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

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

Kalyani Nagaraj
Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061, kalyanin@vt.edu

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

We consider the problem of identifying the solution(s) to an optimization problem 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 multistart algorithm that repeatedly executes a gradient-based 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 seems to be 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 multistarts is proportional to simulation budget per multistart; 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 are important to ensure that cgR-SPLINE exhibits good finite-time performance while retaining asymptotic properties. We demonstrate cgR-SPLINE using three examples, two of which are nontrivial.

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

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

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 $\hat{h}_{i,m}(\boldsymbol{x})$, $1 \leqslant i \leqslant c$. We assume that $\hat{g}_m(\boldsymbol{x})$ and $\hat{h}_{i,m}(\boldsymbol{x})$, $1 \leqslant i \leqslant c$ 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, $1 \leqslant i \leqslant c$, 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 $\boldsymbol{F}_j(\boldsymbol{x}) := (G_j(\boldsymbol{x}), H_{1,j}(\boldsymbol{x}), H_{2,j}(\boldsymbol{x}), \ldots, H_{c,j}(\boldsymbol{x}))$, $j = 1, 2, \ldots,$ of the random vector $\boldsymbol{F}(\boldsymbol{x}) := (G(\boldsymbol{x}), H_1(\boldsymbol{x}), H_2(\boldsymbol{x}), \ldots, H_c(\boldsymbol{x}))$. 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.

Remark 2. The phrase “canonical rate” is used throughout the paper to refer to the fastest achievable convergence rate by an algorithm, under generic Monte Carlo sampling. For the current context, we will say that an algorithm to solve Problem P achieves the canonical rate if the expected optimality gap decays as $O\left(\exp\{-c\sqrt{w}\}\right)$, where w is the total amount of simulation effort expended and $c > 0$ is some constant. A loose sense of why this is the canonical rate follows from a “back of the envelope” calculation: if $r(w)$ exploration steps are performed and each exploration step is allotted an exploitation budget $w/r(w)$, a balancing of the errors due to exploration and exploitation suggests that $r(w) \approx w/r(w)$, giving $r(w) \approx \sqrt{w}$ and a resulting sub-exponential error decay $O\left(\exp\{-c\sqrt{w}\}\right)$. Compare this with the well-known corresponding rates $O(\exp\{-cw\})$ and $O\left(1/\sqrt{w}\right)$ for pure stochastic ordering and local SO on continuous sets respectively. Later in the paper, we establish that cgR-SPLINE achieves a rate that is arbitrarily close to $O\left(\exp\{-c\sqrt{w}\}\right)$.

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) Gaussian 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,

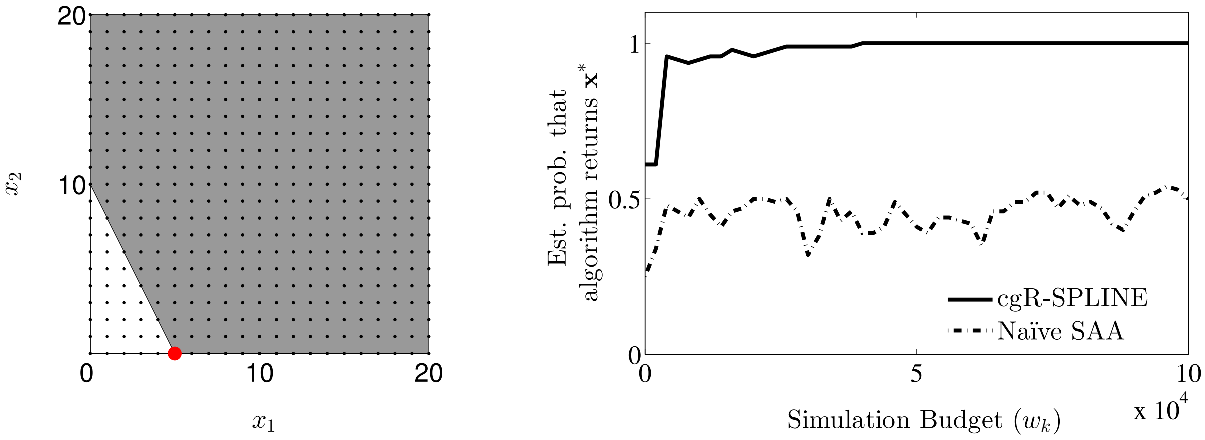


Figure 1 Illustration of a typical complication in a stochastically-constrained SO problem when the solution lies on the boundary of the feasible region. The shaded region on the left represents the feasible region of a stochastically-constrained SO problem and the solid dot at $\boldsymbol{x}^* = (5,0)$ 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 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.

first note that the sampled estimator $\hat{h}_m(\cdot)$ satisfies $\frac{\sqrt{m}}{\sigma}(\hat{h}_m(\boldsymbol{x}) - h(\boldsymbol{x})) \xrightarrow{d} \mathcal{N}(0,1)$ 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 $\hat{h}_m(\boldsymbol{x}^*) > h(\boldsymbol{x}^*) = 0$, 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 are reflected in a deviation from the canonical 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. COMPETING METHODS

Integer-ordered constrained SO problems of type Problem P seem widely prevalent but 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) 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) 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) 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) and what we propose are illustrative. First, the framework in Park and Kim (2011) 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 multistarts 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 canonical rate. It is likely that similar rates are obtainable in Park and Kim (2011) by appropriately setting the geometric sequence of penalties; identifying such sequences in Park and Kim (2011) 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. SUMMARY

We propose a multistart local SO algorithm called cgR-SPLINE to solve Problem P. cgR-SPLINE consists of "outer" iterations during each of which an outer-approximation to a sample-path problem of P is constructed by relaxing the sample-path constraints. The extent of such relaxation is guided by the estimated sample variance of the constraint functions. A locally minimizing SO algorithm, started from a carefully generated initial guess, is then executed on the relaxed sample-path problem until a pre-specified simulation budget is expended (or a local minimizer is obtained to specified accuracy). The points returned by the locally minimizing SO algorithm across the outer iterations are sequentially compared to form an incumbent solution sequence that estimates the global minimizer of Problem P. The extent of the constraint relaxations in the outer approximation, the structure of the locally minimizing SO algorithm along with its multistart initial guesses, and the simulation budgets across outer iterations, are carefully chosen to endow cgR-SPLINE with optimal asymptotic properties while ensuring good finite-time performance.

Remark 3. The algorithm listing and much of the discussion in the ensuing sections assumes that the locally minimizing SO algorithm in use is R-SPLINE (Wang et al. 2013) — this motivates the name cgR-SPLINE. As will become evident, however, any other locally minimizing SO algorithm, e.g., COMPASS (Xu et al. 2010, Hong and Nelson 2006) can be used instead of R-SPLINE.

3.1. Overview of Main Results

Theorem 1 and Theorem 2 provide sufficient conditions to guarantee that the sequence of local solution estimators returned by cgR-SPLINE converges into the set of local solutions of Problem P, implicitly guaranteeing that the issue illustrated through Figure 1 and described in Section 1.1 is averted. Important among the sufficient conditions in Theorem 1 and Theorem 2 is the rate at which the constraint relaxations in the outer-approximation "should be pulled in" relative to the precision with which the objective and constraint functions are evaulated at each point. Both Theorem 1 and Theorem 2 are constructed to guide implementation since they use the estimated standard error of the constraint function estimate to decide the extent to which the constraints should be relaxed in the outer approximation.

Theorem 3 characterizes the asymptotic error probability decay of returning an *incorrect point*, that is, a point that is either infeasible or (locally) suboptimal to Problem P. Theorem 3 demonstrates through a sharp bound that the probability of returning a locally suboptimal point decays sub-exponentially. This result is in contrast to the unconstrained discrete (convex) SO case where the corresponding error probability has been shown to decay exponentially fast with respect to the sample size in use.

While Theorems 1 – 4 relate to local consistency and convergence rates, Theorems 5 – 7 are about global consistency and convergence rates achieved by the incumbent solution sequence in cgR-SPLINE. Theorem 7, for instance, characterizes the rate of convergence of cgR-SPLINE's incumbent solutions to the global minimum in terms of the simulation budgets used in the outer iterations. Theorem 7 is essentially a statement on the exploration-exploitation trade-off in cgR-SPLINE; specifically, in order that cgR-SPLINE achieve the canonical rate, its exploration and exploitation activities should be carefully traded-off by maintaining a square-root relationship between the number of multistarts and the total simulation budget. Analogous prescriptions in the continuous context call for much less exploration. Theorem 5 is the basis of Theorem 7 since it characterizes the rate at which the probability of cgR-SPLINE making an error (when returning a global minimum estimator) decays to zero. Interestingly, this error probability decomposes into two parts corresponding to the sampling error when comparing competing points, and the searching error associated with exploring local minima.

3.1.1. Overview of Implementation Insight The results in Section 6 provide insight on the asymptotic performance of cgR-SPLINE, leading to "boundaries" on algorithm parameters that ensure optimal convergence rates. As is common, however, more is needed for good finite-time performance. Four ideas have proven valuable in this context: (i) using an efficient locally minimizing SO algorithm — we suggest the gradient-based algorithm R-SPLINE (Wang et al. 2013); (ii) using a cost minimization heuristic to trade-off infeasibility against sub-optimality when choosing the constraint relaxations in the outer-approximations; (iii) ensuring that the local solution estimators returned at the end of each outer iteration are feasible with high probability; and (iv) terminating the outer iterations early when a local minimum is attained, or detected to be of poor quality.

As we note in Section 7, the ideas in (ii), (iii), and (iv) are "heuristic," but are used within the confines imposed by the theoretical results of Section 6. This allows cgR-SPLINE to enjoy robust practical performance while retaining guarantees on asymptotic performance. Downloadable C and FORTRAN codes for the algorithms we propose here can be obtained through `http://filebox.vt.edu/users/pasupath/pasupath.htm`.

4. 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 $\boldsymbol{x} := (x_1, x_2, \dots, x_d)$. (ii) By $\Pi_A(x)$ we mean the projection of the point $x \in \mathbb{R}$ into the closed set $A \subset \mathbb{R}$. (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 variables $\{\boldsymbol{X}_n\}$, we say $\boldsymbol{X}_n \xrightarrow{\text{p}} \boldsymbol{X}$ if $\{\boldsymbol{X}_n\}$ converges to $\boldsymbol{X}$ in probability; similarly, we say $\boldsymbol{X}_n \xrightarrow{\text{d}} \boldsymbol{X}$ to mean that $\{\boldsymbol{X}_n\}$ converges to $\boldsymbol{X}$ in distribution, and finally $\boldsymbol{X}_n \xrightarrow{\text{wp1}} \boldsymbol{X}$ to mean that $\{\boldsymbol{X}_n\}$ converges to $\boldsymbol{X}$ with probability one. (v) $\mathbb{Z}^d \subset \mathbb{R}^d$ represents the integer lattice in d-dimensional Euclidean space. (vi) For a sequence of real numbers $\{a_n\}$, we say $a_n = o(1)$ if $\lim_{n\to\infty} a_n = 0$; 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 $0 < \liminf a_n \leq \limsup a_n < \infty$. (vii) For a sequence of random variables $\{X_n\}$, we say $X_n = o_p(1)$ if $X_n \xrightarrow{\text{p}} 0$ as $n \to \infty$; and $X_n = O_p(1)$ if $\{X_n\}$ is stochastically bounded, that is, for given $\epsilon > 0$ there exists $c(\epsilon) \in (0, \infty)$ with $\Pr\{|X_n| < c(\epsilon)\} > 1 - \epsilon$ for large enough n. (viii) The neighborhood $N(\mathbf{0})$ of the d-dimensional origin is defined as any subset of the d-dimensional integer lattice containing the origin. The corresponding neighborhood of any non-zero d-dimensional integer point $\boldsymbol{x}$ is then $N(\boldsymbol{x}) = \{\boldsymbol{y} : (\boldsymbol{y} - \boldsymbol{x}) \in N(\mathbf{0})\}$. (ix) For a given neighborhood definition N and feasible region $\mathbb{X}$, a point $\boldsymbol{x}^* \in \mathbb{X}$ is an *N-local minimizer* of g if the value of the function $g(\cdot)$ at $\boldsymbol{x}^*$ is no larger than at every feasible $\boldsymbol{x}$ in the neighborhood of $\boldsymbol{x}^*$, that is, an *N-local minimizer* is a point in $\mathbb{X}$ that satisfies $g(\boldsymbol{x}) \geq g(\boldsymbol{x}^*),\ \forall \boldsymbol{x} \in N(\boldsymbol{x}^*) \cap \mathbb{X}$.

5. cgR-SPLINE OVERVIEW AND LISTING

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

$$\begin{aligned} P(m, \boldsymbol{\epsilon}):\ \text{minimize}\ & \hat{g}_m(\boldsymbol{x}) \\ \text{subject to}\ & \hat{h}_{i,m}(\boldsymbol{x}) \leqslant \epsilon_i,\ i = 1, \dots, c, \\ & \boldsymbol{x} \in \mathbb{X}. \end{aligned}$$

We see that the sample-path problem $P(m, \boldsymbol{\epsilon})$ is obtained by replacing the objective function g and the constraint functions $h_i, i = 1, 2, \dots, c$ 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 $\boldsymbol{\epsilon} > \mathbf{0}$. We emphasize that, even though we have suppressed explicit notation, the tolerances $\boldsymbol{\epsilon}$ can be dependent on $\boldsymbol{x}$. The feasible, infeasible, and interior regions associated with Problem P and Problem $P(m, \boldsymbol{\epsilon})$ are then $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{X} : \; h_i(\boldsymbol{x}) \leqslant 0, i = 1, \ldots, c\}$, $\mathcal{F}^c = \mathbb{X} \backslash \mathcal{F}$, $\mathcal{F}^\circ = \{\boldsymbol{x} \in \mathbb{X} : \; h_i(\boldsymbol{x}) < 0, i = 1, \ldots, c\}$; and $\mathcal{F}(m, \boldsymbol{\epsilon}) = \{\boldsymbol{x} \in \mathbb{X} : \; \hat{h}_{i,m}(\boldsymbol{x}) \leqslant \epsilon_i, i = 1, \ldots, c\}$, $\mathcal{F}^c(m, \boldsymbol{\epsilon}) = \mathbb{X} \backslash \mathcal{F}(m, \boldsymbol{\epsilon})$ and $\mathcal{F}^\circ(m, \boldsymbol{\epsilon}) = \{\boldsymbol{x} \in \mathbb{X} : \; \hat{h}_{i,m}(\boldsymbol{x}) < \epsilon_i, i = 1, \ldots, c\}$ respectively. Also, let $M_N^* = \{\boldsymbol{x}^* \in \mathcal{F} : \; g(\boldsymbol{x}^*) \leqslant g(\boldsymbol{x}), \; \forall \boldsymbol{x} \in N(\boldsymbol{x}^*) \cap \mathcal{F}\}$ and $M_N^*(m, \boldsymbol{\epsilon}) = \{\boldsymbol{x}^* \in \mathcal{F}(m, \boldsymbol{\epsilon}) : \hat{g}_m(\boldsymbol{x}^*) \leqslant \hat{g}_m(\boldsymbol{x}^*), \; \forall \boldsymbol{x} \in N(\boldsymbol{x}^*) \cap \mathcal{F}(m, \boldsymbol{\epsilon})\}$ denote the set of N-local minima of Problem P and Problem $P(m, \boldsymbol{\epsilon})$ respectively. (Henceforth, we will replace M_N^* and $M_N^*(m, \boldsymbol{\epsilon})$ with M^* and $M^*(m, \boldsymbol{\epsilon})$ respectively for ease of exposition.)

cgR-SPLINE, listed in Figure 2, has a straightforward iterative structure organized into what we call *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 Figure 2) on Problem $P(m_k, \boldsymbol{\epsilon}_k)$, with an appropriately generated initial guess $\boldsymbol{X}_r$. The locally minimizing SO algorithm executes until a specified outer simulation budget b_r is expended, or until a local solution is identified with prespecified precision. At the end of each outer iteration r, the newly estimated local solution $\boldsymbol{Y}_r$ is probabilistically compared (Step 13 in Figure 2) 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 $\{\boldsymbol{Z}_r\}$ form the local and global solution estimators returned by cgR-SPLINE.

Remark 4. cgR-SPLINE as listed in Figure 2 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 Figure 2; it is for this reason that we qualify iterations in cgR-SPLINE as *inner* and *outer* iterations. Although any locally minimizing SO algorithm can replace R-SPLINE in Steps 4–10 of Figure 2, 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.

Key Notation

$\boldsymbol{X}_r$	: multistart initial guess for rth outer iteration
b_r	: simulation budget for rth outer iteration
$\boldsymbol{Y}_r$	: local solution returned after the rth outer iteration
$\boldsymbol{Z}_r$	: incumbent (estimated global) solution after the rth outer iteration
$\boldsymbol{\epsilon}_k$	: constraint relaxation vector used during the kth inner iteraion
m_k	: sample size used during the kth inner iteraion
$\boldsymbol{W}_{k,r}$	: solution returned at the end of the kth inner iteration of the rth outer iteration

Algorithm cgR-SPLINE

Require: budgets $\{b_r\}$; relaxations $\{\boldsymbol{\epsilon}_k\}$; sample sizes $\{m_k\}$; multistarts $\{\boldsymbol{X}_r\}$

Ensure: incumbent solutions $\{\boldsymbol{Z}_r\}$

```
1:  Initialize: outer iteration number $r = 1$, incumbent solution $\boldsymbol{Z}_0 = \boldsymbol{x}_0$, incumbent sample size $t_0 = 0$
2:  Choose: the next (outer) simulation budget $b_r$
3:  Generate: the next multistart initial guess $\boldsymbol{X}_r$.
4:  Initialize: budget utilized $u_0 = 0$, inner iteration number $k = 0$, and warm start $\boldsymbol{W}_{0,r} = \boldsymbol{X}_r$
5:  while $u_k \leqslant b_r$ {budget utilized does not exceed outer simulation budget}
6:      Update: inner iteration $k = k + 1$
7:      Choose: the next sample size $m_k$ and constraint relaxation vector $\boldsymbol{\epsilon}_k$
8:      Solve Problem $P(m_k, \boldsymbol{\epsilon}_k)$: $(N_k, \boldsymbol{W}_{k,r}) = \text{SPLINE}(\boldsymbol{W}_{k-1,r}, m_k)$
9:      Update: utilized budget $u_k = u_{k-1} + N_k$
10: end while
11: Update: returned local solution $\boldsymbol{Y}_r = \boldsymbol{W}_{k,r}$
12: Update: incumbent sample size $t_r = \max\{t_{r-1}, m_k\}$
13: Update: incumbent solution $\boldsymbol{Z}_r = \arg\max\{\hat{g}_{t_r}(\boldsymbol{x}) : \boldsymbol{x} \in \{\boldsymbol{Z}_{r-1}, \boldsymbol{Y}_k\} \cap \mathcal{F}_{t_r}\}$
14: Update: outer iteration $r = r + 1$
15: Go to Step 2
```

Locally minimizing SO algorithm

Inner iterations

Outer iterations

Figure 2 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 sample-path problem. While the above listing of cgR-SPLINE uses the logic of R-SPLINE (Steps 4 – 10) for a locally minimizing algorithm, any other locally minimizing SO algorithm can be used instead. The local extrema estimators obtained across outer iterations are appropriately compared to yield a sequence of global estimators that we call incumbent solutions.

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 in Step 6 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 (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 multistarts 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 multistart. Third, the multistarts, 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 multistarts 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 7 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.

6. MAIN RESULTS

Recall the sets $\mathcal{F}$ and $\mathcal{F}(m,\boldsymbol{\epsilon})$ defined in Section 5. We start with a set of lemmas that describe how the sequence $\{\mathcal{F}(m_k,\boldsymbol{\epsilon}_k)\}$ of sample-path feasible sets 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 the Lemmas 1 – 3 provide 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. $0 < \sigma_* = \inf_{\boldsymbol{x}\in\mathbf{X}}\{\sigma_i(\boldsymbol{x}) : i = 1,2,\ldots,c\} \le \sup_{\boldsymbol{x}\in\mathbf{X}}\{\sigma_i(\boldsymbol{x}) : i = 1,2,\ldots,c\} = \sigma^* < \infty$, *where* $\sigma_i(\boldsymbol{x})$ *is the standard deviation of the random outcomes* $H_{i,l}(\boldsymbol{x})$, $l = 1, 2, \ldots$.

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

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

Assumption 4. *The random vector $\boldsymbol{H}(\boldsymbol{x})$ has a moment-generating function that exists for all $\boldsymbol{x} \in \mathbb{X}$ in some neighborhood of a point $t = 0$.*

Assumption 1 is easily justified if the domain $\mathbb{X}$ is bounded. If $\mathbb{X}$ is unbounded, however, nonpathological 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 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 $\beta > 0$ and all $i = 1, 2, \ldots, c$, then $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k) \overset{wp1}{\to} \mathcal{F}$ 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 $i = 1, 2, \ldots, c$. 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 all $i = 1, 2, \ldots, c$, then $\mathcal{F}(m_k, \boldsymbol{\epsilon}_k) \overset{wp1}{\to} \mathcal{F}$ as $k \to \infty$.*

Lemma 1 notes that the sample-path feasible set converges 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 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:*

The constraint relaxation sequence $\{\epsilon_{i,k}(\boldsymbol{x})\}$ *satisfies*

$$\epsilon_{i,k}(\boldsymbol{x}) = S_{i,k}(\boldsymbol{x}) m_k^{-\delta}, \tag{C.1}$$

where $0 < \delta < 1/2$, $S_{i,k}(\boldsymbol{x}) \in [s_l(\boldsymbol{x}), s_u(\boldsymbol{x})]$, $0 < s_l \leqslant \inf_{\boldsymbol{x}\in\mathbf{X}} s_l(\boldsymbol{x}) < \sup_{\boldsymbol{x}\in\mathbf{X}} s_u(\boldsymbol{x}) \leqslant s_u < \infty$, *and* $S_{i,k}(\boldsymbol{x})$ *is independent of* $\hat{h}_{i,m_k}(\boldsymbol{x})$ *for each* $\boldsymbol{x}$ *wp1;*

The sequence of sample sizes $\{m_k\}$ *satisfies*

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

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

Proof of Lemma 3. Pick $\boldsymbol{x} \in \mathcal{F}$. Then $h_i(\boldsymbol{x}) \leqslant 0$, $i = 1, \ldots, c$, and for large enough k and any $t_i > 0$,

$$\Pr\{\boldsymbol{x} \notin \mathcal{F}^c(m_k, \boldsymbol{\epsilon}_k)\} = \Pr\left\{\bigcup_{i=1}^{c}\left(\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}(\boldsymbol{x})\right)\right\} \leqslant \sum_{i=1}^{c} e^{-(s_l m_k^{-\delta} - h_i(\boldsymbol{x}))t_i} M_{H_i(\boldsymbol{x})}^{m_k}\left(\frac{t_i}{m_k}\right), \tag{1}$$

where $M_{H_i(\boldsymbol{x})}(\cdot)$ is the moment generating function of $H_i(\boldsymbol{x}) - h_i(\boldsymbol{x})$. Minimizing the right hand side of (1) with respect to $(t_1, t_2, \ldots, t_c)$ suggests $t_i^* = \frac{m_k}{\sigma_i^2(\boldsymbol{x})}\left(s_l m_k^{-\delta} - h_i(\boldsymbol{x})\right)$, $1 \leqslant i \leqslant c$. Since $M_{H_i(\boldsymbol{x})}^{m_k}(t_i/m_k) = \left(1 + t_i^2 m_k^{-2} M_{H_i(\boldsymbol{x})}^{(2)}(\xi)/2\right)^{m_k} \to 1$ as $k \to \infty$ for some $\xi \in (0, t_i/m_k)$, we get

$$\Pr\{\boldsymbol{x} \notin \mathcal{F}^c(m_k, \boldsymbol{\epsilon}_k)\} = O\left(e^{-c' m_k^{1-2\delta}}\right) \tag{2}$$

for some $c' > 0$ that is independent of $\boldsymbol{x}$.

Now suppose $\boldsymbol{x} \in \mathcal{F}^c$. Then $h_j(\boldsymbol{x}) > 0$ for some $j \in \{1, \ldots, c\}$. Then for large enough k, $s_u m_k^{-\delta} < h_*$. So for some $c''(h_*) > 0$ we get

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

The result then follows from the application of the Borel-Cantelli lemma (Billingsley 1995). □

The choice $\epsilon_{i,k}(\boldsymbol{x}) = S_{i,k}(\boldsymbol{x}) m_k^{-\delta}, \delta \in (0, 1/2)$ considered in Lemma 3 is of 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 $[s_l(\boldsymbol{x}), s_u(\boldsymbol{x})]$ 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 $s_l(\boldsymbol{x})$ and $s_u(\boldsymbol{x})$, the interval may be of little relevance during implementation. Interestingly, setting $[s_l(\boldsymbol{x}), s_u(\boldsymbol{x})] = (-\infty, \infty)$ seems to result in a much slower error decay rate owing to the tail behavior of $S_{i,k}(\boldsymbol{x})$. 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.

6.1. 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 of true local minima as $k \to \infty$. 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.

Theorem 1. *Let Assumptions 1-4 and conditions C.1, C.2 (listed in Lemma 3) hold. Also, suppose that $\mathcal{F}$ is finite. Then the sequence of local estimators $\{\boldsymbol{Y}_r\}$ converges almost surely to the set M^* of local solutions of Problem P.*

Proof. For ease of exposition, we introduce notation for the 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 the number of steps executed by the solver during the jth inner iteration of the rth outer iteration. Since $U_{j,r}$ is uniformly bounded and $b_r, m_k \in (0, \infty)$ 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 the conditions C.1, C.2 hold, we know that Lemma 3 holds and $\Pr\{\mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r}) \neq \mathcal{F}\ i.o.\} = 0$ as $r \to \infty$. Then since $\mathcal{F}$ is bounded, the sequence $\{\boldsymbol{Y}_r\}$ of local solutions returned by cgR-SPLINE remains bounded with probability 1.

Let $\lambda = \min\{|g(\boldsymbol{x}) - g(\boldsymbol{y})| : (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{F}, g(\boldsymbol{x}) \neq g(\boldsymbol{y})\}$. 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 $K_2(\lambda) \in \mathbb{N}$ such that $|\hat{g}_{m_{k_r}}(\boldsymbol{x}) - g(\boldsymbol{x})| < \lambda/2$ wp1 if $r \geqslant K_2$ for all $\boldsymbol{x} \in \mathcal{F}$. So if $g(\boldsymbol{y}) < g(\boldsymbol{x})$ ($g(\boldsymbol{y}) \geqslant g(\boldsymbol{x})$), then $\hat{g}_{m_{k_r}}(\boldsymbol{y}) < \hat{g}_{m_{k_r}}(\boldsymbol{x})$ ($\hat{g}_{m_{k_r}}(\boldsymbol{y}) \geqslant \hat{g}_{m_{k_r}}(\boldsymbol{x})$) wp1 for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{F}$ if $r \geqslant K_2$. Thus wp1, $\Pr\{\boldsymbol{Y}_r \notin M^*\ i.o.\} = 0$ as $r \to \infty$. □

Theorem 1 guarantees that the sequence of local solution estimators returned by cgR-SPLINE falls 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 $\mathscr{L}_m(\boldsymbol{x})$ 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 5. *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 6. *For* $\boldsymbol{x} \in \mathbb{X}$ *there exists* $\lambda > 0$ *such that* $\mathscr{L}(\boldsymbol{x}, \lambda) = \{\boldsymbol{y} \in \mathbb{X} : g(\boldsymbol{y}) \leqslant g(\boldsymbol{x}) + \lambda\}$ *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 – 6 and conditions C.1, C.2 hold. Then cgR-SPLINE returns a sequence of sample path solutions* $\{Y_r\}$ *that converges wp1 to the set* M^* *of local minima of Problem P.*

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 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."

Theorem 3. *Let $\mathbb{X}$ be finite and suppose that Assumptions 1 – 5 and conditions C.1, C.2 hold. Also, recall the notation k_r, defined in (4), of the number of inner iterations executed by the solver during the rth outer iteration. Then the following hold for some $c' > 0$ as $r \to \infty$.*

(i) The probability that cgR-SPLINE returns an infeasible solution satisfies $Pr\{\boldsymbol{Y}_r \notin \mathcal{F}\} = O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right)$.

(ii) The probability that cgR-SPLINE returns a locally suboptimal but feasible solution satisfies $Pr\{\boldsymbol{Y}_r \notin M^* | \boldsymbol{Y}_r \in \mathcal{F}\} = O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right)$.

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

$$\begin{aligned}\Pr\{\boldsymbol{Y}_r \notin \mathcal{F}\} &= \Pr\{\boldsymbol{Y}_r \notin \mathcal{F}, \mathcal{F} \nsubseteq \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} + \Pr\{\boldsymbol{Y}_r \notin \mathcal{F}, \mathcal{F} \subseteq \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} \\ &\leqslant \sum_{\boldsymbol{y} \in \mathcal{F}} \Pr\{\boldsymbol{y} \in \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})^c\} + \sum_{\boldsymbol{y} \in \mathcal{F}^c} \Pr\{\boldsymbol{y} \in \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r})\} = O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right) + O_p\left(e^{-c'' m_{k_r}}\right), \end{aligned} \tag{5}$$

where the last equality in (5) follows due to the finiteness of $\mathbb{X}$ and from (2) and (3).

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

$$\begin{aligned}\Pr\{\boldsymbol{Y}_r \notin M^* \,|\, \boldsymbol{Y}_r \in \mathcal{F}\} &= \Pr\left\{\boldsymbol{Y}_r \notin M^*, \left(\underset{\boldsymbol{y} \notin N(\boldsymbol{Y}_r) \bigcap \mathcal{F}}{\arg\min}\; g(\boldsymbol{y})\right) \in \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r}) \,|\, \boldsymbol{Y}_r \in \mathcal{F}\right\} + \\ &\qquad \Pr\left\{\boldsymbol{Y}_r \notin M^*, \left(\underset{\boldsymbol{y} \in N(\boldsymbol{Y}_r) \bigcap \mathcal{F}}{\arg\min}\; g(\boldsymbol{y})\right) \in \mathcal{F}(m_{k_r}, \boldsymbol{\epsilon}_{k_r}) \,|\, \boldsymbol{Y}_r \in \mathcal{F}\right\} \\ &= O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right) + O_p\left(e^{-\eta_g m_{k_r}}\right), \end{aligned} \tag{6}$$

where the last equality of (6) follows from the finiteness of $\mathbb{X}$ and from (2) and Assumption 5 (Gäartner-Ellis Theorem (Dembo and Zeitouni 1998)). □

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* as 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, and the error in correct ordering 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. We 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_* is known, the canonical rate is easily achieved by always choosing $\epsilon_{i,k}(\boldsymbol{x}) < h_*$ in Step 7 in Figure 2.

The next result provides a sense of the rate of decay of the expected cost associated with the local search algorithm returning an incorrect solution for both finite and unbounded feasible regions. We do not provide a proof for Theorem 4 because it follows directly from Theorem 3.

Theorem 4. *Let $\mathbb{X}$ be finite and suppose that Assumptions 1 – 5 and conditions C.1, C.2 hold. Also, let $\ell(\boldsymbol{x})$ be a "loss function" associated with cgR-SPLINE returning the solution $\boldsymbol{Y}_r = \boldsymbol{x} \in \mathbb{X}$ that satisfies $\ell(\boldsymbol{x}) \in (0, \infty)$ for $\boldsymbol{x} \in \mathbb{X}$, and $\ell(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in M^*$. Then the expected loss associated with R-SPLINE returning an incorrect solution satisfies $E[\ell(\boldsymbol{Y}_r)] = O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right)$ for some $c' > 0$.*

Theorems 3 and 4 suggest that the convergence rates associated with cgR-SPLINE increase with decreasing δ. This is because, Theorems 3 and 4 recognize that the asymptotic cost due to returning a feasible but suboptimal solution is negligible compared to returning an infeasible solution. 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. As we discuss in Section 7, robust implementation will dictate (somehow) incorporating the loss due to returning feasible but suboptimal solutions.

6.2. 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 6.1 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 exectued 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 $\boldsymbol{y}^* \in M^*$ we also define the "attractor set" $B(\boldsymbol{y}^*)$ and its "reaching probability" $p_r(\boldsymbol{y}^*)$ as $B(\boldsymbol{y}^*) = \{\boldsymbol{x} \in \mathcal{F} : \Pr\{\boldsymbol{Y}_\infty = \boldsymbol{y}^* \,|\, \boldsymbol{X}_r = \boldsymbol{x}\} > 0\}$ 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 ad infinitum; 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 $\boldsymbol{y}^*$ if executed ad infinitum. 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 multistart guesses $\boldsymbol{X}_r$ are not identically distributed across iterations. Each of the outer iterations is however 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, Borel-Cantelli's theorem (Billingsley 1995) ensures that the returned incumbent solutions reach the global extremum almost surely. We need the following assumption.

Assumption 7. *Let $\mathscr{G} = \{\boldsymbol{z}^* \in \mathcal{F} : g(\boldsymbol{z}^*) \leqslant \boldsymbol{x}$ for all $\boldsymbol{x} \in \mathcal{F}\}$ denote the set of all global solutions to Problem P. Then $\bigcup_{z^* \in \mathscr{G}} B(z^*) \neq \emptyset$ and $\liminf_{r\to\infty} p_r(\mathscr{G}) > 0$, where $p_r(\mathscr{G}) := Pr\{\boldsymbol{X}_r \in \bigcup_{z^* \in \mathscr{G}} B(\boldsymbol{z}^*)\}$.*

The set $\bigcup_{z^* \in \mathscr{G}} B(z^*)$ in Assumption 7 connotes the reaching set for the global extrema of Problem P — when executed (ad infinitum) from any point in $\bigcup_{z^* \in \mathscr{G}} B(z^*)$, the locally minimizing SO algorithm has a positive probability of reaching a global extremum. Good locally minimizing algorithms thus tend to have larger sets $\bigcup_{z^* \in \mathscr{G}} B(z^*)$ and higher values of $\sum_{x \in \mathbb{X}} \Pr\{\boldsymbol{Y}_\infty = \boldsymbol{y}^* \,|\, \boldsymbol{X}_r = \boldsymbol{x}\} > 0\}$.

Theorem 5. *Suppose that $\mathbb{X}$ is finite. Furthermore, suppose that Assumptions 1 – 5, Assumption 7, and conditions C.1, C.2 hold. Then*

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

$$\gamma := \begin{cases} \left(\frac{q}{q+1}\right)(1-2\delta) & \text{if } m_k = \Theta(k^q),\ \text{where } q > 0, \\ 1-2\delta & \text{if } m_k = \Theta\left(a_{in}^k\right),\ \text{where } a_{in} > 1; \end{cases} \tag{7}$$

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

(ii) $Pr\{\mathbf{Z}_r \notin \mathscr{G}\, i.o.\} = 0$ *as* $r \to \infty$ *if for* γ *defined as in (7),*

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

Theorem 5 provides important insight about the convergence characteristics of cgR-SPLINE. Specifically, the result in part (i) of Theorem 5 implies that the error probability $\Pr\{\mathbf{Z}_r \notin \mathscr{G}\}$ 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 $O(\tau^r)$ 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 $\Pr\{\mathbf{Z}_r \notin \mathscr{G}\}$ 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 $O(\tau^r)$ parallels a similar term that is obtained in deterministic multistart methods. Part (ii) of Theorem 5 notes that if the outer budget b_r is increased fast enough, then deviations due to mischance are negligible and $\mathbf{Z}_r$ converges to the solution almost surely.

Under the assumption that $\mathbb{X}$ is finite, Theorem 5 leads to the analogous Theorem 6 which measures error $E_r := \max_{z^* \in \mathscr{G}} \mathrm{E}\left[\|g(\mathbf{Z}_r) - g(z^*)\|\right]$ in the function space. Since the proof of Theorem 6 follows in a straightforward way from the proof of Theorem 5, we do not provide one.

Theorem 6. *Suppose that* $\mathbb{X}$ *is finite. Furthermore, suppose that Assumptions 1 – 5, Assumption 7, and conditions C.1, C.2 hold. Then*

(i) $E_r = O\left(e^{-c' b_r^{\gamma}}\right) + O(\tau^r)$ *for* $\tau \in (0, 1)$*,* γ *defined as in (7), and* $\{b_r\}$ *approaching infinity;*

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

Theorems 5 and 6 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 5, Theorem 7 gives broad insight into the convergences rates that are achievable by cgR-SPLINE.

Theorem 7. *Suppose that $\mathbb{X}$ is finite. Furthermore, suppose that Assumptions 1 – 5, Assumption 7, 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 rth outer iteration of cgR-SPLINE and γ is defined as in (7), then*

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

Proof. When the outer budget $\{b_j\}$ grows as $b_j = \Theta(a_{\text{out}}^j)$, we get from Theorem 6 that $E_r = O\left(e^{-c' b_r^\gamma}\right) + O(\tau^r) = O(\tau^r)$. Also, since $w_r = \sum_{j=1}^r b_j = \Theta(e^{r \log a_{\text{out}}})$, conclude that $E_r = O(e^{-\kappa \log w_r})$ where $\kappa = -a_{\text{out}}^{-1} \log \tau$.

Similarly, when the outer budget $\{b_j\}$ grows as $b_j = \Theta(j^q)$, we get from Theorem 6 that $E_r = O\left(e^{-c' r^{q\gamma}}\right)$ if $q\gamma < 1$ and $E_r = O(\tau^r)$ if $q\gamma > 1$. Also, since $w_r = \sum_{j=1}^r b_j = \Theta(r^{q+1})$, conclude that $E_r = O(w_r^{q\gamma/q+1})$ if $q\gamma < 1$ and $E_r = O(w_r^{1/q+1})$ if $q\gamma \leq 1$. □

Theorem 7 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 (7) and the assertion in Theorem 7, 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 $\Theta(j^q)$. Particularly, cgR-SPLINE can be made to achieve a rate that is aribitrarily close to the canonical rate $O(e^{-\kappa \sqrt{w_r}})$ by choosing $q = 1$ and δ close to zero.

An equivalent "fixed budget" interpretation of Theorem 7 is that for fastest possible convergence, the number of multistarts and the budget per multistart 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 multistarts 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 usuing a locally minimizing algorithm other than R-SPLINE, we expect our insights from an analogous Theorem 7 to be similar since we expect the structure of Theorems 5 and 6 to remain unchanged.

7. 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 6. 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.

7.1. Choosing the Constraint Relaxation and Sample Size Sequences

Recall that the constraint relaxations $\boldsymbol{\epsilon}_{k_r}(\mathbf{x})$ in cgR-SPLINE are chosen as $\boldsymbol{\epsilon}_{k_r}(\boldsymbol{x}) = \hat{\boldsymbol{\sigma}}_{k_r}(\boldsymbol{x})/m_{k_r}^{\delta}$, where $\hat{\boldsymbol{\sigma}}_{k_r}(\boldsymbol{x}) := (\hat{\sigma}_{1,k_r}(\boldsymbol{x}), \hat{\sigma}_{2,k_r}(\boldsymbol{x}), \ldots, \hat{\sigma}_{c,k_r}(\boldsymbol{x}))$ is the point estimator of $(\sigma_1, \sigma_2, \ldots, \sigma_c)$. This choice for $\boldsymbol{\epsilon}_{k_r}(\boldsymbol{x})$ makes sense in that it incorporates the observable quantity $\hat{\boldsymbol{\sigma}}_{k_r}(\boldsymbol{x})$ 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 Theorems 3–4 indicate 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 Theorems 3–4 which recognize 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 feasible and infeasible are c_1 and c_2 respectively. Then the optimization problem in ϵ is posed as $\min_\epsilon \mathrm{E}_{\mu,X}[\mathrm{L}(\mu, \delta(\bar{X}))]$ where $\delta(X) = 1$ if $\bar{X} \leq \epsilon$ and 0 otherwise; and $\mathrm{E}[\mathrm{L}(\mu, \delta(X))] = c_1 \int_{-\infty}^{0} \int_{\epsilon}^{\infty} \mathcal{N}_{\bar{x},\hat{\sigma}^2/n}(\mu)\mathcal{N}_{\mu,\hat{\sigma}^2/m}(y)\, dy\, d\mu + c_2 \int_{0}^{\infty} \int_{-\infty}^{\epsilon} \mathcal{N}_{\bar{x},\hat{\sigma}^2/n}(\mu)\mathcal{N}_{\mu,\hat{\sigma}^2/m}(y)\, dy\, d\mu$. Differentiating $\mathrm{E}[\mathrm{L}(\mu, \delta(X))]$ 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 $\epsilon^*_{i,k}(\boldsymbol{x}) = -\hat{h}_{i,m_k}(\boldsymbol{x})$ at the end of every kth inner iteration, where $\boldsymbol{\epsilon}^*_k(\boldsymbol{x}) = (\epsilon^*_{1,k}(\boldsymbol{x}), \epsilon^*_{2,k}(\boldsymbol{x}), \ldots, \epsilon^*_{c,k}(\boldsymbol{x}))$. We modify this slightly to avoid big fluctuations of the constraint relaxations, and recommend setting

$$\epsilon^*_{i,k}(\boldsymbol{x}) = \min\left(\max\left(\frac{\hat{\sigma}_{i,k}(\boldsymbol{x})}{m_k^{0.45}}, -\hat{h}_{i,m_k}(\boldsymbol{x})\right), \frac{\hat{\sigma}_{i,k}(\boldsymbol{x})}{m_k^{0.1}}\right). \tag{8}$$

This also yields $\boldsymbol{\delta}_{k+1} = (\log(m_k))^{-1}\log(\hat{\boldsymbol{\sigma}}_k(\boldsymbol{x})) - \log(\boldsymbol{\epsilon}^*_k(\boldsymbol{x}))$ where $\boldsymbol{\delta}_k = (\delta_{1,k}, \delta_{2,k}, \ldots, \delta_{c,k})$.

7.2. 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 $\alpha_r \to \alpha < 1$ constraint on the probability of feasibility of the local solutions $\{\boldsymbol{Y}_r\}$ returned after outer iterations. For a given local solution $\boldsymbol{Y}_r = \boldsymbol{y}_r$ returned at the end of the rth outer iteration, the probability $\psi(\boldsymbol{y}_r)$ that $\boldsymbol{y}_r$ is truly feasible can be estimated as

$$\psi(\boldsymbol{y}_r) = \Pr\{\boldsymbol{y}_r \in \mathcal{F} \mid \boldsymbol{Y}_r = \boldsymbol{y}_r\} = \Pr\left\{\bigcap_{i=1}^{c}\left(\hat{h}_{i,m_{k_r}}(\boldsymbol{y}_r) \leqslant \epsilon_{i,k_r}(\boldsymbol{y}_r)\right)\right\} \approx \prod_{i=1}^{c} \Phi\left(m_{k_r}^{\frac{1}{2}-\delta} - \frac{\hat{h}_{i,m_{k_r}}(\boldsymbol{y}_r)}{\hat{\sigma}_{i,k_r}(\boldsymbol{y}_r)}\sqrt{m_{k_r}}\right). \tag{9}$$

Upon identification of a local solution $\boldsymbol{Y}_r$, (9) is used to calculate $\psi(\boldsymbol{y}_r)$. $\boldsymbol{Y}_r = \boldsymbol{y}_r$ is returned as the identified local solution if $\psi(\boldsymbol{y}_r) \geq \alpha_r$ or none of $\boldsymbol{Y}_r$’s neighbors $\boldsymbol{y}$ satisfy $\psi(\boldsymbol{y}) \geq \alpha_r$. Otherwise, one of the neighbors of $\boldsymbol{Y}_r$ which satisfies the specified lower bound is returned instead of $\boldsymbol{Y}_r$.

Since the limit $\alpha = \lim_r \alpha_r < 1$, 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 convenience. For example, all our numerical experiments use $\alpha = 0.95$ and $\alpha_r = \alpha(1 - 1/\log b_r)$.

7.3. Premature Stopping of the Local Search Algorithm

Our numerical experience indicates that, sometimes, the locally minimizing SO algorithm in cgR-SPLINE (Steps 4 – 10 in Figure 2) identifies a local minimum well before the allocated budget b_r is expended, returning the same point $\boldsymbol{W}_{k,r}$ in successive inner iterations that execute with increasing sample sizes. In such cases, an obvious way to gain efficiencies is to prematurely terminate the outer iteration. We operationalize this idea by observing the points $\boldsymbol{W}_{k,r}, k = 1, 2, \ldots$ and terminating the rth outer iteration if we find k such that $\boldsymbol{W}_{k-2,r} = \boldsymbol{W}_{k-1,r} = \boldsymbol{W}_{k,r}$. Upon such termination, the usual steps of assigning $\boldsymbol{W}_{k,r}$ to $\boldsymbol{Y}_r$ (Step 11 in Figure 2) and then probabilistically comparing $\boldsymbol{Y}_r$ against the incumbent $\boldsymbol{Z}_{r-1}$ (Steps 12 and 13 in Figure 2) are followed.

The suggested idea for premature termination is a heuristic that mimics criteria for local convergence in deterministic optimization algorithms. We expect it to be effective when there are only a few local minima and they lie on the interior of the feasible region. Also, unlike implementation ideas outlined in Sections 7.1 and 7.2, premature termination affects convergence guarantees under certain pathological conditions that cause the incumbent sample size t_r (in Step 12 of Figure 2) to remain bounded.

8. NUMERICAL EXPERIMENTS

In this section, we illustrate cgR-SPLINE's performance on two nontrivial examples. In each case, we evaluate the performance of cgR-SPLINE by observing two measures as a function of the expended simulation budget: the estimated expected optimality gap expressed as a percentage of the optimal value, and the estimated probability of the incumbent solution being truly feasible. Both measures are calculated simultaneously using independent runs of cgR-SPLINE.

8.1. The 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, and x_3, and having an infinite number of jobs present in front of the first server. The second and third server have finite buffer capacities x_4 and x_5, respectively. The total buffer space $x_4 + x_5$ is restricted to 20 and the individual service rates cannot exceed 20. The objective is to identify the service rates x_1, x_2, and x_3 and the buffer capacities x_4 and x_5 such that the total service rate is minimized, subject to the steady-state expected throughput being at least 5.776 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 $60,647$ feasible solutions of which 123 are local minima. Of these local minima, two points, z_1^* =(6, 7, 7, 12, 8) and z_2^* =(7, 7, 6, 8, 2), are global minima located on the boundary of the feasible region formed by the steady-state throughput function. The function $h(\boldsymbol{x})$ can only be estimated through sampling, as the average number of jobs leaving the system between times $t = 50$ and $t = 1000$. Initial solutions for the multistarts were generated by sampling uniformly from the region $\mathbb{X} = \{\boldsymbol{x} \in \mathbb{Z}_+^5 : x_4 + x_5 = 20; 1 \leqslant x_i \leqslant 20,\ i = 1,2,3; 1 \leqslant x_i \leqslant 19,\ i = 4,5\}$. The constraint relaxation parameter δ was chosen according to suggestions laid out in Section 7.1.

Table 8.1 displays the output log from a *single run* of cgR-SPLINE. The first seven columns in Table 8.1 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 not available to the user. As is evident from Table 8.1, cgR-SPLINE returns an infeasible solution at the end of the first outer iteration, after expending a simulation budget of 1174. Then, over the next four outer iterations, it settles down at a local minimum which is not global. Upon conclusion of the sixth outer iteration and after expending over $15,000$ oracle calls, cgR-SPLINE seems to identify the global minimum. That cgR-SPLINE visits only two of the 123 local minima is interesting and probably explained by the logic of the SPLINE solver; even though SPLINE is a local solver, its search mechanism takes it to regions having good local minima.

Figure 3 summarizes cgR-SPLINE's performance over 95 independent runs. The solid line in Figure 3, dispalying the probability of cgR-SPLINE returning a feasible solution as a function of the total expended simulation budget, suggests that cgR-SPLINE tends to quickly reach a feasible solution that is in the neighborhood of the global minimum. As one might expect in cases where a solution has a binding stochastic

Table 1 The first seven 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 an infeasible solution as being optimal, then gets caught at a local minimum that is not global, and then successfully identifies the global minimum after the sixth outer iteration.

Restart	Incumbent solution					Total work			Feas/
r	$\mathbf{Z}_r$	$\hat{g}_{m_{k_r}}$	$\hat{h}_{m_{k_r}}$	$\hat{\psi}_r$	α_r	w_r	g	h	Opt
1	(12, 7, 7, 1, 19)	26	0.0485	0.1406	0.5486	1174	26	0.0227	I
2	(7, 8, 6, 4, 16)	21	-0.0609	0.9997	0.6891	2856	21	-0.0375	L
3	(7, 8, 6, 4, 16)	21	-0.0435	0.9998	0.7804	5006	21	-0.0375	L
4	(7, 8, 6, 4, 16)	21	-0.0435	0.9998	0.8398	5015	21	-0.0375	L
5	(7, 8, 6, 4, 16)	21	-0.0378	1.0000	0.8784	9308	21	-0.0375	L
6	(6, 7, 7, 12, 8)	20	-0.0132	0.9984	0.9034	15591	20	0	G
7	(6, 7, 7, 12, 8)	20	-0.0150	0.9987	0.9197	25157	20	0	G
8	(6, 7, 7, 12, 8)	20	-0.0132	0.9998	0.9303	38774	20	0	G
9	(6, 7, 7, 12, 8)	20	-0.0100	0.9998	0.9372	58403	20	0	G
10	(6, 7, 7, 12, 8)	20	-0.0052	0.9994	0.9417	88265	20	0	G
11	(6, 7, 7, 12, 8)	20	0.0002	0.9641	0.9446	133505	20	0	G
12	(6, 7, 7, 12, 8)	20	-0.0044	0.9978	0.9465	203685	20	0	G

constraint, much of the simulation effort seems to be expended in deciding amongst a few points around the true solution. Notice also from the dashed line in Figure 3 that, almost always, the relative objective difference diminishes to less than 0.1 in under $10,000$ simulation oracle calls.

8.2. The Bus Scheduling Problem

Passengers arrive at a bus depot between times $t = 0$ and $t = \tau$ according to a homogeneous Poisson process with rate λ. Buses are allowed to depart at time instants 5, 10, 15, …, $5(\lfloor\tau/5\rfloor - 1)$, and $5\lfloor\tau/5\rfloor$. Each bus is assumed to have a large enough capacity to seat all waiting passengers. Additionally, it is assumed that a departure is scheduled at the beginning of the day ($t = 0$) and at the end of the day ($t = \tau$). We define a binary decision variable $\boldsymbol{x} = (x_1, x_2, \ldots, x_{\lfloor\tau/5\rfloor-1})$, where x_i equals 1 if a bus departs from the depot at time $5i$, and 0 otherwise, $i = 1, \ldots, \lfloor\tau/5\rfloor - 1$. The objective is to minimize the fleet size $g(\boldsymbol{x}) = \sum_{i=1}^{\lfloor\tau/5\rfloor-1} x_i$ such

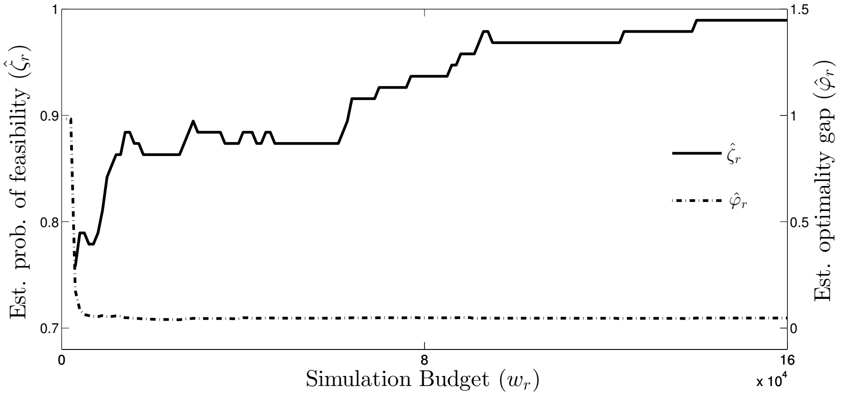
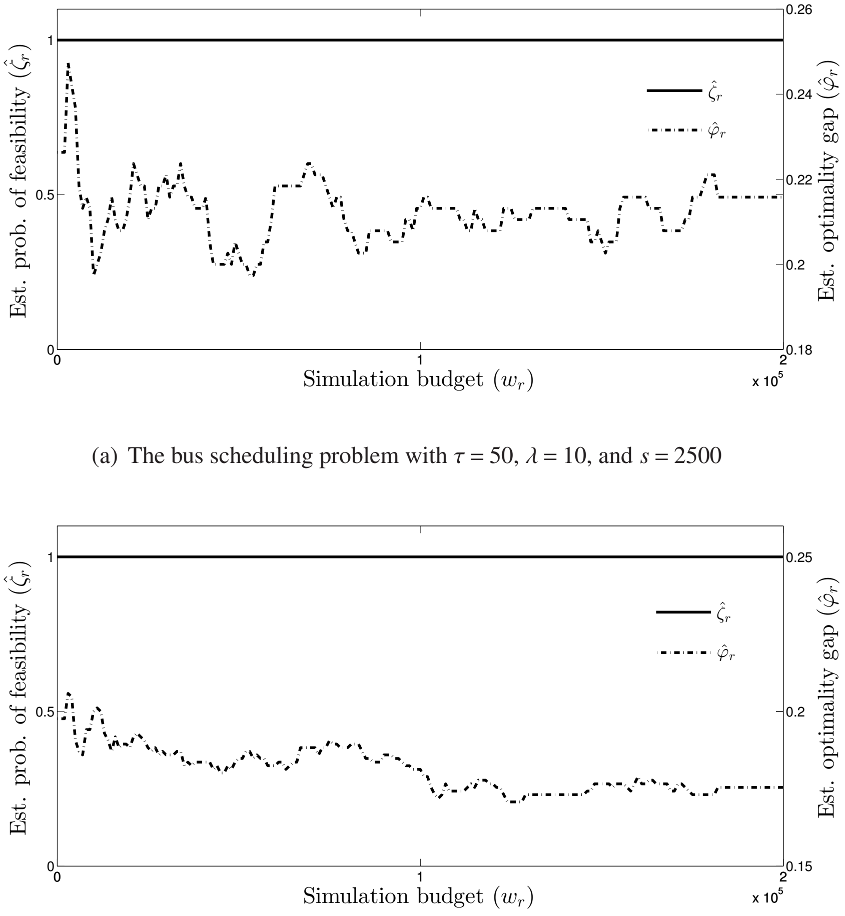


Figure 3 The figure displays the performance of cgR-SPLINE on the three-stage flowline problem. The solid curve plots the estimated probability of feasibility, and the dashed curve the estimated expected relative objective difference, of the sequence of incumbent solutions returned by cgR-SPLINE. The curves were generated from ninety five independent runs of cgR-SPLINE.

that the expected total wait time $h(\boldsymbol{x})$ of all passengers does not exceed s. Given a bus schedule $\boldsymbol{x}$, a Monte Carlo simulation of the arrival and the waiting time process is used to return an estimate of the constraint function $h(\boldsymbol{x})$.

The settings $\tau = 50$, $\lambda = 10$, $s = 2500$, produce a 9-dimensional problem with domain $\mathbb{X}$ and feasible region $\mathcal{F}$ having cardinalities $|\mathbb{X}| = 512$, $|\mathcal{F}| = 199$, 123 local minima, and a unique global minimum $z^* = (0,1,0,1,0,1,0,1,0)$ satisfying $g(z^*) = 4$ and $h(z^*) = 2500$. Setting $\tau = 100$ and $s = 5000$ results 19-dimensional problem having $|\mathbb{X}| = 524,288$, $|\mathcal{F}| = 97,462$, and $27,113$ local minima, of which $26,676$ lie on the boundary $h(\boldsymbol{x}) = 0$. This problem also has a unique optimal solution $z^* = (0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0)$ having $g(z^*) = 9$ and residing on the stochastic boundary of the feasible region.

Figures 4(a) and 4(b) illustrate the performance of cgR-SPLINE for the 9-dimensional and the 19-dimensional bus scheduling problems by summarizing the feasibility and the optimality gap as a function of the expended simulation budget in cgR-SPLINE. Due to the heuristic on returning only solutions that are estimated to be feasible with a high probability, cgR-SPLINE seems to return only truly feasible points. The performance difference between the 9-dimensional and the 19-dimensional problem manifests itself in



(a) The bus scheduling problem with $\tau = 50$, $\lambda = 10$, and $s = 2500$

(b) The bus scheduling problem with $\tau = 100$, $\lambda = 10$, and $s = 5000$

Figure 4 The figure illustrates the peformance of cgR-SPLINE on two versions of the bus scheduling problem. The curves depict, as a function of the total simulation budget w_r, the average relative objective difference and the estimated probability of feasibility of the incumbent solution computed across ninety five independent runs of cgR-SPLINE.

the transient period, with cgR-SPLINE taking a markedly longer time to reach the vicinity of the solution in higher dimensional space.

In general, and as in the three-stage flowline problem, cgR-SPLINE appears to have little difficulty reaching the vicinity of the true solution, but spends a lot of simulation effort trying to identify the best amongst a select few points in the neighborhood of the true solution. We believe that such behavior is desirable and reflective of the fact that the algorithm'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.

9. 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 multistarts 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 infeasibility with suboptimality 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 is 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 multistarts, seems amenable to parallelization. If individual processors are tasked with executing a multistart, the premature termination of a multistart based on the quality of the observed incumbents across processors is an interesting issue that is currently being investigated.

Appendix. Proofs of Lemmas and Theorems

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

$$\begin{aligned}
\Pr\{\boldsymbol{x} \notin \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcup_{i=1}^{c}\left(\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}\right)\right\} \\
&\leqslant \sum_{i=1}^{c} \Pr\{\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}\} \\
&\leqslant \sum_{i=1}^{c} \Pr\{\hat{h}_{i,m_k}(\boldsymbol{x}) \notin (h_i(\boldsymbol{x}) - \epsilon_{i,k}, h_i(\boldsymbol{x}) + \epsilon_{i,k})\} \\
&\leqslant \sum_{i=1}^{c} \frac{\mathrm{Var}\left(\hat{h}_{i,m_k}(\boldsymbol{x})\right)}{\epsilon_{i,k}^2} \\
&= \sum_{i=1}^{c} \frac{\sigma_i^2(x)}{m_k \epsilon_{i,k}^2} \qquad (10)
\end{aligned}$$

where the third inequality of (10) follows from Chebyshev's inequality. Then under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the Borel-Cantelli lemma, there exists K_1 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 $\gamma > 0$ such that $2\gamma < \inf_{\boldsymbol{x} \in \mathcal{F}^c}\{h_i(\boldsymbol{x}) : h_i(\boldsymbol{x}) > 0, i = 1, \ldots, c\}$. 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

$$\begin{aligned}
\Pr\{\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcap_{i=1}^{c}\left(\hat{h}_{i,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}\right)\right\} \\
&\leqslant \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}\} \\
&= \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k},\ \hat{h}_{j,m_k}(\boldsymbol{x}) \in (h_j(\boldsymbol{x}) - \gamma, h_j(\boldsymbol{x}) + \gamma)\} \\
&\quad + \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k},\ \hat{h}_{j,m_k}(\boldsymbol{x}) \notin (h_j(\boldsymbol{x}) - \gamma, h_j(\boldsymbol{x}) + \gamma)\} \\
&\leqslant \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \notin (h_j(\boldsymbol{x}) - \gamma, h_j(\boldsymbol{x}) + \gamma)\} \\
&\leqslant \frac{\mathrm{Var}\left(\hat{h}_{j,m_k}(\boldsymbol{x})\right)}{\gamma^2} \\
&\leqslant \frac{\sigma_j^2(\boldsymbol{x})}{m_k \epsilon_{j,k}^2}. \qquad (11)
\end{aligned}$$

Similarly, under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the 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 $h_i(\boldsymbol{x}) \leqslant 0$, $i = 1, \ldots, c$.

$$\begin{aligned}
\Pr\{\boldsymbol{x} \notin \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcup_{i=1}^{c}\left(\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}\right)\right\} \\
&\leqslant \sum_{i=1}^{c} \Pr\{\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k}\}
\end{aligned}$$

$$\begin{aligned}
&\leqslant \sum_{i=1}^{c} \Pr\{\hat{h}_{i,m_k}(\boldsymbol{x}) > \epsilon_{i,k} + h_i(\boldsymbol{x})\} \\
&\leqslant \sum_{i=1}^{c} e^{-\frac{m_k \epsilon_{i,k}^2}{2\sigma_i^2(\boldsymbol{x})}} \qquad (12)
\end{aligned}$$

Then under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the 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 $\gamma > 0$ such that $2\gamma < \inf_{\boldsymbol{x}\in\mathcal{F}^c}\{h_i(\boldsymbol{x}) : h_i(\boldsymbol{x}) > 0, i = 1, \ldots, c\}$. Then since $\lim_{k\to\infty} \epsilon_{i,k} = 0$, $\epsilon_{i,k} < \gamma$ for all $k \geqslant K_2$(independent of $\boldsymbol{x}$) and

$$\begin{aligned}
\Pr\{\boldsymbol{x} \in \mathcal{F}(m_k, \boldsymbol{\epsilon}_k)\} &= \Pr\left\{\bigcap_{i=1}^{c}\left(\hat{h}_{i,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}\right)\right\} \\
&\leqslant \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{j,k}\} \\
&= \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k},\, \hat{h}_{j,m_k}(\boldsymbol{x}) \in (h_j(\boldsymbol{x}) - \gamma, h_j(\boldsymbol{x}) + \gamma)\} \\
&\quad + \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \leqslant \epsilon_{i,k}\, \hat{h}_{j,m_k}(\boldsymbol{x}) \notin (h_j(\boldsymbol{x}) - \gamma, h_j(\boldsymbol{x}) + \gamma)\} \\
&\leqslant \Pr\{\hat{h}_{j,m_k}(\boldsymbol{x}) \notin (h_j(y) - \gamma, h_j(\boldsymbol{x}) + \gamma)\} \\
&\leqslant 2e^{\frac{\sigma_j^2(\boldsymbol{x})t_j^2}{2m_k} - \gamma t_j} \\
&\leqslant 2e^{-\frac{m_k \epsilon_{j,k}^2}{2\sigma_j^2(\boldsymbol{x})}}. \qquad (13)
\end{aligned}$$

Thus, under Assumption 1 and the minimum rate condition on $m_k\epsilon_{i,k}^2$, and by the 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, $u = \limsup_{k,r} U_{k,r} < \infty$. We define the fixed quantity $\underline{k}_r$ as the *smallest* number of inner iterations executed in the rth outer iteration:

$$\underline{k}_r = \sup\{k : \sum_{j=1}^{k} u m_j \leqslant b_r\}, \qquad (14)$$

and note that $\underline{k}_r \leqslant k_r$ for all r.

By Assumption 6 there exists $\lambda > 0$ such that the level set $\mathscr{L}(\boldsymbol{x}_0, \lambda)$ is finite. Pick $\boldsymbol{y} \in (\mathcal{F} \bigcap \mathscr{L}(\boldsymbol{x}_0, \lambda))^c$. Then either $\boldsymbol{y} \in \mathcal{F}^c$ or $\boldsymbol{y} \in \mathscr{L}(\boldsymbol{x}_0, \lambda)^c \bigcap \mathcal{F}$. Suppose $\boldsymbol{y} \in \mathcal{F}^c$. Then for $r \geqslant K_1$ (independent of $\boldsymbol{x}_0$ and $\boldsymbol{y}$), $\Pr\{\boldsymbol{y} \in \mathscr{L}_{m_{k_r}}(\boldsymbol{x}_0)\} \leqslant \Pr\{\boldsymbol{y} \in \mathcal{F}_{m_{k_r}}\} = O_p(m_{k_r}^{-1/2}) = O(m_{\underline{k}_r}^{-1/2}) = p'_r$ (from (3)). If $\boldsymbol{y} \in \mathscr{L}(\boldsymbol{x}_0, \lambda)^c \bigcap \mathcal{F}$ then $g(\boldsymbol{y}) > g(\boldsymbol{x}_0) + \lambda$. Then for all $r \geqslant K_2$ (independent of $\boldsymbol{y}$, dependent on $\boldsymbol{x}_0$),

$$\Pr\{\boldsymbol{y} \in \mathscr{L}_{m_{k_r}}(\boldsymbol{x}_0)\} = \Pr\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \leqslant \hat{g}_{m_{k_r}}(\boldsymbol{x}_0)\}$$

$$
\begin{aligned}
&\leqslant \Pr\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \leqslant \hat{g}_{m_{k_r}}(\boldsymbol{x}_0),\, \hat{g}_{m_{k_r}}(\boldsymbol{x}_0) \in (g(\boldsymbol{x}_0) - \lambda/4, g(\boldsymbol{x}_0) + \lambda/4)\} \\
&\quad + \Pr\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \leqslant \hat{g}_{m_{k_r}}(\boldsymbol{x}_0),\, \hat{g}_{m_{k_r}}(\boldsymbol{x}_0) \notin (g(\boldsymbol{x}_0) - \lambda/4, g(\boldsymbol{x}_0) + \lambda/4)\} \\
&\leqslant \Pr\{\hat{g}_{m_{k_r}}(\boldsymbol{y}) \notin (g(\boldsymbol{y}) - \lambda/4, g(\boldsymbol{y}) + \lambda/4)\} + \Pr\{\hat{g}_{m_{k_r}}(\boldsymbol{x}_0) \notin (g(\boldsymbol{x}_0) - \lambda/4, g(\boldsymbol{x}_0) + \lambda/4)\} \\
&\leqslant 2e^{-m_{k_r}\eta_g} \\
&\leqslant 2e^{-m_{\underline{k}_r}\eta_g} = p_r''.
\end{aligned} \tag{15}
$$

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

Let $\varepsilon = \min\{|g(\boldsymbol{x}) - g(\boldsymbol{y})| : (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{F} \bigcap \mathscr{L}(\boldsymbol{x}_0, \lambda),\ g(\boldsymbol{x}) \neq g(\boldsymbol{y})\}$. Then since $\mathcal{F} \bigcap\ \mathscr{L}(\boldsymbol{x}_0, \lambda)$ is finite and contained in $\mathcal{F}$, $\hat{g}_{m_{k_r}}$ converges uniformly to g wp1 on the set $\mathcal{F} \bigcap \mathscr{L}(\boldsymbol{x}_0, \lambda)$ as $r \to \infty$. Thus there exists $K_4(\boldsymbol{x}_0) \in \mathbb{N}$ such that $|\hat{g}_{m_{k_r}}(\boldsymbol{x}) - g(\boldsymbol{x})| < \varepsilon/2$ with probability 1 if $r \geqslant K_4$ for all $\boldsymbol{x} \in \mathcal{F} \bigcap \mathscr{L}(\boldsymbol{x}_0, \lambda)$. 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 $\boldsymbol{x}$, $\boldsymbol{y} \in \mathcal{F} \bigcap \mathscr{L}(\boldsymbol{x}_0, \lambda)$ 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 $\boldsymbol{x}$, $\boldsymbol{y} \in \mathcal{F} \bigcap \mathscr{L}(\boldsymbol{x}_0, \lambda)$, $r \geqslant K_4$. And hence $\boldsymbol{Y}_r \in M^*(N)$ wp1 for $r \geqslant \max(K_3, K_4)$. □

Proof of Theorem 5.

$$
\begin{aligned}
\Pr\{\boldsymbol{Z}_r \notin \mathscr{G}\} &= \Pr\ \underbrace{\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\}}_{\text{the event that a global sol. is never attained}}\ + \Pr\ \underbrace{\left\{ \bigcup_{j=2}^{r} \begin{pmatrix} \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} \neq \emptyset, \\ \text{a global solution was last dropped} \\ \text{at the end of the } j\text{th outer iteration} \end{pmatrix} \right\}}_{\text{the event that a global sol. is attained but dropped due to sampling error}} \\
&= \underbrace{\Pr\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\}}_{\text{(I)}} + \underbrace{\sum_{j=2}^{r} \Pr\left\{ (\boldsymbol{Y}_j \in \mathscr{G} \bigcup \boldsymbol{Z}_{j-1} \in \mathscr{G}) \bigcap \left(\boldsymbol{Z}_j \notin \mathscr{G}, \left(\bigcap_{i=j+1}^{r} \boldsymbol{Y}_i \notin \mathscr{G} \right) \right) \right\}}_{\text{(II)}}
\end{aligned} \tag{16}
$$

Then since each restart is performed independently, for all $r > R$ we get

$$
\begin{aligned}
\text{(I)} &= \prod_{j=1}^{r} \Pr\{\boldsymbol{Y}_r \notin \mathscr{G}\} \\
&= \prod_{j=1}^{r} \Bigg(\Pr\left\{ \boldsymbol{Y}_r \notin \mathscr{G},\, \boldsymbol{Y}_\infty \in \mathscr{G} \mid \boldsymbol{X}_j \in \bigcup_{z^* \in \mathscr{G}} B(\boldsymbol{z}^*) \right\} p_j(\mathscr{G}) + \Pr\left\{ \boldsymbol{Y}_r \notin \mathscr{G},\, \boldsymbol{Y}_\infty \in \mathscr{G} \mid \boldsymbol{X}_j \notin \bigcup_{z^* \in \mathscr{G}} B(\boldsymbol{z}^*) \right\} (1 - p_j(\mathscr{G})) \\
&\qquad \Pr\left\{ \boldsymbol{Y}_r \notin \mathscr{G},\, \boldsymbol{Y}_\infty \notin \mathscr{G} \mid \boldsymbol{X}_j \in \bigcup_{z^* \in \mathscr{G}} B(\boldsymbol{z}^*) \right\} p_j(\mathscr{G}) + \Pr\left\{ \boldsymbol{Y}_r \notin \mathscr{G},\, \boldsymbol{Y}_\infty \notin \mathscr{G} \mid \boldsymbol{X}_j \notin \bigcup_{z^* \in \mathscr{G}} B(\boldsymbol{z}^*) \right\} (1 - p_j(\mathscr{G})) \Bigg) \\
&= \prod_{j=R+1}^{r} \left(p_j(\mathscr{G})\, O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right) + (1 - \nu)\, p_j(\mathscr{G}) + (1 - p_j(\mathscr{G})) \right) \\
&= \prod_{j=R+1}^{r} \left(\underbrace{O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right)}_{\text{error in function estimation}} \quad + \quad \underbrace{1 - \nu p_j(\mathscr{G})}_{\text{error due to stochasticity of algorithm}} \right)
\end{aligned}
$$

where $c' > 0$ and $\nu = \liminf_{r\to\infty} p_r(\mathscr{G}) > 0$. Let $\rho_j = 1 - \nu p_j(\mathscr{G})$. Then $\rho_j < 1$ and since $\nu < 1$ and $\liminf_{r\to\infty} p_j(\mathscr{G}) > 0$, $\limsup_{j\to\infty} \rho_j = \rho^* < 1$. Thus

$$\Pr\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\} = \prod_{j=R+1}^{r} \left(O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right) + \rho^*\right).$$

Recall the definition of $\underline{k}_r$ in (14). Then

$$\Pr\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\} = \prod_{j=R+1}^{r} \left(O\left(e^{-c' m_{\underline{k}_r}^{1-2\delta}}\right) + \rho^*\right).$$

Suppose $m_k = \Theta(k^q)$ where q satisfies condition C.2. Then $b_r = \sum_{i=1}^{\underline{k}_r} u m_i = \Theta(\underline{k}_r^{q+1})$ or $\underline{k}_r = \Theta\left(b_r^{1/(q+1)}\right)$, and hence $m_{\underline{k}_r} = \Theta\left(b_r^{q/(q+1)}\right)$. As a result

$$\Pr\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\} = \prod_{j=R+1}^{r} \left(O\left(e^{-c' b_j^{\left(\frac{q}{q+1}\right)(1-2\delta)}}\right) + \rho^*\right).$$

Now suppose $m_k = \Theta\left(a_{in}^k\right)$ where $a_{in} > 1$. Then condition C.2 is satisfied and $b_r = \sum_{i=1}^{\underline{k}_r} u m_i = \Theta\left(a_{in}^{\underline{k}_r+1}\right)$ or $\underline{k}_r = \Theta\left(\log_{a_{in}} b_r\right)$, and hence $m_{\underline{k}_r} = \Theta(b_r)$. As a result

$$\Pr\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\} = \prod_{j=R+1}^{r} \left(O\left(e^{-c' b_j^{1-2\delta}}\right) + \rho^*\right).$$

In either case, we can write

$$\Pr\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\} = \prod_{j=R+1}^{r} \left(O\left(e^{-c' b_j^{\gamma}}\right) + \rho^*\right),$$

where $\gamma = q(1-2\delta)/(q+1))$ if $\{m_k\}$ is increased polynomially and $\gamma = 1 - 2\delta$ if $\{m_k\}$ is increased geometrically.

Then since the sequence $\{b_r\}$ approaches infinity, there exists R' such that $O\left(e^{-c' b_j^{\gamma}}\right) < (1-\rho^*)/2$ for all $j \geqslant R'$. Hence

$$\Pr\left\{\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_r\} \bigcap \mathscr{G} = \emptyset\right\} = \prod_{j=\max\{R,R'\}+1}^{r} O\left(\frac{1-\rho^*}{2} + \rho^*\right) = O(\tau^r), \tag{17}$$

where $\tau = (1-\rho^*)/2 + \rho^* < 1$.

Now,

$$
\begin{aligned}
\text{(II)} &= \sum_{j=2}^{r} \Pr\left\{ \left(\boldsymbol{Y}_j \in \mathscr{G} \bigcup \boldsymbol{Z}_{j-1} \in \mathscr{G} \right) \bigcap \left(\boldsymbol{Z}_j \notin \mathscr{G}, \left(\bigcap_{i=j+1}^{r} \boldsymbol{Y}_i \notin \mathscr{G} \right) \right) \right\} \\
&= \sum_{j=2}^{r} (\Pr\{\boldsymbol{Z}_{j-1} \text{ loses to } \boldsymbol{Y}_j \,|\, \boldsymbol{Z}_{j-1} \in \mathscr{G}, \boldsymbol{Y}_j \notin \mathscr{G}\} \Pr\{\boldsymbol{Z}_{j-1} \in \mathscr{G},\, \boldsymbol{Y}_j \notin \mathscr{G}\} \\
&\qquad + \Pr\{\boldsymbol{Y}_j \text{ loses to } \boldsymbol{Z}_{j-1} \,|\, \boldsymbol{Y}_j \in \mathscr{G}, \boldsymbol{Y}_{j-1} \notin \mathscr{G}\} \Pr\{\boldsymbol{Y}_j \in \mathscr{G}, \boldsymbol{Z}_{j-1} \notin \mathscr{G}\}) \\
&\leqslant \sum_{j=2}^{r} \Big(\Pr\{\boldsymbol{Z}_{j-1} \in \mathcal{F}^{c}_{m_{k_j}},\, \boldsymbol{Y}_j \in \mathcal{F}_{m_{k_j}} \,|\, \boldsymbol{Z}_{j-1} \in \mathscr{G}, \boldsymbol{Y}_j \notin \mathscr{G}\} \\
&\qquad + \Pr\{\hat{g}_{m_{k_j}}(\boldsymbol{Z}_{j-1}) > \hat{g}_{m_{k_j}}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}_{m_{k_j}},\, \boldsymbol{Y}_j \in \mathcal{F}_{m_{k_j}} \,|\, \boldsymbol{Z}_{j-1} \in \mathscr{G},\, \boldsymbol{Y}_j \notin \mathscr{G},\, \boldsymbol{Y}_j \in \mathcal{F}\} \\
&\qquad + \Pr\{\hat{g}_{m_{k_j}}(\boldsymbol{Z}_{j-1}) > \hat{g}_{m_{k_j}}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}_{m_{k_j}},\, \boldsymbol{Y}_j \in \mathcal{F}_{m_{k_j}} \,|\, \boldsymbol{Z}_{j-1} \in \mathscr{G},\, \boldsymbol{Y}_j \notin \mathscr{G},\, \boldsymbol{Y}_j \in \mathcal{F}^{c}\} \\
&\qquad + \Pr\{\boldsymbol{Z}_{j-1} \in \mathcal{F}_{m_{k_j}},\, \boldsymbol{Y}_j \notin \mathcal{F}_{m_{k_j}} \,|\, \boldsymbol{Z}_{j-1} \neq \boldsymbol{z}^{*}, \boldsymbol{Y}_j \in \mathscr{G}\} \\
&\qquad + \Pr\{\hat{g}_{m_{k_j}}(\boldsymbol{Z}_{j-1}) < \hat{g}_{m_{k_j}}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}_{m_{k_j}},\, \boldsymbol{Y}_j \in \mathcal{F}_{m_{k_j}} \,|\, \boldsymbol{Z}_{j-1} \notin \mathscr{G},\, \boldsymbol{Z}_{j-1} \in \mathcal{F},\, \boldsymbol{Y}_j \in \mathscr{G}\} \\
&\qquad + \Pr\{\hat{g}_{m_{k_j}}(\boldsymbol{Z}_{j-1}) < \hat{g}_{m_{k_j}}(\boldsymbol{Y}_j),\, \boldsymbol{Z}_{j-1} \in \mathcal{F}_{m_{k_j}},\, \boldsymbol{Y}_j \in \mathcal{F}_{m_{k_j}} \,|\, \boldsymbol{Z}_{j-1} \notin \mathscr{G},\, \boldsymbol{Z}_{j-1} \in \mathcal{F}^{c},\, \boldsymbol{Y}_j \in \mathscr{G}\} \Big) \\
&= \sum_{j=2}^{r} \left(O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right) + O_p\left(e^{-\eta_g m_{k_j}}\right) + O_p\left(e^{-c'' m_{k_r}}\right) \right) \\
&= O_p\left(e^{-c' m_{k_r}^{1-2\delta}}\right) \\
&= O\left(e^{-c' b_r^{\gamma}}\right),
\end{aligned} \tag{18}
$$

where γ is defined as before. Finally, substituting (17) and (18) in (16) we get

$$
\Pr\{\boldsymbol{Z}_r \notin \mathscr{G}\} = O\left(e^{-c' b_r^{\gamma}}\right) + O(\tau^{r}), \tag{19}
$$

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

Acknowledgments

The authors acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. URL: http://www.arc.vt.edu.

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.

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.

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.

Galombos, J., E. Seneta. 1973. Regularly varying sequences. *Proceedings of the American Mathematical Society* **41**(4) 110–116.

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

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.

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.

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.

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.

Pasupathy, R., S. R. Hunter, N. A. Pujowidianto, L. H. Lee, C.-H. Chen. 2014. Stochastically constrained ranking and selection via SCORE. *ACM TOMACS* To Appear.

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

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

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* To appear.

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.