Pattern Search Ranking and Selection Algorithms for Mixed Variable Simulation-Based Optimization

Todd A. Sriver
Headquarters U.S. Air Force, Manpower and Personnel
Force Management Division
1235 Jefferson Davis Highway, Suite 101
Crystal City, VA 22202 USA
Todd.Sriver@pentagon.af.mil

James W. Chrissis
Department of Operational Sciences
Air Force Institute of Technology
2950 Hobson Way
Wright-Patterson AFB, OH, 45433 USA
James.Chrissis@afit.edu

Mark A. Abramson
Department of Mathematics and Statistics
Air Force Institute of Technology
Mark.Abramson@afit.edu

April 24, 2007

Acknowledgments: This work was sponsored, in part, by the Air Force Office of Scientific Research. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, United States Government, or research sponsors.

Abstract

The class of generalized pattern search (GPS) algorithms for mixed variable optimization is extended to problems with stochastic objective functions, by augmenting it with ranking and selection (R&S). Asymptotic convergence for the algorithm is established, numerical issues are discussed, and performance of the algorithm is studied on a set of test problems.

Key words: pattern search algorithms, mixed variable programming, ranking and selection, stochastic optimization

1 Introduction

We consider the optimization of a stochastic system in which the objective is to minimize some performance measure that is subject to random variation. The system is of sufficient complexity so that no information of the underlying performance measure function is available and therefore cannot be evaluated in a closed form but must be estimated, for example, via "black-box" simulation. Examples of these simulation optimization problems can be found in a variety of applications, including the modeling and design of stochastic manufacturing systems, production-inventory situations, communication or infrastructure networks, logistics system support, and airline operations.

The class of problems we target is further complicated by variables that may be either continuous or *categorical*, the latter of which must take their value from a predefined list or set, which may not even be numeric. For example, a stochastic communication network containing a buffer queue at each router may include a categorical design variable for queue discipline (e.g., first-in-first-out (FIFO), last-in-first-out (LIFO), or priority) at each router. We can use numeric values to represent categorical variables (e.g., 1 = FIFO, 2 = LIFO, and 3 = priority), but the values do not conform to the inherent ordering that the numerical values suggest. The class of optimization problems that includes continuous and categorical variables is known as *mixed variable programming* (MVP) [9].

Given a mixed variable domain Θ and a probability space $(\Omega, \mathcal{F}, P)$ with sample space Ω, sigma-field $\mathcal{F}$ and probability measure P, the stochastic output is modeled as an unknown response function $F : \Theta \times \Omega \to \mathbb{R}$ which depends upon a vector $x \in \Theta$ of controllable design variables and a vector $\omega \in \Omega$ that represents random system effects. The objective function $f : \Theta \to \mathbb{R}$ is the long-run expected performance of the system, given by

$$f(x) = E_P[F(x, \omega)] = \int_\Omega F(x, \omega) P(d\omega).$$

More specifically, we consider the class of stochastic MVP problems, for which the continuous variables are subject to bound and linear constraints; namely,

$$\min_{x \in \Theta} f(x),$$

where Θ is partitioned into continuous and discrete domains Θ^c and Θ^d, respectively. A solution $x \in \Theta$ is denoted as $x = (x^c, x^d)$, where $x^c \in \Theta^c$ is the vector of continuous variables of dimension n^c, and $x^d \in \Theta^d$ is the vector of discrete variables of dimension n^d. The domain of the continuous variables Θ^c can be expressed by $\Theta^c = \{x^c \in \mathbb{R}^{n^c} : \ell \leq Ax^c \leq u\}$, where $A \in \mathbb{R}^{m^c \times n^c}$, $\ell, u \in \mathbb{R}^{m^c}$, and $\ell < u$. The domain of the discrete variables $\Theta^d \subseteq \mathbb{Z}^{n^d}$, which may include categorical variables, is represented as a subset of $\mathbb{Z}^{n^d}$ by mapping each categorical variable value to an integer value (but without taking on any metric associated with integers). Furthermore, since f is of unknown analytical form, it is estimated via Monte Carlo samples of F obtained from a representative model of the stochastic system.

Relaxation techniques commonly used to solve mixed integer problems, such as branch-and-bound, are not applicable to MVP problems because the objective and response functions are defined only at the discrete settings of the categorical variables; therefore, relaxing the "discreteness" of these variables is not possible. Small numbers of categorical variables can sometimes be treated by exhaustive enumeration, which results in more global solutions, but this approach quickly becomes computationally prohibitive for even moderately-sized problems.

Audet and Dennis [9] introduce a convergent pattern search algorithm for solving MVP problems, which is applied in [25] to the design of a thermal insulation system. Lucidi et al. [32] present a more general framework for solving MVP problems by replacing pattern search with an unspecified continuous search procedure that satisfies certain properties (see also [31]). However, other than an extension of Audet and Dennis [9] to MVP problems with nonlinear constraints [1], this represents the entire body of literature on this fairly new class of MVP problems.

Thus we are motivated by the need to rigorously solve mixed variable *simulation optimization* problems. Presently, the predominant methods that provide some assurances of convergence are applicable to search domains that are either entirely continuous or entirely discrete. For continuous problems, approaches with more rigorous convergence analyses primarily include stochastic approximation (SA) [23, 30, 39] and derivative-free variants of SA that use finite differences or simultaneous perturbations [41]. A comprehensive treatment can be found in [42]. Some other approaches include the direct search approach of Anderson and Ferris [6] and the grid-based algorithm of Elster and Neumaier [22]. Methods for discrete problems include those related to simulated annealing [7, 8] and other random search approaches, in which case, convergence analyses typically rely on modeling the problem as a discrete-time Markov chain.

If the number of alternatives in the discrete domain is small enough (≤ 20, for example), ranking and selection (R&S) statistical procedures may be used to select the best of all designs. Of particular interest is the so-called *indifference zone* R&S approach, which guarantees correct selection of the best design with some user-specified probability if it is at least a user-specified amount better than all others. Bechhofer [13] introduced R&S for a set of system designs with unknown true response means and a known, common response variance across all designs. Dudewicz and Dalal [21] and Rinott [38] extended the approach to problems with unknown and unequal response variances. Recently, R&S procedures have been used in an iterative fashion by coupling them with search strategies to enable search of a possibly large solution domain [5, 15, 35, 37].

In this paper, we present and analyze a class of algorithms for optimization of stochastic systems with mixed variables and linear constraints on the continuous variables. The method extends the class of mixed variable *generalized pattern search* (GPS) algorithms [1, 3, 9] to stochastic response functions by employing an R&S procedure in the selection of new iterates as a means of controlling statistical error.

We should note that pattern search methods have not historically been targeted toward large-scale problems, consistent with other pattern search algorithms, the one presented here is not expected to perform efficiently on problems with a large number of variables.

This paper is organized as follows. In the next section, some basic definitions for mixed variable domains are provided, followed by descriptions of pattern search and R&S, respectively. In Section 3, we present and describe an algorithmic framework that combines pattern search with ranking and selection for a class of mixed variable stochastic optimization problems. In Section 4, under certain reasonable assumptions, we prove almost sure convergence of this class of algorithms to limit points that satisfy first-order necessary conditions for optimality. Computational issues are discussed in Section 5, including a study of performance of the algorithm on a set of test problems. Concluding remarks are given in Section 6.

2 Background

2.1 Mixed Variables

Since categorical variables do not necessarily have an inherent ordering, we require notions of local optimality and stationarity in a mixed variable domain. We do this with respect to a set of discrete neighbors that must be defined in the context of the specific problem to be solved. For example, in [25] the optimization problem was to determine the optimal number and types of insulators in a thermal insulation system. In this case, given a design, a discrete neighbor was defined to be any design in which any one insulator was replaced by one of a different material, or a design in which the number of insulators was increased or decreased by one.

The set of discrete neighbors is defined by a set-valued function $\mathcal{N} : \Theta \to 2^{\Theta}$, where 2^{Θ} denotes the power set of Θ. The notation $y \in \mathcal{N}(x)$ means that the point y is a *discrete neighbor* of x. By convention, for each $x \in \Theta$, $x \in \mathcal{N}(x)$ and $\mathcal{N}(x)$ is assumed to be finite. We also require a notion of continuity with respect to the set of discrete neighbors; namely, if $\{x_k\}$ converges to $\hat{x}$, then $\mathcal{N}(x_k)$ converges to $\mathcal{N}(\hat{x})$.

Although there is no metric assumed for categorical variables, convergence in a mixed variable domain is defined as one would expect: a sequence $\{x_i\} = \{(x_i^c, x_i^d)\} \subset \Theta$ converges to $x = (x^c, x^d) \in \Theta$ if x_i^c converges to x^c (under the standard definition) and $x_i^d = x^d$ for all sufficiently large i.

The following definition is due to Audet and Dennis [9].

Definition 2.1 *A point $x = (x^c, x^d) \in \Theta$ is a* local minimizer *of f with respect to the set of neighbors $\mathcal{N}(x) \subset \Theta$ if there exists an $\epsilon > 0$ such that $f(x) \leq f(v)$ for all v in the set*

$$\Theta \cap \bigcup_{y \in \mathcal{N}(x)} (B(y^c, \varepsilon) \times \{y^d\}).$$

Note that this definition is stronger than simply requiring optimality with respect to the continuous variables and also with respect to discrete neighbors. It also requires the local minimizer to have a lower function value than any point in a neighborhood of each discrete neighbor.

The following definition, which is similar in form to that of [32] for unconstrained problems, is implied but not formally stated in [9] and [1]. The notation $\nabla^c f$ represents the gradient of f with respect to the continuous variables while holding the discrete variables constant.

Definition 2.2 *The point $x = (x^c, x^d) \in \Theta$ satisfies* first-order necessary conditions for optimality *if*

1. $(w^c - x^c)^T \nabla^c f(x) \geq 0$ *for every feasible* $(w^c, x^d) \in \Theta$*;*
2. $f(x) \leq f(y)$ *for every discrete neighbor* $y \in \mathcal{N}(x) \subset \Theta$*;*
3. $(w^c - y^c)^T \nabla^c f(y) \geq 0$ *for every discrete neighbor* $y \in \mathcal{N}(x)$ *satisfying* $f(y) = f(x)$ *and for any feasible* $(w^c, y^d) \in \Theta$.

We show in Section 4.3 that, under reasonable assumptions, certain subsequences generated by the class of algorithms introduced in this paper converge with probability one (almost surely) to limit points satisfying Conditions 1–3 of Definition 2.2.

2.2 Pattern search

Pattern search is a subclass of direct search algorithms introduced and analyzed by Torczon [46] for unconstrained problems and extended by Lewis and Torczon to problems with bound constraints [27] and a finite number of linear constraints [28]. In all three results, convergence of a subsequence of iterates to a limit point satisfying first-order necessary conditions is proved. Audet and Dennis [10] generalized this work, adding a hierarchy of results that depend on the local smoothness of the objective function, and limited second-order convergence properties are proved in [2]. Audet and Dennis [9] also extended the work in [27] to bound-constrained MVP problems. Other notable extensions include those to NLP problems with general nonlinear constraints [29, 11, 12], and MVP problems with nonlinear constraints [1, 3].

Pattern search methods are derivative-free, meaning that they do not use explicit or approximate derivatives. They are defined by a finite set of directions D that *positively span* the domain $\mathbb{R}^{n^c}$ [26]. That is, any vector in $\mathbb{R}^{n^c}$ can be represented by a nonnegative linear combination of directions in D [19]. This construction is precisely what is needed to ensure that at least one vector in D must be a direction of descent [19]. The set of directions and a step size parameter are used to construct a conceptual mesh centered about the current iterate (the incumbent). Trial points are selected from the mesh, evaluated, and compared to the incumbent to select the next iterate. If an improvement is found among the trial points, the iteration is declared successful and the mesh is retained or coarsened; otherwise, the mesh is refined and a new set of trial points is constructed. Proofs of convergence are based on showing that the step size parameter becomes arbitrarily small.

Pattern search applied to stochastic optimization problems is rare. Ouali et al. [36] applied multiple repetitions of GPS directly to a stochastic simulation model to seek minimum cost maintenance policies where costs were estimated by the model. In a more rigorous approach, Trosset [47] extended pattern search using traditional statistical tools to deal with variation, the key drawback being that the number of function evaluations needed per iteration to assure convergence becomes very large.

2.2.1 Mixed Variable GPS.

A GPS framework for MVP problems with bound constraints was developed by Audet and Dennis [9] and further extended to linear constraints (as well as to general nonlinear constraints and more general objective functions) in [1]. For each unique combination $i = 1, 2, \ldots, i_{\max}$ of discrete variable values, a set of positive spanning directions D^i is represented as a matrix, such that $d \in D^i$ means that the direction d is a column of D^i. We denote $D = \bigcup_{i=1}^{i_{max}} D^i$. The positive spanning directions must satisfy the restriction,

$$D^i = G_i Z_i, \tag{1}$$

where $G_i \in \mathbb{R}^{n^c \times n^c}$ is a nonsingular generating matrix and $Z_i \in \mathbb{Z}^{n^c \times |D^i|}$. The mesh at iteration k is then formed as the direct product of Θ^d with the union of a finite number of lattices in Θ^c, i.e.

$$M_k = \bigcup_{i=1}^{i_{\max}} M_k^i \times \Theta^d, \qquad M_k^i = \bigcup_{x \in T_k} \{x^c + \Delta_k D^i z : z \in \mathbb{Z}_+^{|D^i|}\} \ \subset \ \mathbb{R}^{n^c}, \tag{2}$$

where $\Delta_k > 0$ is the mesh size parameter, and T_k is the set of all previously evaluated trial points T_k (with T_0 as the set of initial points).

Audet and Dennis explicitly separate each iteration into two distinct steps, a SEARCH step and a POLL step. The optional SEARCH step selects a finite number of trial points on the mesh M_k. This step contributes nothing to the convergence theory, but allows great flexibility in incorporating any desired heuristic to speed convergence. For example, common approaches include randomly selecting a space-filling set of points using a Latin hypercube design or orthogonal arrays, or applying a few iterations of a genetic algorithm. For computationally expensive functions, one common approach [16, 17, 18, 33] is to construct and optimize a less expensive surrogate function on the mesh.

If the SEARCH is unsuccessful, the POLL step evaluates a subset of neighboring mesh points known as the *poll set*. Each point in this set must be evaluated before the iteration can be declared unsuccessful. Polling is conducted in up to three stages: polling with respect to the continuous variables, polling on a set of discrete neighbors, and extended polling around those discrete neighbors whose function value is sufficiently close to that of the incumbent.

For polling with respect to the continuous variables, the poll set centered at $x \in T_k$ is defined as

$$P_k(x) = \{x\} \cup \{x + \Delta_k(d, 0) \in \Theta : d \in D_k(x)\} \ \subset \ M_k. \tag{3}$$

where $D_k(x) \subset D$ is the set of positive spanning directions used at x, and $(d, 0)$ denotes the partitioning into continuous and discrete variables with 0 indicating that the discrete variables remain unchanged, *i.e.*, $x_k + \Delta_k(d, 0) = (x_k^c + \Delta_k d, x_k^d)$. Polling with respect to discrete variables requires a user-defined discrete set of neighbors at x_k, denoted as $\mathcal{N}(x_k) \subset M_k$, where $\mathcal{N}$ is the same set-valued function as previously defined. If the first two polling stages do not yield an improved solution, extended polling may be conducted in the continuous neighborhood about each point y in the set of neighbors $\mathcal{N}(x_k)$ for which the objective function satisfies the extended poll condition $f(x_k) \le f(y) \le f(x_k) + \xi_k$, for some user-specified extended poll trigger $\xi_k \ge \xi > 0$ bounded away from zero.

The rules for updating the mesh size, which allow convergence to be proved without a sufficient decrease condition, are given as follows. Given a fixed rational number $\tau > 1$ and two integers $m^- \le 1$ and $m^+ \ge 0$, the mesh size parameter Δ_k is updated according to the rule,

$$\Delta_{k+1} = \tau^{m_k} \Delta_k, \tag{4}$$

where

$$m_k \in \begin{cases} \{0, 1, \ldots, m^+\}, & \text{if an improved mesh point is found} \\ \{m^-, m^- + 1, \ldots, -1\}, & \text{otherwise.} \end{cases} \tag{5}$$

2.2.2 Bound and Linear Constraints.

Lewis and Torczon [27, 28] showed that, in order to ensure convergence to a stationary point in the presence of linear constraints, search directions used in pattern search must conform to the geometry of the constraints. Specifically, if the current iterate is within $\varepsilon > 0$ of a constraint boundary, the tangent cone $K^{\circ}(x, \varepsilon)$ may be generated as the polar of the cone $K(x, \varepsilon)$ of outward pointing normals for the constraints within ε of x_k. Inclusion of the tangent cone generators in the set of directions used by pattern search is sufficient to preserve convergence properties of the algorithm. An algorithm for computing these directions in the absence of degeneracy is given in [28], while degeneracy is specifically addressed in [4]. It should be noted that, since the target class of problems is restricted to a finite number of linear constraints, there are only a finite number of tangent cone generators for the entire feasible region, which prevents violation of the finiteness of the direction sets $D^i, i = 1, 2, \ldots, i_{\max}$. However, this would not hold in the presence of nonlinear constraints, which is why they are not treated in this paper.

To simplify the convergence analysis in Section 4 and avoid reintroducing the method of [28], the following more general definition from [10] is provided, and the construction and inclusion of tangent cone generators will be assumed.

Definition 2.3 *Let D be a positive spanning set in $\mathbb{R}^n$. A rule for selecting the positive spanning sets $D_k(x_k) \subseteq D$* conforms to *the region $\Psi \subseteq \mathbb{R}^n$ for some $\varepsilon > 0$, if at each iteration k and for each y in the boundary of Ψ for which $\|y - x_k\| < \varepsilon$, the tangent cone $K^{\circ}(x, \varepsilon)$ is generated by nonnegative linear combinations of a subset of the elements of $D_k(x_k)$.*

If directions are chosen at each iteration to conform to Θ^c for some $\varepsilon > 0$, then linear constraints can be treated with a simple "barrier" approach, in which any infeasible point is assigned an objective function value of $+\infty$ without incurring the expense of actually evaluating the objective function value there.

2.3 Ranking and Selection (R&S)

For problems with stochastic response functions, single-sample response comparisons required for traditional pattern search methods can result in erroneous decisions due to variation in the response. Alternative techniques for comparing trial points are necessary to ensure that the iterate selection decision accounts for variation and provides some statistical assurances of correct decisions. In [47], iterate selection via hypothesis testing is suggested in which a binary selection decision between the incumbent and candidate design is based on sufficient statistical evidence. We generalize this approach using R&S so that multiple candidates may be considered simultaneously at reasonable computational cost associated with the requisite sampling. Rather than generating precise estimates, R&S procedures detect the relative order of the candidate solutions, providing computational advantages over precise estimation.

The mechanics of a generic indifference-zone R&S procedure are now briefly described so that this construct may be incorporated into the GPS algorithm in Section 3 (see [45] for a more detailed survey). Generally speaking, we denote by X_k the k-th element of a sequence of random vectors, and by x_k a realization of X_k. Given a finite set of candidate points $C = \{Y_1, Y_2, \ldots, Y_{n_C}\}$ with $n_C \geq 2$, let $f_q = f(Y_q) = E[F(Y_q, \cdot)]$ denote the true mean of the response function F at Y_q for each $q = 1, 2, \ldots, n_C$. The collection of these means can be ordered from minimum to maximum as,

$$f_{[1]} \leq f_{[2]} \leq \cdots \leq f_{[n_C]}, \tag{6}$$

and the notation $Y_{[q]} \in C$ indicates the candidate from C with the q^{th} best (lowest) *true* objective function value.

In an indifference-zone R&S procedure, no distinction is made between two candidate points whose true means are within some $\delta > 0$ of each other. That is, if the two best candidates satisfy $f_{[2]} - f_{[1]} < \delta$, then the procedure is said to be *indifferent* in choosing $Y_{[1]}$ or $Y_{[2]}$ as the best. The probability of correct selection (CS) is defined in terms of the *indifference zone parameter* δ and the significance level $\alpha \in (0, 1)$, as

$$P\{CS\} = P\left\{\text{select } Y_{[1]} \mid f_{[q]} - f_{[1]} \geq \delta;\ q = 2, 3, \ldots, n_C\ \right\} \geq 1 - \alpha. \tag{7}$$

Since $P\{CS\} = \frac{1}{n_C}$ is guaranteed simply by choosing randomly from the alternatives, the significance level must satisfy $0 < \alpha < 1 - \frac{1}{n_C}$.

Of course, true objective function values are not available in practice, so it is necessary to work with sample means of the response F. For each $q = 1, 2, \ldots, n_C$, let s_q be the total number of replications and let $\{F_{qs}\}_{s=1}^{s_q} = \{F(Y_{qs}, W_{qs})\}_{s=1}^{s_q}$ be the set of responses obtained via simulation, where W_{qs} are realizations of the random noise. Then for each $q = 1, 2, \ldots, n_C$, the sample mean $\overline{F}_q$ is computed as,

$$\overline{F}_q = \frac{1}{s_q} \sum_{s=1}^{s_q} F_{qs}. \tag{8}$$

These sample means may be ordered and indexed the same way as in (6). The notation $\hat{Y}_{[q]} \in C$ is used to denote the candidate with the q^{th} best (lowest) *estimated* objective function value as determined by the R&S procedure. The candidate corresponding to the minimum mean response, $\hat{Y}_{[1]} = \arg(\overline{F}_{[1]})$, is chosen as the best point.

To retain generality of the algorithm class of Section 3, we define Procedure RS(C, α, δ) in Figure 1 as a generic R&S procedure that takes as input a candidate set C, significance level α, and indifference zone parameter δ, and returns candidate $\hat{Y}_{[1]} = \arg(\overline{F}_{[1]})$ as the best. The technique used in Step 1 to determine the number of samples for each candidate is dependent on the specific R&S procedure. We discuss such a procedure in Section 5.1.

3 The MGPS-RS algorithm

For stochastic response functions, procedures of the type introduced in Section 2.3 are used within the generalized pattern search framework to select new iterates. This framework is flexible in that a number of specific R&S procedures may be used, so long as they satisfy the probability of correct selection guarantee (7).

A mixed variable GPS ranking and selection (MGPS-RS) algorithm is presented in Figure 2 for mixed variable stochastic optimization problems with linear constraints on the continuous variables. In the algorithm, binary comparisons of incumbent and trial designs used in traditional GPS methods are replaced by R&S procedures in which one candidate is selected from a finite set

Procedure RS(C, α, δ)

Inputs: $C = \{Y_1, Y_2, \ldots, Y_{n_C}\}$, $\alpha \in (0,1)$, $\delta > 0$.

Step 1: For each $Y_q \in C$, use an appropriate statistical technique to determine the number of samples s_q required to meet the probability of correct selection guarantee (7), as a function of α, δ and response variation of Y_q.

Step 2: For each $q = 1, 2, \ldots, n_C$, obtain replicated responses F_{qs}, $s = 1, 2, \ldots, s_q$, and compute the sample mean $\overline{F}_q$, according to (8).

Return: $\hat{Y}_{[1]} = \arg\left(\overline{F}_{[1]}\right)$.

Figure 1: A Generic R&S Procedure.

of candidates considered simultaneously. The R&S procedures provide error control by ensuring sufficient sampling of the candidates so that the best or δ-near-best is chosen with probability $1 - \alpha$ or greater.

The mesh construct of (2) defines the set of points in the search domain Θ from which the candidates are drawn. In the SEARCH step, the flexibility of GPS allows any user-defined procedure to be used in determining which candidates from (2) to consider. In the POLL step, the entire poll set about the incumbent (3) and the discrete neighbor set are considered simultaneously. If SEARCH and POLL are unsuccessful, the EXTENDED POLL step conducts a polling sequence that searches the continuous neighborhood of any discrete neighbor with a response mean $\overline{F}(Y)$ sufficiently close to the response mean $\overline{F}(X_k)$ of the incumbent X_k. This step is divided into sub-steps to account for the sequence of R&S procedures that may be necessary. In Step 3a, each sub-iterate Y_k^j, indexed by sub-iteration counter j and iteration k, is selected as the best candidate from the poll set centered about the previous sub-iterate using the R&S procedure, terminating when the procedure fails to produce a sub-iterate different from its predecessor. The terminal point of the resulting sequence $\{Y_k^j\}_{j=1}^{J_k}$, denoted as $Z_k = Y_k^{J_k}$ and termed an *extended poll endpoint*, is compared to the incumbent via a separate R&S procedure in Step 3b.

If the *extended poll trigger* ξ_k is set too high, more EXTENDED POLL steps result, thus making a solution more "global". However, the additional sampling required at the extra points increases computational expense, particularly with high noise levels in the response output.

The algorithm maintains a separate counter for R&S parameters α_r and δ_r, which are updated according to the formulas,

$$\alpha_{r+1} = \rho_\alpha \alpha_r, \tag{9}$$
$$\delta_{r+1} = \rho_\delta \delta_r, \tag{10}$$

for some $\rho_\alpha, \rho_\delta \in (0,1)$, where $\alpha_0 \in (0,1)$ and $\delta_0 > 0$ are the initial settings. As shown in Section 4, these rules ensure convergence of the algorithm.

The update rules for Δ_k in the algorithm are the same as the deterministic case; i.e., refinement or coarsening is accomplished according to (4)–(5), depending on the success of the

Mixed Variable Generalized Pattern Search - Ranking & Selection (MGPS-RS) Algorithm

Initialization: Choose a feasible starting point $X_0 \in \Theta$. Set $\Delta_0 > 0$, $\xi > 0$, $\alpha_0 \in (0,1)$, $\delta_0 > 0$, $\rho_\alpha \in (0,1)$, and $\rho_\delta \in (0,1)$. Initialize counters $k = 0$ and $r = 0$.

1. SEARCH step (optional): Employ a finite strategy to select a set of mesh points $U_k \subset M_k$ as candidate points to be evaluated. Use Procedure $\mathrm{RS}(U_k \cup \{X_k\}, \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]} \in U_k \cup \{X_k\}$. Update $\alpha_{r+1} = \rho_\alpha \alpha_r$, $\delta_{r+1} = \rho_\delta \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq X_k$ (the step is *successful*), update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$ according to (4)–(5), and $k = k+1$, and repeat Step 1. Otherwise, proceed to Step 2.

2. POLL step: Set extended poll trigger $\xi_k \geq \xi$. Call Procedure RS to get the estimated best solution $\hat{Y}_{[1]} = \mathrm{RS}(P_k(X_k) \cup \mathcal{N}(X_k), \alpha_r, \delta_r)$, where $P_k(X_k)$ is defined in (3) and $\mathcal{N}(X_k) \subset M_k$. Update $\alpha_{r+1} = \rho_\alpha \alpha_r$, $\delta_{r+1} = \rho_\delta \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq X_k$ (the step is *successful*), update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$ according to (4)–(5), and $k = k+1$, and return to Step 1. Otherwise, proceed to Step 3.

3. EXTENDED POLL step: For each discrete neighbor $Y \in \mathcal{N}(X_k)$ that satisfies the extended poll trigger condition $\overline{F}(Y) < \overline{F}(X_k) + \xi_k$, set $j = 1$ and $Y_k^j = Y$ and do the following:

   (a) Call Procedure RS to get the estimated best solution $\hat{Y}_{[1]} = \mathrm{RS}(P_k(Y_k^j), \alpha_r, \delta_r)$. Update $\alpha_{r+1} = \rho_\alpha \alpha_r$, $\delta_{r+1} = \rho_\delta \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq Y_k^j$, set $Y_k^{j+1} = \hat{Y}_{[1]}$ and $j = j+1$, and repeat Step 3a. Otherwise, set $Z_k = Y_k^j$ and proceed to Step 3b.

   (b) Use Procedure $\mathrm{RS}(\{X_k, Z_k\}, \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]} = X_k$ or $\hat{Y}_{[1]} = Z_k$. Update $\alpha_{r+1} = \rho_\alpha \alpha_r$, $\delta_{r+1} = \rho_\delta \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} = Z_k$ (the step is *successful*), update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$ according to (4)–(5), and $k = k+1$, and return to Step 1. Otherwise, repeat Step 3 for another discrete neighbor that satisfies the extended poll trigger condition. If no such discrete neighbors remain, set $X_{k+1} = X_k$, $\Delta_{k+1} < \Delta_k$ according to (4)–(5), and $k = k+1$ and then return to Step 1.

Figure 2: MGPS-RS Algorithm for Stochastic Optimization.

SEARCH, POLL, and EXTENDED POLL steps.

Each execution of the R&S procedure generates an iterate or sub-iterate that is the candidate returned as the "best" by the procedure. When the new iterate (sub-iterate) is different from (presumed "better" than) the incumbent, the iteration (sub-iteration) is termed *successful*; if it remains the same, it is *unsuccessful*. The use of these terms is in keeping with traditional pattern search methods where, in a deterministic setting, a success indicates a strict improvement in the objective function value. Let V_{r+1} denote an iterate or sub-iterate selected from candidate set C of cardinality n_C by the rth R&S procedure of the MGPS-RS algorithm. Each successful or unsuccessful outcome (iteration or sub-iteration) can be divided into three cases, respectively, as follows:

1. The outcome is considered *successful* if one of the following holds:

   (a) indifference zone condition is met and R&S correctly selects a new incumbent; i.e.,
   $$f(Y_{[q]}) - f(V_{r+1}) \geq \delta_r,\ q = 2, 3, \ldots, n_C, \quad V_r \neq V_{r+1} = Y_{[1]}; \tag{11}$$

   (b) indifference zone condition is met but R&S incorrectly selects a new incumbent; i.e.,
   $$f(Y_{[q]}) - f(Y_{[1]}) \geq \delta_r,\ q = 2, 3, \ldots, n_C, \quad V_r \neq V_{r+1} \neq Y_{[1]}; \tag{12}$$

   (c) indifference zone condition is not met and R&S selects a new incumbent; i.e.,
   $$\left|f(Y_{[q]}) - f(Y_{[1]})\right| < \delta_r \text{ for some } q \in \{2, 3, \ldots, n_C\}, \quad V_r \neq V_{r+1}. \tag{13}$$

2. The outcome is *unsuccessful* if one of the following holds:

   (a) indifference zone condition is met and R&S correctly selects the incumbent; i.e.,
   $$f(Y_{[q]}) - f(Y_{[1]}) \geq \delta_r,\ q = 2, 3, \ldots, n_C, \quad V_r = V_{r+1} = Y_{[1]}; \tag{14}$$

   (b) indifference zone condition is met but R&S incorrectly selects the incumbent; i.e.,
   $$f(Y_{[q]}) - f(Y_{[1]}) \geq \delta_r,\ q = 2, 3, \ldots, n_C, \quad V_r = V_{r+1} \neq Y_{[1]}; \tag{15}$$

   (c) indifference zone condition not met and R&S selects the incumbent; i.e.,
   $$\left|f(Y_{[q]}) - f(Y_{[1]})\right| < \delta_r \text{ for some } q \in \{2, 3, \ldots, n_C\}, \quad V_{r+1} = V_r. \tag{16}$$

In the algorithm, X_k and Y_k^j play the role of V_r for iterates and sub-iterates, respectively. Of the possible outcomes for new iterates or sub-iterates, conditions (11) and (14) conform to the traditional GPS methods for deterministic optimization where, in the case of a successful iteration, a trial point from the mesh has a better true objective function value than the incumbent and, in the case of an unsuccessful iteration, the incumbent has the best true objective function value of all candidates considered. Of particular concern for the convergence analysis are the remaining conditions.

Conditions (13) and (16) occur when the difference between true objective function values of a trial point on the mesh and the incumbent is smaller than the indifference zone parameter. This situation can result from either an overly relaxed indifference zone or an objective function whose true surface is "flat" in the region of the search. When this occurs, the probability for correct selection cannot be guaranteed. However, forcing convergence of δ_r to zero via update rules ensures that the indifference zone condition will be met in the limit. A greater concern is the case when the indifference zone condition is met, but the algorithm selects the "wrong" candidate (i.e., it doesn't choose the candidate with the best true objective function value). This represents conditions (12) and (15), and occurs with probability α_r or less for the rth R&S procedure. The convergence analysis of the following section addresses controls placed on the errors presented by these conditions.

4 Proof of Convergence

We now establish convergence results for the the MGPS-RS algorithm. The analysis that follows requires the following assumptions:

A1: All iterates X_k produced by the MGPS-RS algorithm lie in a compact set.

A2: The objective function f is continuously differentiable with respect to the continuous variables when the discrete variables are fixed.

A3: For each set of discrete variables X^d, the corresponding set of directions $D^i = G_i Z_i$, as defined in (1), includes tangent cone generators for every point in Θ^c.

A4: The rule for selecting directions D_k^i conforms to Θ^c for some $\varepsilon > 0$ (see Definition 2.3).

A5: For each $q = 1, 2, \ldots, n_C$, the responses $\{F_{qs}\}_{s=1}^{s_q}$ are independent, identically and normally distributed random variables with mean $f(X_q)$ and unknown variance $\sigma_q^2 < \infty$, where $\sigma_l^2 \neq \sigma_q^2$ whenever $l \neq q$.

A6: For the rth R&S procedure with candidate set $C = \{Y_1, Y_2, \ldots, Y_{n_C}\}$, RS$(C, \alpha_r, \delta_r)$ guarantees correctly selecting the best candidate $Y_{[1]} \in C$ with probability of at least $1 - \alpha_r$ whenever $f(Y_{[q]}) - f(Y_{[1]}) \geq \delta_r$ for any $q \in \{2, 3, \ldots, n_C\}$.

A7: For all but a finite number of MGPS-RS iterations and sub-iterations, the best solution $Y_{[1]} \in C$ is unique, i.e., $f(Y_{[1]}) \neq f(Y_{[q]})$ for all $q \in \{2, 3, \ldots, n_C\}$ where $C = \{Y_1, Y_2, \ldots, Y_{n_C}\} \subset M_k$ at iteration k.

These assumptions warrant a brief discussion. Assumption **A1** is a fairly standard assumption, and is easily enforced by including finite upper and lower bounds on the continuous variables (which is very common in practice). Assumption **A3** ensures that the restriction on the direction set (1) is maintained in the presence of linear constraints, and Assumption **A4** provides for adequate rules to generate conforming directions. A sufficient condition for Assumption **A3** to hold is that $G_i = I$ for each $i \in \{1, \ldots, i_{\max}\}$ and the coefficient matrix A is rational [1]. The independent, normally distributed requirement for responses from a single alternative in Assumption **A5** is common in practice and for R&S techniques. Even in simulation optimization problems, in which data is often correlated, this assumption can be achieved via batched output data or sample averages of independent replications [34]. The assumption of unequal variances between different alternatives is realistic for practical problems and is readily handled with modern R&S procedures. Assumption **A6** provides the correct selection guarantee of the R&S procedure and is required in the absence of identifying a specific method. Most R&S procedures are accompanied by proofs that the correct selection guarantee is met. MGPS-RS is flexible in that any R&S procedure may be used, so long as it satisfies Assumption **A6**. Finally, Assumption **A7** is required to ensure the indifference zone condition is eventually met during the course of the iteration sequence. This assumption may seem restrictive, but the likelihood of two candidate mesh points having exactly the same objective function value is quite rare for non-academic problems.

Since MGPS-RS iterates are random variables, the convergence analysis must be carried out in probabilistic terms. To that end, the following definition is provided.

Definition 4.1 *Let $\Theta \subseteq (\mathbb{R}^{n^c} \times \mathbb{Z}^{n^d})$ be a mixed variable domain, and let $\|\cdot\|$ define a norm on $\mathbb{R}^{n^c}$. A sequence of multivariate random vectors $\{X_k\} \subset \Theta$ is said to* converge almost surely (a.s.) *or* converge with probability 1 *to the limit point $x \in \Theta$ if, for every $\varepsilon > 0$, there exists a positive integer N such that $P(X_k^d = x^d) = 1$ and $P(\|X_k^c - x^c\| < \varepsilon) = 1$ for all $k > N$.*

4.1 Controlling Incorrect Selections

Random variation in the responses can lead to *incorrect selections* in the R&S procedure, as formalized by conditions (12) and (15). In order to establish desirable convergence results, a means of bounding the sequence of incorrect selection events is necessary, so that the sequence of iterates is not dominated by incorrectly selected (and possibly unimproving) candidates. This is accomplished by the update rules for α_r and δ_r given in equations (9) and (10), respectively.

Lemma 4.2 *With probability 1, the subsequence of incorrectly selected iterates and sub-iterates generated by algorithm MGPS-RS is finite.*

Proof. Let $\mathcal{A}_r$ denote the occurrence of the event that the rth R&S procedure incorrectly selects the next iterate or sub-iterate. The complement of Assumption **A6** yields $P(\mathcal{A}_r) \leq \alpha_r$, $r = 1, 2, \ldots$, and (9) ensures that

$$\sum_{r=1}^{\infty} P(\mathcal{A}_r) \leq \sum_{r=1}^{\infty} \alpha_r = \sum_{r=1}^{\infty} \rho_\alpha \alpha_{r-1} = \ldots = \alpha_0 \sum_{r=1}^{\infty} {\rho_\alpha}^r = \frac{\alpha_0}{1 - \rho_\alpha} < \infty.$$

The result follows directly from the first half of the Borel-Cantelli Lemma. ∎

Because of the possibility of incorrect selections, the following lemma is also necessary to establish that MGPS-RS cannot cycle indefinitely on a fixed mesh. Such a condition occurs if and only if there are infinitely many consecutive successful iterations.

Lemma 4.3 *With probability 1, the number of consecutive successful MGPS-RS iterations must be finite.*

Proof. Let K_S represent the number of successful MGPS-RS iterations after iteration k. From conditions (11)–(13), $K_S = K_C + K_I + K_\delta$ where K_C is the number of correctly selected successful iterates (11), K_I is the number of incorrectly selected successful iterates (12), and K_δ is the number of successful iterates until the indifference zone condition (13) is satisfied. Since Assumption **A1** ensures that all iterates are mesh points that lie in a compact set, it must follow that $K_C < \infty$. Furthermore, Assumption **A7** and (10) ensure that $K_\delta < \infty$. Finally, since the number of incorrectly selected successful iterates is a subset of all incorrect selections (successful and unsuccessful), Lemma 4.2 ensures that $P(K_I < \infty) = 1$. It follows that

$$P(K_S < \infty) = P(K_C + K_I + K_\delta < \infty) = P(K_I < \infty) = 1. \quad ∎$$

4.2 Mesh Size Behavior

The main result of this section is that, with probability one, there exists a subsequence of mesh size parameters that goes to zero, i.e., $P\left(\liminf_{k\to+\infty} \Delta_k = 0\right) = 1$, which is independent of any smoothness assumptions on the objective function. This result was first established by Torczon [46] and subsequently modified for MVP problems in [9]. Audet and Dennis [10] later adapted a lemma from [46] to provide a lower bound on the distance between any two mesh points at each iteration, which was then extended in [1] to MVP problems. This lower bound is stated in Lemma 4.4, the result of which is necessary to establish that the mesh size parameter is bounded above in Lemma 4.5. Finally, Theorem 4.6 presents the key result for this section. The proof of Lemma 4.4 is independent of response noise and is proven in [1]. The proofs for Lemma 4.5 and Theorem 4.6 are modified from [1] to account for stochastic responses.

Lemma 4.4 *For any $k \geq 0$, $k \in \mathbb{Z}$, let u and v be any pair of distinct mesh points such that $u^d = v^d$. Then for any norm for which all nonzero integer vectors have norm at least 1,*

$$\|u^c - v^c\| \geq \frac{\Delta_k}{\|G_i^{-1}\|}$$

where the index i corresponds to the combination of discrete variable values defined by $u^d = v^d$.

Lemma 4.5 *With probability 1, there exists a positive integer $b^u < \infty$ such that $\Delta_k \leq \Delta_0 \tau^{b^u}$ for any $k \geq 0$, $k \in \mathbb{Z}$.*

Proof. By Assumption **A1**, the search domain is bounded so the discrete variables can only take on a finite number of values. Let $i_{\max}$ denote this number and let $I = \{1, \ldots, i_{\max}\}$. Also under Assumption **A1**, for each $i \in I$, let Λ_i be a compact set in $\mathbb{R}^{n^c}$ containing all MGPS-RS iterates whose discrete variable values correspond to $i \in I$. Let $\gamma = \max\{\mathrm{diam}(\Lambda_i) : i \in I\}$ and $\beta = \max_{i\in I} \|G_i^{-1}\|$, where $\mathrm{diam}(\cdot)$ denotes the maximum distance between any two points in the set. If $\Delta_k > \gamma\beta$, then by Lemma 4.4 (with $v = X_k$), any mesh point u with $u^c \neq X_k^c$ would be outside of $\bigcup_{i\in I} \Lambda_i$. This can be seen by the following:

$$\begin{aligned} \|u^c - X_k^c\| &\geq \frac{\Delta_k}{\|G_i^{-1}\|} > \frac{\gamma\beta}{\|G_i^{-1}\|} = \frac{\gamma \max_{i\in I} \|G_i^{-1}\|}{\|G_i^{-1}\|} \geq \gamma \\ &> \max\{\mathrm{diam}(\Lambda_i) : i \in I\}. \end{aligned} \tag{17}$$

Thus, $\Delta_k > \gamma\beta$ implies that the continuous part of the mesh is devoid of feasible candidates except for the incumbent. Therefore, $M_k \cap \Theta = \{X_k^c\} \times \Theta^d$ and $P_k(X_k) = \{X_k\}$. Furthermore, the poll set for any discrete neighbor Y of X_k is devoid of candidates except for Y by the same argument as (17) using Lemma 4.4 (with $V = Y$), so the EXTENDED POLL step is avoided.

The algorithm can consider a maximum of $i_{\max}$ different candidates defined by the combinations of Θ^d during a SEARCH or POLL step. The mesh size parameter grows without bound only if it is possible to cycle indefinitely between these $i_{\max}$ solutions. But Lemma 4.3 guarantees

$P(K_S < \infty) = 1$, where K_S is the number of consecutive successful iterations after iteration k. Then the mesh size parameter will have grown, at a maximum, by a factor of $(\tau^{m_{\max}})^{K_S}$ and is thus bounded above by $\gamma\beta(\tau^{m_{\max}})^{K_S}$. Let b^u be large enough so that $\Delta_0\tau^{b^u} \geq \gamma\beta(\tau^{m_{\max}})^{K_S}$. Then $P(K_S < \infty) = 1 \Longrightarrow P(\gamma\beta(\tau^{m_{\max}})^{K_S} < \infty) = 1 \Longrightarrow P(\Delta_0\tau^{b^u} < \infty) = 1 \Longrightarrow P(b^u < \infty) = 1$. ∎

Theorem 4.6 *The mesh size parameters satisfy* $P\left(\liminf_{k\to+\infty}\Delta_k = 0\right) = 1$.

Proof. By way of contradiction, suppose there exists a negative integer b^ℓ such that $\Delta_0\tau^{b^\ell} > 0$ and $P(\Delta_k > \Delta_0\tau^{b^\ell}) = 1$ for all $k \geq 0$, $k \in \mathbb{Z}$. By definition of the update rules, it follows from (4)–(5) that $\Delta_k = \tau^{b_k}\Delta_0$ for some $b_k \in \mathbb{Z}$. Since Lemma 4.5 ensures that b_k is bounded above *a.s.* by b^u, it follows that $b_k \in \{b^\ell, b^\ell + 1, \ldots, b^u\}$ *a.s.* Thus, b_k is an element of a finite set of integers which implies that Δ_k takes on a finite number of values for all $k \geq 0$.

Now, $X_{k+1} \in M_k$ ensures that $X^c_{k+1} = X^c_k + \Delta_k D^i z_k$ for some $z_k \in \mathbb{Z}_+^{|D_i|}$ and some $i \in \{1, 2, \ldots, i_{\max}\}$. Repeated application of this equation leads to the following result over a fixed i at iteration $N \geq 1$, where p and q are relatively prime integers satisfying $\tau = \frac{p}{q}$:

$$\begin{aligned}
X^c_N &= X^c_0 + \sum_{k=0}^{N-1}\Delta_k D^i z_k \\
&= X^c_0 + D^i\sum_{k=0}^{N-1}\Delta_0\tau^{b_k} z_k \\
&= X^c_0 + \Delta_0 D^i\sum_{k=0}^{N-1}\frac{p^{\left(b_k+b^\ell-b^\ell\right)}}{q^{(b_k+b^u-b^u)}} z_k \\
&= X^c_0 + \frac{p^{b^\ell}}{q^{b^u}}\Delta_0 D^i\sum_{k=0}^{N-1} p^{\left(b_k-b^\ell\right)} q^{(b^u-b_k)} z_k
\end{aligned}$$

Since $p^{\left(b_k-b^\ell\right)}$ and $q^{(b^u-b_k)}$ are both integers, then $\sum_{k=0}^{N-1} p^{\left(b_k-b^\ell\right)} q^{(b^u-b_k)} z_k$ is a $\left|D^i\right|$-dimensional vector of integers (recall $z_k \in \mathbb{Z}_+^{\left|D^i\right|}$). So, the continuous part of each iterate, X^c_k, $k = 0, \ldots, N$ having the same discrete variable values defined by i lies on the translated integer lattice generated by X^c_0 and the columns of $\frac{p^{b^\ell}}{q^{b^u}}\Delta_0 D^i$. Furthermore, the discrete part of each iterate, X^d_k, lies on the integer lattice $\Theta^d \subset \mathbb{Z}^{n^d}$. By Assumption **A1**, all iterates belong to a compact set, so there must be only a finite number of possible iterates.

Lemma 4.3 ensures that the algorithm cannot cycle indefinitely between these points (*i.e.* the subsequence of consecutive successful iterations is finite *a.s.*). Thus, as $k \to +\infty$, one of the iterates must be visited infinitely many times *a.s.*, which implies an infinite number of mesh refinements. But this contradicts the hypothesis that $P(\Delta_k > \Delta_0\tau^{b^\ell}) = 1$ as $k \to +\infty$. Therefore, $P(\Delta_k > \Delta_0\tau^{b^\ell}) = 0$, which implies $P\left(\liminf_{k\to+\infty}\Delta_k = 0\right) = 1$. ∎

4.3 Main Results

In this subsection, the existence of limit points for MGPS-RS iterates is proven. In addition, limit points are shown to satisfy the first-order necessary conditions for optimality in Definition 2.2. The results have been modified from [9, 1] to accommodate the new algorithmic framework. The following definition, which distinguishes a subsequence of the unsuccessful iterates, simplifies the analysis.

Definition 4.7 *A subsequence of unsuccessful MGPS-RS iterates $\{X_k\}_{k\in K}$ (for some subset of indices K) is said to be a* refining subsequence *if $\{\Delta_k\}_{k\in K}$ converges almost surely to zero, i.e.,* $P\left(\lim_{k\in K}\Delta_k = 0\right) = 1$.

Since Δ_k shrinks for unsuccessful iterations, Theorem 4.6 guarantees that the MGPS-RS algorithm has, with probability 1, infinitely many such iterations. The next theorem, similar to earlier results [9, 1] but modified here for the probabilistic setting, establishes the existence of certain limit points associated with refining subsequences.

Theorem 4.8 *There exists a point $\hat{x} \in \Theta$ and a refining subsequence $\{X_k\}_{k\in K}$, with associated index set $K \subset \{k : X_{k+1} = X_k\}$ such that $\{X_k\}_{k\in K}$ converges almost surely to $\hat{x}$. Moreover, if $\hat{y}$ belongs to the set of neighbors $\mathcal{N}(\hat{x})$, then there exists a $\hat{z} = (\hat{z}^c, \hat{y}^d) \in \Theta$ such that $\{Y_k\}_{k\in K}$ converges almost surely to $\hat{y}$ and $\{Z_k\}_{k\in K}$ converges almost surely to $\hat{z}$, where each $Z_k \in \Theta$ is an* EXTENDED POLL *endpoint initiated at $Y_k^0 \in \mathcal{N}(X_k)$.*

Proof. Theorem 4.6 guarantees $P\left(\liminf_{k\to+\infty}\Delta_k = 0\right) = 1$; thus there is an infinite subset of indices of unsuccessful iterates $K' \subset \{k : X_{k+1} = X_k\}$, such that the subsequence $\{\Delta_k\}_{k\in K'}$ converges *a.s.* to zero, i.e., $P\left(\lim_{k\in K'}\Delta_k = 0\right) = 1$. Since all iterates X_k lie in a compact set, there exists an infinite subset of indices $K'' \subset K'$ such that the subsequence $\{X_k\}_{k\in K''}$ converges almost surely. Let $\hat{x}$ be the limit point of such a subsequence.

Let $\hat{y} \in \Theta$ be a point in the set of neighbors $\mathcal{N}(\hat{x})$. Recall that a notion of continuity of the sets of neighbors was assumed in Section 2.1. Therefore, $\hat{y}$ is a limit point (almost surely) of a subsequence $Y_k \in \mathcal{N}(X_k)$. Let $\hat{z} \in \Theta$ be a limit point of the sequence $Z_k \in \Theta$ of EXTENDED POLL endpoints initiated at Y_k^0. Choose $K \subset K''$ such that $\{Y_k\}_{k\in K}$ converges *a.s.* to $\hat{y}$ and $\{Z_k\}_{k\in K}$ converges *a.s.*, letting $\hat{z}$ denote the limit point. ∎

For the remainder of the analysis, it is assumed that $\hat{x}$ and K satisfy the conditions of Theorem 4.8. The following lemma establishes the first main result, showing that limit points satisfy necessary condition 2 of Definition 2.2. The direct proof is modified for the stochastic case from [1], where it was presented as an alternative to the contradictory proof in [9].

Lemma 4.9 *The limit point $\hat{x}$ satisfies $f(\hat{x}) \le f(\hat{y})$* a.s. *for all $\hat{y} \in \mathcal{N}(\hat{x})$.*

Proof. From Theorem 4.8, the sequences $\{X_k\}_{k\in K}$ and $\{Y_k\}_{k\in K}$ converge *a.s.* to $\hat{x}$ and $\hat{y}$, respectively. Since $k \in K \subset \{k : X_{k+1} = X_k\}$, each $\{X_k\}_{k\in K}$ meets one of the conditions (14)–(16). Assumption **A7** ensures that the number of iterates satisfying condition (16) is finite.

Furthermore, since the set of iterates meeting condition (15) is a subset of all incorrectly selected iterates, Lemma 4.2 ensures the number of iterates satisfying this condition is finite almost surely. Therefore, the number of correctly selected iterates in $\{X_k\}_{k\in K}$ meeting condition (14) must be infinite. Let k' denote an unsuccessful iteration after the last occurrence of both conditions (15) and (16) and let $K' = K \cap \{k \geq k'\}$ which converges *a.s.* to $\hat{x}$. Since each iterate $\{X_k\}_{k\in K'}$ meets condition (14), $f(X_k) < f(Y_k)$ for all $k \in K'$. By the continuity of $\mathcal{N}$ and Assumption **A2**, $f(\hat{x}) = \lim_{k\in K'} f(X_k) \leq \lim_{k\in K'} f(Y_k) = f(\hat{y})$. ∎

The following lemma is necessary to show stationarity of the iterates X_k, and EXTENDED POLL endpoints Z_k. It merges two lemmas from [9] and modifies the results therein for the new algorithmic framework.

Lemma 4.10 *If $\hat{w}$ is the limit point of a refining subsequence, then $(w^c - \hat{w}^c)^T \nabla^c f(\hat{w}) \geq 0$ a.s. for any feasible $(w^c, \hat{w}^d)$.*

Proof. Let $\hat{w}$ be the limit point of a refining subsequence $\{W_k\}_{k\in K}$. By Assumption **A2**, the mean value theorem applies. Then, a feasible point $V \in P(W_k) \setminus \{W_k\}$ can be expressed as,

$$f(V) = f(W_k + \Delta_k(d,0)) = f(W_k) + \Delta_k d^T \nabla^c f(W_k + \lambda_k^d \Delta_k(d,0))$$

for any any $d \in D_k^i \subseteq D^i$ that is feasible infinitely often and $\lambda_k^d \in [0,1]$ that depends on the iteration k and positive basis vector d. Choose $k \in K$ large enough so that the indifference zone condition is satisfied and incorrect selections have terminated almost surely. Then, by condition (14), $f(V) - f(W_k) \geq \delta_r(k)$ where $\delta_r(k)$ depends on k. Furthermore,

$$\begin{aligned} f(W_k) &\leq \min_{V\in P(W_k)} f(V) - \delta_r(k) \\ &= \min_{d\in D_k^i} \left\{ f(W_k) + \Delta_k d^T \nabla^c f(W_k + \lambda_k^d \Delta_k(d,0)) \right\} - \delta_r(k) \\ &= f(W_k) - \delta_r(k) + \Delta_k \min_{d\in D_k^i} \left\{ d^T \nabla^c f(W_k + \lambda_k^d \Delta_k(d,0)) \right\}, \end{aligned}$$

which implies that

$$\min_{d\in D_k^i} \left\{ d^T \nabla^c f(W_k + \lambda_k^d \Delta_k(d,0)) \right\} \geq \delta_r(k).$$

Taking the limit as $k \to \infty$ (in K) yields $\min_{d\in D^i} \left\{ d^T \nabla^c f(\hat{w}) \right\} \geq 0$ *a.s.* (recall $\lim_{k\to\infty} \delta_r(k) = 0$ by equation (10)). Therefore, $d^T \nabla^c f(\hat{w}) \geq 0$ *a.s.* for any $d \in D^i$ that is feasible infinitely often.

By Assumption **A4**, any feasible direction $(w^c - \hat{w}^c)$ is a nonnegative linear combination of feasible directions in D^i that span the tangent cone of Θ^c at $\hat{w}$. Then for $\beta_j \geq 0$, $j = 1, 2, \ldots, n_d$, $(w^c - \hat{w}^c) = \sum_{i=1}^{n_d} \beta_j d_j$ and

$$(w^c - \hat{w}^c)^T \nabla^c f(\hat{w}) = \sum_{j=1}^{n_d} \beta_j d_j^T \nabla^c f(\hat{w}) \geq 0 \quad a.s. \quad ∎$$

It is now possible to state the second main result. Lemma 4.11 shows that the limit point $\hat{x}$ satisfies condition 1 of Definition 2.2.

Lemma 4.11 *The limit point $\hat{x}$ satisfies $(x^c - \hat{x}^c)^T \nabla^c f(\hat{x}) \geq 0$ a.s. for any feasible $(x^c, \hat{x}^d)$.*

Proof. The result follows directly from Lemma 4.10 by substituting X_k for W_k as the refining subsequence, and from results on the sequence $\{X_k\}_{k\in K}$ of Theorem 4.8. ∎

The remaining result may now be completed. Lemma 4.12 shows that limit points $\hat{x}$ and discrete neighbors $\hat{y}$ that satisfy $f(\hat{y}) = f(\hat{x})$ meet condition 3 of Definition 2.2. Theorem 4.13 collects all the main results into a single theorem.

Lemma 4.12 *The limit point $\hat{x}$ and any point $\hat{y}$ in the set of neighbors $\mathcal{N}(\hat{x})$ satisfying $f(\hat{y}) = f(\hat{x})$, are such that $(y^c - \hat{y}^c)^T \nabla^c f(\hat{y}) \geq 0$ a.s. for any feasible $(y^c, \hat{y}^d)$.*

Proof. Choose $k' \in K$ large enough so that the indifference zone condition is satisfied and incorrect selections have terminated almost surely and let $K' = K \cap \{k \geq k'\}$. Then by condition (11), $f(Y_k^j) < f(Y_k^{j-1})$ for all $k \in K'$, which implies $f(Z_k) < f(Y_k)$ for all $k \in K'$. Furthermore, since K' is a subset of unsuccessful iterates, condition (14) is satisfied, which implies $f(X_k) < f(Z_k)$ for each $k \in K'$. By continuity of f and taking the limit as $k \to \infty$ (in K'), it follows that $f(\hat{x}) \leq f(\hat{z}) \leq f(\hat{y})$. Therefore, $f(\hat{z}) = f(\hat{y})$.

By the differentiability of f, it follows that

$$
\begin{aligned}
(y^c - \hat{y}^c)^T \nabla^c f(\hat{y}) = f'(\hat{y}; (y^c - \hat{y}^c, 0)) &= \lim_{t\to 0} \frac{f(\hat{y} + t(y^c - \hat{y}^c, 0)) - f(\hat{y})}{t} \\
= \lim_{k\in K'} \frac{f(Y_k) - f(\hat{y})}{\Delta_k} &= \lim_{k\in K'} \frac{f(Y_k) - f(\hat{z})}{\Delta_k} \geq \lim_{k\in K'} \frac{f(Z_k) - f(\hat{z})}{\Delta_k} \\
&= (z^c - \hat{z}^c)^T \nabla^c f(\hat{z}) \geq 0,
\end{aligned}
$$

where $f'(\hat{y}; (y^c - \hat{y}^c, 0))$ denotes the directional derivative of f at $\hat{y}$ in the direction $(y^c - \hat{y}^c, 0)$, and the last inequality follows by substituting Z_k for W_k as the refining subsequence in Lemma 4.10. ∎

Theorem 4.13 *The limit point $\hat{x}$ satisfies first-order necessary conditions for optimality* a.s.

Proof. This follows directly from Definition 2.2 and Lemmas 4.9, 4.11, and 4.12. ∎

It is likely that limited second-order behavior of this algorithm could be characterized in a way similar to that of [2], but we do not address this here, as it would lengthen the paper unnecessarily.

5 Computational Issues

In practice, convergence theory is insufficient to ensure a workable algorithm. In particular, as Δ_k (and δ_r) approach zero in accordance with the theory, the number of samples required to ensure that Assumption **A6** holds becomes unbounded. Hence, termination criteria need to ensure sufficient accuracy, while keeping sample sizes under control. This issue is addressed in the subsequent two subsections, in which we analyze the issue more closely and illustrate our ideas on a set of test problems.

5.1 Termination and Sample Size Control

The traditional approach in pattern search algorithms is to simply terminate when the mesh size parameter Δ_k is sufficiently small. As a measure of distance between neighboring mesh points, Δ_k not only provides a reasonable estimate of proximity to a limit point, it is also a reasonable measure of stationarity for deterministic problems [20].

For stochastic problems, it is important to also ensure that α_r is sufficiently small so that errors in correct selection achieve a specified threshold. However, this must be balanced by a computational budget restriction to curb the number of function evaluations. To illustrate how this may be accomplished, we now depart from the generality of previous sections and specify the R&S scheme as Rinott's two-stage indifference-zone procedure [38]. We choose Rinott in part because the analysis is easier, but also because it is based on a least favorable configuration assumption that the best candidate has a true mean exactly δ_r better than all remaining candidates, which are all tied for second best [45]. This means that, compared to other methods in the literature, the procedure tends to over-prescribe the number of samples needed to guarantee a specified probability of correct selection.

For a significance level of α, Rinott's constant g is defined as the solution to the equation,

$$\int_0^\infty \left[\int_0^\infty \Phi\left(\frac{g}{\sqrt{v(1/x+1/y)}}\right) f_v(x)dx\right]^{n_C-1} f_v(y)dy = 1-\alpha,$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and $f_v(\cdot)$ is the χ^2 probability distribution function with v degrees of freedom. Values for g can be computed numerically or are available in tables for commonly used parameter combinations.

In the first of the two Rinott stages, the sample variance S_q^2 for each candidate q is computed from a fixed number s_0 of response samples. In the second stage, for indifference zone parameter δ, $s_q - s_0$ additional samples are collected for each candidate, where

$$s_q = \max\{s_0, \lceil (gS_q/\delta)^2 \rceil\}. \tag{18}$$

The objective function value is then estimated by averaging the response samples over both stages for each candidate.

For n_C candidate points, the number of samples required for each R&S iteration is roughly $n_C \left\lceil \left(\frac{gS}{\delta_r}\right)^2 \right\rceil$, where g increases with n_C and $1-\alpha_r$ and S is the standard deviation of one of the candidate points. This yields an approximate per-iteration budget of $B \approx g^2 n_C \left(\frac{S}{\delta_r}\right)^2$, which can be normalized by setting it equal to a multiple of $g^2 n_C$ (since both depend on problem size), yielding $B = Lg^2 n_C$ for some $L \in \mathbb{R}$. Therefore, $L = \left(\frac{S}{\delta_r}\right)^2$, which can be used to form a real-time budget threshold that estimates a point of "minimal return", yielding the final of three termination criteria for MGPS-RS, given now as

$$\Delta_k \leq \Delta_T \tag{19}$$
$$\alpha_r \leq \alpha_T \tag{20}$$
$$\frac{S_k}{\delta_r} \geq \sqrt{L}, \tag{21}$$

where S_k is the standard deviation of the incumbent at iteration k, and the right-hand sides of (19)–(21) are user-specified values that control termination of the algorithm.

Condition (21) has intuitive appeal because the left-hand side acts as a noise-to-signal ratio, which, when sufficiently large, indicates that continued sampling may return only marginal improvement. In fact, the choice of $L = 1$ in (21) can be interpreted as the point at which the noise overtakes the signal strength. This is illustrated in the numerical results of Section 5.2, the full details of which are given in [43]. Furthermore, if we make the common assumption that $S_k \to 0$ in K (i.e., as $X_k \to \hat{x}$ in K), and if $\delta_r \to 0$ is enforced at a slower rate than that of S_k, then the number of samples required by (18) can be kept under control (i.e., bounded) without enforcing (21).

We should note that problem size clearly plays a major role in determining how many function evaluations are required to achieve a specified degree of accuracy, which means that large-scale problems can require a prohibitive number of function evaluations. However, the usefulness of pattern search as an optimizer is in its treatment of problems for which no derivative information is available, rather than those of large dimension.

5.2 Test Results

The MGPS-RS algorithm was implemented using the two-stage Rinott procedure described in Section 5.1, in which values of g were computed numerically by an adaptation of the code listed in [14]. Numerical tests were performed on 22 bound and linearly constrained NLP problems obtained from [24, 40], along with 4 MVP problems described in [43]. Using (21) (with $L = 1$) as a guide to predict when the per-iteration sampling requirements might grow rapidly, the analysis was conducted by finding the first iteration k' and R&S iteration r' at which (21) holds. Let k_t denote the iteration number at termination. Furthermore, as measures of optimality, let Q and P be defined respectively by

$$Q = \frac{f(x) - f(x_*)}{f(x_0) - f(x_*)},$$

$$P = \begin{cases} \dfrac{\|x - x_*\|}{\|x_0 - x_*\|}, & \text{if } n^d = 0, \\[2ex] \dfrac{\|x^c - x^c_*\| + \min(1, |x^d - x^d_*|)}{\|x^c_0 - x^c_*\| + 1}, & \text{otherwise,} \end{cases}$$

where x is the best iterate found thus far, x_* is the true optimizer, and x_0 is the initial iterate. The measures P and Q are very much related to stopping conditions (19) and (21), respectively.

Responses are obtained by adding a normally distributed, mean-zero noise term to the underlying true objective function; i.e., $F(x, w) = f(x) + w$, where $w \sim N(0, \sigma^2(x, f))$, and the variance σ^2 of the noise depends on the true function. For comparison, we consider high and low noise cases, respectively labeled as Noise Cases 1 and 2, with corresponding standard deviations defined by

$$\begin{aligned}\sigma_1(x,f) &= \min\left(10, \sqrt{f(x) - f(x_*) + 1}\right), \\ \sigma_2(x,f) &= \max\left(0.1, \frac{1}{\sqrt{f(x) - f(x_*) + 1}}\right).\end{aligned}$$

Note that at optimality, $\sigma_1 = \sigma_2 = 1$, but they diverge to values $\sigma_1 = 10$ and $\sigma_2 = 0.1$, respectively, for trial points away from optimality.

R&S parameters used were $\delta_0 = 100$, $\alpha_0 = 0.8$, $\rho_\delta = \rho_\alpha = 0.95$, and $s_0 = 5$. Pattern search parameters were $D = [I, -I]$, $\tau = 2$, $m_k^- = -1$, $m_k^+ = 0$, and no SEARCH. The initial mesh size was different for each problem, ranging from 0.25 to 10. For the MVP problems, the extended poll trigger was also set differently for each problem, ranging from a value of 200-2000 for the first few iterations and 10-20 thereafter.

Numerical results are shown in Table 1, in which certain data were collected for each test problem at iterations k' and k_t, the latter of which occurs when $\Delta_{k_t} < \Delta_T = \frac{1}{100}\Delta_0$ is satisfied, or when the number of function evaluations exceeded 100,000 for problems of less than 30 variables and 400,000 for larger problems. For each noise case, the number of continuous variables (n^c) are given for each problem (the MVP problems each have one categorical variable that may take on 2-3 values), and averages (over 30 replications) for percent reduction in Q ($\%Q$), percent reduction in P ($\%P$), and response samples (RS) are given for iterations k' and k_t.

In seven of the test problems for Noise Case 1 and all 26 problems for Noise Case 2, the termination criteria (20) would have been satisfied for $\alpha_T = 0.01$. In 15 of these cases, plus several others, excellent progress was made toward optimality (97% reduction in Q or higher). Even more telling is that in all cases, very little progress was made between iterations k' and k_t, while using significantly more samples over fewer iterations. This is, in fact, true for nearly all of the problems, which is an indicator that (21) may indeed be a useful means for selecting a stopping point.

In some problems, particularly in Noise Case 1, some mild but not insignificant improvement was observed after iteration k' (e.g., problems 105, 118, 301, 392). In each of these cases, the average step length was still relatively high, indicating that the algorithm may still have been making many successful moves through the design space. In these situations, it would be advantageous to have a parameter update strategy that monitored the decay rate of Δ_k and adjusted the decay rate of α_r and δ_r accordingly. That is, if Δ_k shrinks slowly, then so should α_r and δ_r. This allows the algorithm to continue aggressively exploring the design space before the sampling requirements increase to prohibitive levels.

Problems 289, 300, and 301 performed quite poorly. Problem 289 is challenging because the objective function values at the starting and optimal points differ only by 0.6963. Thus, even in the low noise case, the effect of noise at different candidate designs dominates the difference in true objective function values there. Problems 300 and 301 are both instances of a gradually sloping n-dimensional quadratic function with $n - 1$ cross terms, which hinders performance because the contours of the function do not lie along the coordinate axes, which are the search directions of the algorithm.

The anomalous behavior of Problem 305 (and to a lesser extent, Problem 004, Noise Case 1) indicates that the algorithm tended to move off of a bad point to a point further away from the optimal point, but with an excellent objective function value. This can happen when a function is badly scaled, where some directions are very steep and others are very flat.

Additional (and rather exhaustive) numerical results are given in [43] and [44], in which different R&S schemes (within MGPS-RS) are compared against each other and against other well-known stochastic optimization methods.

6 Conclusion

We have presented a rigorous class of algorithms for the optimization of stochastic systems defined over mixed variable domains, and we believe it is the first-ever algorithm to treat this very general class of optimization problems. The Monte Carlo-based sampling approach is flexible in that any viable ranking and selection method can be inserted to select new iterates within the generalized pattern search framework. Under reasonable conditions, iterates generated by the algorithms converge almost surely to stationary points appropriately defined over the mixed variable domain. Numerical tests show that reasonable termination conditions can be imposed, which can provide a good measure of proximity to a local solution while controlling sample sizes.

An advantage of the approach is that, through manipulation of the R&S parameters ρ_δ and ρ_α, sampling requirements can be increased gradually as the algorithm progresses, so that excessive sampling effort is not wasted at early iterations. Furthermore, more "relaxed" settings of these parameters early in the search may allow the algorithm to avoid entrapment near suboptimal local minima, similar to the cooling schedule in simulated annealing.

References

[1] M. A. Abramson. *Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems*. PhD thesis, Rice University, Department of Computational and Applied Mathematics, 2002.

[2] M. A. Abramson. Second-order behavior of pattern search. *SIAM J. Optim.*, 16(2):515–530, 2005.

[3] M. A. Abramson, C. Audet, and J. E. Dennis, Jr. Filter pattern search algorithms for mixed variable constrained optimization. *Pacific J. Optim.*, to appear, 2007.

[4] M. A. Abramson, O. A. Brezhneva, J. E. Dennis, Jr., and R. L. Pingel. Pattern search in the presence of degeneracy. Technical Report TR03-09, Rice University, Department of Computational and Applied Mathematics, 2003.

[5] M. A. Ahmed and T. M. Alkhamis. Simulation-based optimization using simulated annealing with ranking and selection. *Comput. Oper. Res.*, 29:387–402, 2002.

[6] E. J. Anderson and M. C. Ferris. A direct search algorithm for optimization with noisy function evaluations. *SIAM J. Optim.*, 11(3):837–857, 2001.

[7] S. Andradóttir. Simulation optimization (ch. 9). In J. Banks, editor, *Handbook of Simulation*, pages 307–333, New York, 1998. John Wiley and Sons.

[8] S. Andradóttir. Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Trans. Model. Comput. Sim.*, 9(4):349–380, 1999.

[9] C. Audet and J. E. Dennis, Jr. Pattern search algorithms for mixed variable programming. *SIAM J. Optim.*, 11(3):573–594, 2000.

[10] C. Audet and J. E. Dennis, Jr. Analysis of generalized pattern searches. *SIAM J. Optim.*, 13(3):889–903, 2003.

[11] C. Audet and J. E. Dennis, Jr. A pattern search filter method for nonlinear programming without derivatives. *SIAM J. Optim.*, 14(4):980–1010, 2003.

[12] C. Audet and J. E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17(2):188–217, 2006.

[13] R. E. Bechhofer. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Annals Math. Stat.*, 25:16–39, 1954.

[14] R. E. Bechhofer, T. J. Santner, and D. M. Goldsman. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons.* John Wiley and Sons, New York, 1995.

[15] J. Boesel, B. L. Nelson, and S. H. Kim. Using ranking and selection to "clean up" after simulation optimization. *Oper. Res.*, 51(5):814–825, 2003.

[16] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. W. Moore, and D. B. Serafini. Managing surrogate objectives to optimize a helicopter rotor design — further experiments. Technical Report 98-4717, AIAA, St. Louis, September 1999.

[17] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, and V. Torczon. Optimization using surrogate objectives on a helicopter test example. In J. Borggaard, J. Burns, E. Cliff, and S. Schreck, editors, *Optimal Design and Control*, Progress in Systems and Control Theory, pages 49–58, Cambridge, Massachusetts, 1998. Birkhäuser.

[18] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Struct. Optim.*, 17(1):1–13, 1999.

[19] C. Davis. Theory of positive linear dependence. *Amer. J. Math.*, 76(4):733–746, 1954.

[20] E. D. Dolan, R. M. Lewis, and V. Torczon. On the local convergence of pattern search. *SIAM J. Optim.*, 14(2):567–583, 2003.

[21] E. J. Dudewicz and S. R. Dalal. Allocation of observations in ranking and selection. *Indian J. Stat.*, 37B(1):28–78, 1975.

[22] C. Elster and A. Neumaier. A grid algorithm for bound constrained optimization of noisy functions. *IMA J. Numer. Anal.*, 15:585–608, 1995.

[23] M. C. Fu. Optimization for simulation: Theory vs. practice. *INFORMS J. Comput.*, 14(3):192–215, 2002.

[24] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Lecture Notes in Economics and Mathematical Systems, No. 187. Springer-Verlag, Berlin, 1981.

[25] M. Kokkolaras, C. Audet, and J. E. Dennis, Jr. Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system. *Optim. Eng.*, 2(1):5–29, 2001.

[26] R. M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. Technical Report ICASE 96-71, NASA Langley Research Center, 1996.

[27] R. M. Lewis and V. Torczon. Pattern search algorithms for bound constrained minimization. *SIAM J. Optim.*, 9(4):1082–1099, 1999.

[28] R. M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM J. Optim.*, 10(3):917–941, 2000.

[29] R. M. Lewis and V. Torczon. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM J. Optim.*, 12(4):1075–1089, 2002.

[30] L. Ljung, G. Pflug, and H. Walk. *Stochastic Approximation and Optimization of Random Systems*. Birkhäuser Verlag, Berlin, 1992.

[31] S. Lucidi and V. Piccialli. A derivative-based algorithm for a particular class of mixed variable optimization problems. *Optim. Methods Softw.*, 19(3-4):371–387, 2004.

[32] S. Lucidi, V. Piccialli, and M. Sciandrone. An algorithm model for mixed variable programming. *SIAM J. Optim.*, 15(4):1057–1084, 2005.

[33] A. L. Marsden, M. Wang, J. E. Dennis, Jr., and P. Moin. Optimal aeroacoustic shape design using the surrogate management framework. *Optim. Eng.*, 5(2):235–262, 2004.

[34] B. L. Nelson, J. Swann, D. Goldsman, and W. Song. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Oper. Res.*, 49(6):950–963, 2001.

[35] S. Ólafsson. Iterative ranking-and-selection for large-scale optimization. In P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, editors, *Proc. 1999 Winter Sim. Conf.*, pages 479–485, 1999.

[36] M. S. Ouali, H. Aoudjit, and C. Audet. Optimisation des stratégies de maintenance. *Journal Européen des Systèmes Automatisés*, 37(5):587–605, 2003.

[37] J. Pichitlamken and B. L. Nelson. A combined procedure for optimization via simulation. *ACM Trans. Modeling Comput. Simulation*, 13(2):155–179, 2003.

[38] Y. Rinott. On two-stage selection procedures and related probability-inequalities. *Comm. Statist.*, A7(8):799–811, 1978.

[39] H. Robbins and S. Monro. A stochastic approximation method. *Annals Math. Stat.*, 22:400–407, 1951.

[40] K. Schittkowski. *More Test Examples for Nonlinear Programming Codes.* Lecture Notes in Economics and Mathematical Systems, No. 282. Springer-Verlag, Berlin, 1987.

[41] J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

[42] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control.* John Wiley and Sons, Hoboken, New Jersey, 2003.

[43] T. A. Sriver. *Pattern Search Ranking and Selection Algorithms for Mixed Variable Stochastic Systems.* PhD thesis, Air Force Institute of Technology, Graduate School of Engineering and Management, Wright-Patterson AFB, Ohio, 2004.

[44] T. A. Sriver and J. W. Chrissis. Combined pattern search and ranking and selection for simulation optimization. In R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, editors, *Proc. 2004 Winter Sim. Conf.*, 2004.

[45] J. R. Swisher, S. H. Jacobson, and E. Yücesan. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Trans. Modeling Comput. Simulation*, 13(2):134–154, 2003.

[46] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optim.*, 7(1):1–25, 1997.

[47] M. W. Trosset. On the use of direct search methods for stochastic optimization. Technical Report TR00-20, Department of Computational and Applied Mathematics, Rice University, Houston Texas, June 2000.

Table 1: Results for MGPS-RS using Rinott.

Noise Case 1:

					$k \leq k'$			$k' < k \leq k_t$		
Problem	n^c	Δ_T	$\Delta_{k'}$	$\alpha_{r'}$	%Q	%P	RS	%Q	%P	RS
003	2	.0050	0.00003	.00824	83.64	2.24	4127	0.00	0.00	95873
004	2	.0025	0.00001	.00886	41.55	-23.07	4113	0.00	0.00	95888
005	2	.0050	0.00006	.00912	80.30	61.20	4314	0.00	0.00	95686
025	3	.0200	0.00499	.01534	91.62	2.75	4905	0.03	0.00	95095
036	3	.0100	0.00002	.00995	99.98	99.96	4994	0.00	0.00	95006
105	8	.0025	0.07324	.08216	32.49	0.92	5787	6.51	0.11	94213
110	10	.0010	0.01633	.01390	22.42	13.44	22742	1.18	0.92	77258
118	15	.0400	1.13333	.05841	79.43	28.90	10165	5.81	3.65	89835
224	2	.0050	0.00005	.01097	99.51	89.62	4043	0.00	0.00	95958
244	3	.0200	0.00897	.01080	54.47	29.48	5308	-0.01	-0.04	94692
256	4	.0100	0.00125	.00970	99.82	85.45	8042	0.00	0.03	91958
275	4	.0100	0.01257	.00916	99.40	52.72	8311	0.01	0.40	91689
281	10	.0050	0.04167	.02213	47.04	29.31	19651	1.66	1.83	80349
287	20	.0100	0.23542	.06347	99.90	55.26	28682	0.05	0.35	71318
288	20	.0100	0.14583	.01733	99.71	81.92	55894	0.06	1.49	45449
289	30	.0010	0.05125	.01226	-0.65	-0.64	71202	-0.05	-0.04	367696
297	30	.0200	0.04401	.02796	99.90	94.16	68156	0.08	0.35	379523
300	20	.0010	0.06542	.03924	-3.45	0.00	29006	-0.10	0.03	70994
301	50	.0200	0.77083	.07253	-18.31	0.30	58528	11.69	0.08	373597
305	100	.0200	0.07917	.04353	100.00	-360.14	412328	0.00	-0.07	57354
314	2	.0025	0.00001	.00801	95.36	52.44	4539	0.00	0.00	95461
392	30	.1000	3.76823	.07644	42.11	4.97	9512	3.08	3.66	400878
MVP 1	4	.0050	0.02834	.01037	99.60	77.97	20735	0.02	0.36	79265
MVP 2	4	.0050	0.13319	.01152	99.38	76.80	45588	0.21	1.64	56787
MVP 3	20	.0050	0.02160	.02343	97.74	34.41	224914	0.01	-0.06	267036
MVP 4	20	.0050	0.21095	.02684	98.97	47.37	431675	0.02	0.15	78998

Noise Case 2:

003	2	.0050	0.00007	.00709	86.28	2.47	4529	0.00	0.00	95471
004	2	.0025	0.00000	.00747	72.49	27.73	4032	0.00	0.00	95968
005	2	.0050	0.00002	.00688	92.37	76.28	4588	0.00	0.00	95412
025	3	.0200	0.00060	.00359	87.22	20.66	8472	0.01	0.00	91528
036	3	.0100	0.00001	.00698	99.99	99.98	6313	0.00	0.00	93687
105	8	.0025	0.22292	.00138	79.87	11.23	32540	0.98	1.10	67460
110	10	.0010	0.01331	.00621	61.03	44.48	27200	0.50	0.75	72800
118	15	.0400	0.17663	.00302	95.78	48.61	35190	0.23	0.93	65495
224	2	.0050	0.00001	.00688	99.94	97.27	4502	0.00	0.00	95499
244	3	.0200	0.00152	.00708	70.75	36.50	6116	0.00	0.01	93884
256	4	.0100	0.00106	.00804	99.92	90.32	8323	0.00	0.02	91677
275	4	.0100	0.00714	.00810	99.48	53.62	8139	0.00	-0.01	91861
281	10	.0050	0.00523	.00655	93.84	88.81	24282	0.34	0.08	75718
287	20	.0100	0.00422	.00160	99.96	55.49	93786	0.00	0.00	12951
288	20	.0100	0.05443	.00791	99.97	93.41	49772	0.00	0.23	50228
289	30	.0010	0.04958	.00738	-0.17	-0.18	88385	0.06	0.06	350677
297	30	.0200	0.01393	.00757	99.99	98.06	73016	0.01	0.02	355564
300	20	.0010	0.02372	.00217	5.69	0.90	90101	0.06	0.01	17971
301	50	.0200	0.06302	.00133	9.23	1.26	341258	0.07	0.01	110212
305	100	.0200	0.00993	.00228	100.00	-345.67	493679	0.00	0.01	8654
314	2	.0025	0.00002	.00770	97.95	64.75	4182	0.00	0.00	95818
392	30	.1000	1.25000	.00096	53.67	15.29	74923	0.59	0.64	368260
MVP 1	4	.0050	0.01477	.00787	99.83	81.58	8617	0.01	0.02	91383
MVP 2	4	.0050	0.14147	.00487	99.77	80.65	52684	0.02	0.37	54461
MVP 3	20	.0050	0.00329	.00687	99.62	84.18	300380	0.00	0.02	195053
MVP 4	20	.0050	0.00767	.00666	99.79	82.25	312967	0.00	0.02	182272