Proceedings of the 2004 Winter Simulation Conference
R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

COMBINED PATTERN SEARCH AND RANKING AND SELECTION FOR SIMULATION OPTIMIZATION

Todd A. Sriver
James W. Chrissis

Department of Operational Sciences
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765, U.S.A.

ABSTRACT

A new algorithm class is presented for optimization of stochastic simulation models. The algorithms, which combine generalized pattern search (GPS) with ranking and selection (R&S), require "black-box" simulation evaluations and are applicable to problems with mixed variables (continuous, discrete numeric, and categorical). Implementation of the Mixed-variable Generalized Pattern Search with Ranking and Selection (MGPS-RS) algorithm with three different R&S procedures is demonstrated and tested on a small set of standard test functions. Results of this preliminary performance evaluation are summarized and compared with existing search methods.

1 DISCLAIMER

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

2 INTRODUCTION

Consider the optimization of a stochastic system in which the objective is to find the combination of controllable system parameters (design variables) that minimizes a system performance measure of merit. For highly complex systems, simulation is often used as a modelling and analysis tool in which performance is evaluated from a set of sampled output responses that estimate the performance measure of interest. We consider the *simulation optimization* problem of the following form,

$$\min_{x \in \Theta} E[F(x,\omega)], \qquad (1)$$

where the minimum performance is sought for the output response $F(x,\omega)$, x is an n-dimensional design vector from feasible domain Θ, and ω is a vector of random elements representing inherent variation in the system.

We treat simulation as a "black-box" procedure that takes as input the vector x and produces the output response $F(x,\omega)$. The fundamental properties of problem (1) that make it difficult are that the analytical form of $E[F(x,\cdot)]$ is unknown and the response $F(x,\cdot)$ is "contaminated" by random variation. Thus, $E[F(x,\cdot)]$ cannot be evaluated exactly, but must instead be estimated via samples of $F(x,\cdot)$, thereby reducing the number of designs that can be visited given a fixed computational budget.

Additional complications arise when elements of the design vector can be discrete, either discrete-numeric (*e.g.*, integer-valued) or *categorical*. Categorical variables are those which must take their value from a predefined list or set. Such values may not even be numeric. These restrictions are common for realistic stochastic systems. For example, a 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. The class of optimization problems that includes continuous, discrete-numeric and categorical variables is known as *mixed variable programming* (MVP) problems (Audet and Dennis 2000). We group discrete-numeric and categorical variables into a "discrete" variable class by mapping categorical variables to discrete numerical values. For example, integer values are assigned to the queue discipline categorical variable (*e.g.*, 1 = FIFO, 2 = LIFO, and 3 = priority), even though the values do not conform to the inherent ordering that the numerical value suggests.

To define a mixed-variable domain, Θ is partitioned into continuous and discrete subdomains Θ^c and Θ^d, respectively. A design point $x \in \Theta$ 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 continuous variable domain is expressed as $\Theta^c = \{x^c \in \mathbb{R}^{n^c} : \ell \leq Ax^c \leq u\}$, where $A \in \mathbb{R}^{m^c \times n^c}$,

$\ell, u \in \mathbb{R}^{m^c}$, and $\ell < u$. Hence, Θ^c is subject to bound and linear constraints. The discrete variable domain is expressed as a subset of the integers, $\Theta^d \subseteq \mathbb{Z}^{n^d}$. The iterates of sequential search methods for such problems may be considered a sequence of random vectors. Hence, we use the conventional notation X_k to denote a random quantity for the design at iteration k to distinguish it from the x_k used to denote a realization of X_k.

Numerous search techniques with rigorous convergence analyses have been devised for "black-box" simulation optimization problems. Presently, the predominant methods are applicable to search domains that are either entirely continuous or entirely discrete. For example, over continuous domains, gradient-free versions of stochastic approximation (SA) that use finite differences (Kiefer and Wolfowitz 1952) or simultaneous perturbations (Spall 1992) to approximate the direction of steepest descent have well-developed convergence theories. Methods for discrete domains include certain random search approaches (Andradóttir 1999) in which the convergence analyses typically rely on modelling the iterate sequence as a discrete time Markov chain.

Mixed-variable problems present unique challenges for search routines. Relaxation techniques commonly used for mixed integer problems, such as branch and bound, are not applicable to the mixed-variable case because the response output is 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 be treated by exhaustively enumerating their possible values, but this approach quickly becomes computationally prohibitive. More desirable is a method that can systematically search over the mixed-variable domain.

In this paper, we present a method for simulation optimization over mixed variables, for which the continuous variable values are restricted by bound and linear constraints. The method extends the class of mixed variable *generalized pattern search* (GPS) algorithms described in Audet and Dennis (2000) and Abramson (2002) to stochastic response functions by employing ranking and selection (R&S) procedures in the selection of new iterates as a means of error control during the search. In Sriver, Chrissis, and Abramson (2004), we show that the method converges almost surely to appropriately defined stationary points over the mixed-variable domain. In this paper, we focus on implementation issues and report on some initial computational results.

The remainder of this paper is organized as follows. Section 3 introduces pattern search and its extensions to MVPs. Section 4 presents a new class of GPS algorithms for simulation optimization and Section 5 discusses important implementation considerations for the algorithms. Section 6 reports some initial computational results and Section 7 offers conclusions.

3 PATTERN SEARCH

Pattern search is a subclass of direct search algorithms, which involve the direct comparison of objective function values and do not require the use of explicit or approximate derivatives. Torczon (1997) introduced the general class of pattern search methods for unconstrained optimization, demonstrating that the class of methods unified various distinct direct search techniques, such as the pattern search of Hooke and Jeeves (1961).

3.1 Continuous Variable GPS

Pattern search over continuous variables is defined via a finite set of directions used at each search iteration. The direction set and a step length parameter define a conceptual mesh centered about the current iterate (the incumbent). Trial points are selected from the mesh, evaluated, and compared to the incumbent in order 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.

The key to generating the mesh is the definition of the direction set. This set must be sufficiently rich to ensure that at least one of the directions is one of descent. Lewis and Torczon (1996) applied the theory of positive linear dependence (Davis 1954) to establish criteria for the direction set; namely, that the set of directions must *positively span* the domain $\mathbb{R}^{n^c}$. A *positive spanning set* allows any vector originating from the design point to be formed as a nonnegative linear combination of the set. Therefore, if the gradient of the unknown objective function is nonzero at any iterate, at least one element of the direction set is a descent direction.

For example, consider the commonly used direction set $D = [I, -I]$ where the columns of D denote the positive and negative coordinate axes as the search directions. This situation is demonstrated for two dimensions in Figure 1 where x_k is the incumbent iterate and the points $\{a, b, c, d\}$ are trial points along the search directions parameterized by step length Δ_k.

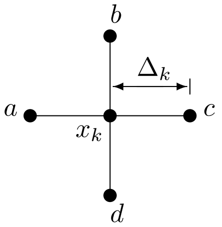


Figure 1: Example Pattern

Lewis and Torczon extended pattern search to bound constrained (1999) and problems with a finite number of linear constraints (2000). In these situations, the direction set is required to include search directions that conform to the geometry of the constraint boundaries. Such directions are referred to as *tangent cone generators* and an algorithm for computing these directions in the absence of degeneracy is given in Lewis and Torczon (2000). With conforming directions included, linear constraints can be treated with the simple "barrier" approach. That is, if a linear constraint is violated at a trial point, then a function value of $+\infty$ is assigned without computing the objective function value there, thus saving computational expense.

3.2 Mixed Variable GPS

A GPS framework for MVP problems with bound constraints was developed by Audet and Dennis (2000) and further extended to linear constraints in Abramson (2002). In this case, a set of positive spanning directions is denoted as D^i for each unique combination $i = 1, 2, \ldots, i_{\max}$ of discrete variable values.

Audet and Dennis explicitly separate the search into two distinct steps, a search step and a poll step. The optional search step selects a finite number of trial points from an implicitly defined mesh that spans the entire search space. This step contributes nothing to the convergence theory, but allows the user great flexibility to apply any desired heuristic to speed convergence. In this paper, we employ an empty search step and therefore exclude it from the algorithm listing of Section 4.

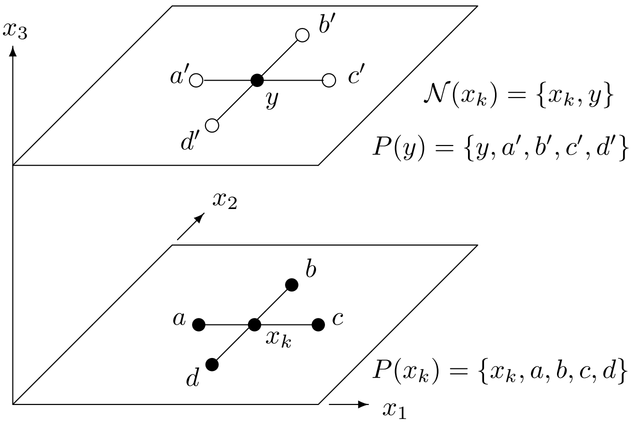
The poll step evaluates a subset of neighboring mesh points known as the *poll set*. Each point in this set must be evaluated before the iteration can be declared unsuccessful. Polling is conducted in up to three stages: polling with respect to the continuous variables, polling on a set of discrete neighbors, 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, trial points are defined by,

$$P_k = \{x_k + \Delta_k(d, 0) : d \in D_k^i\}, \tag{2}$$

where $(d, 0)$ denotes the partitioning into continuous and discrete variables and 0 means the discrete variables remain unchanged, *i.e.*, $x_k + \Delta_k(d, 0) = (x_k^c + \Delta_k d, x_k^d)$. The notation $d \in D_k^i$ means that d is a column of D_k^i. Polling with respect to discrete variables requires a user-defined discrete set of neighbors at x_k, denoted as $\mathcal{N}(x_k)$. If the first polling stage does 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 $f(x_k) < f(y) \leq f(x_k) + \xi_k$, for some user-specified extended poll trigger $\xi_k > \xi > 0$ bounded away from zero.

An example poll step is depicted in Figure 2 for two continuous variables (x_1, x_2) and one discrete variable x_3. The two planes represent each of two settings for the discrete variable. The poll set about the incumbent consists of the points $\{x_k, a, b, c, d\}$ and the discrete neighbor set consists of the points $\{x_k, y\}$. If the poll set and discrete neighbor set fail to produce an improvement on x_k, and if discrete neighbor y produces an objective function value sufficiently close to that of the incumbent, then a polling sequence is initiated about y. In the figure, the set of points $\{y, a', b', c', d'\}$ indicates the initial poll set about y to be evaluated. The extended polling sequence continues until a point is found in the upper plane that improves upon x_k or until no further improvement in that plane is possible.



x_3
b'
a'
c'
y
d'
$\mathcal{N}(x_k) = \{x_k, y\}$
$P(y) = \{y, a', b', c', d'\}$
x_2
b
a
c
x_k
d
$P(x_k) = \{x_k, a, b, c, d\}$
x_1

Figure 2: Poll Step Example

The mesh is updated (refined, coarsened, or retained) according to a strict set of rules. Refinement must satisfy,

$$\Delta_{k+1} = \tau^{m_k^-}\Delta_k, \tag{3}$$

where $\tau > 1$ is rational and fixed over all iterations, $0 < \tau^{m_k^-} < 1$, and m_k^- is an integer satisfying $m_{\min} \leq m_k^- \leq -1$ for some fixed integer $m_{\min} \leq -1$.

Coarsening after a successful poll or extended poll step is accomplished by,

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

where $\tau > 1$ is defined as above and m_k^+ is an integer satisfying $0 \leq m_k^+ \leq m_{\max}$ for some fixed integer $m_{\max} \geq 0$.

4 THE MGPS-RS ALGORITHM

For problems with random responses, single-sample response comparisons used in traditional pattern search can lead to many erroneous iterate selection decisions. Alternative comparison techniques are necessary to ensure that these decisions account for variation and provide some statistical assurances of correct decisions. In an approach by Trosset (2000), 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. The new class of algorithms is called Mixed-variable Generalized Pattern Search with Ranking and Selection (MGPS-RS). Prior to presenting the algorithms in detail in Section 4.2, we first discuss the general considerations necessary to include R&S in a GPS framework.

4.1 Ranking and Selection

Ranking and selection procedures are useful for selecting the best system design from among a set of a small number of alternatives. For a detailed survey of such procedures, see Swisher, Jacobson, and Yücesan (2003). 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 [*e.g.* see Pichitlamken and Nelson (2003) and Ahmed and Alkhamis (2002)].

Within the pattern search framework, we employ *indifference zone* R&S to select the best candidate design from a set of trial points considered simultaneously. Let $C = \{Y_1, Y_2, \ldots, Y_{n_C}\}$ be a set of candidate designs, including the incumbent, such that $n_C \geq 2$. For each $q = 1, 2, \ldots, n_C$, let $f_q = E[F(Y_q, \cdot)]$ denote the true mean of the response $F(Y_q, \cdot)$. The collection of these means can be ordered from minimum to maximum as,

$$f_{[1]} \leq f_{[2]} \leq \cdots \leq f_{[n_C]}. \tag{5}$$

If one or more candidates have true means within a practical tolerance of the true best candidate, *i.e.* $f_{[i]} - f_{[1]} < \delta$ for some $\delta > 0$ and $2 \leq i \leq n_C$, then the procedure is said to be *indifferent* in choosing Y_i as the best. 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, 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

$$\begin{aligned} P\{CS\} &= P\{\text{select } Y_{[1]} \mid f_{[q]} - f_{[1]} \geq \delta; 2 \leq q \leq n_C\} \\ &\geq 1 - \alpha, \end{aligned} \tag{6}$$

where δ and α are user specified.

Of course, when using simulation to evaluate system performance, it is necessary to work with sample means of the response $F(\cdot, \cdot)$. For each $q = 1, 2, \ldots, n_C$, let s_q be the total number of samples and let $\{F_{qs}\}_{s=1}^{s_q} = \{F(Y_q, \cdot)\}_{s=1}^{s_q}$ be the set of responses obtained via simulation. For each q, we assume the responses $\{F_{qs}\}_{s=1}^{s_q}$ are independent, identically and normally distributed random variables with mean f_q and unknown variance $\sigma_q^2 < \infty$, where $\sigma_\ell^2 \neq \sigma_q^2$ whenever $\ell \neq q$.

The R&S procedure determines how many samples s_q are required to guarantee $P\{CS\}$ (6). Then for each q, the sample means $\overline{F}_q = s_q^{-1} \sum_{s=1}^{s_q} F_{qs}$ are computed, ordered and indexed the same way as in (5). 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. For the purpose of inclusion in a class of algorithms defined in Section 4.2, we denote procedure $RS(C, \alpha, \delta)$ as a generic R&S procedure that takes as input a candidate set, significance level, and indifference zone parameter setting and returns $\hat{Y}_{[1]} = \arg(\overline{F}_{[1]})$ as the candidate having the δ-near-best mean.

4.2 Algorithm Listing

The Mixed-variable GPS Ranking and Selection (MGPS-RS) algorithm class follows for mixed-variable simulation optimization problems. The algorithms replace binary comparisons of incumbent and trial designs from traditional GPS methods with R&S procedures. The R&S procedures provide error control by ensuring sufficient sampling of the candidates so that the best or near-best is chosen with a user specified probability. The algorithm class 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.

Step 0 : *Initialization*. Set the iteration counter k to 0. Set the R&S counter r to 0. Choose a feasible starting point $X_0 \in \Theta$. Set $\Delta_0 > 0$, $\xi > 0$, $\alpha_0 \in (0, 1)$, and $\delta_0 > 0$.

Step 1 : `poll` step. Set extended poll trigger $\xi_k \geq \xi$. Use procedure RS($P_k(X_k) \cup \mathcal{N}(X_k), \alpha_r, \delta_r$) where $P_k(X_k)$ is defined in (2) to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, 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), and $k = k + 1$ and return to Step 1. Otherwise, proceed to Step 2.

Step 2 : `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) Use procedure RS($P_k(Y_k^j), \alpha_r, \delta_r$) to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, 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 2a. Otherwise, set $Z_k = Y_k^j$ and proceed to Step 2b.

(b) Use procedure RS($X_k \cup Z_k, \alpha_r, \delta_r$) to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, 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), and $k = k+1$ and return to Step 1. Otherwise, repeat Step 2 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 (3), and $k = k + 1$ and return to Step 1.

In the `poll` step, the entire poll set about the incumbent (2) and the discrete neighbor set are considered simultaneously. If the `poll` step is unsuccessful, the `extended poll` step conducts a polling sequence that searches the continuous neighborhood of any discrete neighbor with a response mean sufficiently close to the response mean of the incumbent. This step is divided into sub-steps in order to account for the sequence of R&S procedures that may be necessary. In Step 2a, 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 2b.

The *extended poll trigger* ξ_k has important implications for solution quality and algorithm performance. If it is set too high, this results in more extended poll steps and thus a potentially better solution. However, the additional sampling required at the extra points increases computational expense, particularly with high noise levels in the response output.

The update rules for Δ_k in the algorithm are the same as for the deterministic case. Refinement (3) is accomplished after both the `poll` and `extended poll` steps are unsuccessful. Coarsening (4) is accomplished after any successful `poll` or `extended poll` step.

The algorithm maintains a separate counter for R&S parameters α_r and δ_r in order to enforce strict rules on these parameters that are updated after each execution of the R&S procedure. The rules ensure that $\alpha_r \to 0$ and $\delta_r \to 0$ as the number of iterations approaches infinity, which is critical to the convergence theory. An additional restriction on α_r is that the infinite series $\sum \alpha_r$ converges; that is, $\sum_{r=1}^{\infty} \alpha_r < \infty$.

The restriction on δ_r ensures that the indifference zone condition is satisfied as r gets large if the true best design from candidate set C considered by $RS(C, \alpha_r, \delta_r)$ is a singleton for all but a finite number of iterations and sub-iterations. The restriction on α_r ensures that, with probability one, the number of incorrect selections by the algorithm is finite. Furthermore, by assuming all iterates belong to a compact set, the number of consecutive successful `poll` or `extended poll` steps is finite with probability one. By coupling the "finiteness" of incorrectly selected iterates with existing GPS convergence theory, we are able to show that MGPS-RS iterates converge almost surely to limit points that satisfy first-order necessary conditions for optimality defined over the mixed-variable domain (Sriver, Chrissis, and Abramson 2004).

In practice, the restriction on the R&S parameters can be enforced via an appropriately selected update rule. For example, the update rule $\alpha_r = \alpha_0 \rho^r$ for $0 < \rho < 1$ and $\alpha_0 > 0$ results in a convergent geometric series since $\sum_{r=1}^{\infty} \alpha_r = \frac{\alpha_0}{1-\rho} < \infty$. Using this rule with $\alpha_0 < 1$, as required by the R&S procedure, the rate at which α_r converges to zero can be controlled by the parameter ρ.

5 IMPLEMENTATION CONSIDERATIONS

Of primary concern for implementation is the selection of specific R&S procedures. Since we allow for unknown and unequal variances, a procedure of at least two stages is required so that the sample variance can be computed in an initial stage. Three such procedures were selected for implementation in this study.

The first, Rinott's two-stage procedure (Rinott 1978), is a well-known simple procedure that satisfies (6). In the first stage, the sample variance S_q^2 for each candidate q is computed from a fixed number s_0 of response samples for each candidate. Then, $s_q - s_0$ additional samples are prescribed for the second stage, where

$$s_q = \max\{s_0, \lceil (hS_q/\delta)^2 \rceil\} \tag{7}$$

that depends on Rinott's constant $h = h(n_C, \alpha, s_0)$. Although tabulated values for h have been published for commonly used parameter combinations, it was computed numerically in this study by adapting code listed in (Bechhofer, Santner, and Goldsman 1995) to accommodate changing parameter settings.

Rinott's procedure can be computationally inefficient because it is constructed based on the *least favorable configuration* assumption that the best candidate has a true mean exactly δ better than all remaining candidates, which are tied for second best (Swisher, Jacobson, and Yücesan 2003). As a result, the procedure can overprescribe the number of required second stage samples in order to guarantee the $P\{CS\}$. Furthermore, the procedure has no mechanism to consider the sample mean of the responses after the first stage, and therefore cannot eliminate clearly inferior candidates prior to conducting additional sampling. These characteristics are especially problematic in the present setting since the R&S procedure is executed repeatedly and the number of unnecessary samples accumulates at each iteration, limiting the progress of the algorithm relative to a fixed budget of response samples.

The second procedure implemented for this study alleviates some of the computational concerns of Rinott's procedure. The screen-and-select procedure of Nelson et al. (2001) combines Rinott's procedure with a screening procedure that can eliminate some solutions after the first stage. For an overall significance level α, significance levels α_1 and α_2 are chosen for screening and selection, respectively, such that $\alpha = \alpha_1 + \alpha_2$. After collecting s_0 samples of each candidate in the first stage, those candidates with a sample mean that is significantly inferior to the best of the rest are eliminated from further sampling. The set of surviving candidates is guaranteed to contain the best with probability at least $1 - \alpha_1$ as long as the indifference zone condition is met. Then, $s_q - s_0$ second stage samples are required only for the survivors according to (7) except at significance level α_2 instead of α. Nelson et al. (2001) prove that the combined procedure satisfies (6).

The third procedure implemented for this study extends the notion of intermediate elimination of inferior solutions. The Sequential Selection with Memory (SSM) procedure of Pichitlamken and Nelson (2001) is a *fully sequential* procedure specifically designed for simulation optimization search routines. A fully sequential procedure is one that takes one sample at a time from every candidate still in play and eliminates clearly inferior ones as soon as their inferiority is apparent. In SSM, an initial stage of sampling is conducted to estimate the variances between each pair of candidates. This is followed by a sequence of screening steps that eliminate candidates whose cumulative sums exceed the best of the rest plus a tolerance level that depends on the variances and parameters δ and α. Between each successive screening step, one additional sample is taken from each survivor and the tolerance level decreases. The procedure terminates when only one survivor remains or after exceeding a maximum number of samples determined after the initial stage. In the latter case, the survivor with the minimum sample mean is selected as the best. Pichitlamken (2002) proves that SSM satisfies (6). An advantage of this method is that previously sampled responses can be re-used if a design point is revisited, leading to further computational savings. If previous samples are not used, the procedure is the same as the one in Kim and Nelson (2001).

A final implementation consideration concerns the choice of starting values and the rate of decay for R&S parameters δ_r and α_r. Smaller values for either parameter lead to larger sampling requirements. For example, this can be seen for Rinott's procedure by observing (7) and noting that h increases with $(1 - \alpha)$. In practice, it is desirable to avoid excessive sampling in regions of the search space far from optimality. An advantage of the MGPS-RS algorithms is that through manipulation of these parameters, the sampling requirements can be increased gradually as the algorithm progresses, so that excessive sampling effort is not wasted at early iterations. In our numerical experiments, the initial values δ_0 and α_0 are set very "loose" so that, in the early iterations, no samples are taken beyond the initial s_0 required for each candidate in all three procedures used. Each parameter is reduced geometrically with r so that error control of iterate selection increases as the search moves toward the region of optimality.

6 NUMERICAL EXPERIMENTS

Algorithm performance was conducted for unconstrained problems over continuous variables for each of three implementations, which we designate as MGPS-RIN (Rinott), MGPS-SAS (screen-and-select), and MGPS-SSM (sequential selection with memory). For comparison to other methods, three additional algorithms were also included in the experiments: finite-difference stochastic approximation (FDSA), simultaneous perturbation stochastic approximation (SPSA), and simplex search (NM) of Nelder and Mead (1965). Code for FDSA and SPSA was obtained from the web site associated with (Spall 2003). Code for NM was adapted from the `Matlab` function `fminsearch`.

The test functions included the extended Rosenbrock function and the extended Powell singular function, both of which are described in (Moré, Garbow, and Hillstrom 1981). The dimension can be varied, so a 4- and 20-dimensional version of each function was tested. To imitate random responses from a simulation, noise was added to each function evaluation according to a zero-mean normal distribution $N(0, \sigma^2(f(x)))$. To compare two different random noise scenarios, the standard deviation of the error term $\sigma(f(x))$ was either proportional or inversely proportional to f, but bounded on the range (.1, 10):

$$\sigma_1(f(x)) = \min\left(10, \sqrt{f(x)}\right), \quad \text{or}$$
$$\sigma_2(f(x)) = \max\left(0.1, \frac{1}{\sqrt{f(x)}}\right).$$

We refer to these test cases as noise cases 1 and 2, respectively. The test functions were slightly altered to make the optimal objective function value equal unity so that the standard deviation approached one from above (below) for noise case 1 (case 2) as algorithms neared optimality.

For the MGPS algorithms, the direction set consisted of the coordinate axes $D = [I, -I]$ for all k. The step size parameters were set to $\tau = 2$, $m_k^- = -1$, and $m_k^+ = 1$ for all k so that $\Delta_{k+1} = \frac{1}{2}\Delta_k$ for refinement and $\Delta_{k+1} = 2\Delta_k$ for coarsening. The initial step size was set to $\Delta_0 = 2$. The R&S parameters were updated as $\delta_r = \delta_0 \rho^r$ and $\alpha_r = \alpha_0 \rho^r$ using $\rho = 0.95$ and initial values $\delta_0 = 100$ and $\alpha_0 = 0.8$. First stage samples were set to $s_0 = 5$.

As recommended in Spall (2003), the step sizes for FDSA and SPSA were updated according to $a_k = a/(k + 1 + A)^{0.602}$ with stability constant $A = 100$ for FDSA and $A = 1,000$ for SPSA. The initial 500 response samples were used to estimate a by the method suggested in Spall (2003, p. 165) and using the initial desired change in magnitude of the elements in the design vector of 0.25 for SPSA and, in the case of FDSA, 0.5 (Rosenbrock) or 1.5 (Powell). The perturbation distance parameter was updated as $c_k = 0.447/(k + 1)^{0.101}$. For SPSA the random perturbation direction vector was determined from a Bernoulli ± 1 distribution with probability $\frac{1}{2}$ for each outcome. Each point in the differencing formula was averaged over $s_0 = 5$ response samples.

The Nelder-Mead algorithm was used in its original form with the following modifications. As suggested by Barton and Ivey (1996), the shrink parameter was adjusted to 0.9 (from 0.5) and the best point was resampled after a shrink. In addition, each point was evaluated based on an average over $s_0 = 5$ samples. The initial simplex was constructed using the starting point plus n points a distance of 4 units in the direction of the coordinate axes from the starting point.

For each combination of test function, dimension, and noise case, each algorithm was run for 100,000 response evaluations using the starting points in (Moré Garbow, and Hillstrom. 1981). The difference in true objective function value from the optimum was measured at three stages of computational budget: 1,000, 10,000 and 100,000 response evaluations. The results, averaged over 30 replications, are presented in Tables 1 and 2.

The results demonstrate the computational improvements achieved using screening and fully sequential R&S approaches for the high variance case (noise case 1). In all four test function and dimension combinations under this case, MGPS-SAS achieved a better mean objective function value than MGPS-RIN after 10,000 response evaluations where MGPS-SSM achieved a better result in three of the combinations. For the 20-dimensional Powell function, MGPS-SSM was hampered by homogeneity among a subset of candidate points in early iterations, causing an increase in sampling required to select the best and thus slowing progress relative to the other two methods. By 100,000 response evaluations, the standard deviation nears unity and the distinction between the three methods is less clear. The near equivalence of all three methods in the lower variance case (noise case 2) indicates that not many samples beyond the initial stage samples were necessary and the methods become essentially the same.

In every noise, test function, and dimension combination, SPSA outperformed all of the other algorithms. This is a tribute to its efficient method of estimating the gradient, in which only $2 \times s_0$ samples are needed at each iteration. The FDSA algorithm outperformed all MGPS algorithms for the Powell function but was nearly equivalent for the Rosenbrock function. This seems reasonable since Rosenbrock's function is known to be difficult for gradient-based methods because the gradient between subsequent iterations become nearly orthogonal along the valley en route to the optimum. Also, FDSA samples at $n \times s_0$ points which is nearly the same as MGPS using $D_k = [I, -I]$ when the noise is low. One weakness of FDSA and SPSA is that if the initial desired change in magnitude of the design vector is set too high, the algorithms can diverge badly. Careful tuning is required beforehand to ensure this doesn't happen. The MGPS algorithms are robust to such initial parameter settings, although the performance can be significantly degraded with poorly chosen settings.

Finally, all MGPS algorithms outperform the similar direct search method NM in this limited numerical study, especially with increasing noise. As discussed in Barton and Ivey (1996), NM can be adversely affected in noisy cases because inaccurate relative ranks of the points in the simplex can lead to inappropriate contract and shrink steps, causing it to converge much too early. This is evident from Tables 1 and 2: the objective function value does not improve after 10,000 response evaluations for any of the test cases.

7 CONCLUSIONS

We have presented a convergent, gradient-free class of algorithms for simulation optimization that employs ranking and selection to control the error of selecting iterates from a set of candidate designs. It is flexible in that any viable R&S method can be inserted within the pattern search framework. It is also more general than many existing algorithms because it allows for simple linear constraints and mixed variables. In a limited computational evaluation, The MGPS-RS class of algorithms performed better than a similar direct search method and comparably with a finite-differencing stochastic approximation algorithm.

Future work will focus on techniques to accelerate convergence, developing effective termination criteria, and a more comprehensive computational evaluation.

Table 1: True Objective Function Difference From Optimum Averaged Over 30 Replications – Rosenbrock Test Function

n	Noise Case	Budget	MGPS-RIN	MGPS-SAS	MGPS-SSM	FDSA	SPSA	NM
4	1	1,000	0.72	0.62	0.66	5.75	4.88	1.24
		10,000	0.73	0.44	0.22	0.71	0.15	1.19
		100,000	0.16	0.18	0.11	0.44	0.05	1.19
	2	1,000	0.42	0.36	0.38	5.69	4.64	0.73
		10,000	0.15	0.20	0.16	0.72	0.05	0.71
		100,000	0.09	0.10	0.10	0.44	0.02	0.71
20	1	1,000	57.1	56.9	56.9	43.6	32.6	45.0
		10,000	11.6	9.71	9.18	12.1	1.71	42.5
		100,000	2.81	2.96	1.89	3.20	0.54	42.5
	2	1,000	56.9	56.9	56.9	41.0	28.0	42.4
		10,000	2.15	2.23	2.22	8.10	0.29	14.6
		100,000	1.29	1.29	1.17	3.03	0.01	14.6

Table 2: True Objective Function Difference From Optimum Averaged Over 30 Replications – Powell Test Function

n	Noise Case	Budget	MGPS-RIN	MGPS-SAS	MGPS-SSM	FDSA	SPSA	NM
4	1	1,000	0.82	0.52	0.95	7.86	8.28	9.00
		10,000	0.43	0.18	0.13	0.34	0.11	8.87
		100,000	0.10	0.06	0.04	0.005	0.002	8.87
	2	1,000	0.13	0.21	0.20	7.77	8.15	0.18
		10,000	0.08	0.09	0.08	0.34	0.11	0.17
		100,000	0.04	0.04	0.03	0.004	8e-5	0.17
20	1	1,000	820	820	820	117	57.6	58.2
		10,000	16.9	13.4	22.8	15.5	2.64	41.7
		100,000	7.24	3.74	7.92	0.47	0.02	41.7
	2	1,000	819	819	819	115	59.4	54.2
		10,000	15.3	14.6	15.0	15.1	2.59	3.99
		100,000	1.38	0.80	1.26	0.42	0.003	3.99

ACKNOWLEDGMENTS

This work was sponsored, in part, by the Air Force Office of Scientific Research.

REFERENCES

Abramson, M. A. 2002. Pattern search algorithms for mixed variable general constrained optimization problems. PhD Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas.

Ahmed, M. A., and T. M. Alkhamis. 2002. Simulation-based optimization using simulated annealing with ranking and selection. *Computers and Operations Research* 29 : 387–402.

Andradóttir, S. 1999. Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation* 9 (4) : 349–380.

Audet, C., and J. E. Dennis, Jr. 2000. Pattern search algorithms for mixed variable programming. *SIAM Journal on Optimization* 11 (3) : 573–594.

Barton, R. R., and J. S. Ivey, Jr. 1996. Nelder-Mead simplex modifications for simulation optimization. *Management Science* 42 (7) : 954–973.

Bechhofer, R. E., T. J. Santner, and D. M. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Se-*

lection, Screening, and Multiple Comparisons. New York: John Wiley & Sons.

Davis, C. 1954. Theory of positive linear dependence. *American Journal of Mathematics* 76 (4) : 733–746.

Hooke, R., and T. A. Jeeves. 1961. "Direct search" solution of numerical and statistical problems. *Journal of the Association of Computing Machinery* 8 : 212–229.

Kiefer, J., and J. Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics* 23 : 462–466.

Kim, S. -H., and B. L. Nelson. 2001. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation* 11 (3) : 251–273.

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

Lewis, R. M., and V. Torczon. 1999. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization* 9 (4) : 1082–1099.

Lewis, R. M., and V. Torczon. 2000. Pattern search algorithms for linearly constrained minimization. *SIAM Journal on Optimization* 10 (3) : 917–941.

Moré, J. J., B. S. Garbow, and K. E. Hillstrom. 1981. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software* 7 (1) : 17–41.

Nelder, J. A., and R. Mead. 1965. A simplex method for function minimization. *The Computer Journal* 7 (4) : 308–313.

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

Pichitlamken, J. 2002. A combined procedure for optimization via simulation. PhD Thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois.

Pichitlamken, J., and B. L. Nelson. 2001. Selection-of-the-best procedures for optimization via simulation. In *Proceedings of the 2001 Winter Simulation Conference* ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 401–407. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Pichitlamken, J., and B. L. Nelson. 2003. A combined procedure for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 13 (2) : 155–179.

Rinott, Y. 1978. On two-stage selection procedures and related probability-inequalites. *Communications in Statistics* A7 (8) : 799–811.

Spall, J. C. 1992. Multivariate stochastic approximation using simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control* 37 (3) : 332–341.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization*. New Jersey: John Wiley & Sons. `Matlab` code available online via `<http://www.jhuapl.edu/ISSO/>` [accessed March 19, 2004].

Sriver, T. A., J. W. Chrissis, and M. A. Abramson. 2004. Pattern search ranking and selection algorithms for mixed variable stochastic optimization. Working paper, Department of Operational Sciences, Air Force Institute of Technology, Dayton, Ohio, submitted to *Mathematical Programming*.

Swisher, J. R., S. H. Jacobson, and E. Yücesan. 2003. Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures. *ACM Transactions on Modeling and Computer Simulation* 13 (2): 134–154.

Torczon, V. 1997. On the convergence of pattern search algorithms. *SIAM Journal on Optimization* 7 (1) : 1–25.

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

AUTHOR BIOGRAPHIES

TODD A. SRIVER is a Ph.D. candidate in Operations Research at the Air Force Institute of Technology (AFIT). He received a B.S. in Aeronautical and Astronautical Engineering from Purdue University (1990) and a M.S. in Operations Research from AFIT (1998). His research focuses on the application of direct search to optimization of stochastic systems. His e-mail address is `<Todd.Sriver@afit.edu>`.

JAMES W. CHRISSIS is an Associate Professor of Operations Research at the Air Force Institute of Technology (AFIT). He has a B.S. in Mathematics from the University of Pittsburgh (1975) and a M.S. (1977) and Ph.D. (1980) in Industrial Engineering and Operations Research from Virginia Tech. His research interests include industrial engineering and operations research, engineering optimization, mathematical programming, stochastic systems, and simulation. His e-mail address is `<James.Chrissis@afit.edu>`.