subsection 5.1, we can estimate the Lipschitz constant $L \le 40$ and for given a_l, $l = 1, \ldots, 20$ calculate also a sharper constant. There, we have presented numerical results with $\alpha = 0.025$ corresponding to the rough estimate on L. In Fig. A1, we present further experiments to demonstrate how alternative step sizes influence the results.

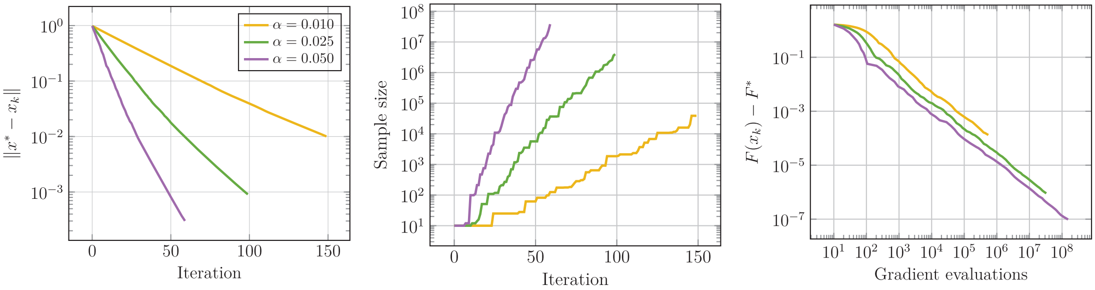


Fig. A1. Numerical results for step sizes $\alpha = 0.010, 0.025, 0.050$ while all other parameters are kept fixed as in Subsection 5.1 with $\theta = 0.5$.

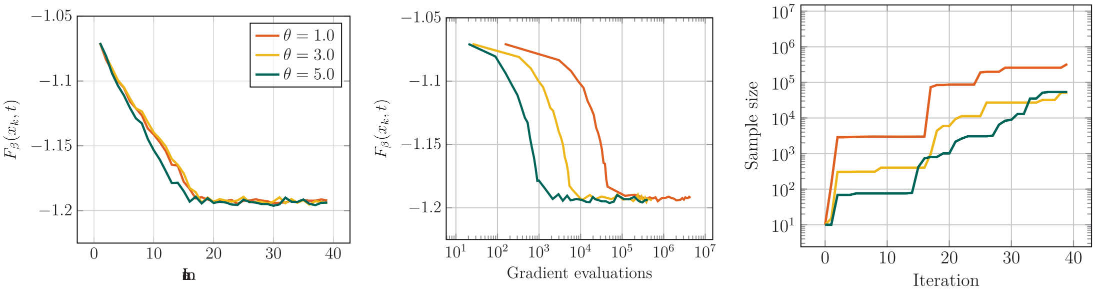


Fig. A2. Numerical results for the portfolio optimization problem (5.4) using Algorithm 1 for the sampling rates $\theta = 1.0, 3.0, 5.0$. Note that this problem is equivalent to (5.5) with $\beta = 0.0$.

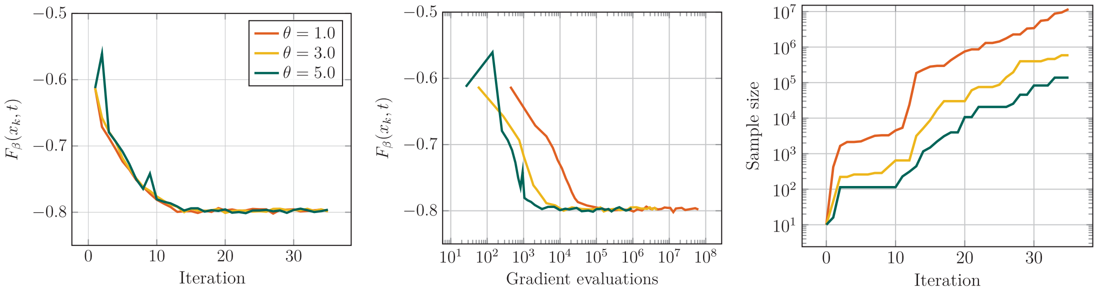


FIG. A3. Numerical results for the portfolio optimization problem (5.5) with $\beta = 0.5$ using Algorithm 1 for the sampling rates $\theta = 1.0, 3.0, 5.0$.

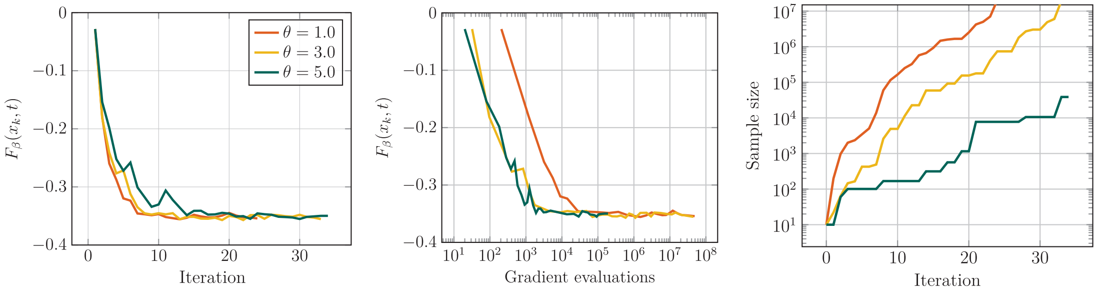


FIG. A4. Numerical results for the portfolio optimization problem (5.5) with $\beta = 0.9$ using Algorithm 2 for the sampling rates $\theta = 1.0, 3.0, 5.0$.

Qualitatively, q-linear convergence is observed for each step size in Fig. A1. Nevertheless, the largest step size ($\alpha = 0.050$) results in both the most iteration-efficient and sample-efficient choice. Although Algorithm 1 is used here, these results are in line with the analysis in Theorem 2.10.

A.2 *Portfolio optimization: the sampling rate*

In Subsection 5.2.2, we have analyzed different choices of the risk-averseness parameter β. Distinct choices of β change the optimization problem and, therefore, we used a different adaptive sampling rate θ for each β. We now study the influence of θ for three different fixed values of β, namely $\beta = 0, 0.5, 0.9$. Since Algorithm 2 performed better than Algorithm 1 in Subsection 5.2, we will focus our investigation on that particular algorithm when $\beta > 0$. The remaining problem set-up is identical to Subsection 5.2.2.

In Figs A2–A4, the influence of the sampling rate θ becomes visible: the sample size grows faster as θ decreases. The rate of growth of the sample size becomes more pronounced as β grows. At the same time, the growth pattern becomes smoother and there are fewer big jumps in the sample size for one iteration to the next.