Submitted to *Operations Research*
manuscript (Please, provide the manuscript number!)

Stochastically Constrained Simulation Optimization On Integer Lattices: The cgR-SPLINE Algorithm

Kalyani Nagaraj
School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078, kalyani.nagaraj@okstate.edu

Raghu Pasupathy
Department of Statistics, Purdue University, West Lafayette, IN 47907, pasupath@purdue.edu

We consider optimization problems whose domain is a subset of the integer lattice, and whose objective and constraint functions can only be observed using a stochastic simulation. Such problems seem particularly prevalent (see `www.simopt.org`) within service systems having capacity or service-level constraints. We present cgR-SPLINE — a random restarts algorithm that repeatedly executes a gradient-based simulation optimization (SO) routine on strategically relaxed sample-path problems, to return a sequence of local solution estimators at increasing precision; the local solution estimators are probabilistically compared to update an incumbent solution sequence that estimates the global minimum. Four issues are salient. (i) Solutions with binding stochastic constraints render naïve sample-average approximation inconsistent; consistency in cgR-SPLINE is guaranteed through sequential relaxation of the stochastic constraints. (ii) Light-tailed convergence that is characteristic of SO problems on unconstrained discrete spaces is weakened here; the general convergence rate is shown to be sub-exponential. (iii) An exploration-exploitation characterization demonstrates that cgR-SPLINE achieves the fastest convergence rate when the number of restarts is proportional to the simulation budget per restart; this is in contrast with the continuous context where much less exploration has been prescribed. (iv) Certain heuristics on choosing constraint relaxations, solution reporting, and premature stopping ensure that cgR-SPLINE exhibits good finite-time performance while retaining asymptotic properties. We demonstrate the functioning of cgR-SPLINE on two nontrivial examples; downloadable code can be obtained at `http://iem.okstate.edu/nagaraj`.

Key words: stochastic constraints; integer-ordered simulation optimization; cgR-SPLINE

1. INTRODUCTION

We consider the problem of solving a constrained optimization problem over an integer-ordered lattice, when the objective and constraint functions involved in the problem can only be observed (consistently) through a stochastic simulation. Formally, we consider problems of the form

$$\begin{aligned} P: \text{ minimize } & g(\boldsymbol{x}) \\ \text{subject to } & h_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, \ell, \\ & \boldsymbol{x} \in \mathbb{X}, \end{aligned}$$

where the set $\mathbb{X}$ is a subset of the d-dimensional integer lattice $\mathbb{Z}^d$, and the functions $g(\boldsymbol{x})$ and $h_i(\boldsymbol{x})$ are estimated through simulation-based function estimators $\hat{g}_m(\boldsymbol{x})$ and $\hat{h}_{i,m}(\boldsymbol{x})$, $1 \leqslant i \leqslant \ell$. We assume that $\hat{g}_m(\boldsymbol{x})$ and $\hat{h}_{i,m}(\boldsymbol{x})$, $1 \leqslant i \leqslant \ell$ are well-defined random functions that for each $\boldsymbol{x} \in \mathbb{X}$ satisfy $\lim_{m\to\infty} \hat{g}_m(\boldsymbol{x}) = g(\boldsymbol{x})$ with probability one (wp1) and $\lim_{m\to\infty} \hat{h}_{i,m}(\boldsymbol{x}) = h_i(\boldsymbol{x})$ wp1, $1 \leqslant i \leqslant \ell$, with m representing some measure of simulation effort. A frequently occurring setting involves $g(\boldsymbol{x}) = \mathrm{E}[G(\boldsymbol{x})], h_i(\boldsymbol{x}) = \mathrm{E}[H_i(\boldsymbol{x})]$ and a stochastic simulation that, for each $\boldsymbol{x} \in \mathbb{X}$, generates independent and identically distributed (iid) copies $\boldsymbol{F}_j(\boldsymbol{x}) := (G_j(\boldsymbol{x}), H_{1,j}(\boldsymbol{x}), H_{2,j}(\boldsymbol{x}), \ldots, H_{\ell,j}(\boldsymbol{x})), j = 1, 2, \ldots$, of the random vector $\boldsymbol{F}(\boldsymbol{x}) := (G(\boldsymbol{x}), H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_\ell(\boldsymbol{x}))$. The function estimators in this context are the simple sample means $\hat{g}_m(\boldsymbol{x}) = m^{-1}\sum_{j=1}^m G_j(\boldsymbol{x})$, $\hat{h}_{i,m}(\boldsymbol{x}) = m^{-1}\sum_{j=1}^m H_{i,j}(\boldsymbol{x})$.

We emphasize that the statement of Problem P implies that only function estimates of $g(\boldsymbol{x}), h(\boldsymbol{x})$ are available through the simulation oracle. Any solution procedure to Problem P will thus likely be a numerical algorithm that generates a random sequence of iterates to approximate a solution. Whether this generated sequence of random iterates converges to a correct solution in some rigorous sense, and if so, its convergence-rate as expressed in terms of the total expended simulation effort, will concern us. Apart from the two asymptotic measures, *convergence* and *convergence-rate*, empirical evidence of good finite-time algorithm performance will be used as practical evidence of a solution procedure's effectiveness.

REMARK 1. For the purposes of this paper, the notion of a "simulation" is broadly interpreted. For instance, settings where an existing large database of randomly generated scenarios (or collected data) that can be used towards constructing the estimators $\hat{g}_m(\boldsymbol{x})$ and $\hat{h}_{i,m}(\boldsymbol{x})$ fall within the scope of Problem P.

Integer-ordered simulation-optimization (SO) problems of the type Problem P are widely prevalent, appearing in decision-making settings such as production systems (Sarin and Jaiprakash 2010), call center staffing (Gans et al. 2003), bus or rail fleet-size management (Bish 2011, Bish et al. 2011), communication network design (Hou et al. 2014), and vaccine allocation within epidemic spreading models (Eubank et al. 2004). Surprisingly, more than sixty percent of the problems submitted to the SO problem library (`www.simopt.org`) are integer-ordered SO problems. (Specific examples, including downloadable oracle code, of integer-ordered and other SO problems are available through `www.simopt.org`.)

1.1. Questions Considered

What makes solving Problem P difficult? To answer this question in part, consider a simple version of the problem P where the objective and constraint functions are $g(\boldsymbol{x}) = x_1 + 3x_2^2$ and $h(\boldsymbol{x}) = 10 - 2x_1 - x_2$, and the domain $\mathbb{X}$ equals $\mathbb{Z}_+^2$. Let the estimators $\hat{g}_m(\boldsymbol{x}) = g(\boldsymbol{x})$, and $\hat{h}_m(\boldsymbol{x}) = m^{-1}\sum_{j=1}^{m} H_{1,j}(\boldsymbol{x})$, where $H_{1,j}(\boldsymbol{x}), j = 1, 2, \ldots, m$ are independent and identically distributed (iid) random variables having mean $h(\boldsymbol{x})$ and variance $\sigma^2 > 0$. As the left panel of Figure 1 illustrates, the unique solution to this problem is $(5, 0)$, and importantly, it lies on the boundary of the feasible region. This latter feature — the solution to the optimization problem lies on (or near) the boundary of the feasible region — is a crucial complication. To see this, first note that $\frac{\sqrt{m}}{\sigma}(\hat{h}_m(\boldsymbol{x}) - h(\boldsymbol{x}))$ converges weakly to $\mathcal{N}(0, 1)$ for all $\boldsymbol{x}$, where $\mathcal{N}(0, 1)$ is a standard normal random variable. Now suppose an algorithmic procedure encounters the solution $\boldsymbol{x}^* = (5, 0)$ and attempts to evaluate its feasibility. Since $\hat{h}_m(\boldsymbol{x}^*)$ is all that is observed by the procedure, and $\hat{h}_m(\boldsymbol{x}^*)$ is approximately normally distributed with mean 0 and variance σ^2/m, there is roughly a 0.5 probability that

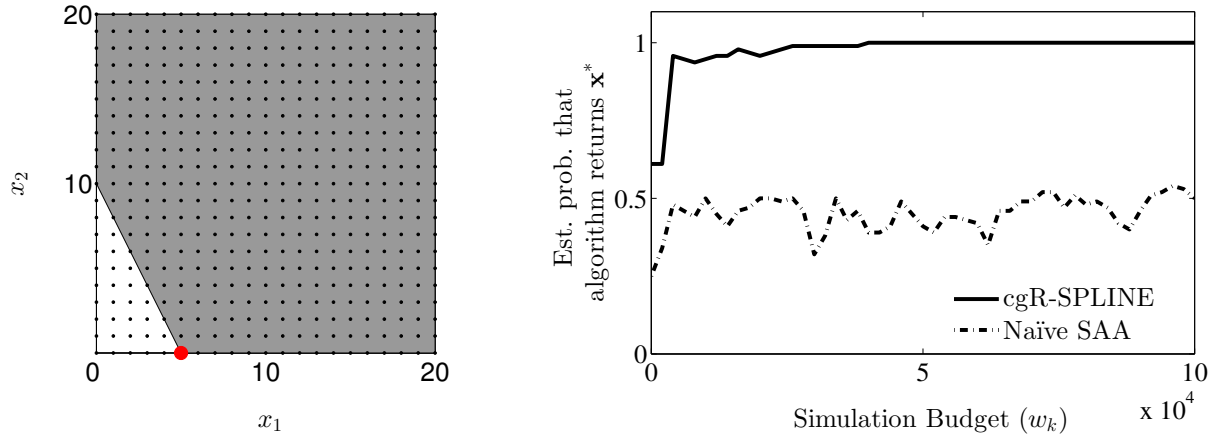

Figure 1 Illustration of a typical complication in a stochastically-constrained SO problem when the solution lies on the boundary of the feasible region. The shaded region on the left represents the feasible region of a stochastically-constrained SO problem and the solid dot at $\boldsymbol{x}^* = (5, 0)$ is its solution. In the above situation, it can be shown that under widely prevalent conditions, irrespective of how much sampling is performed, the solution $\boldsymbol{x}^* = (5, 0)$ cannot be identified as being feasible with certainty. In fact, there is roughly only a fifty percent chance that the solution $\boldsymbol{x}^* = (5, 0)$ will be deemed feasible even as the sample size tends to infinity! This renders naïve sample–average approximation (SAA) procedures inconsistent as shown by the dashed curve on the right. cgR-SPLINE addresses the problem of algorithmic consistency (as shown by the solid curve on the right) by relaxing the stochastic constraint and "pulling them in" at a carefully specified rate.

$\hat{h}_m(\boldsymbol{x}^*) > 0$, that is, there is (roughly) a fifty percent chance that $\boldsymbol{x}^* = (5, 0)$ is deemed infeasible. What is worse, this is irrespective of the sample size m; in fact, the probability of $\boldsymbol{x}^* = (5, 0)$ being deemed infeasible is exactly 0.5 as $m \to \infty$, as illustrated by the right panel of Figure 1.

The challenge associated with detecting feasibility is not pathological to the example we have just described. Instead, we believe it is the norm in problems with resource or service-level constraints, where the solution to the optimization problem often lies on a boundary that can only be estimated by simulation. Algorithms for solving Problem P thus have to do something special in order to successfully recognize a solution (at least asymptotically) and produce iterates that are consistent.

Q.1 Given that the objective and constraint functions in Problem P can only be estimated through a stochastic simulation, what is a consistent algorithmic procedure for solving problems of

the type P, particularly one that tackles settings having binding stochastic constraints at the solution(s)?

An iterative algorithm that addresses $Q.1$ will likely involve following four repeating steps: (i) use a strategy to identify the next point to visit in $\mathbb{X}$; (ii) at the visited point, estimate the objective and constraint function values to specified precision by "executing" the simulation with "adequate sampling effort"; (iii) update the estimated solution; and (iv) update objects (e.g., derivative estimates) that will be used within step (i) during the subsequent iteration. At a broad level, steps (i) – (iv) are no different from any numerical procedure to solve a deterministic optimization problem. At a detail level, however, a crucial difference arises in step (ii) where the procedure needs to decide how much simulation effort to expend at a particular visited point. Exerting "too little" simulation sampling effort at a point leads to poor precision in the resulting objective and constraint function estimates, and consequent loss in guarantees of consistency; exerting "too much" simulation effort at each point, on the other hand, can lead to inefficiencies, that is deviations from the fastest achievable convergence rate. So, we ask:

Q.2 How should sampling be performed within a procedure that addresses $Q.1$ so that the resulting iterates converge to the solution of P at the fastest possible rate?

Q.3 What is the fastest achievable rate (as a function of the total simulation effort expended) at which solution(s) to Problem P can be identified?

While we seek algorithms that are endowed with desirable asymptotic properties as reflected in questions $Q.1$, $Q.2$, and $Q.3$, we also seek algorithms that exhibit good finite-time performance without user-tuning of algorithm parameters. This leads us to ask:

Q.4 What heuristics ensure that algorithms to solve Problem P exhibit good finite-time performance, while retaining "optimal" asymptotic properties?

We recognize that an answer to $Q.4$ can only be empirical, and the heuristics devised in response to $Q.4$ will be "common sense" strategies that will automatically make algorithmic decisions to aid good empirical performance without having the need for user intervention.

2. RELATED LITERATURE

The challenge posed by feasible regions defined by stochastic constraints is well-recognized and has seen considerable treatment in recent years, particularly within stochastic programming. Two formulations have become popular. The first, usually called *probabilistic constraints* or *chance constraints*, defines the feasible region $\mathcal{F} \triangleq \{\boldsymbol{x} : \Pr\{H_i(\boldsymbol{x}) \leq 0, i = 1, 2, \ldots, \ell\} \geq 1 - \alpha\}$, implying that the feasibility of a point $\boldsymbol{x}$ is decided based on whether $\boldsymbol{x}$ is feasible with respect to at least "$1 - \alpha$ fraction" of the sample-path constraint functions. While attractive, it is well-known that chance-constrained formulations are non-convex in general, unless strong assumptions are imposed on the probability measures driving the sample-path constraints (Shapiro et al. 2009). Accordingly, methods to solve probabilistically constrained formulations typically construct convex and conservative approximations of $\mathcal{F}$ in such a way that the resulting problems are easily solved, and yield solutions that are feasible (with prescribed probability) to the original problem. Notable examples of such treatment include Calafiore and Campi (2005, 2006), Nemirovski and Shapiro (2006a,b). Since a more detailed discussion of such treatment will take us far afield, we refer the interested reader to Chapter 4 in Shapiro et al. (2009).

The second popular treatment, and the one we consider here within a more stylized setting, is what has traditionally been called *expected value constraints*. It seems that expected value constrained problems were first studied by Prékopa (1973), Prékopa (1995), but the last two decades have seen further development of specific algorithms derived by imposing structural conditions. Here again, we do not go into further detail owing to lack of direct relevance but note in passing that algorithms in such contexts usually exploit some imposed structural conditions. For example, O'Brien (2000) develops specific algorithms for the context where the distributions underlying the sample-path outcomes have known finite support, Kuhn (2009) develops algorithms for multistage stochastic programs with expected value constraints assuming that the sample-path constraint functions are convex, and Atlason et al. (2008) develop cutting-plane SO algorithms for call-center staffing problems assuming that the constraint functions (or service-level functions) are either discrete concave or discrete pseudo-concave.

More recently, Wang and Ahmed (2008) consider SAA algorithms with expected value constraints and develop minimum sample size results to guarantee the quality of the resulting sample-path feasible regions. Specifically, Wang and Ahmed (2008) construct inner and outer bounding sets by ϵ-displacing the (true) constraint functions of the original problem inward and outward respectively, towards guaranteeing that the resulting sample-path feasible region lies "within" such bounding sets with a specified probability. Such constraint displacement ideas also appear in Hernández-Lerma and Lasserre (1998a,b) for approximating certain optimal control problems with a sequence of finite linear programs. As we shall see, the algorithms we propose also use constraint displacements; however, as in Hernández-Lerma and Lasserre (1998a,b), the displacements we propose are strategically chosen and introduced in a sequential form across iterations. Such a sequential refinement complicates asymptotic analysis but is usually a powerful way to avoid the inefficiencies that are known to result from solving a single "large" representative problem. See Kim et al. (2014) for more on similar sequential methods in the unconstrained context, and Siegmund (1985) for general ideas on sequential estimation.

The specific problem class we consider in this paper is SO on integer-ordered feasible regions having expected value constraints. Such problems seem widely prevalent but, interestingly, methods for their solution are still relatively few. One example is (Lim 2012, Luo and Lim 2013), where stochastic approximation (SA) is generalized to work on discrete sets through an appropriate extension of the functions involved in the problem. Luo and Lim (2013) is particularly relevant since it incorporates stochastic constraints, through the use of a Lagrangian. Like SA, methods in Lim (2012) and Luo and Lim (2013) guarantee almost sure convergence to a local minimizer, although it is unclear as to what convergence rates are guaranteed by these methods.

Two other noteworthy competitors, Park and Kim (2011, 2015) and Li et al. (2009), use a penalty function formulation to deal with stochastic constraints, with penalties being updated as the algorithm evolves. The key stipulation in Park and Kim (2011, 2015) is that the imposed function penalties be chosen carefully; specifically, the penalties (across visits) due to a constraint

should be chosen in such a way that they converge to zero or diverge depending on whether or not the constraint is satisfied. For satisfying this stipulation, Park and Kim (2011, 2015) introduce a geometric parameter sequence that attains the correct limit based on observed constraint function estimates.

A few points of comparison between Park and Kim (2011, 2015) and what we propose are illustrative. First, the framework in Park and Kim (2011, 2015) assumes the use of a global SO algorithm. Two examples of such algorithms are Nested Partitions (Shi and Ólafsson 2000) and Random Search (Andradóttir 2006). By contrast, we choose to use random restarts of a local SO algorithm such as R-SPLINE (Wang et al. 2013) and COMPASS (Xu et al. 2010, Hong and Nelson 2006). Our choice is dictated in part by analogous debates in the deterministic context (Pardalos and Romeijn 2002) on the use of branch and bound adaptations versus the repeated use of fast local solvers from different starting points. Second, the penalty function weights in Park and Kim (2011) are analogous to the constraint relaxation amounts in the proposed method. We believe that both penalty function weights and constraint relaxations are influential parameters in their respective algorithms in the sense that their settings crucially affect algorithm performance. Accordingly, our theoretical results and numerical illustrations incorporate a certain function of the sample variance (observable) in the constraint estimates when setting the constraint relaxations. Third, as we will demonstrate, our methods guarantee almost sure convergence at rates that are arbitrarily close to the fastest achievable convergence rate for unconstrained, discrete simulation optimization problems (Kleywegt et al. 2001). It is likely that similar rates are obtainable in Park and Kim (2011, 2015) by appropriately setting the geometric sequence of penalties; identifying such sequences in Park and Kim (2011, 2015) appears to be an unresolved question.

We remind the reader that the now popular COMPASS (Xu et al. 2010, Hong and Nelson 2006), Industrial Strength COMPASS (Xu et al. 2010), and R-SPLINE (Wang et al. 2013) are all algorithms constructed for functioning on integer-ordered spaces. However, their scope includes only settings where the constraint set is either deterministic or void. In fact, COMPASS and R-SPLINE are both local solvers that can be adapted for use within the framework that we propose.

The entire recent literature on ranking and selection (R&S) in the presence of stochastic constraints (Andradóttir and Kim 2010, Andradóttir et al. 2005, Batur and Kim 2005, Hunter and Pasupathy 2010, Hunter et al. 2011, Pasupathy et al. 2014, Hunter and Pasupathy 2013) applies to the class of integer-ordered SO problems as long as the domain is finite. R&S algorithms are, however, constructed for settings that assume no topology in the domain; hence, they are designed to sample from *every* system (albeit to varying extents) in the feasible space in order to make inferences on optimality. For this reason, in the current integer-ordered setting, R&S algorithms will likely be uncompetitive with tailored algorithms that exploit known structure.

3. NOTATION AND CONVENTION

We will adopt the following notation through the paper. (i) If $\boldsymbol{x} \in \mathbb{R}^d$ is a vector, then its components are denoted through $\boldsymbol{x} := (x_1, x_2, \ldots, x_d)$. (ii) $\mathbb{Z}^d \subset \mathbb{R}^d$ represents the integer lattice in d-dimensional Euclidean space. (iii) If $\mathcal{F}$ represents a finite set, then $|\mathcal{F}|$ represents the cardinality of the set $\mathcal{F}$. (iv) For a sequence of random vectors $\{\boldsymbol{X}_n\}$, we say $\boldsymbol{X}_n \overset{\text{wp1}}{\to} \boldsymbol{x}$ to mean that $\{\boldsymbol{X}_n\}$ converges to $\boldsymbol{x}$ with probability one. (v) For a sequence of real numbers $\{a_n\}$, we say $a_n = o(1)$ if $\lim_{n\to\infty} a_n = 0$ (or, $a_n = o^{-1}(1)$ if $a_n^{-1} = o(1)$); and $a_n = O(1)$ if $\{a_n\}$ is bounded, i.e., $\exists c \in (0,\infty)$ with $|a_n| < c$ for large enough n. We say that $a_n = \Theta(1)$ if $0 < \liminf a_n \leq \limsup a_n < \infty$. (vi) For a fixed set $\mathcal{F}$ and a sequence of random sets $\{\mathcal{F}_k\}$, each a subset of $\mathbb{X} \subseteq \mathbb{Z}^d$, define a sequence of indicator functions $\{I_k\}_{k=1}^{\infty}$ as follows. For each $\boldsymbol{x} \in \mathbb{X}$, $I_k(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in (\mathcal{F} \bigcap \mathcal{F}_k) \bigcup (\mathcal{F}^c \bigcap \mathcal{F}_k^c)$, otherwise $I_k(\boldsymbol{x}) = 1$. We then say $\mathcal{F}_k \overset{\text{wp1}}{\to} \mathcal{F}$ uniformly on $\mathbb{X}$ if $I_k \overset{\text{wp1}}{\to} 0$ uniformly on $\mathbb{X}$. (vii) The neighborhood $N(\mathbf{0})$ of the d-dimensional origin is defined as any subset of $\mathbb{Z}^d$ containing the origin. The corresponding neighborhood of any non-zero d-dimensional integer point $\boldsymbol{x}$ is the set $N(\boldsymbol{x}) = \{\boldsymbol{y} : (\boldsymbol{y} - \boldsymbol{x}) \in N(\mathbf{0})\}$. (viii) For a given neighborhood definition N and a function $g : \mathbb{X} \subseteq \mathbb{Z}^d \to \mathbb{R}$, we say a point $\boldsymbol{x}^* \in \mathbb{X}$ is an *N-local minimizer* of g if $g(\boldsymbol{x}^*) \leq g(\boldsymbol{x})$ for all $\boldsymbol{x} \in N(\boldsymbol{x}^*) \cap \mathbb{X}$.

4. cgR-SPLINE OVERVIEW AND LISTING

Fundamental to cgR-SPLINE is the notion of a relaxed sample-path problem $P(m, \boldsymbol{\epsilon})$ defined as

$$P(m, \boldsymbol{\epsilon}) : \text{ minimize } \hat{g}_m(\boldsymbol{x})$$

$$\text{subject to } \hat{h}_{i,m}(\boldsymbol{x}) \leqslant \epsilon_i,\ i = 1, \ldots, \ell,$$

$$\boldsymbol{x} \in \mathbb{X}.$$

We see that the sample-path problem $P(m, \boldsymbol{\epsilon})$ is obtained by replacing the objective function g and the constraint functions $h_i, i = 1, 2, \ldots, \ell$ in Problem P by their corresponding estimators $\hat{g}_m$ and $\hat{h}_m$ obtained using a sample size m. Importantly, the constraints appearing in Problem P are relaxed by an amount $\boldsymbol{\epsilon} > \mathbf{0}$. As we shall see, the extent of such relaxation can be guided by the standard error of the constraint estimators. Although we have suppressed explicit notation for now, the tolerances $\boldsymbol{\epsilon}$ will eventually dependent on both $\boldsymbol{x}$ and the sample size m.

The feasible and infeasible regions associated with Problem P and Problem $P(m, \boldsymbol{\epsilon})$ are then given by $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{X} :\ h_i(\boldsymbol{x}) \leqslant 0, i = 1, \ldots, \ell\}$ and $\mathcal{F}^c = \mathbb{X} \backslash \mathcal{F}$, and $\mathcal{F}(m, \boldsymbol{\epsilon}) = \{\boldsymbol{x} \in \mathbb{X} :\ \hat{h}_{i,m}(\boldsymbol{x}) \leqslant \epsilon_i, i = 1, \ldots, \ell\}$ and $\mathcal{F}^c(m, \boldsymbol{\epsilon}) = \mathbb{X} \backslash \mathcal{F}(m, \boldsymbol{\epsilon})$, respectively. Also, for a given neighborhood N, let $\mathcal{M}_N = \{\boldsymbol{x}^* \in \mathcal{F} :\ g(\boldsymbol{x}^*) \leqslant g(\boldsymbol{x}),\ \forall \boldsymbol{x} \in N(\boldsymbol{x}^*) \cap \mathcal{F}\}$ (henceforth referred to as $\mathcal{M}$ for notational simplicity) denote the set of N-local minima of Problem P.

cgR-SPLINE, listed in Algorithm 1, has a straightforward iterative structure organized into what we call *inner* and *outer iterations*. During each outer iteration r, an estimate $\boldsymbol{Y}_r$ of a local solution is identified by executing a locally minimizing SO algorithm (Steps 4 – 10 in Algorithm 1) with an appropriately generated initial guess $\boldsymbol{X}_r$. cgR-SPLINE uses R-SPLINE (Wang et al. 2013) as the locally minimizing SO algorithm. R-SPLINE is itself an iterative algorithm as shown in Steps 4 – 10 of Algorithm 1; it is for this reason that we qualify iterations in cgR-SPLINE as *inner* and *outer* iterations. During each outer iteration r, R-SPLINE is capable of generating a sequence of solutions $\{\boldsymbol{W}_{k,r}\}$ to relaxed sample-path Problems $\{P(m_k, \boldsymbol{\epsilon}_k)\}$ formed by progressively increasing the sample size m_k and simultaneously tightening the constraint relaxation $\boldsymbol{\epsilon}_k$. The inner iterations of cgR-SPLINE (indexed by k) correspond to the *retrospective* iterations of R-SPLINE (see Remark 2) and are performed independently in the sense that different sample-path problems $P(m_k, \boldsymbol{\epsilon}_k)$ use different random numbers.

Key Notation

$\boldsymbol{X}_r$: initial guess for rth outer iteration

b_r : simulation budget for rth outer iteration

m_k : sample size used during the kth inner iteration

$\boldsymbol{\epsilon}_k$: constraint relaxation vector used during the kth inner iteration

$\boldsymbol{W}_{k,r}$: solution returned at the end of the kth inner iteration of the rth outer iteration

$\boldsymbol{Y}_r$: local solution returned after the rth outer iteration

$\boldsymbol{Z}_r$: incumbent (estimated global) solution at the end of the rth outer iteration

Algorithm cgR-SPLINE

Require: initial guesses $\{\boldsymbol{X}_r\}$; restart budgets $\{b_r\}$; inner sample sizes $\{m_k\}$; constraint relaxations $\{\boldsymbol{\epsilon}_k\}$

Ensure: incumbent solutions $\{\boldsymbol{Z}_r\}$, local minimizers $\{\boldsymbol{Y}_r\}$

1: Initialize: outer iteration number $r = 1$
2: Choose: the outer simulation budget b_r
3: Generate: an initial guess $\boldsymbol{X}_r$ to the rth restart
4: Initialize: inner iteration number $k = 0$, restart budget utilized $B_r = 0$, $\boldsymbol{W}_{k,r} = \mathbf{X}_r$
5: **while** $B_r < b_r$ and $\mathbf{W}_{k,r} \neq \emptyset$ {outer simulation budget not exceeded and Line 8 returns sample-path feasible solution}
6: Update: inner iteration $k = k + 1$
7: Choose: the next sample size m_k and constraint relaxation vector $\boldsymbol{\epsilon}_k$
8: Obtain: Solution $\boldsymbol{W}_{k,r}$ to Problem $P(m_k, \boldsymbol{\epsilon}_k)$ after expending $N_{k,r}$ oracle calls and using a warm start $\boldsymbol{W}_{k-1,r}$, where $\boldsymbol{W}_{0,r} = \boldsymbol{X}_r$
9: Update: utilized budget $B_r = B_r + N_{k,r}$
10: **end while**
11: Update: returned local solution $\boldsymbol{Y}_r = \boldsymbol{W}_{k,r}$
12: Update: incumbent sample size $t_r = \max\{t_{r-1}, m_k\}$, where $t_0 = 0$
13: Update: incumbent solution $\boldsymbol{Z}_r = \arg\min\{\hat{g}_{t_r}(\boldsymbol{x}) : \boldsymbol{x} \in \{\boldsymbol{Z}_{r-1}, \boldsymbol{Y}_r\} \cap \mathcal{F}(t_r, \boldsymbol{\epsilon}(t_r))\}$, where $\mathbf{Z}_0 = \mathbf{X}_1$ {compare $\mathbf{Y}_k$ and $\mathbf{Z}_{r-1}$ at sample size t_r}
14: Update: outer iteration $r = r + 1$
15: Go to: Step 2

Locally minimizing SO algorithm: Steps 4–10 (Inner iterations); Steps 2–15 (Outer iterations)

Algorithm 1 cgR-SPLINE has *outer iterations*, each of which returns an estimator of a local extremum by partially solving a strategically relaxed sample-path problem using a locally minimizing algorithm. While the above listing of cgR-SPLINE uses the logic of R-SPLINE (Steps 4 – 10), other suitably adapted locally minimizing SO algorithms may be used instead. Estimators of local extrema obtained across outer iterations are appropriately compared to yield a sequence of global estimators that we call incumbent solutions. A detailed discussion of the SPLINE algorithm that performs Step 8 and how it handles infeasible points appears in Section EC.2 of the e-companion.

REMARK 2. Although any locally minimizing SO algorithm that can be adapted to handle stochastic constraints can replace R-SPLINE in Steps 4–10 of Algorithm 1, we recommend R-SPLINE due to its many desirable properties. R-SPLINE is placed within an iterative framework called retrospective approximation (RA) that facilitates the use of common random numbers for function smoothness, and warm-starts for efficiency, during sample-path optimization. We do not go into any further detail about R-SPLINE or RA.

The solution $\boldsymbol{W}_{k,r}$ returned at the end of the kth inner iteration during the rth outer iteration acts as the initial solution to Problem $P(m_{k+1}, \boldsymbol{\epsilon}_{k+1})$. R-SPLINE executes until a specified outer restart simulation budget b_r is expended, or until a local solution is identified with prespecified precision, and $\boldsymbol{Y}_r$ is set equal to the solution $\boldsymbol{W}_{k_r,r}$ obtained upon conclusion of the inner iterations of the local solver during the rth outer iteration. k_r here denotes the random number of inner iterations executed by R-SPLINE during the rth outer iteration of cgR-SPLINE. At the end of each outer iteration r, the newly estimated local solution $\boldsymbol{Y}_r$ is probabilistically compared (Step 13 in Algorithm 1) against the incumbent solution $\boldsymbol{Z}_{r-1}$ to decide whether the incumbent should be updated. The sequence of local solution estimators $\{\boldsymbol{Y}_r\}$ and the incumbent sequence $\{\boldsymbol{Z}_r\}$ thus form the local and global solution estimators returned by cgR-SPLINE.

REMARK 3. The algorithm listing and much of our discussion assumes that the locally minimizing SO algorithm in use is R-SPLINE (Wang et al. 2013). This motivates the name cgR-SPLINE, with the prefix "cg" signifying the extension to the constrained, global context. As stated previously, any other locally minimizing SO algorithm that can be adapted to handle constrained problems, e.g., COMPASS (Xu et al. 2010, Hong and Nelson 2006), can be used instead of R-SPLINE.

The general features of cgR-SPLINE should not come as a surprise — they mimic multistart algorithms (Pardalos and Romeijn 2002) that have been applied with success in the deterministic global optimization context. Some specific features of cgR-SPLINE, however, are noteworthy. First, the constraints are relaxed by the amount $\boldsymbol{\epsilon}_k$ compared to the original Problem P. As we shall see, through the careful choice of the sequence $\{\boldsymbol{\epsilon}_k\}$, we avoid the inconsistency issue described in

Q.1 of Section 1.1. Second, the outer simulation budget b_r (set in Step 2) implicitly determines the exploration-exploitation trade-off within the algorithm. A slow increase in the outer simulation budget leads to more frequent restarts of the local SO algorithm, connoting a stronger focus on exploration; a faster increase, by contrast, connotes a focus on exploitation due to the increased effort devoted to the local search from each restart. Third, the restarts, particularly their locations, should be introduced to ensure that the local SO algorithm in use identifies all local minima asymptotically.

The constraint relaxation parameter sequence $\{\boldsymbol{\epsilon}_k\}$ in Step 7, the outer simulation budget sequence $\{b_r\}$ in Step 2, and the location of restarts $\{\boldsymbol{X}_r\}$ of the local SO algorithm in Step 3 affect cgR-SPLINE at the multiple levels of consistency, efficiency, and finite-time performance, and should hence be chosen carefully. Towards guiding such choice, the results in the ensuing section characterize cgR-SPLINE's asymptotic behavior as a function of these algorithm parameters, thereby identifying "boundaries" on parameter choice that ensure optimal (asymptotic) performance. Section 6 goes further and provides heuristics that, while ensuring confinement within the boundaries prescribed by the theoretical results, identify algorithm parameters to ensure uniformly good finite-time performance.

5. MAIN RESULTS

Recall the sets $\mathcal{F}$ and $\mathcal{F}(m, \boldsymbol{\epsilon})$ defined in Section 4 and which refer to the feasible region of Problem P and its sample-path analogue, respectively. We start with a set of lemmas that describe how the sequence $\{\mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\}$ of sample-path feasible sets generated by the inner iterations of cgR-SPLINE relates to the true feasible set $\mathcal{F}$. Specifically, in the three lemmas that follow we lay down sufficient conditions to ensure that $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$ converges to $\mathcal{F}$ in a certain rigorous sense. Each of Lemmas 1 – 3 provides the same result but under alternative sets of assumptions on the sample sizes $\{m_k\}$, the constraint estimators $\{\hat{\boldsymbol{h}}_{m_k}\}$, and the constraint error tolerances $\{\boldsymbol{\epsilon}_k\}$. Unlike Lemma 3, Lemma 1 and Lemma 2 are of a book-keeping nature and of limited implementation value. Accordingly, we relegate their proofs to the Appendix. We first state a few assumptions, the first four of which are assumed to hold throughout the rest of the paper.

ASSUMPTION 1. $0 < \sigma_* = \inf_{\boldsymbol{x}\in\mathbb{X}}\{\sigma_i(\boldsymbol{x}) : i = 1,2,\ldots,\ell\} \le \sup_{\boldsymbol{x}\in\mathbb{X}}\{\sigma_i(\boldsymbol{x}) : i = 1,2,\ldots,\ell\} = \sigma^* < \infty$, *where* $\sigma_i(\boldsymbol{x})$ *is the standard deviation of the random outcomes* $H_{i,j}(\boldsymbol{x})$, $j = 1, 2, \ldots$.

ASSUMPTION 2. $h_* = \inf\{h_i(\boldsymbol{x}) : h_i(\boldsymbol{x}) > 0, i = 1,\ldots,\ell\} > 0$.

ASSUMPTION 3. *For each* $\boldsymbol{x} \in \mathbb{X}$, *the sequence of random outcomes of the constraint function simulator* $\{\boldsymbol{H}_j(\boldsymbol{x})\}_{j=1}^{\infty}$ *is independent and identically distributed (iid).*

ASSUMPTION 4. *For each constraint* $i \in \{1,\ldots,\ell\}$ *and every* $\boldsymbol{x} \in \mathbb{X}$, *let* $M_{H_i(\boldsymbol{x})}(t)$ *be the moment generating function of* $H_i(\boldsymbol{x}) - h_i(\boldsymbol{x})$. *There exists a constant* $\tilde{h} > 0$ *such that*

(i) $M_{H_i(\boldsymbol{x})}(t)$ *is finite for all* $t \in (-\tilde{h}, \tilde{h})$, *all* $\boldsymbol{x} \in \mathbb{X}$, *and all* $i = 1,2,\ldots,\ell$; *and*

(ii) $m_i^{(2)} := \sup_{t\in(-\tilde{h},\tilde{h}),\boldsymbol{x}\in\mathbb{X}} M^{(2)}_{H_i(\boldsymbol{x})}(t) < \infty$.

Assumption 1 is easily justified if the domain $\mathbb{X}$ is bounded. If $\mathbb{X}$ is unbounded, however, non-pathological cases where Assumption 1 is violated can be constructed. In such cases, we will see that Assumption 1 can be relaxed without diminishing the strength of the results that we present. Assumption 2 is about the behavior of the constraint function at the boundary of the feasible region. Since $\mathbb{X}$ is a subset of the integer lattice, the set $\mathcal{F}^c$ has no boundary points. This means that Assumption 2 precludes functions $h_i(\boldsymbol{x})$ that "flatten out" to zero at infinity. Assumptions 3 and 4 hold widely. Nevertheless, they are made only for convenience of exposition and most of our results that rely on these assumptions can be generalized using less stringent assumptions that preclude extreme dependence across outputs from the simulation.

LEMMA 1. *Suppose that Assumptions 1 and 2 hold. If, for every* $i \in \{1,2,\ldots,\ell\}$, *the sequences* $\{m_k\}$ *and* $\{\epsilon_{i,k}\}$ *satisfy* $\limsup_{k\to\infty} \frac{k^{1+\beta}}{m_k\epsilon_{i,k}^2} = 0$ *for some* $\beta > 0$, *then* $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k) \overset{wp1}{\to} \mathcal{F}$ *uniformly on* $\mathbb{X}$ *as* $k \to \infty$.

LEMMA 2. *Suppose that Assumptions 1 and 2 hold, and that* $\hat{h}_{i,m_k}(\boldsymbol{x}) \sim \mathcal{N}\left(h_i(\boldsymbol{x}), \frac{\sigma_i^2(\boldsymbol{x})}{m_k}\right)$ *for all* $\boldsymbol{x} \in \mathbb{X}$ *and all* $i \in \{1,2,\ldots,\ell\}$. *If the sequences* $\{m_k\}$ *and* $\{\epsilon_{i,k}\}$ *satisfy* $\limsup_{k\to\infty} \frac{\log k}{m_k\epsilon_{i,k}^2} = 0$ *for each* $i \in \{1,2,\ldots,\ell\}$, *then* $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k) \overset{wp1}{\to} \mathcal{F}$ *uniformly on* $\mathbb{X}$ *as* $k \to \infty$.

Lemma 1 notes that the sample-path feasible set converges (see Section 3 for the definition of set convergence) to the true feasible set if the constraint relaxation sequence $\{\boldsymbol{\epsilon}_k\}$ is not too small compared to the reciprocal of the sample-size sequence $\{m_k\}$. Specifically, it notes that the constraints should not be brought in faster than $\sqrt{k/m_k}$. Lemma 2 is a variation on the same result obtained by assuming more about the nature of the constraint estimators. Specifically, by assuming that they are light-tailed, the sufficient conditions on the constraint relaxation parameters are further relaxed.

Lemmas 1 and 2 are useful but it seems that robust implementation will dictate that the constraint relaxations should somehow depend on the estimated standard errors of the constraint estimators. Lemma 3 is one such "implementer's version" of Lemmas 1 and 2 in the sense that it provides a more concrete recommendation on the choice of the constraint relaxations (based on estimated standard errors) to ensure that the sample-path feasible set converges to the true feasible set.

LEMMA 3. *Suppose that Assumptions 1–4 and the following two conditions C.1 and C.2 hold:*

(C.1) For each $\boldsymbol{x} \in \mathbb{X}$ and for each constraint $i \in \{1, \ldots, \ell\}$, the constraint relaxation sequence $\{\epsilon_{i,k}(\boldsymbol{x})\}$ satisfies

$$\epsilon_{i,k}(\boldsymbol{x}) = V_{i,k}(\boldsymbol{x}) m_k^{-\delta},\ k = 1, 2, \ldots,$$

where $\delta \in (0, 1/2)$ and the random variable $V_{i,k}(\boldsymbol{x}) \in [v_l(\boldsymbol{x}), v_u(\boldsymbol{x})]$ wp1 as $k \to \infty$. Moreover, there exist constants $\underline{v}$ and $\overline{v}$ such that $0 < \underline{v} \leqslant \inf_{\boldsymbol{x}\in\mathbb{X}} v_l(\boldsymbol{x}) < \sup_{\boldsymbol{x}\in\mathbb{X}} v_u(\boldsymbol{x}) \leqslant \overline{v} < \infty$.

(C.2) For δ is defined in C.1 and $\beta \in (0, 1-2\delta)$, the sequence of sample sizes $\{m_k\}$ satisfies

$$\limsup_{k\to\infty} \frac{\log k}{m_k^{1-2\delta-\beta}} = 0.$$

Then $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k) \overset{wp1}{\to} \mathcal{F}$ uniformly on $\mathbb{X}$ as $k \to \infty$.

Proof of Lemma 3. Pick $\boldsymbol{x} \in \mathcal{F}$. Then $h_i(\boldsymbol{x}) \leqslant 0$ for all $i = 1, 2, \ldots, \ell$. After noting that $V_{i,k}(\boldsymbol{x}) \geq \underline{v}$ wp1 past a certain k (independent of $\boldsymbol{x}$), we get the following for a large enough k.

$$\Pr\{\boldsymbol{x} \in \mathcal{F}^c(m_k, \boldsymbol{\epsilon}_k)\} = \Pr\left\{\bigcup_{i=1}^{\ell} \left(\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}(\boldsymbol{x})\right)\right\}$$

$$
\begin{aligned}
&\leqslant \sum_{i=1}^{\ell} \Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}(\boldsymbol{x})\right\} \\
&= \sum_{i=1}^{\ell} \left(\int_{-\infty}^{\underline{v}} \Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x}) > s m_k^{-\delta}\right\} dF_{V_{i,k}(\boldsymbol{x})}(v) \right. \\
&\qquad \left. + \int_{\underline{v}}^{\infty} \Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x}) > s m_k^{-\delta}\right\} dF_{V_{i,k}(\boldsymbol{x})}(v) \right) \\
&\leqslant \sum_{i=1}^{\ell} \Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x}) > \underline{v} m_k^{-\delta}\right\} \\
&\leqslant \sum_{i=1}^{\ell} e^{-(\underline{v} m_k^{-\delta} - h_i(\boldsymbol{x}))t} M_{H_i(\boldsymbol{x})}^{m_k}\left(\frac{t}{m_k}\right), \qquad (1)
\end{aligned}
$$

where $M_{H_i(\boldsymbol{x})}$ is the moment generating function of $H_i(\boldsymbol{x}) - h_i(\boldsymbol{x})$ and $t > 0$. Now set $t = m_k^{1+a}$ where $a \in (-(1-\delta), -\delta)$. Then by Assumption 4, $M_{H_i(\boldsymbol{x})}\left(\frac{t}{m_k}\right)$ exists for large enough k for all i. Additionally, $M_{H_i(\boldsymbol{x})}^{m_k}(t/m_k) = \left(1 + t^2 m_k^{-2} M_{H_i(\boldsymbol{x})}^{(2)}(\xi_{i,\boldsymbol{x}}(t))/2\right)^{m_k}$ for some $\xi_{i,\boldsymbol{x}}(t) \in (0, t/m_k)$. Then for large enough k, since $t = m_k^{1+a}$, $h_i(\boldsymbol{x}) \leqslant 0$ and $a > -(1-\delta)$,

$$
\begin{aligned}
\Pr\{\boldsymbol{x} \in \mathcal{F}^c(m_k, \boldsymbol{\epsilon}_k)\} &\leqslant \sum_{i=1}^{\ell} e^{-\underline{v} m_k^{1+a-\delta} + h_i(\boldsymbol{x}) m_k^{1+a}} \left(1 + \frac{m_k^{2a}}{2} M_{H_i(\boldsymbol{x})}^{(2)}(\xi_{i,\boldsymbol{x}}(t))\right)^{m_k} \\
&= \sum_{i=1}^{\ell} O\left(e^{-\underline{v} m_k^{1+a-\delta}}\right) \left(1 + \frac{m_k^{2a}}{2} M_{H_i(\boldsymbol{x})}^{(2)}(\xi_{i,\boldsymbol{x}}(t))\right)^{m_k}.
\end{aligned}
$$

Also, $M_{H_i(\boldsymbol{x})}^{(2)}(\xi_{i,\boldsymbol{x}}(t)) \leqslant m_i^{(2)}$ by Assumption 4, which gives for all $\beta \in (0, 1-2\delta)$,

$$
\begin{aligned}
\Pr\{\boldsymbol{x} \in \mathcal{F}^c(m_k, \boldsymbol{\epsilon}_k)\} &\leqslant \sum_{i=1}^{\ell} O\left(e^{-\underline{v} m_k^{1+a-\delta}}\right) \left(1 + \frac{m_i^{(2)}}{2} m_k^{2a}\right)^{m_k} \\
&= O\left(e^{-\underline{v} m_k^{1+a-\delta}} e^{\frac{m^{(2)}}{2} m_k^{1+2a}}\right), \quad m^{(2)} = \max\{m_i^{(2)} : i = 1, 2, \ldots, \ell\} \\
&= O\left(e^{-\underline{v} m_k^{1-2\delta-\beta}}\right), \qquad (2)
\end{aligned}
$$

where the first equality in (2) above follows after noting that $1 + x \leq e^x$, and the last equality follows since $a < -\delta$. Then under the minimum rate condition on m_k (condition C.2) and from the application of the Borel-Cantelli lemma (Billingsley 1995), there exists K_1(independent of $\boldsymbol{x}$) such that for $k \geqslant K_1$, $\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$ wp1 for any $\boldsymbol{x} \in \mathcal{F}$. In other words, $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k)^c \subseteq \mathcal{F}^c$ wp1 if $k \geqslant K_1$.

Now suppose $\boldsymbol{x} \in \mathcal{F}^c$. Then $h_j(\boldsymbol{x}) > 0$ for some $j \in \{1, \ldots, \ell\}$. Then for large enough k, $\epsilon_{i,k}(\boldsymbol{x}) \leq \overline{v} m_k^{-\delta} < \frac{h_*}{2}$ wp1, where h_* is defined in Assumption 2. So for some $\tilde{c} > 0$ we get

$$
\Pr\{\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\} = \Pr\left\{\bigcap_{i=1}^{\ell} \left(\hat{h}_{i,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}(\boldsymbol{x})\right)\right\} \leqslant \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \frac{h_*}{2}\right\} = O\left(e^{-\tilde{c} m_k}\right), \qquad (3)
$$

where the last equality in (3) follows from Assumption 4 and Cramér's theorem (Dembo and Zeitouni 1998). Once again, under the minimum rate condition on m_k specified in condition C.2 and by invoking the Borel-Cantelli lemma, we conclude that there exists K_2 (independent of $\boldsymbol{x}$) such that for $k \geqslant K_2$, $\boldsymbol{x} \in \mathcal{F}^c(m_k, \boldsymbol{\epsilon}_k)$ wp1 for any $\boldsymbol{x} \in \mathcal{F}^c$. That is, $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k) \subseteq \mathcal{F}$ wp1 for $k \geqslant K_2$, and the result follows. $\square$

The choice $\epsilon_{i,k}(\boldsymbol{x}) = V_{i,k}(\boldsymbol{x}) m_k^{-\delta}, \delta \in (0, 1/2)$ considered in Lemma 3 is of much relevance during implementation because it is inspired by the standard error estimate $\hat{se}(\hat{h}_{i,m_k}(\boldsymbol{x})) = m_k^{-1}\sqrt{\sum_{j=1}^{m_k}(H_{i,j}(\boldsymbol{x}) - \hat{h}_{i,m_k}(\boldsymbol{x}))^2}$ of the constraint estimator $\hat{h}_{i,m_k}(\boldsymbol{x}) = m_k^{-1}\sum_{j=1}^{m_k} H_{i,j}(\boldsymbol{x})$. As can be seen, such a choice for $\epsilon_{i,k}(\boldsymbol{x})$ dictates a rather weak stipulation on the growth rate of the sample sizes to ensure consistency. A projection interval $[v_l(\boldsymbol{x}), v_u(\boldsymbol{x})]$ has been introduced in the expression for $\epsilon_{i,k}(\boldsymbol{x})$ to enhance the decay rate of the error probability, although, depending on the choice of $v_l(\boldsymbol{x})$ and $v_u(\boldsymbol{x})$, the interval may be of little relevance during implementation. Interestingly, setting $[v_l(\boldsymbol{x}), v_u(\boldsymbol{x})] = (-\infty, \infty)$ seems to result in a much slower error decay rate owing to the tail behavior of $V_{i,k}(\boldsymbol{x})$.

Also worthy of note is the stipulation $\delta < 1/2$ which implies the constraints should be "pulled in" slower than the rate at which the standard error of the constraint estimator decays to zero in order to guarantee consistency. To illustrate this point, consider the following counterexample. In each inner iteration k, let $\epsilon_k = 1/\sqrt{m_k}$ (obtained by setting $\delta = 0$). Consider a problem with a single stochastic constraint $h(\boldsymbol{x}) \leq 0$. Also suppose $\sqrt{m_k}\hat{h}_{m_k}(\boldsymbol{x})/\sigma(\boldsymbol{x}) \sim \mathcal{N}(0,1)$ for all $\boldsymbol{x} \in \mathbb{X}$ and at every k. Then for any point $\boldsymbol{x}' \in \mathbb{X}$ with $h(\boldsymbol{x}') = 0$, the probability of incorrectly classifying $\boldsymbol{x}'$ as sample-path infeasible is given by

$$\Pr\{\boldsymbol{x}' \notin \mathcal{F}(m_k, \epsilon_k)\} = \Pr\left\{\hat{h}_{m_k}(\boldsymbol{x}') > \epsilon_k\right\} = \Pr\left\{\frac{\hat{h}_{m_k}(\boldsymbol{x}')}{\sigma(x')/\sqrt{m_k}} > 1/\sigma(x')\right\} = 1 - \Phi(1/\sigma(x')),$$

which is constant for *all* k. A similar analysis can be performed when $\hat{h}_{m_k}(\boldsymbol{x})$ is not Gaussian but satisfies Assumption 4.

Finally, we recognize that the precise form of conditions C.1 and C.2 in Lemma 3 (or for that matter, their counterparts in Lemmas 1 and 2) is a consequence of the algorithm framework:

performing constraint relaxations in an RA setting. We still expect to see error rates similar to the ones in (2) and (3) in the case of other local solvers capable of handling stochastic constraints. Also, uniform convergence of the sample-path feasible sets in Lemmas 1–3 is required to ensure consistency of cgR-SPLINE, as will become evident from the proof of Theorem 2.

5.1. Local Convergence and Rate

Given the above implementer's version of the feasibility result, and assuming that the feasible region is finite, it seems that the sequence of estimated local minima $\{\boldsymbol{Y}_r\}$ identified across the outer iterations of cgR-SPLINE should converge "into" the set $\mathcal{M}$ of true local minima as $b_r \to \infty$. This is because, as the sample size used within an inner iteration diverges, points in $\mathcal{F}$ order themselves correctly even when measured in terms of their sample-path objective functions. This is proved rigorously in the result that follows. We begin with a set of assumptions about the locally minimizing SO algorithm used within cgR-SPLINE.

ASSUMPTION 5. *Let $U_{k,r}$ denote the random number of steps executed during the kth inner iteration of the rth outer iteration of cgR-SPLINE. Then (i) $U_{k,r}$ is uniformly bounded, that is, $u = \limsup_{k,r} U_{k,r} < \infty$ wp1, and (ii) $\hat{g}_{m_k}(\boldsymbol{W}_{k,r}) \leqslant \hat{g}_{m_k}(\boldsymbol{W}_{k-1,r})$ wp1 for any r.*

$U_{j,r}$ represents the number of steps taken by the jth inner iteration during the rth outer iteration. We actually think there is no reason to believe that $U_{j,r}$ will grow with r. As r grows, the budget b_r grows, but this only has the effect of increasing the number of inner iterations of R-SPLINE. Each iteration j within R-SPLINE is stochastically identical. Having made the above observation, we do think that the assumption of uniformly bounded $U_{j,r}$ can be relaxed, although at the expense of non-trivially increasing the complexity of the resulting proofs. Such generalization will involve making assumptions on the nature of $u_k = \sup_\omega U_{k,r}(\omega)$, something that we wish to avoid.

Since $U_{k,r}$ is uniformly bounded for any outer iteration r, cgR-SPLINE returns a solution $\boldsymbol{W}_{k,r}$ during its kth inner iteration in finite time. Consequently, as long as each successive outer iteration r is executed with a finite budget b_r, cgR-SPLINE returns an infinite sequence of local solution estimates $\{\boldsymbol{Y}_r\}$ and global solution estimates $\{\boldsymbol{Z}_r\}$.

THEOREM 1. *Let Assumptions 1-5 and conditions C.1, C.2 listed in Lemma 3 hold. Also, suppose that $\mathcal{F}$ is finite. Then for $b_r \to \infty$, the sequence of local estimators $\{\boldsymbol{Y}_r\}$ converges almost surely to within the set $\mathcal{M}$ of local solutions of Problem P as $r \to \infty$. That is, $Pr\{\boldsymbol{Y}_r \notin \mathcal{M} \ \ i.o.\} = 0$.*

Proof. For ease of exposition, we introduce notation for the random number of inner iterations k_r executed during the rth outer iteration:

$$k_r = \sup\{k : \sum_{j=1}^{k} U_{j,r} m_j \leqslant b_r\}, \tag{4}$$

where $U_{j,r}$ is defined in Assumption 5. Since $U_{j,r}$ is uniformly bounded, $b_r, m_k \in (0,\infty)$, and $b_r \to \infty$, we infer that $0 < k_r < \infty$ and $k_r \to \infty$ wp1.

Recall also that $\boldsymbol{W}_{k,r} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$ is the solution obtained by the local solver at the end of the kth inner iteration, and $\boldsymbol{Y}_r$ is set equal to the solution $\boldsymbol{W}_{k,r}$ obtained upon conclusion of the inner iterations during the rth outer iteration, that is, $\boldsymbol{Y}_r = \boldsymbol{W}_{k_r,r}$. Now, since Assumptions 1–4 and conditions C.1, C.2 hold, we know that Lemma 3 holds and $\Pr\{\mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r}) \neq \mathcal{F} \ \text{ i.o.}\} = 0$ as $r \to \infty$. Furthermore, since $\mathcal{F}$ is bounded, the sequence $\{\boldsymbol{Y}_r\}$ of local solutions returned by cgR-SPLINE remains bounded with probability one.

Let $\Delta = \min\{|g(\boldsymbol{x}) - g(\boldsymbol{y})| : \boldsymbol{x}, \boldsymbol{y} \in \mathcal{F}, g(\boldsymbol{x}) \neq g(\boldsymbol{y})\}$. Then since $\mathcal{F}$ is finite, $\hat{g}_{m_{k_r}}$ converges uniformly to g wp1 on the set $\mathcal{F}$ as $r \to \infty$. Thus, there exists a positive integer K_2 (dependent on Δ) such that $|\hat{g}_{m_{k_r}}(\boldsymbol{x}) - g(\boldsymbol{x})| < \Delta/2$ wp1 if $r \geqslant K_2$ for all $\boldsymbol{x} \in \mathcal{F}$. So if $g(\boldsymbol{y}) < g(\boldsymbol{x})$ (or $g(\boldsymbol{y}) \geqslant g(\boldsymbol{x})$), then $\hat{g}_{m_{k_r}}(\boldsymbol{y}) < \hat{g}_{m_{k_r}}(\boldsymbol{x})$ (or $\hat{g}_{m_{k_r}}(\boldsymbol{y}) \geqslant \hat{g}_{m_{k_r}}(\boldsymbol{x})$) wp1 for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{F}$ if $r \geqslant K_2$. And we conclude that $\Pr\{\boldsymbol{Y}_r \notin \mathcal{M} \ \text{ i.o.}\} = 0$. □

Theorem 1 guarantees that the sequence of local solution estimators returned by cgR-SPLINE falls almost surely within the set of true local solutions to Problem P after a finite number of outer iterations. This is an attractive minimum guarantee for an implementer who does not insist on a global extremum but is instead content with a "good" local extremum.

Theorem 1 was proved for finite feasible regions $\mathcal{F}$. In order to extend Theorem 1 to account for unbounded feasible regions, we impose further structural assumptions on the objective function g to prevent "chase-offs" to infinity. Towards this, let $\mathcal{L}_m(\boldsymbol{x})$ denote the set $\{\boldsymbol{y} \in \mathbb{X} : \hat{g}_m(\boldsymbol{y}) \leqslant \hat{g}_m(\boldsymbol{x}), \boldsymbol{y} \in \mathcal{F}(m, \boldsymbol{\epsilon})\}$ for each $\boldsymbol{x} \in \mathcal{F}$, and let us make the following further assumptions.

ASSUMPTION 6. *Let the sequence of random variables* $\{g(\boldsymbol{x}) - \hat{g}_{m_k}(\boldsymbol{x})\}$ *be governed by a large-deviation principle with rate function* $I_{\boldsymbol{x}} : \mathbb{R} \to \mathbb{R}$ *such that for any* $\varepsilon > 0$, $\inf_{\boldsymbol{x}\in\mathbb{X}} \min(I_{\boldsymbol{x}}(-\varepsilon), I_{\boldsymbol{x}}(\varepsilon)) = \eta_g > 0$.

ASSUMPTION 7. *For each* $\boldsymbol{x} \in \mathbb{X}$, *there exists* $\lambda > 0$ *such that* $\mathcal{L}(\boldsymbol{x}, \lambda) = \{\boldsymbol{y} \in \mathbb{X} : g(\boldsymbol{y}) \leqslant g(\boldsymbol{x}) + \lambda\}$ *is finite.*

The analogues to Lemma 3 and Theorem 1 for unbounded feasible regions are consolidated into the following single result, with a proof provided in the Appendix.

THEOREM 2. *Suppose that Assumptions 1 – 7 and conditions C.1, C.2 hold. Then for* $b_r \to \infty$, *cgR-SPLINE returns a sequence of sample path solutions* $\{Y_r\}$ *that converges wp1 to within set* $\mathcal{M}$ *of local minima of Problem P as* $r \to \infty$. *That is,* $Pr\{\boldsymbol{Y}_r \notin \mathcal{M} \ \ i.o.\} = 0$.

Theorems 1 and 2 prove the almost sure convergence of cgR-SPLINE's iterates $\{\boldsymbol{Y}_r\}$ to a true local minimum. How fast does such convergence happen? In other words, can anything be said about the rate at which the probability of cgR-SPLINE returning an infeasible solution decays to zero, and what is the corresponding rate for returning a truly feasible solution that is suboptimal? The following two results assert that these rates are sub-exponential, and dependent on the rate at which the constraints are "pulled in."

First, recall the definition of k_r in (4) — it denotes the random number of inner iterations executed during the rth outer iteration. Since $U_{k,r}$ is uniformly bounded with $u := \limsup_{k,r} U_{k,r} < \infty$, we may define the fixed quantity $\underline{k}_r$ as the *smallest* number of inner iterations executed in the rth outer iteration:

$$\underline{k}_r = \sup\{k : \sum_{j=1}^{k} um_j \leqslant b_r\}. \tag{5}$$

Since $b_r \to \infty$ and $m_k > 0$, we infer that $\underline{k}_r \to \infty$.

THEOREM 3. *Let* $\mathbb{X}$ *be finite and suppose that Assumptions 1 – 6 and conditions C.1, C.2 hold. Then for* δ *defined in condition C.1, the following hold for some* $c' > 0$ *and as* $r \to \infty$.

(i) The probability that cgR-SPLINE returns an infeasible solution $Pr\{\boldsymbol{Y}_r \notin \mathcal{F}\} = O\left(e^{-c' m_{\underline{k}_r}^{1-2\delta-\beta}}\right)$*, where* $\beta > 0$ *is arbitrarily close to zero.*

(ii) The probability that cgR-SPLINE returns a locally suboptimal but feasible solution $Pr\{\boldsymbol{Y}_r \notin \mathcal{M} | \boldsymbol{Y}_r \in \mathcal{F}\} = O\left(e^{-c' m_{\underline{k}_r}^{1-2\delta-\beta}}\right)$*, where* $\beta > 0$ *is arbitrarily close to zero.*

Proof of Theorem 3(i). For large enough r

$$\begin{aligned}
\Pr\{\boldsymbol{Y}_r \notin \mathcal{F}\} &= \Pr\{\boldsymbol{Y}_r \notin \mathcal{F}, \mathcal{F} \nsubseteq \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} + \Pr\{\boldsymbol{Y}_r \notin \mathcal{F}, \mathcal{F} \subseteq \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} \\
&= \Pr\{\boldsymbol{Y}_r \notin \mathcal{F}, \mathcal{F} \nsubseteq \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} + \Pr\{\boldsymbol{Y}_r \notin \mathcal{F}, \mathcal{F} \subset \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} \\
&\leqslant \sum_{\boldsymbol{y} \in \mathcal{F}} \Pr\{\boldsymbol{y} \in \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})^c\} + \sum_{\boldsymbol{y} \in \mathcal{F}^c} \Pr\{\boldsymbol{y} \in \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} \\
&= O\left(e^{-c' m_{\underline{k}_r}^{1-2\delta-\beta}}\right) + O\left(e^{-c'' m_{\underline{k}_r}}\right) \qquad (6)
\end{aligned}$$

where the last equality follows due to the finiteness of $\mathbb{X}$ and from (2) and (3).

Proof of Theorem 3(ii). Once again for large enough r,

$$\begin{aligned}
\Pr\{\boldsymbol{Y}_r \notin \mathcal{M} \,|\, \boldsymbol{Y}_r \in \mathcal{F}\} = \Pr\left\{\boldsymbol{Y}_r \notin \mathcal{M}, \left(\underset{\boldsymbol{y} \in N(\boldsymbol{Y}_r) \bigcap \mathcal{F}}{\arg\min}\, g(\boldsymbol{y})\right) \in \mathcal{F}^c(m_{k_r}, \boldsymbol{\epsilon}_{k_r}) \,|\, \boldsymbol{Y}_r \in \mathcal{F}\right\} + \\
\Pr\left\{\boldsymbol{Y}_r \notin \mathcal{M}, \left(\underset{\boldsymbol{y} \in N(\boldsymbol{Y}_r) \bigcap \mathcal{F}}{\arg\min}\, g(\boldsymbol{y})\right) \in \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r}) \,|\, \boldsymbol{Y}_r \in \mathcal{F}\right\} \\
= O\left(e^{-c' m_{\underline{k}_r}^{1-2\delta-\beta}}\right) + O\left(e^{-\eta_g m_{\underline{k}_r}}\right), \qquad (7)
\end{aligned}$$

where the last equality follows from the finiteness of $\mathbb{X}$, equation (2), and Assumption 6. □

Theorem 3 is important in that it asserts that the rate at which a suboptimal solution is returned in the current context is *not exponential*, unlike what has been shown by Kleywegt et al. (2001) in the discrete unconstrained context. The difference is that, in the discrete unconstrained context, the error associated with suboptimality relates only to correct ordering which typically exhibits light-tailed decay. The error in the current context, however, is dominated by the error due to incorrect assessment of feasibility in the presence of stochastic constraints, resulting in a deteriorated convergence rate compared to pure ordering. While such deterioration is a result of our algorithm design, we believe that some sort of deterioration in rate is inevitable as a price of having to estimate the feasibility of a solution where contraints may be binding. We also note that

the extent of the deterioration of this error rate from exponential is a function of the algorithm parameter δ and can thus be made negligibly small by choosing a very small positive value for δ. In fact, if h_* (defined in Assumption 2) is somehow known, an exponential rate of decay is easily achieved by always choosing $\epsilon_{i,k}(\boldsymbol{x}) < h_*$ in Step 7 in Figure 1. It is important to note that the error due to infeasibility vanishes in the limit precisely due to the assumed problem domain (integer-ordered) and the constraint structure endowed by Assumption 2; for instance, such an h_* satisfying Assumption 2 does not exist in the continuous context.

Theorem 3 suggests that the convergence rates associated with cgR-SPLINE increase with decreasing δ. From an implementation standpoint, however, the implied directive — "pull the constraints in as slowly as possible," that is, set δ as close to zero as possible — is only of limited use. In Section 6, we discuss robust implementation of cgR-SPLINE that follows directives on choosing an optimal δ based on an analysis of trade-off between the loss resulting from incorrectly assessing feasible and infeasible points.

5.2. Global Convergence and Rate

Recall the broad structure of cgR-SPLINE: during the rth iteration, a randomly restarted locally minimizing SO algorithm executes until a budget b_r is expended and returns an estimator $\boldsymbol{Y}_r$ of a local extremum to Problem P. The local solution estimator $\boldsymbol{Y}_r$ is then probabilistically compared against the incumbent $\boldsymbol{Z}_{r-1}$ to produce the updated incumbent $\boldsymbol{Z}_r$. Section 5.1 was about the behavior of the sequence $\{\boldsymbol{Y}_r\}$; in this section, we study the behavior of the incumbents $\{\boldsymbol{Z}_r\}$ which estimate the global minimum to Problem P.

For ease of exposition of what follows, let $\boldsymbol{Y}_\infty$ denote the point that would be attained by the locally minimizing SO algorithm in use within cgR-SPLINE if any specific outer iteration is executed with an infinite budget. (We ignore the measurability concerns of $\boldsymbol{Y}_\infty$ and assume that $\boldsymbol{Y}_\infty$ is well defined.) Of course, $\boldsymbol{Y}_\infty$ is not observed since the budget b_r for any specific iteration r is finite. For each $\boldsymbol{y}^* \in \mathcal{M}$ we also define the "attractor set" $B(\boldsymbol{y}^*)$ and its "reaching probability" $p_r(\boldsymbol{y}^*)$ as $B(\boldsymbol{y}^*) = \{\boldsymbol{x} \in \mathcal{F} : \Pr\{\boldsymbol{Y}_\infty = \boldsymbol{y}^* \mid \boldsymbol{X}_r = \boldsymbol{x}\} > 0\}$ and $p_r(\boldsymbol{y}^*) = \Pr\{\boldsymbol{X}_r \in B(\boldsymbol{y}^*)\}$, respectively. The

fixed set $B(\boldsymbol{y}^*)$ has the interpretation of *the set of points* from which the locally minimizing SO algorithm should be started in order that it attain the local extremum $\boldsymbol{y}^*$ if executed with an infinite budget; the (fixed) quantity $p_r(\boldsymbol{y}^*)$ represents the *probability* of starting the locally minimizing SO algorithm from the attractor region of $\boldsymbol{y}^*$ and is hence related to the probability of the local SO algorithm attaining $\boldsymbol{y}^*$ in the limit. The set $B(\boldsymbol{y}^*)$ is independent of r because the locally minimizing SO algorithm within cgR-SPLINE is assumed to not change across outer iterations r. The probability $p_r(\boldsymbol{y}^*)$, on the other hand, depends on r to the extent that the restart guesses $\boldsymbol{X}_r$ may not be identically distributed across iterations. Since the initial guesses $\boldsymbol{X}_r$ are, however, chosen independently, and possibly ahead of time, the outer iterations are assumed to be executed independently.

The following result characterizes the rate at which the probability of cgR-SPLINE returning an incumbent not lying in the set of true global minima decays to zero. Since this probability is shown to decay exponentially in the outer iteration number r, the first Borel-Cantelli lemma (Billingsley 1995) ensures that the returned incumbent solutions reach the global extremum almost surely. We need the following assumption on the local solver within cgR-SPLINE and the initial guesses $\{\boldsymbol{X}_r\}$.

ASSUMPTION 8. *Let* $\mathcal{G} = \{\boldsymbol{z}^* \in \mathcal{F} : g(\boldsymbol{z}^*) \leqslant g(\boldsymbol{x})$ *for all* $\boldsymbol{x} \in \mathcal{F}\}$ *denote the set of all global solutions to Problem* P. *Then* $\bigcup_{z^* \in \mathcal{G}} B(z^*) \neq \emptyset$ *and* $\liminf_{r\to\infty} p_r(\mathcal{G}) > 0$, *where* $p_r(\mathcal{G}) := Pr\{\boldsymbol{X}_r \in \bigcup_{\boldsymbol{z}^* \in \mathcal{G}} B(\boldsymbol{z}^*)\}$.

The condition $\liminf_{r\to\infty} \Pr\{\boldsymbol{X}_r \in \bigcup_{\boldsymbol{z}^* \in \mathcal{G}} B(\boldsymbol{z}^*)\} > 0$ ensures that a restart originates in the attraction region of a global solution infinitely often. One easy way to satisfy this condition is to sample restart locations uniformly on $\mathbb{X}$. The set $\bigcup_{\boldsymbol{z}^* \in \mathcal{G}} B(\boldsymbol{z}^*)$ in Assumption 8 connotes the reaching set for the global extrema of Problem P — when executed from any point in $\bigcup_{z^* \in \mathcal{G}} B(z^*)$, the locally minimizing SO algorithm has a positive probability of reaching a global extremum (in the limit). Good locally minimizing algorithms thus tend to have larger sets $\bigcup_{z^* \in \mathcal{G}} B(z^*)$ and higher values of $\Pr\{\boldsymbol{Y}_\infty \in \mathcal{G} \,|\, \boldsymbol{X}_r \in \bigcup_{z^* \in \mathcal{G}} B(z^*)\}$.

THEOREM 4. *Suppose that* $\mathbb{X}$ *is finite. Furthermore, suppose that Assumptions 1 – 6, Assumption 8, and conditions C.1, C.2 hold. Then as* $r \to \infty$*, for* δ *defined in C.1 and* $\beta > 0$ *arbitrarily close to zero,*

(i) $Pr\{\boldsymbol{Z}_r \notin \mathcal{G}\} = O\left(e^{-c' b_r^{\gamma}}\right) + O\left(\tau^r\right)$*, where* $c' > 0, \tau \in (0,1)$*, and*

$$\gamma := \begin{cases} \left(\frac{q}{q+1}\right)(1-2\delta-\beta) & \text{if } m_k = \Theta\left(k^q\right), \text{ where } q > 0 \text{ and} \\ 1-2\delta-\beta & \text{if } m_k = \Theta\left(a_{in}^k\right), \text{ where } a_{in} > 1; \end{cases} \tag{8}$$

if the sequence of outer iteration budgets $\{b_r\}$ *approaches infinity;*

(ii) $Pr\{\boldsymbol{Z}_r \notin \mathcal{G} \text{ i.o..}\} = 0$ *for* γ *defined in (8) and*

$$b_r = o^{-1}\left(\log^{\frac{1}{\gamma}} r\right). \tag{C.3}$$

Theorem 4 provides important insight about the convergence characteristics of cgR-SPLINE. Specifically, the result in part (i) of Theorem 4 implies that the error probability $\Pr\{\boldsymbol{Z}_r \notin \mathcal{G}\}$ of finding the global solution decomposes into two parts. The first part $O\left(e^{-c' b_r^{\gamma}}\right)$ represents the sampling error due to the locally minimizing algorithm in use, and the second part $O\left(\tau^r\right)$ represents the probability of the locally minimizing algorithm operating in the wrong attraction region, that is, an attraction region that does not contain the global minimum. (If a locally minimizing algorithm other than R-SPLINE is used within cgR-SPLINE, we conjecture that $\Pr\{\boldsymbol{Z}_r \notin \mathcal{G}\}$ will still decompose into the same two parts except that the form of the constant γ will differ.) Also, we note that the second part $O\left(\tau^r\right)$ parallels a similar term that is obtained in deterministic multistart methods. Part (ii) of Theorem 4 notes that if the outer budget b_r is increased fast enough, then deviations due to mischance become negligible and $\boldsymbol{Z}_r$ converges to the solution almost surely.

Under the assumption that $\mathbb{X}$ is finite, Theorem 4 leads to the analogous Theorem 5 which measures error $E_r := \max_{\boldsymbol{z}^* \in \mathcal{G}} \mathrm{E}\left[\|g\left(\boldsymbol{Z}_r\right) - g\left(\boldsymbol{z}^*\right)\|\right]$ in the function space. Since the proof of Theorem 5 follows in a straightforward way from the proof of Theorem 4, we do not provide one.

THEOREM 5. *Suppose that* $\mathbb{X}$ *is finite. Furthermore, suppose that Assumptions 1 – 6, Assumption 8, and conditions C.1, C.2 hold. Then as* $r \to \infty$*,*

(i) $E_r = O\left(e^{-c' b_r^{\gamma}}\right) + O\left(\tau^r\right)$, *where* $\tau \in (0,1), \gamma$ *defined as in (8), and* $b_r \to \infty$*; and*

(ii) $E_r \to 0$ *wp1 if condition (C.3) also holds.*

Theorems 4 and 5 form the basis for deducing cgR-SPLINE's convergence rate under different choices of the sequence $\{b_r\}$. Together with the expression for γ derived in Theorem 4, Theorem 6 gives broad insight into the convergence rates that are achievable by cgR-SPLINE.

THEOREM 6. *Suppose that* $\mathbb{X}$ *is finite. Furthermore, suppose that Assumptions 1 – 6, Assumption 8, and conditions C.1, C.2 hold. If* $w_r := \sum_{j=1}^{r} b_j$ *denotes the total simulation effort expended by end of the* r*th outer iteration of cgR-SPLINE and* γ *is defined as in (8), then as* $r \to \infty$*,*

$$-\log E_r = \begin{cases} O(\log w_r) & \text{if } b_j = \Theta(a_{out}^j), \text{ where } a_{out} > 1; \\ O(w_r^{1/(1+q_{out})}) & \text{if } b_j = \Theta(j^{q_{out}}), \text{ where } \gamma q_{out} \geq 1; \\ O(w_r^{\gamma q_{out}/(1+q_{out})}) & \text{if } b_j = \Theta(j^{q_{out}}), \text{ where } \gamma q_{out} < 1. \end{cases} \cdot$$

Proof. When the outer budget $\{b_j\}$ grows as $b_j = \Theta(a_{\text{out}}^j)$, we get from Theorem 5 that $E_r = O\left(e^{-c' b_r^{\gamma}}\right) + O\left(\tau^r\right) = O\left(\tau^r\right)$. Also, since $w_r = \sum_{j=1}^{r} b_j = \Theta(e^{r \log a_{\text{out}}})$, we conclude that $E_r = O(e^{-\kappa_1 \log w_r})$ where $\kappa_1 = \log \tau / \log a_{\text{out}}$.

Similarly, when the outer budget $\{b_j\}$ grows as $b_j = \Theta(j^{q_{\text{out}}})$, we get from Theorem 5 that $E_r = O\left(e^{-c' r^{\gamma q_{\text{out}}}}\right)$ if $\gamma q_{\text{out}} < 1$ and $E_r = O\left(\tau^r\right)$ if $\gamma q_{\text{out}} > 1$. Also, since $w_r = \sum_{j=1}^{r} b_j = \Theta(r^{1+q_{\text{out}}})$, we conclude that $-\log E_r = O(w_r^{\gamma q_{\text{out}}/(1+q_{\text{out}})})$ if $\gamma q_{\text{out}} < 1$ and $-\log E_r = O(w_r^{1/(1+q_{\text{out}})})$ if $\gamma q_{\text{out}} \geq 1$.

□

Theorem 6 implies that cgR-SPLINE's convergence rate is dependent on three factors: the value of $\delta \in (0, 1/2)$ used in the constraint relaxation parameter sequence given in (C.1), the rate at which the inner sample sizes $\{m_k\}$ are increased, and the rate at which the outer budgets $\{b_r\}$ are increased. From the expression for γ in (8) and the assertion in Theorem 6, it is clear that the inner sample sizes $\{m_k\}$ should be increased exponentially (e.g., by a fixed percentage during each iteration) and the outer budgets $\{b_r\}$ as $\Theta(r^{q_{\text{out}}})$. Particularly, cgR-SPLINE can be made to

achieve a rate that is arbitrarily close to the canonical rate $O(e^{-\kappa\sqrt{w_r}})$ by choosing $q_{\text{out}} = 1$ and δ close to zero.

An equivalent "fixed budget" interpretation of Theorem 6 is that for fastest possible convergence, the number of restarts and the budget per restart should each bear a roughly square-root relationship with the total simulation budget. This last insight is remarkable in that it is a prescription for greater exploration than in the continuous context where optimal convergence seems to stipulate that the number of restarts be logarithmic in the total simulation budget. Intuitively, the faster exponential convergence of the local SO algorithm in the integer-ordered context (compared to the $O(1/\sqrt{\text{work}})$ convergence in the continuous setting) affords more exploration. When using a locally minimizing algorithm other than R-SPLINE, we expect our insights from an analogous Theorem 6 to be similar since we expect the structure of Theorems 4 and 5 to remain unchanged.

6. IMPLEMENTATION HEURISTICS

In this section we provide some directives that have proven useful for cgR-SPLINE's robust implementation. The directives we discuss here do not affect cgR-SPLINE's asymptotic performance and to this extent did not appear as part of the asymptotic theory presented in Section 5. The motivation for these directives is that cgR-SPLINE's asymptotic theory says only "part of the story" in that the stipulations imposed by optimal asymptotic performance still leave a lot of room for algorithmic decision-making. We emphasize that, while the directives we propose have a theoretical foundation, they are still heuristics in the sense that the evidence for their effect on cgR-SPLINE's improved finite-time functioning is only empirical.

6.1. Choosing the Constraint Relaxation Sequence

For each restart r, recall that the constraint relaxations $\boldsymbol{\epsilon}_k(\boldsymbol{x})$ in cgR-SPLINE are chosen as $\boldsymbol{\epsilon}_k(\boldsymbol{x}) = \hat{\boldsymbol{\sigma}}_k(\boldsymbol{x})/m_k^{\delta}$, where $\hat{\boldsymbol{\sigma}}_k(\boldsymbol{x}) := (\hat{\sigma}_{1,k}(\boldsymbol{x}), \hat{\sigma}_{2,k}(\boldsymbol{x}), \ldots, \hat{\sigma}_{\ell,k}(\boldsymbol{x}))$ is the point estimator of $(\sigma_1, \sigma_2, \ldots, \sigma_\ell)$. This choice for $\boldsymbol{\epsilon}_k(\boldsymbol{x})$ makes sense in that it incorporates the observable quantity $\hat{\boldsymbol{\sigma}}_k(\boldsymbol{x})$ which is a measure of the uncertainty in the constraint function value. The constant δ is introduced to ensure

that the constraints are pulled in slower than the rate at which the standard error of the constraint estimate at a point drops to zero. cgR-SPLINE's consistency dictates $\delta \in (0, 1/2)$, and Theorem 3 indicates that it is most efficient to set δ as close to zero as possible.

While the recommendations proposed by theory are useful, they present complications during implementation. That δ should be as close to zero as possible (while remaining above it) follows from Theorem 3 which recognizes that the convergence rate is dictated by the probability of incorrectly deeming solutions (with binding constraints) as being infeasible; that is, such probabilities asymptotically dominate the probability of an infeasible point incorrectly being deemed feasible. From a practical standpoint, however, the probability of an infeasible point being deemed feasible is a distinct concern, especially when the sample sizes in effect are small. As a way around this dilemma, we present a directive which trades-off infeasibility and suboptimality through a Bayesian cost minimization.

For ease of exposition of the cost minimization, we pose the problem for determining δ in the following slightly more general framework. Suppose we wish to decide if an unknown (but fixed) parameter $\mu = \mathrm{E}[X] \in \mathbb{R}$ is *feasible* ($\mu \leq 0$) or *infeasible* ($\mu > 0$) after observing m iid copies of the random variable X that is distributed as $\mathcal{N}(\mu, \sigma^2)$. We impose a subjective probability distribution $\mathcal{N}(\overline{x}, \sigma^2/n)$ on μ, where $\overline{x}$ and σ^2 are known. Say we deem μ as being feasible if $\bar{X} = m^{-1}\sum_{i=1}^{m} X_i \leq \epsilon$ and infeasible otherwise, making ϵ the decision variable of the optimization problem. Suppose also that the cost of incorrectly deeming μ as being infeasible and feasible are c_1 and c_2 respectively. Then the optimization problem in ϵ is posed as $\min_\epsilon \mathrm{E}_{\mu,X}[\mathrm{L}(\mu, f(\bar{X}))]$ where $f(\bar{X}) = 1$ if $\bar{X} \leq \epsilon$ and 0 otherwise, and $\mathrm{E}[\mathrm{L}(\mu, f(\bar{X}))] = c_1 \int_{-\infty}^{0}\int_{\epsilon}^{\infty} \mathcal{N}_{\overline{x},\hat{\sigma}^2/n}(\mu)\mathcal{N}_{\mu,\hat{\sigma}^2/m}(y)\,dy\,d\mu + c_2 \int_{0}^{\infty}\int_{-\infty}^{\epsilon} \mathcal{N}_{\overline{x},\hat{\sigma}^2/n}(\mu)\mathcal{N}_{\mu,\hat{\sigma}^2/m}(y)\,dy\,d\mu$. Differentiating $\mathrm{E}[\mathrm{L}(\mu, f(\bar{X}))]$ with respect to ϵ and equating to zero yields an equation for ϵ. If we choose $c_1 = c_2$ and $m = n$, for instance, we get $\epsilon = -\bar{X}$.

For the context of cgR-SPLINE, the above analysis suggests setting $\epsilon^*_{i,k}(\boldsymbol{x}) = -\hat{h}_{i,m_k}(\boldsymbol{x})$, where $\boldsymbol{\epsilon}^*_k(\boldsymbol{x}) = (\epsilon^*_{1,k}(\boldsymbol{x}), \epsilon^*_{2,k}(\boldsymbol{x}), \ldots, \epsilon^*_{\ell,k}(\boldsymbol{x}))$. We modify this slightly to avoid big fluctuations of the constraint relaxations, and recommend setting

$$\epsilon^*_{i,k}(\boldsymbol{x}) = \min\left(\max\left(\frac{\hat{\sigma}_{i,k}(\boldsymbol{x})}{m_k^{0.45}}, -\hat{h}_{i,m_k}(\boldsymbol{x})\right), \frac{\hat{\sigma}_{i,k}(\boldsymbol{x})}{m_k^{0.1}}\right), \quad \text{for all } i = 1, 2, \ldots, c. \tag{9}$$

This yields $\delta^*_{i,k}(\boldsymbol{x}) = \log(\hat{\sigma}_{i,k}(\boldsymbol{x})/\epsilon^*_{i,k}(\boldsymbol{x}))/\log(m_k)$. Instead of calculating δ at every point $\boldsymbol{x} \in \mathbb{X}$ visited by cgR-SPLINE, we recommend updating δ once at the end of every inner iteration after observing the sample-path local solution $\boldsymbol{W}_{k,r}$. In other words, set $\delta_{i,k+1}(\boldsymbol{x}) = \delta^*_{i,k}(\boldsymbol{W}_{k,r})$, for all $\boldsymbol{x} \in \mathbb{X}, i = 1, 2, \ldots, \ell$.

6.2. Solution Reporting

Implementers often find it desirable to have a probabilistic guarantee on solutions reported by an algorithm. Accordingly, we suggest imposing a "soft" lower bound α_r constraint on the probability of feasibility of the local solutions $\{\boldsymbol{Y}_r\}$ returned after outer iterations. For a given local solution $\boldsymbol{Y}_r = \boldsymbol{\epsilon}_r$ returned at the end of the rth outer iteration, the probability $\psi(\boldsymbol{\epsilon}_r)$ that $\boldsymbol{\epsilon}_r$ is truly feasible can be estimated as $\psi(\boldsymbol{\epsilon}_r) = \Pr\{\boldsymbol{\epsilon}_r \in \mathcal{F} \,|\, \boldsymbol{Y}_r = \boldsymbol{\epsilon}_r\} = \Pr\left\{\bigcap_{i=1}^{\ell} \left(\hat{h}_{i,m_{k_r}}(\boldsymbol{\epsilon}_r) \leqslant \epsilon_{i,k_r}(\boldsymbol{\epsilon}_r)\right)\right\} \approx \prod_{i=1}^{\ell} \Phi\left(m_{k_r}^{\frac{1}{2}-\delta} - \frac{\hat{h}_{i,m_{k_r}}(\boldsymbol{\epsilon}_r)}{\hat{\sigma}_{i,k_r}(\boldsymbol{\epsilon}_r)}\sqrt{m_{k_r}}\right)$. $\boldsymbol{Y}_r = \boldsymbol{\epsilon}_r$ is returned as the identified local solution if $\psi(\boldsymbol{\epsilon}_r) \geq \alpha_r$ or none of $\boldsymbol{Y}_r$'s neighbors $\boldsymbol{\epsilon}$ satisfy $\psi(\boldsymbol{\epsilon}) \geq \alpha_r$. Otherwise, one of the neighbors of $\boldsymbol{Y}_r$ which satisfies the specified lower bound is returned instead of $\boldsymbol{Y}_r$.

By further requiring $\alpha_r \uparrow \alpha$, where $\alpha \in [0, 1)$, it is easy to see that $\boldsymbol{Y}_r$ will satisfy the lower bound constraint as $r \to \infty$ wp1, implying that the above heuristic does not affect the asymptotics of cgR-SPLINE. To this extent, the proposed guideline on local solution reporting is only to provide a "reasonable solution" during the early iterations.

The constant α and the sequence $\{\alpha_r\}$ are chosen according to implementer convenience. One suggestion is to use $\alpha_r = \alpha(1 - \alpha_0^r)$, where $\alpha_0 \in (0, 1)$. When α is close to one and α_0 is close to zero, cgR-SPLINE searches in the interior of the search space where points have a high likelihood of being deemed feasible. On the other hand, setting $\alpha = 0$ "switches off" the feasibility heuristic within cgR-SPLINE. Boundary solutions are more likely to be discovered under this setting. Numerical experiments detailing the effects of the feasibility heuristic on cgR-SPLINE's performance can be found in the e-companion.

A similar guarantee on solution optimality (for example, statistical bounds on the optimality gap of the reported solution) would likewise benefit the user. Assessment of solution optimality in

simulation optimization being a well researched topic, we refer the reader to Shapiro et al. (2009) for a comprehensive treatment of solution quality assessment in the SAA context and to Xu et al. (2010) and Boesel et al. (2003) for suggested post–run clean–up procedures in the discrete SO context.

Finally, we suggest a heuristic for premature termination of restarts, whose details we list (along with numerical results) in the e-companion. The suggested procedure is loosely based on local convergence criteria in deterministic optimization algorithms. We expect this idea to be effective when there are only a few local minima that lie in the interior of the feasible region. Unlike implementation ideas outlined in Sections 6.1 and 6.2, premature termination affects convergence guarantees under certain pathological conditions that cause the incumbent sample size t_r (in Step 12 of Algorithm 1) to remain bounded.

7. NUMERICAL EXPERIMENTS

In this section, we illustrate cgR-SPLINE's performance on two nontrivial examples, a three-stage flowline problem (adapted from Wang et al. 2013) and an (s, S) inventory problem (adapted from Park and Kim 2015), each formulated to result in solutions with binding constraints. For each problem, we evaluate the performance of cgR-SPLINE by observing estimates of three measures: (i) the expected optimality gap expressed as a fraction of the optimal objective value, $\mathrm{E}[\|g(\mathbf{Z}_r) - g(\mathbf{z}^*)\|]/g(\mathbf{z}^*)$, (ii) the probability of the incumbent solution being truly feasible, $\Pr\{\mathbf{Z}_r \in \mathcal{F}\}$, and (iii) the probability of the incumbent solution being globally optimal, $\Pr\{\mathbf{Z}_r \in \mathcal{G}\}$, as a function of the expended simulation budget. All measures are calculated simultaneously using independent runs of cgR-SPLINE. Finally, we test the recently developed SCORE algorithm (Pasupathy et al. 2014) for stochastically constrained R&S as a competitor to cgR-SPLINE.

To ensure the fastest possible rate of convergence of cgR-SPLINE, the analysis in Section 5.2 suggests increasing the inner sample sizes m_k geometrically in the inner iteration number k when the restart budget b_r is increased linearly in the number of restarts r. Consequently, for the experiments presented in this section, we set $m_{k+1} = 1.1m_k$ and $b_r = 500 \times r^{1.1}$. The constraint relaxation

parameter δ is chosen according to suggestions laid out in Section 6.1, and the feasibility heuristic described in Section 6.2 is deactivated (by setting $\alpha_r = 0$ for all r) to illustrate cgR-SPLINE's behavior when solutions have binding constraints. Results from additional experiments that test cgR-SPLINE's performance on different choices of $\{b_r\}$ and $\{\alpha_r\}$, and an early termination heuristic are listed in Section EC.3 of the e-companion. MATLAB code for cgR-SPLINE can be downloaded from `http://iem.okstate.edu/nagaraj`.

7.1. A Three-Stage Flowline Problem

Consider a stochastically constrained version of the three-stage flowline problem (Xu et al. 2010, Wang et al. 2013) consisting of three serial servers with exponential service rates $x_1, x_2, x_3 \in \{8, 9, \ldots, 20\}$. The second and third server have finite buffer capacities x_4 and x_5, respectively, with a total buffer space $x_4 + x_5 = 20$. We assume there are an infinite number of jobs in front of the first server. The objective is to identify service rates x_1, x_2, and x_3 and buffer capacities x_4 and x_5 that minimize the total service rate $g(\boldsymbol{x})$, subject to the steady-state expected throughput $h(\boldsymbol{x}) \geq 8.8584$ units.

Our choice of the throughput constraint is not arbitrary; it is chosen to make the problem difficult by placing the global minima on the boundary of the feasible region. This problem turns out to be nontrivial, having 27,181 feasible solutions of which 205 are local minima. Of the local minima, the point $\boldsymbol{z}^* = (10, 10, 10, 10, 10)$ located on the boundary formed by the throughput constraint represents the global solution. At any point $\boldsymbol{x}$ in the search space, the corresponding throughput $h(\boldsymbol{x})$ can only be estimated through sampling as the average number of jobs leaving the system between times $t = 50$ and $t = 1000$. Initial solutions to the restarts are generated by sampling uniformly from the region $\mathbb{X} = \{\boldsymbol{x} \in \mathbb{Z}_+^5 : x_4 + x_5 = 20; 8 \leqslant x_i \leqslant 20,\ i = 1, 2, 3; 1 \leqslant x_i \leqslant 19,\ i = 4, 5\}$.

Table 1 displays the output log from a *single run* of cgR-SPLINE. The first six columns in Table 1 are displayed to the user. The last three columns — displaying the true objective and constraint functions, and whether or not the identified solution is truly local, global, or infeasible — are typically not available to the user. As is evident from Table 1, cgR-SPLINE returns a local

solution at the end of first outer iteration after expending a simulation budget of 504. Then, over the next several outer iterations, it gets stuck at an infeasible solution, followed by a two local minima which are not global. Upon conclusion of the fifty third outer iteration and after expending over a million oracle calls, cgR-SPLINE seems to identify the global minimum. That cgR-SPLINE visits one of the 205 local minima is interesting and probably explained by the logic of the SPLINE solver; even though SPLINE is a local sample-path solver, its search mechanism takes it to regions having good local minima.

Table 1 The first six columns display the output log from a single run of cgR-SPLINE. The last three columns display the true objective function and constraint function values as well as whether the solutions are infeasible (I), locally optimal (L), or globally optimal (G). Notice how cgR-SPLINE first identifies a local solution, then gets caught at an infeasible solution, followed by a local minimum that is not global, and then successfully identifies the global minimum after fifty three outer iterations. Although this implementation of cgR-SPLINE does not employ the feasibility heuristic (α_r is set to zero for all r), the code still reports an estimated probability of feasibility $\hat{\psi}_r$ for each incumbent solution $\boldsymbol{Z}_r$.

Restart	Incumbent solution				Total work			Feasible/
r	$\boldsymbol{Z}_r$	$\hat{g}_{m_{k_r}}$	$\hat{h}_{m_{k_r}}$	$\hat{\psi}_r$	w_r	g	h	Optimal
1	(11, 9, 16, 14, 6)	36	11.0146	0.65	504	36	10.9930	L
2	(11, 9, 14, 11, 9)	34	8.8431	0.60	1630	34	8.8210	I
3	(9, 12, 11, 10, 10)	32	8.8497	0.74	3604	32	8.8512	I
⋮								
11	(9, 12, 11, 10, 10)	32	8.8449	0.52	44046	32	8.8512	I
12	(10, 10, 12, 16, 4)	32	8.8852	1.00	52051	32	8.8820	L
13	(10, 10, 12, 16, 4)	32	8.8853	1.00	60934	32	8.8820	L
14	(10, 10, 12, 16, 4)	32	8.8911	1.00	71006	32	8.8820	L
15	(10, 11, 10, 12, 8)	31	9.0876	1.00	81358	31	9.0968	L
⋮								
52	(10, 11, 10, 12, 8)	31	9.0937	1.00	1026351	31	9.0968	L
53	(10, 10, 10, 10, 10)	30	8.8573	0.86	1068509	30	8.8584	G
54	(10, 10, 10, 10, 10)	30	8.8573	0.86	1111533	30	8.8584	G
55	(10, 10, 10, 10, 10)	30	8.8573	0.86	1153787	30	8.8584	G

Figure 2 summarizes cgR-SPLINE's performance over 100 independent runs. Curves in the right hand side panel of Figure 2 display the probability of cgR-SPLINE returning a feasible solution and the true solution as a function of the total expended simulation budget. Although not shown here, a similar curve — one that estimates the probability of the incumbent solution being locally

optimal $\Pr\{\mathbf{Z}_r \in \mathcal{M}\}$ — is found to closely track the curve for feasibility. For reasons mentioned previously, we attribute this behavior to SPLINE and conclude, as one might expect in cases where a solution has a binding stochastic constraint, that much of the simulation effort seems to be expended in deciding amongst the true solution and a few competing local solutions.

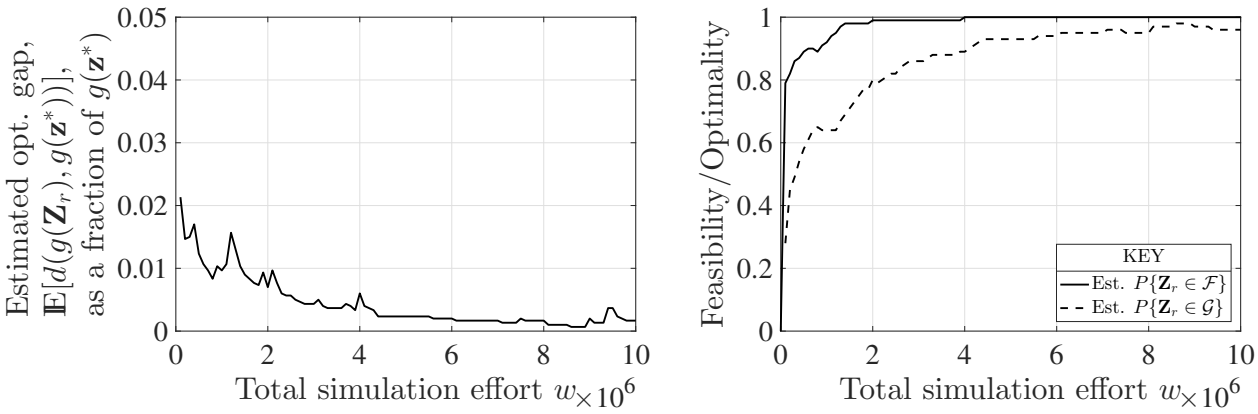

Figure 2 The figures display the performance of cgR-SPLINE on the three-stage flowline problem. Curves in the panel on the right depict the probability of an incumbent solution being (1) feasible (solid black line) and (2) optimal (dashed black line), estimated from one hundred independent runs of cgR-SPLINE, as a function of the total simulation budget expended. The left-hand side panel displays the estimated relative optimality gap of the sequence of incumbent solutions returned by cgR-SPLINE.

7.2. The (s, S) Inventory Problem

Next, we consider an optimization problem based on the well-known periodic (s, S) inventory policy (see, e.g., Hadley and Whitin 1963). At the start of each period, if the inventory level is found to be below a threshold s, an order is placed to replenish the inventory stock up to a level S, where $S > s$. Random demand is realized during the course of the period, and the inventory level is readjusted (to account for the satisfied demand) at the end of the period. Demand in each period, independently of other periods, follows a Poisson distribution with rate $\lambda = 25$. Placing an order at the start of a period incurs a cost of $c = 3$ for each unit ordered and one time order cost $A = 32$. Order delivery lag is assumed to be zero and any leftover inventory at the end of each period is

stored at a cost $h = 1$ per unit. Finally, a penalty cost $p = 5$ is incurred for each unmet unit of demand.

The objective is to find a policy $(s, S) \in \mathbb{X}$, where $\mathbb{X} = \{(s, S) \in \mathbb{Z}^2 : 20 \leq s \leq 80, 40 \leq S \leq 100, s < S\}$, that minimizes the long-term expected inventory cost per period $g(s, S)$, such that the probability of not meeting demand in any period, $h(s, S)$, is at most 0.00998. The problem has eleven local minimizers and one global solution $(s^*, S^*) = (31, 61)$ with $g(s^*, S^*) = 117.34$ and $h(s^*, S^*) = 0.00998$. Once again, the bound on the stochastic constraint is chosen deliberately so that the global solution lies on the boundary of the feasible region. Given a policy (s, S), a simulation oracle estimates the long-term expected inventory cost $g(s, S)$ as the average cost incurred in periods $n = 100$ to $n = 130$. Similarly, the shortage probability $h(s, S)$ is estimated as the fraction of periods (from $n = 100$ to 130) when demand is not completely satisfied due to insufficient inventory.

Compared to the flowline problem, the inventory problem has a small search space ($|\mathbb{X}| = 2860$), which makes it a suitable candidate for a R&S algorithm like SCORE. As will become evident, the probabilistic nature of the constraint, however, poses a considerable challenge to performing SO. In particular, we note that estimating the probability of feasibility of a candidate policy whose shortage probability (unknown to any optimization algorithm) is close to zero (and conversely, the probability of infeasibility of a policy with shortage probability close to one) is akin to estimating a rare-event probability. Such policies are detrimental to SCORE's performance as it must first obtain enough pilot samples to generate nonzero standard error estimates of the shortage probability estimator for *all* points in the search space. SCORE then iteratively updates its candidate solution based on a suboptimality-infeasibility measure called "score" that it calculates for every point in the search space. In the absence of any variance reduction measures within the oracle, SCORE is unable to complete the pilot sampling stage and return an initial candidate solution even after ten million oracle calls. cgR-SPLINE, on the other hand, identifies feasible, near-optimal solutions within 2×10^6 oracle calls, as can be seen in Figure 3. In fact, 98 percent of the solutions reported

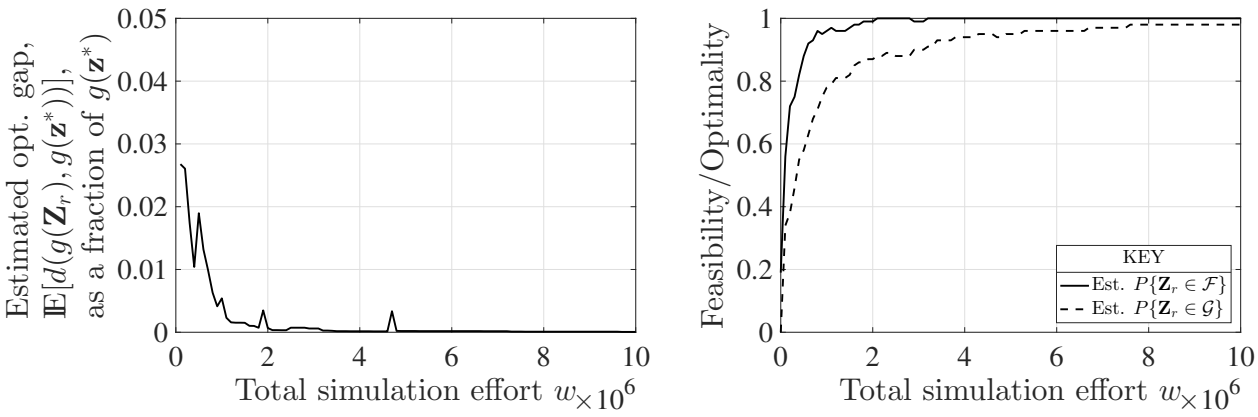

Figure 3 Figure depicts the average performance of cgR-SPLINE, calculated from 100 independent replications on the (s^*, S^*) inventory problem with $|\mathbb{X}|$=2860.

by cgR-SPLINE past the two million mark are locally optimal and within 0.5 percent of the optimal objective value.

The probability of optimality curve (dashed black line) in the right-hand side panel of Figure 3, and the corresponding curve for the flowline problem, together seem to reflect the combined influence of problem dimensionality, the number of local solutions relative to the size of the problem domain, the SPLINE local solver, and cgR-SPLINE's restart mechanism on its error rate through the implied constant τ appearing in Theorem 4. From a practical standpoint, it is worth noting, for example, that as problem dimensionality increases, τ will likely approach one and significantly affect the observed error rate.

Next, to effectively compare cgR-SPLINE's performance to that of SCORE, we construct four instances of the inventory problem, given by S_1, S_2, S_3, and S_4, with increasing problem sizes. Specifically, we define the problem instances in a way such that their search regions satisfy $\mathbb{X}_{S_1} \subseteq \mathbb{X}_{S_2} \subseteq \mathbb{X}_{S_3} \subseteq \mathbb{X}_{S_4} \subseteq \mathbb{X}$ with $|\mathbb{X}_{S_1}| = 1280$, $|\mathbb{X}_{S_1}| = 1454$, $|\mathbb{X}_{S_1}| = 1565$, and $|\mathbb{X}_{S_4}| = 1619$. All four problem instances employ the same constraint $h(s, S) \leq 0.00998$ and result in the same global minimizer $(s^*, S^*) = (31, 61)$.

Figure 4 compares the the average performance of cgR-SPLINE and SCORE on the four problem instances. Each curve is constructed from 100 independent replications of one of two algorithms on a problem instance. As noted before, SCORE does not report a solution until it has reached

the end of its pilot sampling stage. Hence, as would be expected, the number of pilot samples required by SCORE increases with problem size. For example, SCORE generates an initial solution to problem S_4 after approximately 4×10^6 oracle calls. While SCORE quickly identifies a feasible solution after the pilot stage (top right panel of Figure 4), it identifies the global solution less than forty percent of the time (bottom left panel of Figure 4) in each problem instance, which is worse than the roughly fifty percent that is expected when sampling "naïvely" (as illustrated on the toy example in Section 1.1). This is because SCORE is not designed to handle solutions with binding constraints. In fact, SCORE identifies the global solution with decreasing frequency as the simulation effort is increased. We suspect this happens because SCORE penalizes a point (in terms of the simulation budget the point gets allocated) if its "score" is very large. So if the global solution receives a high score in the pilot stage — which is expected of points with low shortage probabilities — it is becomes increasingly unlikely that SCORE will eventually recover and allocate enough samples to deem the point even feasible.

On the other hand, cgR-SPLINE seems resilient to varying problem sizes — the four solid curves in Figure 4 appear closely clustered. This behavior is once again attributed to SPLINE, which, due to its line search mechanism, is capable of identifying regions with good local solutions fairly quickly, regardless of the problem size. Furthermore, performing constraint relaxations seems to benefit cgR-SPLINE as all local solutions to the inventory problem lie close to the stochastic boundary. This is evidenced by cgR-SPLINE's ability to identify local extrema, more so than SCORE, as seen in the bottom-right panel of Figure 4. We infer that SCORE is perhaps suitable for small problem contexts and in cases where a near-optimal feasible solution will suffice. It is important to note, however, that SCORE loses its competitive edge fairly quickly as the problem size increases. More importantly, SCORE does not guarantee consistency in the presence of boundary solutions.

Two broad features of cgR-SPLINE stand out. First, a significant amount of sampling effort goes into determining solution feasibility (top right panels of Figures 2-4). This is to be expected when the solutions have binding constraints. As suggested by Theorem 3, the cost for determining

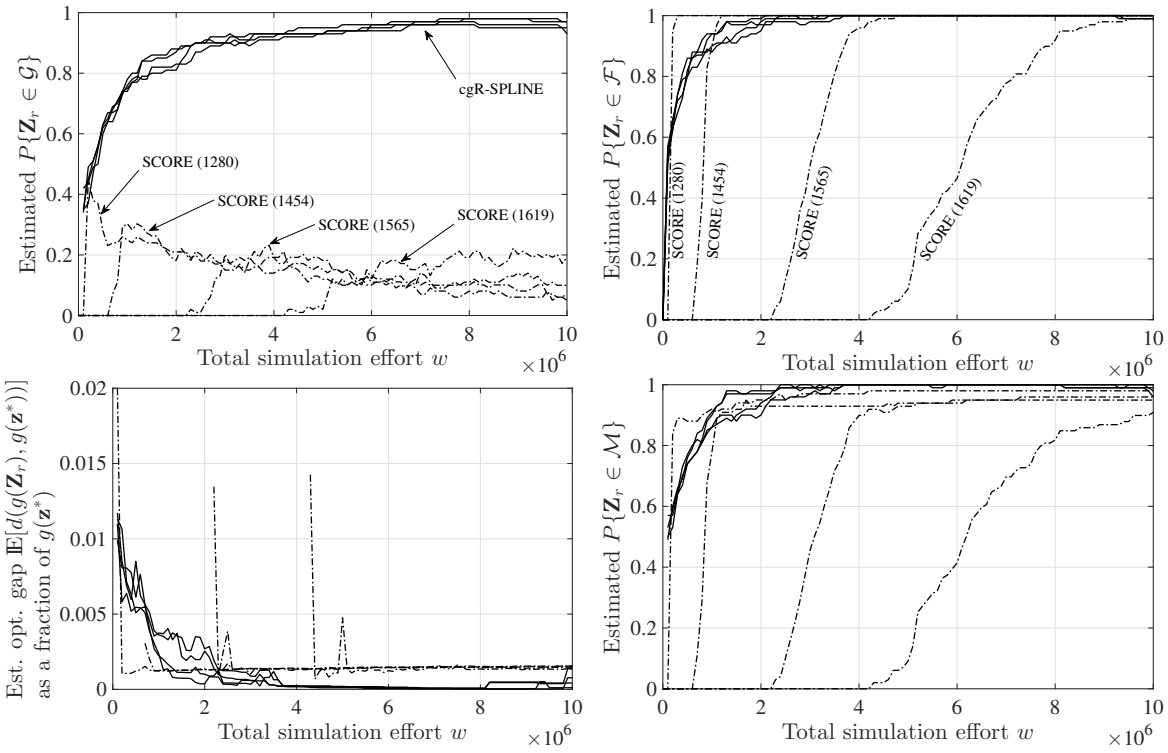

Figure 4 The figure depicts the performance of cgR-SPLINE (solid black curves) and SCORE (dashed curves) averaged over one hundred independent runs of each algorithm. Although SCORE is a R&S algorithm, it is designed specifically to handle stochastic constraints, and as such serves as a good comparator for cgR-SPLINE on smaller problems. One additional performance measure, namely, the probability of obtaining a local solution (bottom right panel) illustrate each algorithm's ability to return a *good* solution as a function of total simulation effort. We also note in passing that the the curves in the top right panel can be viewed as depicting the probability of correct selection in a R&S setting.

feasibility of boundary points is the primary cause for the deterioration of the error rate. Second, cgR-SPLINE appears to have little difficulty reaching the vicinity of the true solution in both test problems, but spends a lot of simulation effort trying to identify the best amongst a select few points competing with the true solution. We believe that such behavior is desirable and reflective of the fact that cgR-SPLINE's search routine is effective. Even when the domain is large, cgR-SPLINE seems to invariably reduce the problem to a ranking and selection problem among a few alternatives.

8. CONCLUDING REMARKS

Stochastic constraints in SO pose special challenges that cannot be addressed through sampling alone. Mechanisms such as strategic constraint relaxation or the use of penalty functions should be combined with adequate sampling to consistently solve such problems. cgR-SPLINE is an algorithm that combines strategic constraint relaxation with random restarts of a (gradient-based) locally minimizing SO algorithm to achieve (local and global) consistency. Unsurprisingly, the large-deviation type exponential convergence that is associated with SO on unconstrained discrete spaces is replaced by the slower sub-exponential convergence in the current context. Moreover, in order to achieve such a rate, the number of restarts and the total simulation budget should obey a sublinear relationship parameterized by the algorithm constants.

Apart from the quality of the locally minimizing SO algorithm in use within the cgR-SPLINE framework, we have found through fairly extensive numerical experimentation that cgR-SPLINE's finite-time performance is enhanced by three heuristics. The first relates to trading-off the costs from incorrectly assessing feasibility and infeasibility when deciding constraint relaxations; the second to returning (only) solutions that are estimated to be feasible with high probability; and the third to prematurely stopping iterations that seem to not be improving. Although the scope of our asymptotic theory covers only the first two heuristics, an extension to cover the third seems apparent.

Three other issues relating to ongoing research are worthy of note.

(i) The sequence of (outer and inner) simulation budgets are assumed to come from a deterministic sequence. Our theoretical results provide directives on the optimal speed of increase of such sequences, but this still leaves room for choice. Inspired by our work in a slightly different context, ongoing work attempts to make the choice of these simulation budgets fully adaptive to the historical algorithm trajectory. The analysis of such algorithms is nuanced but there appears to be the possibility of gains in finite-time efficiency with such an approach.

(ii) Our results on the convergence rate of cgR-SPLINE are of the $O(\cdot)$ variety; while results that more precisely characterize the convergence rate could be obtained, they will likely involve further potentially non-verifiable assumptions on algorithmic behavior.

(iii) The setting of the current paper is integer-ordered spaces. Extending cgR-SPLINE to mixed spaces presents special challenges because the crucial Assumption 2 can no longer be expected to hold. Specifically, the strategy of outer relaxation of stochastic constraints will fail owing to the presence of infeasible points that are arbitrarily close (as measured by the constraint function) to the boundary.

(iv) The structure of cgR-SPLINE, especially the use of restarts, seems amenable to parallelization. If individual processors are tasked with executing a restart, the premature termination of a restart based on the quality of the observed incumbents across processors is an interesting issue that is currently being investigated.

References

Andradóttir, S. 2006. An overview of simulation optimization via random search. S. G. Henderson, B. L. Nelson, eds., *Simulation*. Handbooks in Operations Research and Management Science, Elsevier, 617–631.

Andradóttir, S., D. Goldsman, S. -H. Kim. 2005. Finding the best in the presence of a stochastic constraint. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, J. A. Joines, eds., *Proceedings of the 2005 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 732–738.

Andradóttir, S., S. -H. Kim. 2010. Fully sequential procedures for comparing constrained systems via simulation. *Naval Research Logistics* (57) 403–421.

Atlason, J., M. Epelman, S. Henderson. 2008. Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Science* **54**(2) 295–309.

Batur, D., S. -H. Kim. 2005. Procedures for feasibility detection in the presence of multiple constraints. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, J. A. Joines, eds., *Proceedings of the 2005 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey, 692–698.

Billingsley, P. 1995. *Probability and Measure*. Wiley, New York, NY.

Bish, D. R. 2011. Planning for a bus-based evacuation. *OR Spectrum* **33**(3) 629–654.

Bish, D. R., E. Agca, R. Glick. 2011. Decision support for hospital evacuation and emergency response. *Annals of Operations Research* To appear.

Boesel, J., B. L. Nelson, S-H. Kim. 2003. Using ranking and selection to "clean up" after simulation optimization. *Operations Research* **51**(5) 814–825.

Calafiore, G., M. Campi. 2005. Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming* **102** 25–46.

Calafiore, G., M. Campi. 2006. The scenario approach to robust control design. *Automatic Control, IEEE Transactions on* **51** 742?–753.

Dembo, A., O. Zeitouni. 1998. *Large Deviations Techniques and Applications*. Springer-Verlag, New York, NY.

Eubank, S., H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z Toroczkai, N. Wang. 2004. Modelling disease outbreaks in realistic urban social networks. *Nature* **429** 180–184.

Gans, Noah, Ger Koole, Avishai Mandelbaum. 2003. Telephone call centers: Tutorial, review, and research prospects. *Mgmt* **5** 79–141.

Hadley, George, Thomson M Whitin. 1963. *Analysis of inventory systems*. Prentice Hall.

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

Hernández-Lerma, O., J. Lasserre. 1998a. Approximation schemes for infinite linear programs. *SIAM Journal on Optimization* **8**(4) 973–988.

Hernández-Lerma, O., J. Lasserre. 1998b. Linear programming approximations for markov control processes in metric spaces. *Acta Applicandae Mathematica* **51**(2). doi:10.1023/A:1005826226226.

Hong, J., B. L. Nelson. 2006. Discrete optimization via simulation using compass. *Operations Research* **54**(1) 115–129.

Hou, Y. Thomas, Yi Shi, Hanif D. Sherali. 2014. *Applied Optimization Methods for Wireless Networks*. Cambridge University Press.

Hunter, S. R., R. Pasupathy. 2010. Large-deviation sampling laws for constrained simulation optimization on finite sets. B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, E. Yücesan, eds., *Proceedings of the 2010 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey.

Hunter, S. R., R. Pasupathy. 2013. Optimal sampling laws for stochastically constrained simulation optimization. *INFORMS Journal on Computing* **25**(3) 527–542.

Hunter, S. R., N. A. Pujowidianto, L. H. Lee, C. H. Chen, R. Pasupathy. 2011. Optimal sampling laws for constrained simulation optimization on finite sets: The bivariate normal case. S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, M. Fu, eds., *Proceedings of the 2011 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey.

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

Kleywegt, A. J., A. Shapiro, T. Homem-de-Mello. 2001. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* **12** 479–502.

Kuhn, D. 2009. Convergent bounds for stochastic programs with expected value constraints. *Journal of Optimization Theory and Applications* **141**(3) 597–618.

Li, J., S. Sava, X. Xie. 2009. Simulation-based discrete optimization of stochastic discrete event systems subject to non closed-form constraints 54:2900–2904.

Lim, E. 2012. Stochastic approximation over multidimensional discrete sets with applications to inventory systems and admission control of queueing networks. *ACM TOMACS* **22**(4) 19:1–19:23.

Luo, Y., E. Lim. 2013. Simulation-based optimization over discrete sets with noisy constraints. *IIE Transactions* **45**(7) 699–715.

Nemirovski, A., A. Shapiro. 2006a. Convex approximations of chance constrained programs. *SIAM Journal on Optimization* **17** 969–996.

Nemirovski, A., A. Shapiro. 2006b. Scenario approximations of chance constraints. Giuseppe Calafiore, Fabrizio Dabbene, eds., *Probabilistic and Randomized Methods for Design under Uncertainty*. Springer London, 3–47.

O'Brien, M. 2000. Techniques for incorporating expected value constraints into stochastic programs. Ph.D. thesis, Stanford University.

Pardalos, P. M., H. E. Romeijn, eds. 2002. *Handbook of Global Optimization*, vol. 2. Kluwer Academic Publishers.

Park, C., S. -H. Kim. 2011. Handling stochastic constraints in discrete optimization via simulation. S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, M. Fu, eds., *Proceedings of the 2011 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers: Piscataway, New Jersey.

Park, C., S.-H. Kim. 2015. Penalty function with memory for discrete optimization via simulation with stochastic constraints. *Operations Research* **63**(5) 1195–1212.

Pasupathy, R., S. R. Hunter, N. A. Pujowidianto, L. H. Lee, C.-H. Chen. 2014. Stochastically constrained ranking and selection via SCORE. *ACM TOMACS* **25**(1) 1–26.

Prékopa, A. 1973. Contributions to the theory of stochastic programming. *Mathematical Programming* **4**(1) 202–221.

Prékopa, A. 1995. *Stochastic Programming*. Mathematics and Its Applications, Springer Netherlands.

Sarin, Subhash C., Puneet Jaiprakash. 2010. *Flow Shop Lot Streaming*. Springer, New York.

Shapiro, A., D. Dentcheva, A. Ruszczynski. 2009. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, PA.

Shi, L., S. Ólafsson. 2000. Nested partitions method for stochastic optimization. *Methodology and Computing in Applied Probability* **2** 271–291.

Siegmund, D. 1985. *Sequential Analysis: Tests and Confidence Intervals*. Springer, New York, NY.

Wang, H., R. Pasupathy, B. W. Schmeiser. 2013. Integer-ordered simulation optimization using R-SPLINE: Retrospective search using piecewise-linear interpolation and neighborhood enumeration. *ACM TOMACS* **23**(3).

Wang, W, S Ahmed. 2008. Sample average approximation of expected value constrained stochastic programs. *Operations Research Letters* **36**(5) 515–519.

Xu, J., L. J. Hong, B. L. Nelson. 2010. Industrial strength compass: A comprehensive algorithm and software for optimization via simulation. *ACM TOMACS* (20) 1–29.

APPENDIX

EC.1. Proofs of Lemmas and Theorems

Proof of Lemma 1. Pick $\boldsymbol{x} \in \mathcal{F}$. Then $h_i(\boldsymbol{x}) \leqslant 0$, $i = 1, \ldots, \ell$.

$$\begin{aligned}
\Pr\{\boldsymbol{x} \notin \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcup_{i=1}^{\ell}\left(\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}\right)\right\} \\
&\leqslant \sum_{i=1}^{\ell} \Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}\right\} \\
&\leqslant \sum_{i=1}^{\ell} \Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x}) \notin (h_i(\boldsymbol{x}) - \epsilon_{i,k}, h_i(\boldsymbol{x}) + \epsilon_{i,k})\right\} \\
&\leqslant \sum_{i=1}^{\ell} \frac{\mathrm{Var}\left(\hat{h}_{i,m_k}(\boldsymbol{x})\right)}{\epsilon_{i,k}^2} \\
&= \sum_{i=1}^{\ell} \frac{\sigma_i^2(x)}{m_k \epsilon_{i,k}^2} \qquad \text{(EC.1)}
\end{aligned}$$

where the third inequality of (EC.1) follows from Chebyshev's inequality. Then under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the first Borel-Cantelli lemma, there exists K_1(independent of $\boldsymbol{x}$) such that for $k \geqslant K_1$, $\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$ wp1 for any $\boldsymbol{x} \in \mathcal{F}$. Thus $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k)^c \subseteq \mathcal{F}^c$ wp1 when $k \geqslant K_1$.

Now pick $\boldsymbol{x} \in \mathcal{F}^c$. Then $h_{j,m_k}(\boldsymbol{x}) > 0$ for some $j \in \{1, \ldots, c\}$. Under Assumption 2 there exists $\Delta' > 0$ such that $2\Delta' < \inf_{\boldsymbol{x} \in \mathcal{F}^c}\{h_i(\boldsymbol{x}) : h_i(\boldsymbol{x}) > 0, i = 1, \ldots, \ell\}$. Since $\lim_{k\to\infty} \epsilon_{i,k} = 0$, there exists K_2(independent of $\boldsymbol{x}$) such that for all $k \geqslant K_2$, $\epsilon_{i,k} < \gamma$, and thus

$$\begin{aligned}
\Pr\{\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcap_{i=1}^{\ell}\left(\hat{h}_{i,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}\right)\right\} \\
&\leqslant \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}\right\} \\
&= \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}, \hat{h}_{j,m_k}(\boldsymbol{x}) \in (h_j(\boldsymbol{x}) - \Delta', h_j(\boldsymbol{x}) + \Delta')\right\} \\
&\quad + \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}, \hat{h}_{j,m_k}(\boldsymbol{x}) \notin (h_j(\boldsymbol{x}) - \Delta', h_j(\boldsymbol{x}) + \Delta')\right\} \\
&\leqslant \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x}) \notin (h_j(\boldsymbol{x}) - \Delta', h_j(\boldsymbol{x}) + \Delta')\right\} \\
&\leqslant \frac{\mathrm{Var}\left(\hat{h}_{j,m_k}(\boldsymbol{x})\right)}{\Delta'^2} \\
&\leqslant \frac{\sigma_j^2(\boldsymbol{x})}{m_k \epsilon_{j,k}^2}. \qquad \text{(EC.2)}
\end{aligned}$$

Similarly, under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the first Borel-Cantelli lemma, $\mathcal{F}(m_k,\boldsymbol{\epsilon}_k)\subseteq\mathcal{F}$ wp1 for all $k\geqslant K_2$, and the result follows. □

Proof of Lemma 2. Pick $\boldsymbol{x}\in\mathcal{F}$. Then $h_i(\boldsymbol{x})\leqslant 0, i=1,\ldots,\ell$.

$$\begin{aligned}\Pr\{\boldsymbol{x}\notin\mathcal{F}(m_k,\boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcup_{i=1}^{\ell}\left(\hat{h}_{i,m_k}(\boldsymbol{x})>\epsilon_{i,k}\right)\right\}\\ &\leqslant \sum_{i=1}^{\ell}\Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x})>\epsilon_{i,k}\right\}\\ &\leqslant \sum_{i=1}^{\ell}\Pr\left\{\hat{h}_{i,m_k}(\boldsymbol{x})>\epsilon_{i,k}+h_i(\boldsymbol{x})\right\}\leqslant\sum_{i=1}^{\ell}e^{-\frac{m_k\epsilon_{i,k}^2}{2\sigma_i^2(\boldsymbol{x})}}. \end{aligned}\tag{EC.3}$$

Then under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the first Borel-Cantelli lemma, for $k\geqslant K_1$(independent of $\boldsymbol{x}$), $\boldsymbol{x}\in\mathcal{F}(m_k,\boldsymbol{\epsilon}_k)$ wp1 and hence $\mathcal{F}(m_k,\boldsymbol{\epsilon}_k)^c\subseteq\mathcal{F}^c$ wp1 uniformly on $\mathbb{X}$.

Now pick $\boldsymbol{x}\in\mathcal{F}^c$. Then $h_j(\boldsymbol{x})>0$ for some $j\in\{1,\ldots,c\}$. Under Assumption 2 there exists $\Delta''>0$ such that $2\Delta''<\inf_{\boldsymbol{x}\in\mathcal{F}^c}\{h_i(\boldsymbol{x}):h_i(\boldsymbol{x})>0,i=1,\ldots,\ell\}$. Then since $\lim_{k\to\infty}\epsilon_{i,k}=0,\epsilon_{i,k}<\Delta''$ for all $k\geqslant K_2$ (ndependent of $\boldsymbol{x}$) and

$$\begin{aligned}\Pr\{\boldsymbol{x}\in\mathcal{F}(m_k,\boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcap_{i=1}^{\ell}\left(\hat{h}_{i,m_k}(\boldsymbol{x})\leqslant\epsilon_{i,k}\right)\right\}\\ &\leqslant \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x})\leqslant\epsilon_{j,k}\right\}\\ &= \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x})\leqslant\epsilon_{i,k},\,\hat{h}_{j,m_k}(\boldsymbol{x})\in(h_j(\boldsymbol{x})-\Delta'',h_j(\boldsymbol{x})+\Delta'')\right\}\\ &\quad+\Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x})\leqslant\epsilon_{i,k}\,\hat{h}_{j,m_k}(\boldsymbol{x})\notin(h_j(\boldsymbol{x})-\Delta'',h_j(\boldsymbol{x})+\Delta'')\right\}\\ &\leqslant \Pr\left\{\hat{h}_{j,m_k}(\boldsymbol{x})\notin(h_j(y)-\Delta'',h_j(\boldsymbol{x})+\Delta'')\right\}\\ &\leqslant 2e^{-\frac{m_k\Delta''^2}{2\sigma_j^2(\boldsymbol{x})}}\leqslant 2e^{-\frac{m_k\epsilon_{j,k}^2}{2\sigma_j^2(\boldsymbol{x})}}. \end{aligned}\tag{EC.4}$$

Thus, under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the application of the first Borel-Cantelli lemma, $\mathcal{F}(m_k,\boldsymbol{\epsilon}_k)\subseteq\mathcal{F}$ wp1 when $k\geqslant K_2$, and the result follows. □

Proof of Theorem 2. Recall that k_r denotes the (random) number of inner iterations executed during the rth outer iteration. Since $U_{k,r}$, the number of steps executed by the local SO solver during the kth inner iteration of any rth outer iteration, is uniformly bounded, $u = \limsup_{k,r} U_{k,r} < \infty$. Also recall that $\underline{k}_r$ denotes the *smallest* number of inner iterations executed in the rth outer iteration. Note that $\underline{k}_r \leqslant k_r$ wp1 for all r and $\underline{k}_r \to \infty$ since $b_r \to \infty$ and $m_k > 0$.

Pick $\boldsymbol{x}_0 \in \mathcal{F}$. Then by Assumption 7 there exists $\lambda > 0$ such that the level set $\mathcal{L}(\boldsymbol{x}_0, \lambda)$ is finite. Pick $\boldsymbol{y} \in (\mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda))^c$. Then either $\boldsymbol{y} \in \mathcal{F}^c$ or $\boldsymbol{y} \in \mathcal{L}(\boldsymbol{x}_0, \lambda)^c \bigcap \mathcal{F}$. Suppose $\boldsymbol{y} \in \mathcal{F}^c$. Then for $r \geqslant K_1$ (independent of $\boldsymbol{x}_0$ and $\boldsymbol{y}$), $\Pr\left\{\boldsymbol{y} \in \mathcal{L}_{m_{k_r}}(\boldsymbol{x}_0)\right\} \leqslant \Pr\left\{\boldsymbol{y} \in \mathcal{F}_{m_{k_r}}\right\} = O(e^{-\tilde{c} m_{\underline{k}_r}}) = a'_r$ (from (3)). If $\boldsymbol{y} \in \mathcal{L}(\boldsymbol{x}_0, \lambda)^c \bigcap \mathcal{F}$ then $g(\boldsymbol{y}) > g(\boldsymbol{x}_0) + \lambda$. For all $r \geqslant K_2$ (independent of $\boldsymbol{y}$, dependent on $\boldsymbol{x}_0$),

$$
\begin{aligned}
\Pr\left\{\boldsymbol{y} \in \mathcal{L}_{m_{k_r}}(\boldsymbol{x}_0)\right\} &= \Pr\left\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \leqslant \hat{g}_{m_{k_r}}(\boldsymbol{x}_0)\right\} \\
&\leqslant \Pr\left\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \leqslant \hat{g}_{m_{k_r}}(\boldsymbol{x}_0),\, \hat{g}_{m_{k_r}}(\boldsymbol{x}_0) \in (g(\boldsymbol{x}_0) - \lambda/4, g(\boldsymbol{x}_0) + \lambda/4)\right\} \\
&\quad + \Pr\left\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \leqslant \hat{g}_{m_{k_r}}(\boldsymbol{x}_0),\, \hat{g}_{m_{k_r}}(\boldsymbol{x}_0) \notin (g(\boldsymbol{x}_0) - \lambda/4, g(\boldsymbol{x}_0) + \lambda/4)\right\} \\
&\leqslant \Pr\left\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \notin (g(\boldsymbol{y}) - \lambda/4, g(\boldsymbol{y}) + \lambda/4)\right\} + \Pr\left\{\hat{g}_{m_{k_r}}(\boldsymbol{x}_0) \notin (g(\boldsymbol{x}_0) - \lambda/4, g(\boldsymbol{x}_0) + \lambda/4)\right\} \\
&\leqslant 2e^{-m_{k_r}\eta_g} \\
&\leqslant 2e^{-m_{\underline{k}_r}\eta_g} = a''_r. \qquad \text{(EC.5)}
\end{aligned}
$$

Let $a_r = \max(a'_r, a''_r)$. Then for any $\boldsymbol{y} \in (\mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda))^c$, $\Pr\{\boldsymbol{y} \in \mathcal{L}_{m_{k_r}}(\boldsymbol{x}_0)\} \leqslant a_r$ if $r \geqslant K_3 = \max(K_1, K_2)$. Then under condition C.2 and by the first Borel-Cantelli lemma (Billingsley 1995), $\Pr\{y \in \mathcal{L}_{m_{k_r}}(\boldsymbol{x}_0)\ i.o.\} = 0$. In other words $\Pr\{\mathcal{L}_{m_{k_r}}(\boldsymbol{x}_0) \nsubseteq \mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda)\, i.o.\} = 0$. Then $\Pr\{\boldsymbol{Y}_r \notin \mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda)\ \text{i.o.}\ \} = 0$ as $\boldsymbol{Y}_r \in \mathcal{L}_{m_{k_r}}(\boldsymbol{x}_0)$. And since $\mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda)$ is finite, cgR-SPLINE returns a sequence of solutions that are bounded wp1.

Let $\lambda' = \min\{|g(\boldsymbol{x}) - g(\boldsymbol{y})| :\ (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda), g(\boldsymbol{x}) \neq g(\boldsymbol{y})\}$. Then since $\mathcal{F} \bigcap\ \mathcal{L}(\boldsymbol{x}_0, \lambda)$ is finite and contained in $\mathcal{F}$, $\hat{g}_{m_{k_r}}$ converges uniformly to g wp1 on the set $\mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda)$ as $r \to \infty$. Thus there exists $K_4(\boldsymbol{x}_0) \in \mathbb{N}$ such that $|\hat{g}_{m_{k_r}}(\boldsymbol{x}) - g(\boldsymbol{x})| < \lambda'/2$ with probability 1 if $r \geqslant K_4$ for all $\boldsymbol{x} \in \mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda)$. So if $g(\boldsymbol{y}) < g(\boldsymbol{x})$ then with probability 1, $\hat{g}_{m_{k_r}}(\boldsymbol{y}) < \hat{g}_{m_{k_r}}(\boldsymbol{x})$ for all $\boldsymbol{x}$, $\boldsymbol{y} \in \mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda)$ if $r \geqslant K_4$. This in turn implies that with probability 1 if $\hat{g}_{m_{k_r}}(\boldsymbol{y}) \geqslant \hat{g}_{m_{k_r}}(\boldsymbol{x})$ then

$g(\boldsymbol{y}) \geqslant g(\boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{F} \bigcap \mathcal{L}(\boldsymbol{x}_0, \lambda)$, $r \geqslant K_4$. Hence, $\boldsymbol{Y}_r \in M^*(N)$ wp1 for $r \geqslant \max(K_3, K_4)$, and the result follows. □

Proof of Theorem 4.

$$\Pr\{\boldsymbol{Z}_r \notin \mathcal{G}\} = \underbrace{\Pr\left\{\bigcap_{j=1}^{r} (\boldsymbol{Y}_j \notin \mathcal{G})\right\}}_{\substack{\text{the event that} \\ \text{a global solution} \\ \text{is never attained}}} + \underbrace{\Pr\left\{\bigcup_{j=2}^{r} \left(\begin{array}{l} \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathcal{G} \neq \emptyset, \\ \text{a global solution was last dropped} \\ \text{at the end of the } j\text{th outer iteration} \end{array}\right)\right\}}_{\substack{\text{the event that a global solution is attained} \\ \text{but dropped due to sampling error}}}$$

$$= \underbrace{\Pr\left\{\bigcap_{j=1}^{r} (\boldsymbol{Y}_j \notin \mathcal{G})\right\}}_{\text{(I)}} + \underbrace{\sum_{j=2}^{r} \Pr\left\{ (\boldsymbol{Y}_j \in \mathcal{G} \bigcup \boldsymbol{Z}_{j-1} \in \mathcal{G}) \bigcap \left(\boldsymbol{Z}_j \notin \mathcal{G}, \left(\bigcap_{i=j+1}^{r} \boldsymbol{Y}_i \notin \mathcal{G}\right)\right) \right\}}_{\text{(II)}} \tag{EC.6}$$

Let $B(\mathcal{G})$ denote the set $\bigcup_{\boldsymbol{z}^* \in \mathcal{G}} B(\boldsymbol{z}^*)$. Then since each restart is performed independently, the first term in (EC.6) can be written as

$$\begin{aligned} \text{(I)} = \prod_{j=1}^{r} \Pr\{\boldsymbol{Y}_j \notin \mathcal{G}\} = \prod_{j=1}^{r} \big(& \Pr\{\boldsymbol{Y}_j \notin \mathcal{G}, \boldsymbol{Y}_\infty \in \mathcal{G} \mid \boldsymbol{X}_j \in B(\mathcal{G})\} p_j(\mathcal{G}) \\ & + \Pr\{\boldsymbol{Y}_j \notin \mathcal{G}, \boldsymbol{Y}_\infty \in \mathcal{G} \mid \boldsymbol{X}_j \notin B(\mathcal{G})\} (1 - p_j(\mathcal{G})) \\ & + \Pr\{\boldsymbol{Y}_j \notin \mathcal{G}, \boldsymbol{Y}_\infty \notin \mathcal{G} \mid \boldsymbol{X}_j \in B(\mathcal{G})\} p_j(\mathcal{G}) \\ & + \Pr\{\boldsymbol{Y}_j \notin \mathcal{G}, \boldsymbol{Y}_\infty \notin \mathcal{G} \mid \boldsymbol{X}_j \notin B(\mathcal{G})\} (1 - p_j(\mathcal{G}))) \end{aligned}$$

Let $\nu = \inf_{\boldsymbol{x} \in B(\mathcal{G})} \Pr\{\boldsymbol{Y}_\infty \in \mathcal{G} | \boldsymbol{X}_j = \boldsymbol{x}\}$ for any restart j. Then from Theorem 3, there exists $R \in \mathbb{Z}$ such that for all $r > R$ and for some $c' > 0$ and $\beta > 0$ that is arbitrarily close to zero,

$$\begin{aligned} \text{(I)} &\leqslant \prod_{j=R+1}^{r} \left(p_j(\mathcal{G}) O\left(e^{-c' m_{\underline{k}_j}^{1-2\delta-\beta}}\right) + (1-\nu) p_j(\mathcal{G}) + (1 - p_j(\mathcal{G}))\right) \\ &= \prod_{j=R+1}^{r} \left(\underbrace{O\left(e^{-c' m_{\underline{k}_j}^{1-2\delta-\beta}}\right)}_{\substack{\text{error in function} \\ \text{estimation}}} + \underbrace{1 - \nu p_j(\mathcal{G})}_{\substack{\text{error due to} \\ \text{stochasticity} \\ \text{of algorithm}}} \right) \end{aligned}$$

Let $\rho_j = 1 - \nu p_j(\mathcal{G})$ and $\rho^* = \limsup_{j\to\infty} \rho_j$. Then from Assumption (8), $\rho^* < 1$ and

$$(\mathrm{I}) \leqslant \prod_{j=R+1}^{r} \left(O\left(e^{-c' m_{\underline{k}_j}^{1-2\delta-\beta}} \right) + \rho^* \right).$$

Now suppose $m_k = \Theta(k^q)$ where q satisfies condition C.2. Then $b_j = \sum_{i=1}^{\underline{k}_j} u m_i = \Theta(\underline{k}_j^{q+1})$ or $\underline{k}_j = \Theta\left(b_j^{1/(q+1)}\right)$. Consequently, $m_{\underline{k}_j} = \Theta\left(b_j^{q/(q+1)}\right)$. If $m_k = \Theta(a_{in}^k)$ where $a_{in} > 1$, then condition C.2 is satisfied and $b_j = \sum_{i=1}^{\underline{k}_j} u m_i = \Theta\left(a_{in}^{\underline{j}_r+1}\right)$ or $\underline{k}_j = \log_{a_{in}} b_j + \Theta(1)$. This gives $m_{\underline{k}_j} = \Theta(b_j)$. For either choice of the inner sample size sequence $\{m_k\}$, we can express the first term in (EC.6) as

$$(\mathrm{I}) \leqslant \prod_{j=R+1}^{r} \left(O\left(e^{-c' b_j^{\gamma}} \right) + \rho^* \right)$$

where $\gamma = q(1-2\delta-\beta)/(q+1))$ if $\{m_k\}$ is increased polynomially and $\gamma = 1-2\delta-\beta$ if $\{m_k\}$ is increased geometrically. Since the sequence $\{b_r\}$ approaches infinity, there exists some R' such that $c'' e^{-c' b_j^{\gamma}} < (1-\rho^*)/2$ for all $j \geqslant R'$, which gives

$$(\mathrm{I}) \leqslant \prod_{j=\max\{R,R'\}+1}^{r} \left(\frac{1-\rho^*}{2} + \rho^* \right) = O(\tau^r) \tag{EC.7}$$

where $\tau = (1-\rho^*)/2 + \rho^* \in (0,1)$. The constant τ can be viewed as representing the probability of not achieving the global solution in any one restart and can be attributed to a combination of multiple factors: error in function estimation, error in selecting restarts locations, and error due to the local solver.

Now consider the second term in equation (EC.6). Recall that t_r, defined in Step 12 of Algorithm 1, represents the sample size at which the incumbent solution $\boldsymbol{Z}_{r-1}$ is compared with the local solution $\boldsymbol{Y}_r$, and the set $\mathcal{F}(t_r, \boldsymbol{\epsilon}(t_r))$ represents the corresponding relaxed sample-path feasible

region. Thus, for large enough r,

$$
\begin{aligned}
\text{(II)} = \sum_{j=2}^{r} \big(& \Pr\{\boldsymbol{Z}_{j-1} \text{ loses to } \boldsymbol{Y}_j \,|\, \boldsymbol{Z}_{j-1} \in \mathcal{G}, \boldsymbol{Y}_j \notin \mathcal{G}\} \Pr\{\boldsymbol{Z}_{j-1} \in \mathcal{G},\, \boldsymbol{Y}_j \notin \mathcal{G}\} \\
& + \Pr\{\boldsymbol{Y}_j \text{ loses to } \boldsymbol{Z}_{j-1} \,|\, \boldsymbol{Y}_j \in \mathcal{G}, \boldsymbol{Y}_{j-1} \notin \mathcal{G}\} \Pr\{\boldsymbol{Y}_j \in \mathcal{G}, \boldsymbol{Z}_{j-1} \notin \mathcal{G}\}\big) \Pr\left\{ \bigcap_{i=j+1}^{r} (\boldsymbol{Y}_i \notin \mathcal{G}) \right\} \\
\leqslant \sum_{j=2}^{r} & \left(\prod_{i=j+1}^{r} \Pr\{\boldsymbol{Y}_i \notin \mathcal{G}\} \right) \big(\Pr\{\boldsymbol{Z}_{j-1} \in \mathcal{F}^c(t_j, \boldsymbol{\epsilon}(t_j)),\, \boldsymbol{Y}_j \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)) \,|\, \boldsymbol{Z}_{j-1} \in \mathcal{G}, \boldsymbol{Y}_j \in \mathcal{F}\backslash\mathcal{G}\} \\
& + \Pr\left\{\hat{g}_{t_j}(\boldsymbol{Z}_{j-1}) > \hat{g}_{t_j}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)),\, \boldsymbol{Y}_j \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)) \,|\, \boldsymbol{Z}_{j-1} \in \mathcal{G},\, \boldsymbol{Y}_j \in \mathcal{F}\backslash\mathcal{G}\right\} \\
& + \Pr\left\{\hat{g}_{t_j}(\boldsymbol{Z}_{j-1}) > \hat{g}_{t_j}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)),\, \boldsymbol{Y}_j \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)) \,|\, \boldsymbol{Z}_{j-1} \in \mathcal{G},\, \boldsymbol{Y}_j \in \mathcal{F}^c\right\} \\
& + \Pr\{\boldsymbol{Z}_{j-1} \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)),\, \boldsymbol{Y}_j \in \mathcal{F}^c(t_j, \boldsymbol{\epsilon}(t_j)) \,|\, \boldsymbol{Z}_{j-1} \in \mathcal{F}\backslash\mathcal{G}, \boldsymbol{Y}_j \in \mathcal{G}\} \\
& + \Pr\left\{\hat{g}_{t_j}(\boldsymbol{Z}_{j-1}) < \hat{g}_{t_j}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)),\, \boldsymbol{Y}_j \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)) \,|\, \boldsymbol{Z}_{j-1} \in \mathcal{F}\backslash\mathcal{G},\, \boldsymbol{Y}_j \in \mathcal{G}\right\} \\
& + \Pr\left\{\hat{g}_{t_j}(\boldsymbol{Z}_{j-1}) < \hat{g}_{t_j}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)),\, \boldsymbol{Y}_j \in \mathcal{F}(t_j, \boldsymbol{\epsilon}(t_j)) \,|\, \boldsymbol{Z}_{j-1} \in \mathcal{F}^c,\, \boldsymbol{Y}_j \in \mathcal{G}\right\}\big).
\end{aligned}
$$

Moreover, since $t_j \geqslant m_{k_j} \geqslant m_{\underline{k}_j}$ wp1, we obtain

$$
\begin{aligned}
\text{(II)} &\leqslant \sum_{j=2}^{R} \left(\prod_{i=j+1}^{r} \Pr\{\boldsymbol{Y}_i \notin \mathcal{G}\} \right) \\
&\quad + \sum_{j=R+1}^{r} \left(\prod_{i=j+1}^{r} \Pr\{\boldsymbol{Y}_i \notin \mathcal{G}\} \right) \left(O\left(e^{-c' m_{\underline{k}_j}^{1-2\delta-\beta}}\right) + O\left(e^{-\eta_g m_{\underline{k}_j}}\right) + O\left(e^{-c'' m_{\underline{k}_j}}\right) \right) \\
&= O(\tau^r) + \sum_{j=R+1}^{r} O\left(\tau^{r-j}\right) O\left(e^{-c' m_{\underline{k}_j}^{1-2\delta-\beta}}\right) \\
&= O(\tau^r) + O(\tau^r) \sum_{j=R+1}^{r} O\left(e^{j \log \tau - c' m_{\underline{k}_j}^{1-2\delta-\beta}}\right) \\
&= O(\tau^r) + O(\tau^r) \sum_{j=R+1}^{r} O\left(e^{-c' m_{\underline{k}_j}^{1-2\delta-\beta}}\right) && \text{(From condition C.2)} \\
&= O(\tau^r) + O(\tau^r) O\left(e^{-c' m_{\underline{k}_r}^{1-2\delta-\beta}}\right) \\
&= O(\tau^r) + O\left(e^{r \log \tau - c' m_{\underline{k}_r}^{1-2\delta-\beta}}\right) \\
&= O(\tau^r) + O\left(e^{-c' m_{\underline{k}_r}^{1-2\delta-\beta}}\right) && \text{(From condition C.2)}
\end{aligned}
$$

where the first equality follows from Assumption 6 and Theorem 3. The constant τ is defined in line (EC.7), $c' > 0$ is independent of $\boldsymbol{x}$, and $\beta > 0$ is arbitrarily close to zero. Finally substituting

for $m_{\underline{k}_r}$, we get

$$(\text{II}) = O\left(\tau^r\right) + O\left(e^{-c' b_r^{\gamma}}\right), \tag{EC.8}$$

where γ is as defined in (8). Finally, substituting (EC.7) and (EC.8) back in (EC.6) we get

$$\Pr\left\{\boldsymbol{Z}_r \notin \mathcal{G}\right\} = O\left(e^{-c' b_r^{\gamma}}\right) + O\left(\tau^r\right), \tag{EC.9}$$

giving us the result in (i). The result in (ii) follows from condition C.3 by the application of the first Borel-Cantelli lemma (Billingsley 1995). □

EC.2. The SPLINE Algorithm

We provide details of the SPLINE algorithm in this section with the purpose of aiding the reader's understanding of the inner iterations of cgR-SPLINE. Heuristics described in Section 6 are excluded from this listing as we believe their addition will needlessly complicate the discussion. MATLAB code for cgR-SPLINE, complete with heuristic capabilities, can be downloaded from `http://iem.okstate.edu/nagaraj`.

The SPLINE algorithm in cgR-SPLINE is structurally identical to its counterpart in R-SPLINE (Wang et al. 2013). During the kth inner iteration of the rth restart, cgR-SPLINE solves the sample path Problem $P(m_k, \boldsymbol{\epsilon}_k)$ by calling SPLINE, which works as follows. SPLINE calls two routines, SPLI followed by NE, repeatedly. SPLI performs a gradient-based search on a continuous piecewise linear interpolation of the function estimate $\hat{g}_{m_k}$, and NE asserts local optimality of the solution returned by SPLI by enumerating all points in its neighborhood. SPLINE terminates when NE identifies a local solution, or if the simulation budget assigned to SPLINE is exhausted.

SPLINE differs, however, in the way its handles solution feasibility. If the initial solution to SPLINE is sample-path infeasible, the algorithm traces back its steps to a previously visited point, but only within the ongoing restart, and continues doing so until it finds one that is sample-path feasible. This backtracking of steps occurs at the start of the SPLINE routine (Steps 1-6 of Algorithm 2). A point $\boldsymbol{x} \in \mathbb{X}$ encountered in the kth inner iteration of any restart is deemed sample-path feasible, denoted $\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$, if it satisfies

$$\hat{h}_{i,m_k}(\boldsymbol{x}) \leq \frac{\widehat{\text{s.e.}}(\hat{h}_{i,m_k}(\boldsymbol{x}))}{m_k^{\delta}} \text{ for all } i \in \{1, 2, \ldots, \ell\}.$$

If no (sample-path) feasible solution is found, cgR-SPLINE terminates the inner iterations and restarts the local search (after incrementing r) from a new location $\boldsymbol{X}_r$.

The **SPLINE** Algorithm: Returns a local solution to Problem $P(m_k, \boldsymbol{\epsilon}_k)$

Require: Initial solution $\boldsymbol{X}_0$ and solutions $\{\mathbf{W}_1, \ldots, \mathbf{W}_{k-1}\}$ to sample path problems $P(m_1, \boldsymbol{\epsilon}_1), \ldots, P(m_{k-1}, \boldsymbol{\epsilon}_{k-1})$; sample size m_k; SPLINE budget n_k; constraint relaxation $\boldsymbol{\epsilon}_k$

Ensure: Sample-path local solution $\boldsymbol{X}_{\text{SPLINE}}$, budget utilized N_k

Initialize: $\mathbf{W}_0 = \mathbf{X}_0$, $j = k-1$, and $N_k = 0$
repeat
  Set: $\boldsymbol{X} = \mathbf{W}_j$
  Observe: $\hat{g}_{m_k}(\mathbf{X}), \hat{h}_{i,m_k}(\mathbf{X}), i = i, \ldots, \ell$ {Update function estimates of previously identified solutions with sample size m_k}
  Update: $N_k = N_k + m_k$
  Set: $j = j - 1$
until $\mathbf{X} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$ or $j = -1$ {Identify a feasible warm start}
if $\mathbf{X} \notin \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$ **then**
  Set: $\boldsymbol{X}_{\text{SPLINE}} = \emptyset$ and return $[\boldsymbol{X}_{\text{SPLINE}}, N_k]$ {Terminate inner iterations of cgR-SPLINE and restart local search from a new location}
end if
Initialize: $\boldsymbol{X}_{\text{NE}} = \boldsymbol{X}$
repeat
  $[N_{\text{SPLI}}, \boldsymbol{X}_{\text{SPLI}}] = \text{SPLI}(\boldsymbol{X}_{\text{NE}}, m_k, n_k - N_k, \boldsymbol{\epsilon}_k)$ {Continuous line-search based on phantom gradients}
  $[N_{\text{NE}}, \boldsymbol{X}_{\text{NE}}] = \text{NE}(\boldsymbol{X}_{\text{SPLI}}, m_k, \boldsymbol{\epsilon}_k)$ {Enumerate neighborhood $N(\boldsymbol{X}_{\text{SPLI}})$ of $\boldsymbol{X}_{\text{SPLI}}$}
  Update: $N_k = N_k + N_{\text{SPLI}} + N_{\text{NE}}$ {Update number of oracle calls expended}
until $\hat{g}_{m_k}(\boldsymbol{X}_{\text{NE}}) = \hat{g}_{m_k}(\boldsymbol{X}_{\text{SPLI}})$ or $N_k > n_k$ {Line search ends on a local solution}
Set: $\boldsymbol{X}_{\text{SPLINE}} = \boldsymbol{X}_{\text{NE}}$ and return $[\boldsymbol{X}_{\text{SPLINE}}, N_k]$

Algorithm 2 SPLINE can be viewed as a deterministic local solver for the sample-path Problem $P(m_k, \boldsymbol{\epsilon}_k)$ *after* randomness has been realized. A call to SPLINE is placed in each inner iteration of cgR-SPLINE, which corresponds to a retrospective iteration of R-SPLINE. As such the inner iterations of cgR-SPLINE are identical across all restarts. Consequently, for notational simplicity, we drop the explicit reference to the outer iteration r wherever convenient. For example, the solution $\mathbf{W}_{k,r}$ returned by SPLINE in the kth inner iteration of the rth restart is written simply as $\mathbf{W}_k$ inside SPLINE. Similarly, the simulation budget $N_{k,r}$ expended to solve Problem $P(m_k, \boldsymbol{\epsilon}_k)$ is written as N_k inside SPLINE.

In addition to the set of previous sample-path local solutions $\{\mathbf{W}_1, \ldots, \mathbf{W}_{k-1}\}$, the initial location $\mathbf{X}_0$, sample size m_k, and constraint relaxation $\boldsymbol{\epsilon}_k$, SPLINE also accepts as input an upper bound n_k on the number of oracle calls. This input parameter is implicit in the call to SPLINE on

Line 8 of Algorithm 1, mostly because we regard such an upper bound a numerical necessity; it safeguards the local search from an infinite loop that can result, for example, when the problem is unbounded. The upper bound n_k is found to be nonbinding past a certain k (wp1) when $n_k \to \infty$, and convergence of cgR-SPLINE is not affected.

Finally, we note that our code for the inner iterations of cgR-SPLINE searches for an N_1-local minimizer. That is to say, NE determines local optimality by comparing a candidate solution with at most $2d$ of its sample-path feasible neighbors that are within unit distance on the integer lattice $\mathbb{Z}^d$. Additionally, SPLI and NE always return sample-path feasible solutions because each relies on a sample-path feasible point as input. As both routines are identical in structure to their counterparts in R-SPLINE, and vary only in the way they account for feasibility, we refer the reader to Wang et al. (2013) for their detailed lisitings. As was previously mentioned, any point $\boldsymbol{x} \in \mathbb{X}$ encountered by one of the procedures inside cgR-SPLINE is deemed feasible only if $\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)$.

EC.3. Numerical Experiments On Algorithm Paramater Choices

EC.3.1. Choice of inner sample size sequence $\{m_k\}$ and restart budget sequence $\{b_r\}$

Efficiency results for cgR-SPLINE (Section 5.2) dictate that (i) the inner sample size m_k be increased exponentially in the number of inner iterations k, that is, as $\Theta(a_{in}^k)$, where $a_{in} > 1$, and (ii) the restart budget b_r be increased as $\Theta(r^q)$, where $q > 1$, in the number of restarts r.

Our choice of the inner sample size sequence, namely $m_k = 1.1m_{k-1}$, which conforms to the above directive, is informed by the second author's numerical experience with R-SPLINE (see Wang et al. (2013) for details). We also believe a ten precent increase in sample size after each inner iteration is practicable in most SO problem settings; the resulting sample size seqeunce has consistently resulted in good numerical performance of R-SPLINE and cgR-SPLINE. A more sophisticated version of cgR-SPLINE may perhaps utilize a variable sample size at each visited point (to account for the heteroskedasticity of the estimated objective and constraint functions across the decision variable space $\mathbb{X}$), but this places the algorithm outside the scope of an RA framework.

As previously noted, cgR-SPLINE can be made to achieve its fastest convergence rate (that is arbitrarily close to the canonical rate $O(e^{-\kappa\sqrt{w_r}})$) by choosing b_r as $\Theta(r^q)$ with q close to 1. This agrees with our numerical observations summarized in Figures 5 and 6. That being said, we note that an early termination routine (see Section EC.3.3 for a complete description), when implemented, can safeguard cgR-SPLINE from bad instances of b_r that can occur if the budget is increased too fast. As a practical consideration, we caution against implementing the early termination routine without also returning a guarantee on solution feasibility as suggested in Section 6.2. For example, consider a problem context in which the objective function is deterministic and has a negative "gradient" at the stochastic boundary. In such cases, cgR-SPLINE is likely to incorrectly identify a local solution while operating in an infeasible region at low sample sizes, and consequently, have a low confidence on solution feasibility.

Figures 5 and 6 display cgR-SPLINE's performance on the inventory problem and the three stage flowline problem, respectively, for different choices of b_r. Each curve is obtained from 100

independent runs of cgR-SPLINE using common random numbers. In both test cases, cgR-SPLINE appears to display the best convergence rates when b_r is increased almost linearly and with a sufficiently small initial budget of $b_1 = 500$ (solid black curves in Figures 5 and 6).

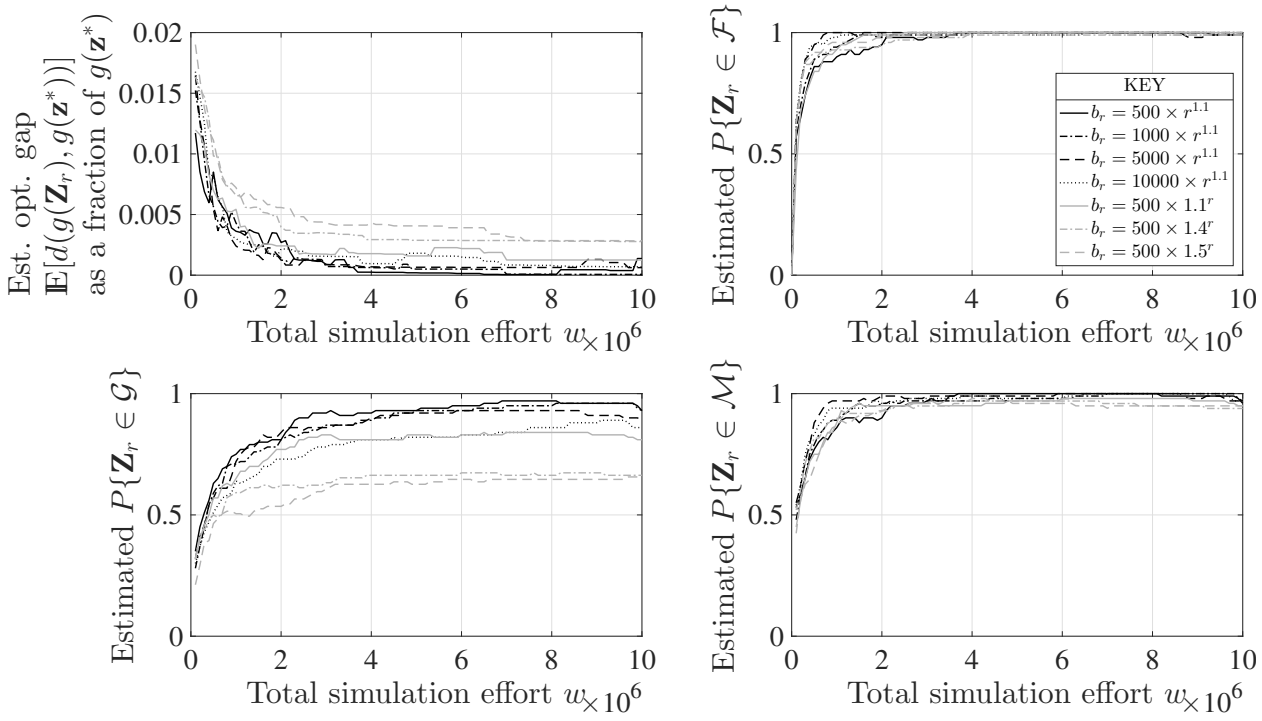

Figure 5 The figures display the performance of cgR-SPLINE on the inventory problem with $|\mathbb{X}| = 1619$. The problem has eleven local solutions, of which the global solution $\boldsymbol{z}^* = (31, 61)$ lies on the stochastic boundary. By setting $\alpha_r = 0$ in the feasibility heuristic, we notice cgR-SPLINE searches closer to the stochastic boundary and returns infeasible points earlier in the search (when the expended budget is under 10^6).

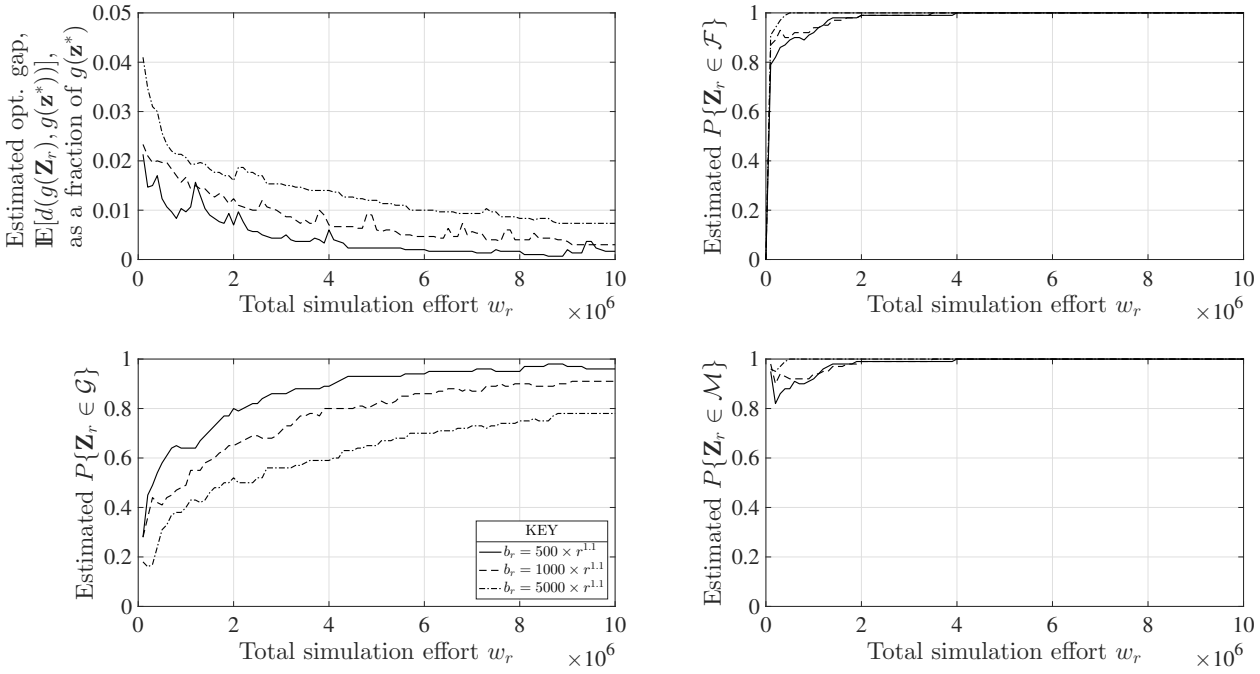

Figure 6 The three-stage flowline problem poses a significant computational challenge to cgR-SPLINE. The size of the search space is nontrivial ($|X| = 41743$), and the problem has 205 local solutions and one global solution $\boldsymbol{z}^* = (10, 10, 10, 10, 10)$ with a binding stochastic constraint. The probability of optimality curves in the bottom left panel, when compared to their counterparts for the inventory problem, seem to reflect the combined influence of problem dimensionality, the number of local solutions relative to the size of the problem domain, and cgR-SPLINE's restart mechanism on its error rate.

EC.3.2. Choice of α_r in the feasibility heuristic

At the end of each outer iteration r, cgR-SPLINE estimates the probability of the candidate local solution $\boldsymbol{X}_r$ being truly feasible. It then reports the solution only if the estimated probability is at least some predetermined amount α_r. Otherwise, cgR-SPLINE returns a neighboring point that satisfies the feasibility threshold. We test cgR-SPLINE on the three-stage flowline problem and the inventory problem with (1) $\alpha_r = 0.95(1 - 0.65^{r+1})$, and (2) $\alpha_r = 0$ for all r.

The first of the two α_r configurations represents a conservative choice in the sense that it directs the local search to regions with a high likelihood of feasibility, especially during earlier restarts. When the optimal point lies on the stochastic boundary, this translates into cgR-SPLINE returning good local solutions from the interior of the search space. On the other hand, by setting $\alpha_r = 0$, cgR-SPLINE "switches off" the feasibility heuristic and, depending on the problem structure, is more likely to return infeasible points earlier in the search. It turns out, cgR-SPLINE's behavior is sensitive to the choice of α_r, as will become evident in the discussion that follows.

Both test problems have binding constraints at their respective global extrema. Not surprisingly then, in each test case, the first choice of α_r returns solutions that are feasible with probability at least 0.97 past a mere 50,000 oracle calls (dashed black lines in the top right panels of Figures 7 and 8). On the other hand, cgR-SPLINE returns the global solution roughly half as often (and well under fifty percent of the time) as under the second configuration (bottom left panels of Figures 7 and 8). This sensitivity to the choice of α_r is not surprising given the significant amount of sampling that would be required to establish feasibility of a boundary solution with high probability.

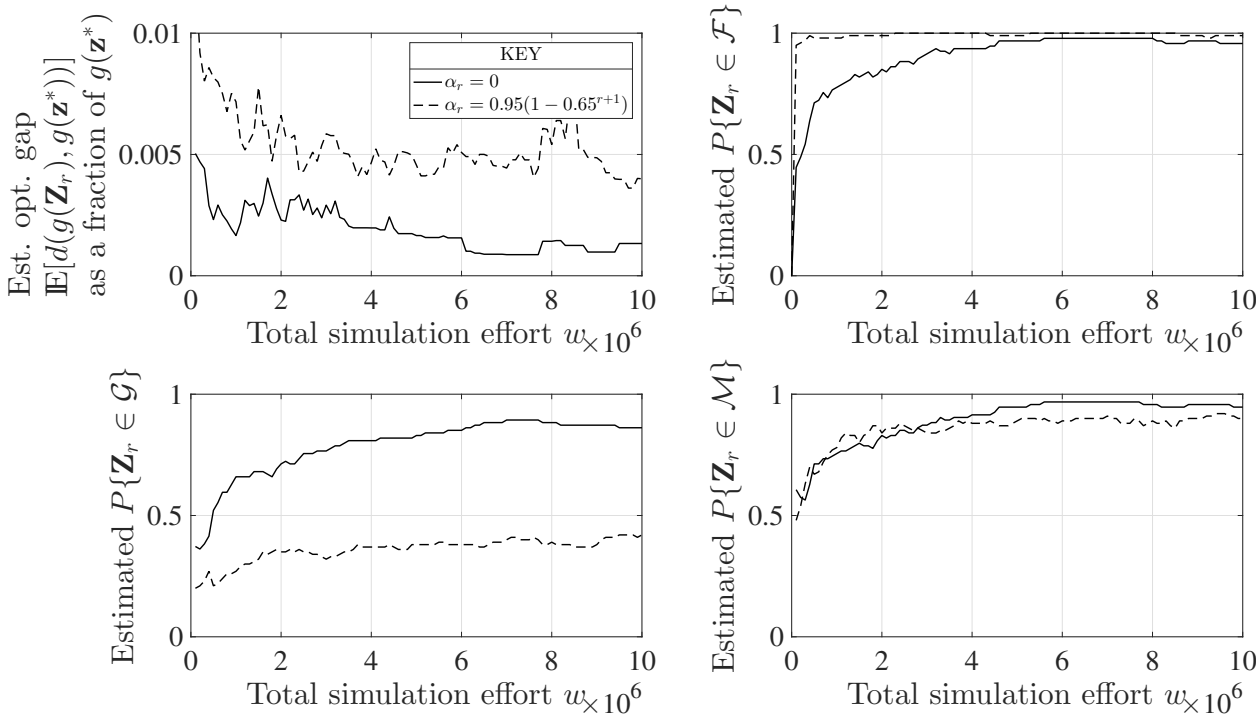

Figure 7 We test two configurations of α_r, (i) $\alpha_r = 0.95 \times (1 - 0.65^{r+1})$, and (ii) $\alpha_r = 0$, on the (s^*, S^*) inventory problem with $|\mathbb{X}|$=1619. In each experiment, $m_k = 8 \times 1.1^k$ and $b_r = 500 \times r^{1.1}$. We notice that the value of α_r does not significantly affect cgR-SPLINE's probability of returning a *local* solution, which is possibly explained by the problem structure: all except one local solution (namely, the global) lie in the interior of the feasible region. We also observe that when $\alpha_r = 0$, cgR-SPLINE returns infeasible solutions with better objective values earlier in the search due to a negative gradient on the objective function near the stochastic boundary. This explains the small optimality gap depicted by solid black curves in the top left panel.

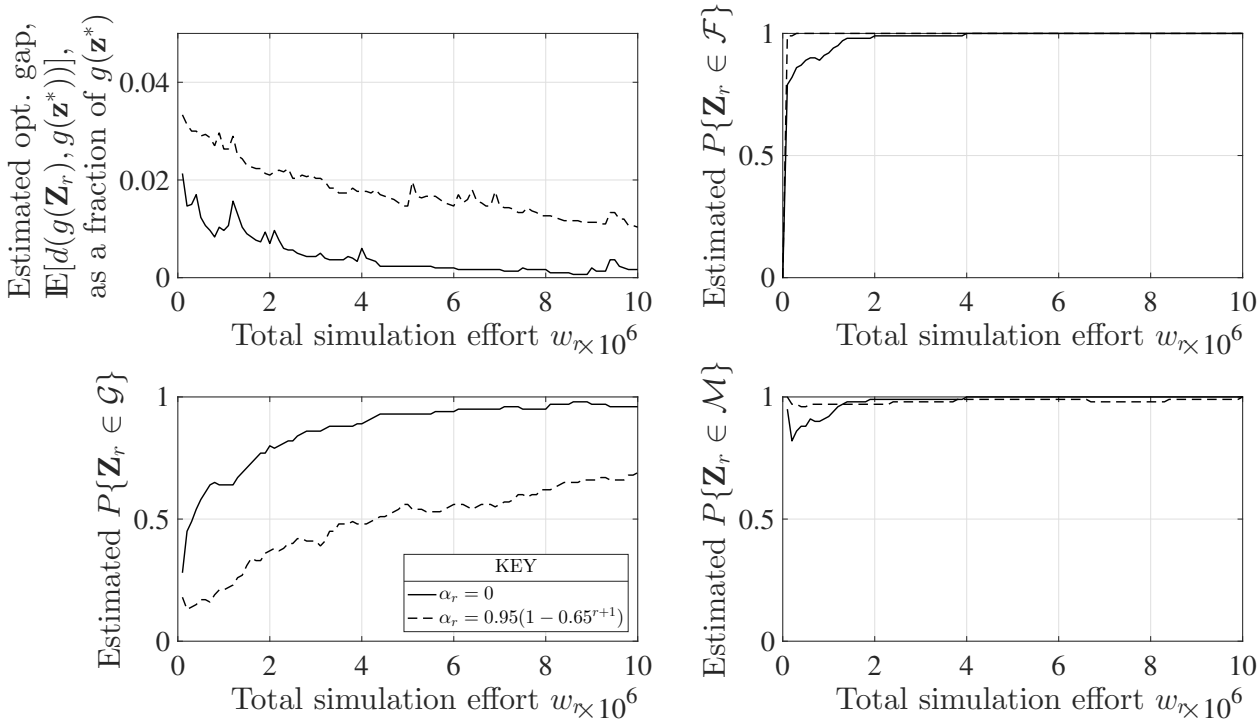

Figure 8 Figures illustrate the effect of the feasibility heuristic on the three-stage flowline problem. In each of the two experiments, $m_k = 8 \times 1.1^k$ and $b_r = 500 \times r^{1.1}$.

EC.3.3. An early termination heuristic

Oftentimes, the locally minimizing SO algorithm (Steps 4 – 10 in Algorithm 1) identifies a local minimum well before the restart budget is expended. This is particularly true when the budget b_r allocated to restart r is very large. Specifically, after some k' inner iterations, we find $\boldsymbol{W}_{k',r} = \boldsymbol{W}_{k'+1,r} = \boldsymbol{W}_{k'+2,r} = \ldots = \boldsymbol{W}_{k_r,r} = \boldsymbol{x}^*$, where $\boldsymbol{x}^*$ is some local solution. One way to guard against what is possibly a poor choice of the outer budget sequence is to prematurely terminate the outer iteration. Building on ideas for adaptive sampling by Hashemi et al. (2014), we propose the following.

After identifying a local solution $\boldsymbol{W}_{k,r}$ at the end of the kth inner iteration of the rth restart (Step 8 in Algorithm 1), cgR-SPLINE obtains objective and constraint function estimates at all points in the neighborhood of $\boldsymbol{W}_{k,r}$. It then terminates the outer iteration if $\boldsymbol{W}_{k,r}$ is deemed sample-path locally optimal with a large enough probability. Formally, the outer iteration r is terminated after k inner iterations if, for all $\boldsymbol{x} \in N(\boldsymbol{W}_{k,r}) \cap \mathcal{F}(m_{k_r}, \epsilon_{k_r})$, we observe

$$\pi(p, m_{k_r})\hat{\text{se}}(\hat{g}_{m_{k_r}}(\boldsymbol{W}_{k,r}) - \hat{g}_{m_{k_r}}(\boldsymbol{x})) \leq \|\hat{g}_{m_{k_r}}(\boldsymbol{W}_{k,r}) - \hat{g}_{m_{k_r}}(\boldsymbol{x})\|.$$

The constant $\pi(p, m_{k_r})$ represents the p-quantile of a Student's t distribution with $m_{k_r} - 1$ degrees of freedom. For a finite neighborhood and a fixed value of p, it is possible to show that the above inequality is satisfied infinitely often past a certain sample size if $\boldsymbol{W}_{k,r}$ is truly a local solution.

We test the early termination routine on the inventory problem for three choices of p, namely $0.75, 0.9$, and 1. For smaller values of p, we expect the outer iterations will terminate too early too often. Whereas, when $p = 1$, the above inequality is never satisfied and a restart is terminated only when the outer budget b_r is exhausted.

The curves in Figure 9 seems to indicate that there is no apparent advantage to terminating restarts prematurely. We are uncertain why this might be the case, but suspect the benefits of early termination become evident only when the outer simulation budget is increased much faster. In practice, we recommend setting p to a sufficiently large value.

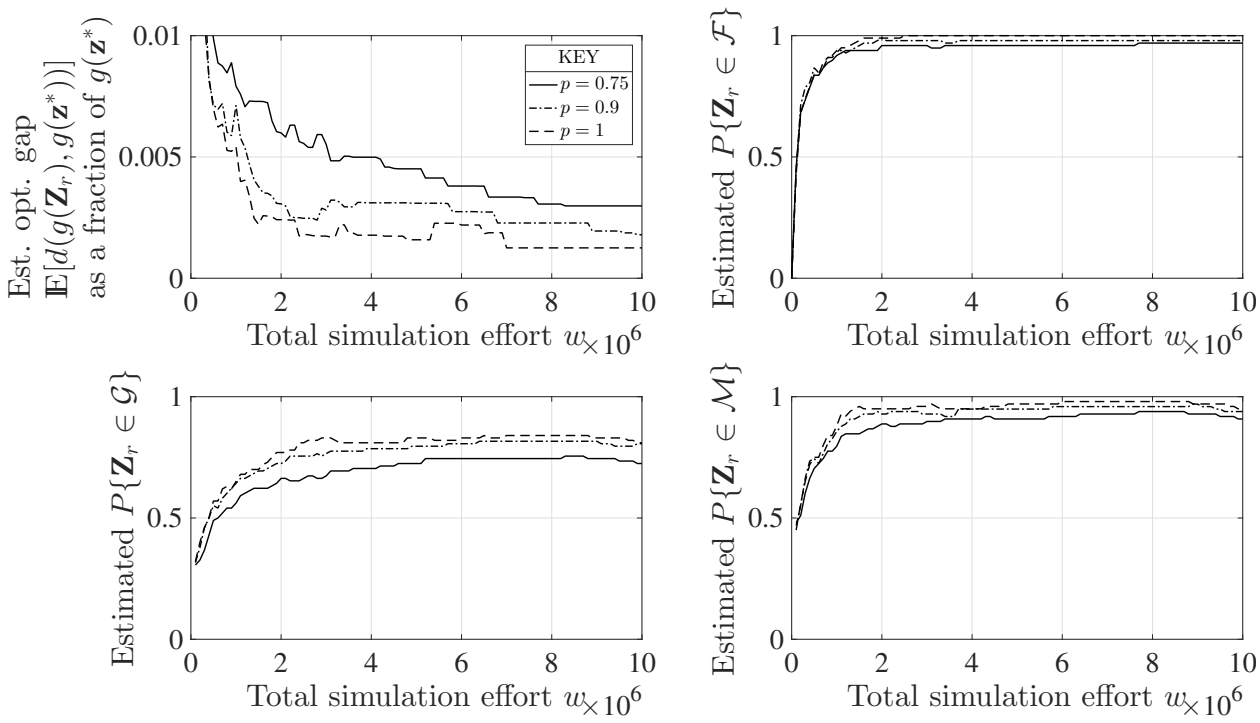

Figure 9 The figure illustrates the effect of the early termination routine with $p = 0.75, 0.9, 1$ on the (s^*, S^*) inventory problem with $|\mathbb{X}|$=1619. For this experiment, we set $m_k = 8 \times 1.1^k$, $b_r = 500 \times 1.1^r$, and $\alpha_r = 0$.