

A Framework for Mixed-Variable Optimization Under Uncertainty Using Surrogates and Statistical Selection

Todd A. Srivier * and James W. Chrissis †

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

A framework for engineering design optimization is presented that combines generalized pattern search (GPS) with ranking and selection (R&S) and surrogate function approximations. This framework is applied to problems with mixed variables (continuous, discrete numeric, and categorical), that are characterized by inherent variation in system performance. The class of algorithms will be described and illustrated on a small numerical example.

I. Introduction

Realistic engineering design models are characterized by several distinguishing features. In the optimization of such models, traditional approaches have assumed away some of these features, the goal being to simplify and reduce the expense of search procedures. In this paper, we present a rigorous framework for optimization of design models that relax some restrictions previously enforced. In particular, the approach allows for (a) a mixture of design variable types, and (b) uncertainty in the design in the form of inherent variability in system performance. The target problem class is defined as,

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

where the minimum long-run expected performance is sought for the performance measure $F(x, \omega)$, x is an n -dimensional design vector from a feasible set Θ of the design variable domain, and ω is a vector of random elements representing the inherent variation in performance. The performance F is *stochastic* in the sense that repeated evaluation for a given design vector will produce different, random values.

Variables of problem (1) include continuous, discrete-numeric (*e.g.*, integer-valued) or discrete-categorical variables. Categorical variables are those that can only take on values from a predefined list and have no ordinal relationship to one another. For example, the type of material used in an engineering model may be a categorical variable. The class of optimization problems that includes continuous, discrete-numeric and categorical variables is known as *mixed variable programming* (MVP) problems. Discrete-numeric and categorical variables are grouped into a “discrete” variable class by mapping categorical variables to discrete numerical values, even though the values do not conform to the inherent ordering that the numerical value suggests.

For problem (1), it is assumed that the underlying geometry of the *true* performance measure function $\mathbb{E}[F]$ is unknown and therefore cannot be evaluated analytically; it must be evaluated via “black-box” simulation. Thus, we refer to the output F as the *response function* of a simulation run. Furthermore, due to random variation, true performance cannot be evaluated exactly but must be estimated by aggregating repeated samples of the response F . In a sequential search strategy with a fixed sampling budget, an important tradeoff is the need to balance improved accuracy at each design point (via increased sampling) with a wider range of design space exploration (visiting more design points).

Rigorous gradient-based search methods, such as stochastic approximation using finite-differencing¹ or simultaneous perturbations,² exist for stochastic optimization problems. However, such methods are not

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.

*Doctoral Candidate, Department of Operational Sciences, Todd.Srivier@afit.edu, Student Member AIAA.

†Associate Professor, Department of Operational Sciences, James.Chrissis@afit.edu, Member AIAA

applicable unmodified in the presence of categorical variables. Relaxation methods that iteratively solve subproblems in which the “discreteness” restrictions are relaxed, such as branch-and-bound, are inadequate when considering MVP problems because system performance is defined only at the discrete settings of the categorical variables. Thus, relaxing the “discreteness” of these variables may force a situation where performance cannot be evaluated because the design is undefined. Due to prohibitive expense, enumeration methods that solve problem (1) for every combination of discrete variable values are impractical when the number of categorical variables and/or the number of settings for such variables is large. More desirable is a method that can search for optimal solutions over the mixed variable domain.

The generalized pattern search (GPS) class of gradient-free algorithms provides a useful framework to treat MVP problems. In this paper, we present an extension to mixed-variable GPS for problems with stochastic response functions. An algorithm class is defined that replaces the binary comparisons of incumbent and trial designs used in traditional GPS methods with a class of statistical selection procedures known as ranking and selection (R&S) in which one solution is selected from a finite number of candidates considered simultaneously. In addition, the framework allows for the incorporation of surrogate functions that approximate the objective function of problem (1). The surrogate function is used as an inexpensive means to evaluate potential trial points and thus accelerate algorithm convergence to a near-optimal region of the design space without sacrifice to the convergence theory.

Section II of this paper presents the algorithmic framework in which generalized pattern search and ranking and selection are each described and then combined to provide a search methodology. Section III describes how a particular class of surrogates is embedded within this framework as a means to accelerate the search. Section IV provides a detailed algorithm description and then illustrates the algorithm on a simple numerical example. Finally, section V offers conclusions.

II. Algorithmic Framework

A general class of pattern search algorithms was introduced by Torczon³ for unconstrained, deterministic optimization problems. These algorithms are defined through a finite set of directions used at each iteration. The direction set and step-length parameter are used to construct a conceptual mesh centered about the current iterate (the incumbent). Trial points are selected from this discrete 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. Torczon showed that via specific rules, the direction set may be constructed in such a way to ensure that a component of the steepest descent direction can be captured by at least one direction in the set when the current iterate is not a stationary point.

Audet and Dennis⁴ extended pattern search to MVP problems by incorporating user-defined discrete neighborhoods into the definition of the mesh. Audet and Dennis also explicitly separate the search in the continuous domain into two distinct steps, an optional SEARCH step and a POLL step.⁴ The POLL step consists of a local search of mesh points in a *poll set* surrounding the incumbent and must be conducted before terminating the iteration. The SEARCH step allows for a more global search of the set of possible points on the mesh in the hope of accelerating convergence and avoiding local minima. Audet and Dennis also allow a search in the continuous domain of discrete neighbors that are sufficiently close in objective function value to the incumbent design. This step, called the EXTENDED POLL step, enables extension of the convergence theory to problems with mixed variables.

These concepts are made more precise using the following notation. A design point is partitioned into continuous and discrete components $x = (x^c, x^d)$ where $x^c \in \Theta^c \subseteq \Theta$, $x^d \in \Theta^d \subseteq \Theta$, and Θ^c (Θ^d) is the continuous (discrete) portion of Θ . At iteration k , the mesh is defined as,

$$M_k(x_k) = \Theta^d \times \bigcup_{i=1}^{i_{\max}} \left\{ x_k^c + \Delta_k D_k^i z \in \Theta^c : z \in \mathbb{Z}_+^{|D_k^i|} \right\}, \quad (2)$$

where $i \in \{1, \dots, i_{\max}\}$ is a combination of discrete variable values, i_{\max} is the total number of such combinations, Δ_k is the step length parameter, and D_k^i is the direction set for the i th combination of discrete variable values. The columns of D_k^i are search directions represented by a vector of the same dimension as x^c . In the SEARCH step, a user-defined strategy is used to nominate points from M_k for comparison to the incumbent.

The *continuous* poll set is defined as,

$$P_k = \{x_k + \Delta_k(d, 0) : d \in D_k^i\} \subseteq M_k, \quad (3)$$

where $(d, 0)$ denotes the partitioning into continuous and discrete variables and 0 means the discrete variables remain unchanged, *i.e.*, $x_k + \Delta_k(d, 0) = (x_k^c + \Delta_k d, x_k^d)$. A *discrete* poll set is constructed via a user-defined set of neighbors at x_k , denoted as $\mathcal{N}(x_k)$, where the elements of $\mathcal{N}(x_k)$ include changes to the discrete variable values. If the SEARCH step did not yield improvement, then points from the set $\{P_k \cup \mathcal{N}(x_k)\}$ are evaluated during the POLL step until improvement is found or the set is exhausted.

In the event a point from $\{P_k \cup \mathcal{N}(x_k)\}$ does not yield an improvement in the objective function, the EXTENDED POLL step is invoked in which a polling sequence is initiated in the continuous domain of any discrete neighbor $y \in \mathcal{N}(x_k)$ with an objective function value within ξ_k of the incumbent design point. The parameter ξ_k is termed the *extended poll trigger*.

For the stochastic case, we combine mixed-variable GPS with the ranking and selection (R&S) class of statistical selection procedures in the selection of new iterates. Ranking and selection procedures are methods that select the “best” system, or a subset of systems that includes the best system, from a set of competing alternatives [5, p. 273]. We introduce notation necessary to incorporate R&S in a pattern search framework. A detailed exposition of R&S theory is beyond the scope of this paper (see Bechhofer⁶ for a comprehensive coverage); however, it is constructive to think of R&S as a means of controlling the error involved in selecting new iterates. Traditional R&S procedures are guaranteed to select an alternative from a finite set that is at least a user-specified amount better than all others with a user-specified probability. With the *indifference-zone* R&S approach, the procedures guarantee selecting the best design alternative with probability $1 - \alpha$ or better, whenever the difference between the true best and all the rest exceeds the user-specified quantity δ , termed the *indifference zone parameter*.

Within the pattern search framework, we employ R&S to select the estimated best candidate design from a set of trial points considered simultaneously. Let $C = \{y_1, y_2, \dots, y_{n_C}\} \subset M_k$ be a set of candidate designs, including the incumbent, such that $n_C \geq 2$. For each $q = 1, 2, \dots, n_C$, 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,

$$f_{[1]} \leq f_{[2]} \leq \dots \leq f_{[n_C]}. \quad (4)$$

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 δ and the significance level $\alpha \in (0, 1)$, as

$$P\{CS\} = P\{\text{select } y_{[1]} \mid f_{[q]} - f_{[1]} \geq \delta; q = 2, \dots, n_C\} \geq 1 - \alpha \quad (5)$$

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 . For each $q = 1, 2, \dots, n_C$, 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 $\ell \neq q$.

In essence, the R&S procedure prescribes the number of samples s_q that are required to guarantee $P\{CS\}$ (5). After the samples have been collected, the sample means $\bar{F}_q = s_q^{-1} \sum_{s=1}^{s_q} F_{qs}$ for each q are computed, ordered and indexed the same way as in (4). The notation $\hat{y}_{[q]} \in C$ is used to denote the candidate with the q th best (lowest) *estimated* objective function value after sampling the n_C design candidates. For the purpose of inclusion in a class of algorithms defined in Section IV, 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 $\hat{y}_{[1]} = \arg(\bar{F}_{[1]})$ as the candidate having the δ -near-best mean.

In Srivier *et al.*,⁷ we show that the class of Mixed-variable Generalized Pattern Search with Ranking & Selection (MGPS-RS) algorithms produces iterates that converge almost surely to limit points that satisfy stationary point conditions appropriately defined over a mixed-variable domain. In this paper, we emphasize the use of surrogates during the SEARCH step to accelerate algorithm convergence. In particular, previously sampled points are used to construct a surrogate function that approximates $\mathbb{E}[F]$ which can then be

optimized inexpensively to generate trial points in the SEARCH step. This technique is advantageous when simulation responses are particularly expensive to generate, as is common for high-fidelity engineering design analyses. Similar approaches have been previously incorporated in a pattern search setting (*e.g.* see Booker *et al.*⁸) without sacrifice to the convergence theory.

III. Augmenting The Search Using Surrogates

Paramount to the construction of surrogates is the selection of a family of plausible functions for use in approximating the true objective function. Because we do not wish to assume a specific parametric representation of the underlying structure of $\mathbb{E}[F]$, we use a technique from the *nonparametric regression* literature; in particular, the *Nadaraya-Watson estimator*^{9,10} is used to approximate the objective function at a point x ,

$$\hat{f}(x) = \frac{\sum_{j=1}^N \bar{F}_j K_h(x - x_j)}{\sum_{j=1}^N K_h(x - x_j)}, \quad (6)$$

where $\{x_1, \dots, x_N\}$ are previously sampled design “sites” (points in the design space), \bar{F}_j is the mean of the sampled responses at site x_j , and K_h is an appropriately selected *kernel function* that depends on parameter h and the distance from x to each design site. As it has its roots in probability density estimation, the kernel function must integrate to unity, *i.e.*, $\int K_h(x) = 1$. We use the multivariate Gaussian kernel function, originally proposed in univariate form by Parzen.¹¹ This results in the following regression equation,

$$\hat{f}(x) = \frac{\sum_{j=1}^N \bar{F}_j \exp(-\frac{D_j^2}{2h^2})}{\sum_{j=1}^N \exp(-\frac{D_j^2}{2h^2})} \quad (7)$$

where $D_j^2 = (x - x_j)^T(x - x_j)$ represents the squared Euclidean distance from x to x_j and h is a smoothing parameter that determines the width of the kernel centered at each site x_j . For this reason, h is often called the *bandwidth*. We refer to \hat{f} as the *surrogate* function. The regression function (7) has also been described in the context of *generalized regression neural networks*.¹²

The estimate $\hat{f}(x)$ can be thought of as the weighted average of all response means, \bar{F}_j , where the weight received by \bar{F}_j depends on the distance between the corresponding x_j and the estimation point. The smoothing parameter h determines the degree of nonlinearity in the approximating function. A larger value of h forces \hat{f} to be a smooth function and approaches a multivariate Gaussian with covariance $h^2 I$.¹² If h is very large, \hat{f} is a constant that assumes the mean value all \bar{F}_j , *i.e.*, $N^{-1} \sum_{j=1}^N \bar{F}_j$. Smaller values of h allow more curvature in \hat{f} but can cause outliers to have too great an effect on the estimate. If h is zero, \hat{f} assumes the value of \bar{F}_j for the corresponding x_j that is nearest the estimation point.

An advantage of the kernel regression approach is its simplicity. To evaluate a surrogate function at a design point, all that is required is storage of the pairs (x_j, \bar{F}_j) and an appropriate setting for the lone bandwidth parameter. However, care must be taken in the practical consideration of selecting an “appropriate” setting. We use the well-known *leave one out, cross-validation* method [13, p. 152] to select the bandwidth. Using this method, the bandwidth is set to a fixed value, then the estimator \hat{f} is computed according to (7) at design site x_j except that x_j is excluded from (*left out of*) the summand, and the squared error $(\hat{f} - \bar{F}_j)^2$ is recorded. The sum of squared error (SSE) accumulated over all sites x_j , $j = 1, \dots, N$ is then used as a criterion for evaluating h . This procedure is repeated over a range of bandwidth values and the setting that delivers the smallest SSE is selected.

In order to build the original surrogate function prior to initiating the search, it is necessary to select design sites x_1, \dots, x_N via some appropriate experimental design technique. For this purpose, we use *latin hypercube sampling* (LHS).¹⁴ In LHS, a total of p equally-spaced values for each of the n continuous variables are used as components of the design site vectors. These values are randomly matched to form p design sites. If $N = p$, then each of the p values is represented exactly once in the set of design sites x_1, \dots, x_N , and we say that the design is of *strength* one. We typically use designs of strength two ($N = 2p$) so that the design space is sampled more densely; the random matching operation is performed twice.

Once the surrogate function is built, it can be utilized in the pattern search framework as an inexpensive means to nominate trial points from the mesh M_k in the SEARCH step of the algorithm. After trial points are evaluated, they are then added as design sites that enhance the accuracy of the surrogate function. A

straightforward approach is simply to minimize \hat{f} directly using any deterministic search routine. However, such a greedy approach may inhibit improvements in accuracy of the surrogate function because the trial points will cluster in a particular region of the design space. Alternatively, we use the technique of Torczon and Trosset¹⁵ that seeks to find improvements in \hat{f} while simultaneously seeking “space-filling” points that could improve the accuracy of the surrogate function. They propose a bi-objective function of the form,

$$m(x) = \hat{f}(x) - \rho d(x), \quad (8)$$

where $d(x) = \min \|x - x_j\|_2$ is the distance from x to the nearest previously sampled design site and $\rho \geq 0$ determines the relative weight placed on the space-filling objective. We refer to $m(x)$ as the *merit* function.

The surrogate function \hat{f} is a smooth approximation to the unknown objective function with respect to the continuous variables. If one or more discrete variables change, then the true objective function may have an entirely different structure. Therefore, when using surrogates in a mixed-variable pattern search framework, it is necessary to maintain a surrogate function for each combination of discrete variables $i = 1, \dots, i_{\max}$. Consequently, we index the surrogate function, merit function, bandwidth, and space-filling parameter as f_i , m_i , h_i , and ρ_i , respectively.

IV. Algorithm Listing and Example Problem

An algorithm for Mixed-variable Generalized Pattern Search with Ranking & Selection (MGPS-RS) and surrogates may now be stated.

MGPS-RS Algorithm using Surrogates

Initialization: For each $i = 1, \dots, i_{\max}$, do the following

- Select N design sites and set the number of samples s per design site.
- Perform an initial computer experiment and compute the mean responses \overline{F}_j based on s samples for each design site x_1, \dots, x_N .
- Calibrate h_i using the leave one out, cross-validation method, construct \hat{f}_i , and set space-filling parameter $\rho_i \geq 0$.

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 $\delta_0 > 0$.

Until done, do the following:

1. SEARCH step: Find a candidate y on the mesh M_k that minimizes $\cup_{i=1}^{i_{\max}} m_i$. Use procedure $\text{RS}(\{y, x_k\}, \alpha_r, \delta_r)$ to return the estimated best solution $\hat{y}_{[1]}$. Add y as a design site and recalibrate the appropriate \hat{f}_i , 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 $x_{k+1} = \hat{y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$, and $k = k + 1$, and repeat Step 1. Otherwise, proceed to Step 2.
2. POLL step: Set extended poll trigger $\xi_k \geq \xi$. Use procedure $\text{RS}(P_k(x_k) \cup \mathcal{N}(x_k), \alpha_r, \delta_r)$ 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 $\hat{y}_{[1]} \neq x_k$, the step is *successful*, add $\hat{y}_{[1]}$ as a design site and recalibrate the appropriate \hat{f}_i , update $x_{k+1} = \hat{y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$, 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) Use procedure $\text{RS}(P_k(y_k^j), \alpha_r, \delta_r)$ 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 3a. Otherwise, set $z_k = y_k^j$ and proceed to Step 3b.
 - (b) Use procedure $\text{RS}(\{x_k, z_k\}, \alpha_r, \delta_r)$ 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*, add $\hat{y}_{[1]}$ as a design site and recalibrate the appropriate \hat{f}_i , update $x_{k+1} = \hat{y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$, 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 $x_{k+1} = x_k$, $\Delta_{k+1} < \Delta_k$, and $k = k + 1$ and return to Step 1.

Prior to search steps 1, 2, and 3 in the algorithm, initial surrogate functions are constructed for each combination of discrete variable values. If N design sites are selected for each combination, each requiring s samples of the random response function, then a budget of $N \times s \times i_{\max}$ response samples is required during initialization.

The algorithm consists of a SEARCH step, a POLL step, and an EXTENDED POLL step. In the SEARCH step, the merit functions are searched to nominate a trial point for evaluation. Any deterministic search routine may be used for this purpose, so long as the resulting point is from the mesh M_k . We use a mixed-variable pattern search of the mesh points to nominate a trial point. A R&S procedure is used to select the best alternative from among the incumbent and the trial point. If the trial point is best, the incumbent is updated and the mesh coarsened (or retained). Regardless the outcome of the SEARCH step, the appropriate surrogate function is updated and recalibrated by adding the newly evaluated point as a design site. If the incumbent is the best, the POLL step is executed.

In the POLL step, the R&S procedure selects the best from the set of alternative designs in the poll set P_k and user-defined discrete neighbor set \mathcal{N}_k . If the best is different from the incumbent, the incumbent is updated, the mesh coarsened (or retained), and the appropriate surrogate function is updated and recalibrated. Otherwise, the EXTENDED POLL step is executed.

In the EXTENDED POLL step each discrete neighbor that satisfies the extended poll trigger condition initiates a polling sequence in the continuous neighborhood of the discrete neighbor. Each subiterate in this sequence is selected via the R&S procedure. The terminal point in the sequence is compared to the incumbent via an additional R&S procedure. If the best is different from the incumbent, the incumbent is updated, the mesh coarsened (or retained), and the appropriate surrogate function is updated and recalibrated. Otherwise the mesh is refined by setting $\Delta_{k+1} < \Delta_k$ and the algorithm returns to the SEARCH step.

The update rules for Δ_k are the same as the deterministic case. Refinement requires that $\Delta_{k+1} < \Delta_k$ and is accomplished after SEARCH, POLL, and EXTENDED POLL are all unsuccessful. Coarsening requires $\Delta_{k+1} \geq \Delta_k$ and is accomplished after any successful SEARCH, POLL, or EXTENDED POLL step. Typically, the step size is halved after unsuccessful iterations and retained or doubled after successful iterations. For specific restrictions on these rules, see Audet and Dennis.⁴

The algorithm maintains a separate counter for R&S parameters α_r and δ_r to provide strict enforcement of the rules on these parameters that are updated after each execution of the R&S procedure. The rules ensure that each parameter tends to zero as the number of iterations approaches infinity. These restrictions are critical for convergence and are described in Srivier *et al.*⁷

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

The space-filling parameters ρ_i need not remain constant throughout the search. In fact, as Torczon and Trosset¹⁵ suggest, these parameters should tend to zero to ensure convergence to a local solution. We use initial settings that are multiples of the maximum difference between mean responses of the initial design sites for each combination of discrete variable values. As the algorithm progresses, the parameters are halved after each SEARCH step.

We illustrate the MGPS-RS algorithm with a very simple example that has two continuous variables and one discrete (binary) variable where each continuous variable is bounded on the range $[-10, 10]$. Consider the response function

$$F(x) = f(x) + N(0, \sigma^2)$$

where $N(0, \sigma^2)$ is a normally distributed, mean-zero noise term added to an underlying true objective function. The true objective function is of the form,

$$f(x) = g_1(x_1, x_2)(1 - x_3) + g_2(x_1, x_2)x_3$$

where the linear function g_1 and quadratic function g_2 overlap in the continuous domain. Therefore, when

the discrete variable x_3 is 0 (1), the function takes a linear (quadratic) form. The functions are defined as,

$$g_1(x_1, x_2) = 21 - x_1 - x_2,$$

$$g_2(x_1, x_2) = x_1^2 + x_2^2.$$

The optimum is located at $x^* = (x_1^*, x_2^*, x_3^*) = (0, 0, 1)$ with $f(x^*) = 0$ and the starting point was set to $x_0 = (-5, -5, 0)$ with $f(x_0) = 31$. The standard deviation of the noise term was set to $\sigma = 2$ throughout the design space although the algorithm is equipped to handle a more complicated noise structure.

A progression of the algorithm is illustrated Figure 1. For comparison purposes, the true function is shown in the upper left of the figure. In the upper right, the initial surrogate surfaces are shown for each of the two binary variable settings. We used a strength one LHS design with $p = 10$ to determine the initial design sites and five samples at each design site. Therefore, a budget of $10 \times 5 \times 2 = 100$ response samples was necessary to build the original surrogates.

The lower left graphic of the figure shows the surrogate surfaces after nine iterations of the algorithm. By this point, 500 response samples had been generated and eight new design sites had been added to the surrogates. It can be seen how the space-filling parameter has forced the algorithm to evaluate points relatively far from the minimal point on either surrogate surface. The lower right graphic shows the surfaces after 17 iterations and 2000 response samples. Now, the search has begun to cluster near the optimal point, as desired. Additionally, the form of the surface approximating the quadratic function appears to more accurately predict the true response.

Due to its simplicity, no notable improvements in the speed of convergence are attained for this example by using surrogates. However, the importance of improvements in surrogate accuracy is clearly illustrated. Note, for example, that in Figure 1b, the minimum on the surrogate surfaces is actually located on the surface corresponding to the linear function g_1 . By forcing the search to evaluate “space-filling” points, the accuracy of the surface corresponding to g_2 was eventually improved so that the surrogates correctly predicted a minimum on the surface of g_2 . However, this behavior is not always guaranteed and presents some complications for problems with mixed variables. Fortunately, the algorithm provides the fail-safe EXTENDED POLL step that can ensure a search of alternative surfaces provided the extended poll trigger is set sufficiently large. In fact, it is good practice to set this parameter large enough in early iterations to ensure that enough design points are sampled to identify the combination of discrete variables that contains the minimum.

V. Conclusions

We have presented a framework for mixed-variable optimization under uncertainty for realistic engineering design problems. This framework exploits the flexibility of the pattern search class of algorithms by incorporating statistical selection methods to select iterates and the use of surrogates to accelerate the search without sacrifice to the convergence theory.

Through a simple example, the importance in building accuracy in the surrogate function was demonstrated. Although convergence acceleration was not realized for this small-dimensional example, these benefits become more readily achievable as the dimension of the problem increases. This is due to the fact that, in the POLL step, the number of trial points to evaluate grows linearly with the number of dimensions, which becomes very costly. A surrogate search in these situations provides a means to inexpensively nominate high quality trial points so that the search can more rapidly progress to improving designs before expending the computational budget. Future work will focus on comprehensive numerical testing on a range of test problems to quantify these benefits.

Acknowledgments

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

References

- ¹Kiefer, J. and Wolfowitz, J., “Stochastic Estimation of the Maximum of a Regression Function,” *Annals of Mathematical Statistics*, Vol. 23, 1952, pp. 462-466.

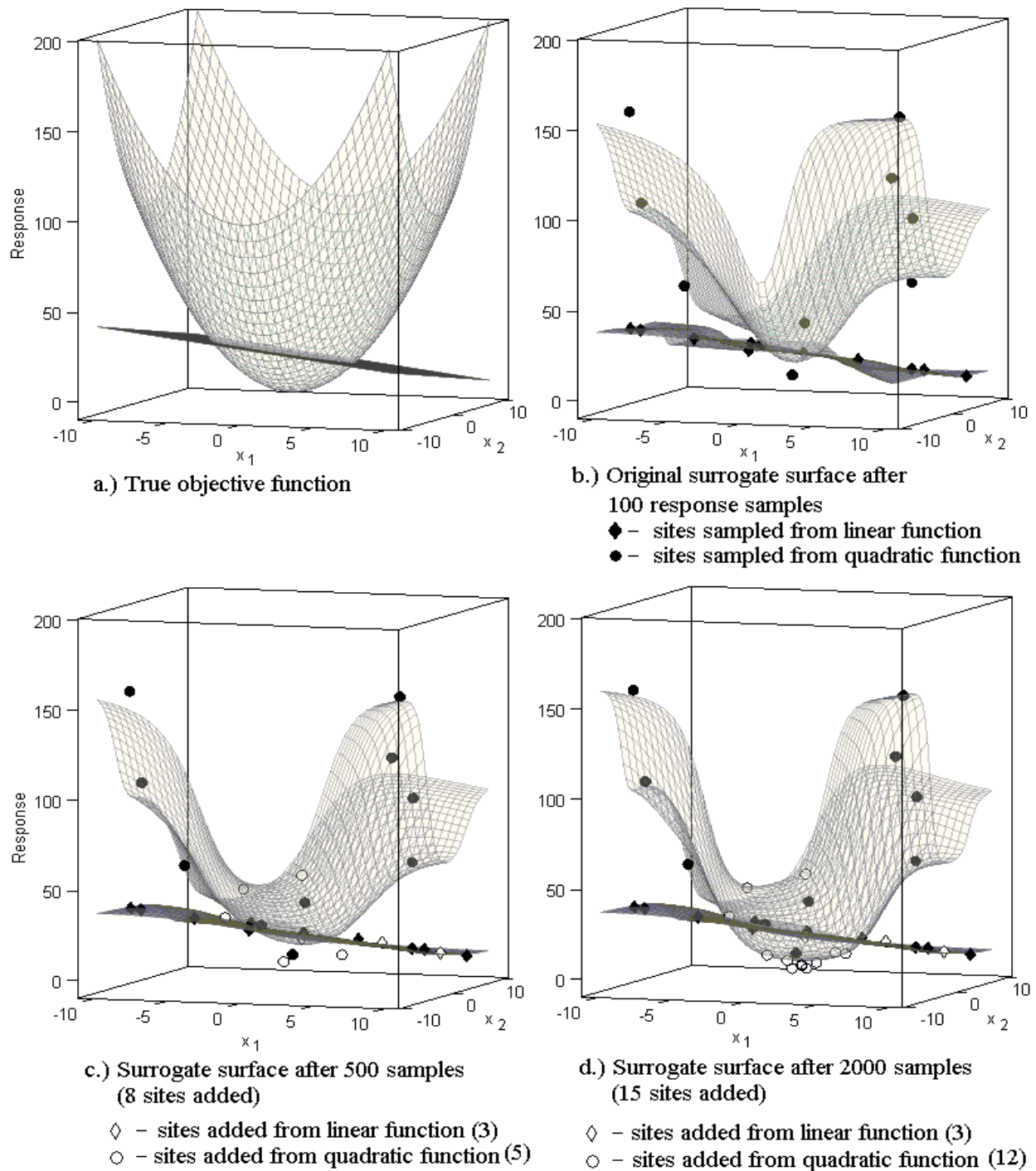


Figure 1. Demonstration of the surrogate building process as the MGPS-RS algorithm progresses for a simple problem with 2 continuous variables and 1 discrete (binary) variable. The true function is shown in (a). When the discrete variable is 0 (1), the function takes a linear (quadratic) form. The original surrogate after initialization is shown in (b). The surface is also shown after 9 iterations in (c) and after 17 iterations in (d).

²Spall, J. C., "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Transactions on Automatic Control*, Vol. 37, No. 3, 1992, pp. 332-341.

³Torczon, V., "On the Convergence of Pattern Search Algorithms," *SIAM Journal on Optimization*, Vol. 7, No. 1, 1997, pp. 1-25.

⁴Audet, C. and Dennis, Jr., J. E., "Pattern Search Algorithms for Mixed Variable Programming," *SIAM Journal on*

Optimization, Vol. 11, No. 3, 2000, pp. 573–594.

⁵Goldsman, D. and Nelson, B. L., “Comparing Systems Via Simulation (Chap. 8),” *Handbook of Simulation*, edited by J. Banks, John Wiley and Sons, New York, 1998, pp. 273–306.

⁶Bechhofer, R. E., Santner, T. J., and Goldsman, D. M., *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*, John Wiley and Sons, New York, 1995.

⁷Sriver, T. A., Chrissis, J. W., and Abramson, M. A., “Pattern Search Ranking and Selection Algorithms for Mixed Variable Stochastic Optimization,” 2004, Working Paper, Submitted to *Mathematical Programming*.

⁸Booker, A. J., Dennis, Jr., J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., “A Rigorous Framework for Optimization of Expensive Functions by Surrogates,” *Structural Optimization*, Vol. 17, No. 1, 1999, pp. 1–13.

⁹Nadaraya, E. A., “On Estimating Regression,” *Theory of Probability and Its Applications*, Vol. 9, 1964, pp. 141–142.

¹⁰Watson, G. S., “Smooth Regression Analysis,” *Sankhyā, Series A*, Vol. 26, 1964, pp. 359–372.

¹¹Parzen, E., “On Estimation of a Probability Density and Mode,” *Annals of Mathematical Statistics*, Vol. 35, 1962, pp. 1065–1076.

¹²Specht, D. F., “A General Regression Neural Network,” *IEEE Transactions on Neural Networks*, Vol. 2, No. 6, November 1991, pp. 568–576.

¹³Härdle, W., *Applied Nonparametric Regression*, Cambridge University Press, New York, 1990.

¹⁴McKay, M. D., Beckman, R. J., and Conover, W. J., “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

¹⁵Torczon, V. and Trosset, M. W., “Using Approximations to Accelerate Engineering Design Optimization,” *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998, pp. 738–748, AIAA paper 98-4800.