Proceedings of DETC'03
ASME 2003 Design Engineering Technical Conference and Computers and Information in Engineering Conference
Chicago, Illinois USA, September 2-6, 2003

DETC2003/DAC-48763

A NEW GLOBAL OPTIMIZATION METHOD FOR SIMULTANEOUS COMPUTATION ON EXPENSIVE BLACK-BOX FUNCTIONS

Liqun Wang
Dept. of Statistics
The university of Manitoba
Winnipeg, MB
Canada, R3T 2N2

Songqing Shan
Dept. of Mech. & Indus. Engineering
The University of Manitoba
Winnipeg, MB
Canada, R3T 5V6

G. Gary Wang*
Dept. of Mech. & Indus. Engineering
The University of Manitoba
Winnipeg, MB, Canada, R3T 5V6
Tel: 204-474-9463 Fax: 204-275-7507
Email: gary_wang@umanitoba.ca

ABSTRACT

The presence of black-box functions in engineering design, which are usually computation-intensive, demands efficient global optimization methods. This work proposes a new global optimization method for black-box functions. The global optimization method is based on a novel mode-pursuing sampling (MPS) method which systematically generates more sample points in the neighborhood of the function mode while statistically covers the entire search space. Quadratic regression is performed to detect the region containing the global optimum. The sampling and detection process iterates until the global optimum is obtained. Through intensive testing, this method is found to be effective, efficient, robust, and applicable to both continuous and discontinuous functions. It supports simultaneous computation and applies to both unconstrained and constrained optimization problems. Because it does not call any existing global optimization tool, it can be used as a standalone global optimization method for inexpensive problems as well. Limitation of the method is also identified and discussed.

Keywords: Global Optimization, Metamodeling, Black-box function, Design of Experiments

INTRODUCTION

In today's engineering design, as computer modeling capabilities increase dramatically, product behaviors are modeled and analyzed using the finite element analysis (FEA) and computational fluid dynamics (CFD) techniques. These analyses and simulation processes are usually computationally expensive. The design optimization based on these computation-intensive processes is challenging from several aspects.

- First, the overall optimization time should be acceptable by the increasingly impatient manufacturing industry. The optimization time relates to two issues, i.e., the number of total expensive function evaluations, and the amount of simultaneous computation. The term *simultaneous computation* is carefully chosen to be distinctive from *parallel computation*. S*imultaneous computation* means the possibility of having multiple simultaneous computing processes, not necessarily involving interactions between these computing processes, which is the characteristic of *parallel computation*.
- Second, for FEA or CFD, an explicit expression of optimization objective and constraint functions with respect to design variables is not available. Also gradients computed from these processes are usually unreliable or expensive [5]. To a design engineer, these processes are like a black-box, i.e., only the input and output can be obtained without *a priori* knowledge about the function.
- Third, a global design optimum is always preferred over a local optimum, if the computation cost is acceptable.

Numerous global optimization methods can be found in the literature. A recent survey is given in the Ref. [8]. However, few of them is suitable for the above expensive black-box function problems. Current global optimization methods can be possibly classified into two groups, deterministic and stochastic methods. Deterministic methods require *a priori* knowledge of the objective function, e.g., its shape, expression, gradient, Lipshitz constant, and so on. Thus they are not directly applicable to black-box functions. Most stochastic methods,

* Corresponding author

Copyright © 2003 by ASME

such as simulated annealing, genetic algorithms, Tabu search, multistart (including clustering), and many others, require a large number of function evaluations even for a simple 2-D design problem, though they do not demand *a priori* knowledge of the objective function. In addition, most of these methods are not developed to maximize the amount of simultaneous computation. Therefore, most of existing stochastic methods are not efficient in saving the total optimization time for expensive functions.

In the emerging area of metamodeling-based optimization, methods have been developed to address the computation efficiency problem. Those methods are based on the idea of sampling in the design space, building approximation models on these sampling points, and then performing optimization on the approximation function. Researches focus on developing better sampling strategies, approximation models, or methods dealing with the sampling, modeling, and optimization as a whole. Detailed surveys on researches in this direction can be found in the third author's previous work [12; 14]. Metamodeling-based optimization methods entail many attractive ideas for optimizing expensive black-box function problems, such as the idea of sampling and approximation. However, in general these methods rely on the structure of the approximation model. Furthermore, the choice of the best approximation model is problem dependent. So far, few methods have been developed for the global optimization. One successful development is in Refs. [6; 10], where the authors apply Bayesian method to estimate a kriging model, and then gradually identifies points in the space to update the model and perform the optimization. Their method, however, pre-assumes a continuous objective function and a correlation structure among sample points. The identification of a new point requires a complicated optimization process. The construction of the kriging model usually requires a global optimization process.

The third author of this paper and his colleagues have developed a number of global optimization strategies for the expensive black-box functions [12-14]. These methods focus on strategies to gradually reduce the search space. In the reduced final region, an existing global optimization method is applied on the approximation model to locate the optimum. In this work, a new global optimization method for expensive black-box functions is proposed, assuming the design space cannot be confidently reduced. In contrast to the methods in Ref. [6; 10], this method does not assume any properties of the black-box function; it works for both continuous and discontinuous functions; and it does not call any existing global optimization process or tool in its optimization process.

Before the global optimization strategy is discussed, a new mode-pursuing sampling method is to be introduced as it forms the core of the proposed global optimization method.

MODE-PURSUING SAMPLING METHOD (MPS)

In this section we introduce the so-called mode-pursuing sampling algorithm. It is an extension of the random-discretization based sampling method of Fu and Wang [3], which is a general-purpose algorithm to draw a random sample from any given multivariate probability distribution. This algorithm requires only the knowledge of the probability density function, up to a normalizing constant. This sampling method has been successfully implemented in many high-dimensional random sampling and numerical integration problems. This section will first describe the proposed MPS algorithm, which will be elaborated and explained with a sampling example. Then the properties of the MPS method will be given and proved.

Algorithm of the MPS Method

Suppose we are given a d-dimensional probability density function $g(x)$ with a compact support $S(g) \subset \Re^d$. Fu and Wang's algorithm [3] consists of three steps. In the first step, the discretization step, a discrete space $S_N(g)$ is generated consisting of N uniformly distributed base points in $S(g)$. Usually N is large and should be larger if the dimension of $g(x)$, d, is higher. These uniform base points may be generated using either deterministic or stochastic procedures. Then in the second step, the contourization step, the base points of $S_N(g)$ are grouped into K contours $\{E_1, E_2, \ldots, E_K\}$ with equal size according to the relative height of the function $g(x)$. For example, the first contour E_1 contains the $[N/K]$ points having the highest function values among all base points, whereas the last contour E_K contains the $[N/K]$ points having the lowest function values. Also in this step, a discrete distribution $\{P_1, P_2, \ldots, P_K\}$ over the K contours is constructed, which is proportional to the average functional heights of the contours. Finally, a sample is drawn from the set of all base points $S_N(g)$ according to the discrete distribution $\{P_1, P_2, \ldots, P_K\}$ and the discrete uniform distribution within each contour. As has been shown in Fu and Wang [3], the sample drawn according to their algorithm is independent and has an asymptotic distribution $g(x)$. The approximation gets better for larger values of N and K.

In this work, we incorporate the Fu and Wang's algorithm as a component of our new sampling method for optimization problems. Following the convention of engineering optimization, we refer to the minimum as the function mode.

Concretely, we wish to minimize a n-dimensional black-box function $f(x)$ over a compact set $S(f) \subset \Re^n$. To simplify notation, let us assume that $S(f) = [a, b]^n$, where $-\infty < a < b < \infty$ are known, and $f(x)$ is positive on $S(f)$. In general, if $f(x)$ is negative for some $x \in S(f)$, then we can always add a positive number to $f(x)$, so that it becomes positive on $S(f)$. Note that to minimize $f(x)$ is equivalent to maximize $-f(x)$. The proposed mode-pursuing sampling algorithm consists of the following four steps:

Step 1. First, generate m initial points $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ that are uniformly distributed in $S(f)$ (m is usually small).

Step 2. Use the m function values $f(x^{(1)}), f(x^{(2)}), \ldots, f(x^{(m)})$ to fit a linear spline function

$$\hat{f}(x) = \sum_{i=1}^{m} a_i \| x - x^{(i)} \|, \tag{1}$$

such that $\hat{f}(x^{(i)}) = f(x^{(i)})$, $i = 1, 2, \ldots, m$.

Step 3. Define $g(x) = c_0 - \hat{f}(x)$, where c_0 is any constant such that $c_0 \geq \hat{f}(x)$, for all x in $S(f)$. Since $g(x)$ is nonnegative on

Copyright © 2003 by ASME

$S(f)$, it can be viewed as a probability density function, up to a normalizing constant, with the modes located at $x^{(i)}$'s where the function values are the lowest among $\{f(x^{(i)})\}$. Then apply the sampling algorithm of Fu and Wang [3] to draw a random sample $x^{(m+1)}, x^{(m+2)}, \ldots, x^{(2m)}$ from $S(f)$ according to $g(x)$. These sample points have the tendency to concentrate on the maximum of $g(x)$, which corresponds to the minimum of $\hat{f}(x)$.

Step 4. Combine the sample points obtained in Step 3 with the initial points in Step 1 to form the set $x^{(1)}, x^{(2)}, \ldots, x^{(2m)}$ and repeat Steps 2–3 until a certain stopping criterion is met.

Remark. Note that in the Step 2 above, a linear spline function is used to fit the expensive points. It is based on the following reasons:

1. The linear spline function, or the radial basis function (RBF), is the simplest function that passes all the expensive points.
2. The linear spline function also prevents unnecessary "curvatures" added to the unknown surface as in the case of other models such as kriging.
3. Moreover, it preserves the minimum among the expensive points, i.e., $\min \hat{f}(x) = \min\{f(x^{(i)}),\ i = 1, \cdots, m\}$ because its linearity nature. This feature ensures more expensive points are generated around the current minimum of $f(x)$, rather than being biased because of the approximation model $\hat{f}(x)$.

This algorithm is to be further elaborated with a sampling example.

A Sampling Example

For ease of understanding, the mode-pursuing sampling method is illustrated with the well-known six-hump camel-back (SC) problem. The expression of SC is given in Eq. 2.

$$f_{sc}(x) = 4x_1^2 - \frac{21}{10}x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,\ (x_1, x_2) \in [-2\ \ 2]^2 \quad (2)$$

The contour plot of the SC function is shown in Figure 1, where H's represent the local optima, and H_2 and H_5 are two global optima at points (0.090, -0.713) and (-0.090, 0.713) with an equal function value $f_{\min} = -1.032$.

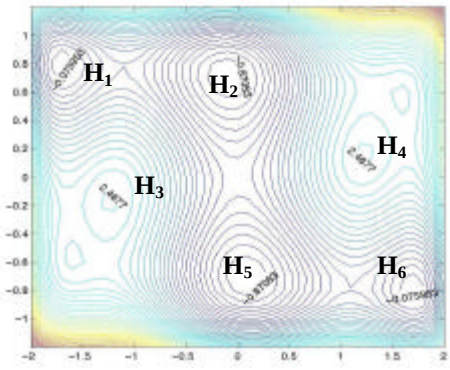

Figure 1 Contour plot of the SC function.

First, assume we start with $m = 6$ initial random points $x^{(1)}, x^{(2)}, \ldots, x^{(6)} \in [-2\ \ 2]^2$. Then $\hat{f}(x)$ is computed by fitting Eq. 1 to $f(x^{(1)}), f(x^{(2)}), \ldots, f(x^{(6)})$. Further, the function $g(x)$ is obtained by using the maximum of $\{f(x^{(i)}), i=1, \ldots, m\}$ as c_0.

Now we apply Fu and Wang's algorithm to draw a sample as follows. First, $N = 10^4$ uniform base points are generated to form $S_N(g)$, the discretized version of the sample space $[-2\ \ 2]^2$. Note that the base points in $S_N(g)$ are cheap points, in contrast to the original $m = 6$ expensive points used to build $\hat{f}(x)$. Further, without loss of generality, suppose the points in $S_N(g)$ are sorted by ascending order of the values of function $\hat{f}(x)$. The sequence of the corresponding function values of $\hat{f}(x)$ is plotted in Figure 2(a), whereas the function $g(x)$ is plotted in Figure 2(b).

According to Fu and Wang's [3] method, the ordered 10^4 base points are then grouped into $K = 10^2$ contours $\{E_1, E_2, \ldots, E_{100}\}$, with each having $N/K = 100$ points. For example, the first contour E_1 contains the 100 points at which the values of function $\hat{f}(x)$ are the lowest, whereas the last contour E_{100} contains the 100 points at which the values of $\hat{f}(x)$ are the highest. Let $\tilde{g}(i)$ be the average of $\hat{f}(x)$ over E_i, $i = 1, 2, \ldots, 100$. The function $\tilde{g}(i)$, $i = 1, 2, \ldots, 100$ is plotted in Figure 2(c) and its cumulative distribution function $G(i)$ is displayed in Figure 2(d).

Finally, $m = 6$ contours are drawn with replacement using the inverse transformation of the distribution $\{G(i), i = 1, 2, \ldots, 100\}$. If the contour E_i occurs $m_i > 0$ times in these six draws, then m_i points are randomly drawn from E_i. All points so obtained form the new sample $x^{(m+1)}, x^{(m+2)}, \ldots, x^{(2m)}$.

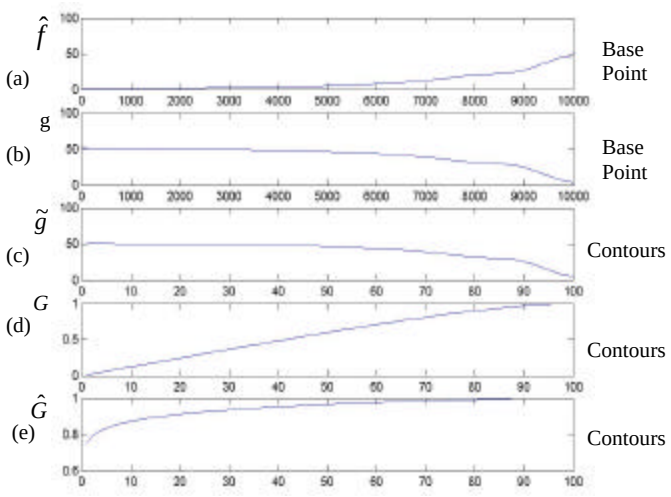

Figure 2 A screen shot of functions $\hat{f}$, g, $\tilde{g}$, G and $\hat{G}$ for the SC problem.

As one can see from Figure 2(d), the contours from E_{80}~E_{100} (corresponding to high $\hat{f}$ values) will have a lower probability to be picked for further sampling than other contours, since the G curve is relatively flat in this area. However, such a probability for each contour is always larger than zero. On the other hand, it is generally desired to increase the probability of the first few contours as they correspond to low $\hat{f}$ values. In order to have a better control of the sampling, a speed control factor is introduced, which is to be discussed later in the Speed Control Factor subsection. Figure

Copyright © 2003 by ASME

2(e) shows $\hat{G}(i)$, which is obtained by applying the speed control factor to $G(i)$ in Figure 2(d). From Figure 2(e), one can see that the first few contours have a high probability to be picked for the new sampling while the contours from E_{40} ~E_{100} have a low probability. This curve shows an aggressive sampling step as much more new sample points will have function value close to the current minimum of $f(x)$ as compared to the sampling based on Figure 2(d).

The whole procedure is repeated eight times, so that totally 48 sample points are generated. Figure 3 shows these 48 sample points, where the circle dots indicate attractive design points having a function value less than –0.5. Even with 48 sample points, many attractive points have already shown up around H_2 and H_5. It can be seen from the figure that points spread out in the design space with a high density around one of the function mode H_2 (global minimum). The mode-pursuing characteristic is demonstrated.

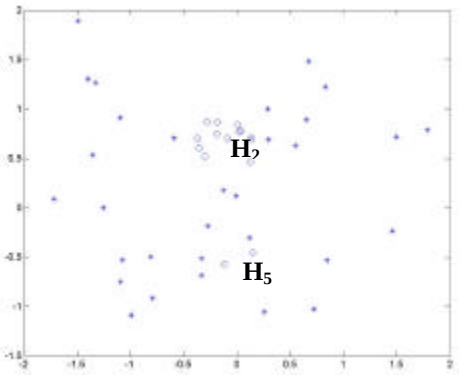

Figure 3 Sample points of SC generated by using the mode-pursuing sampling method, where "o" indicates its function value less than –0.5; and H_2 and H_5 are the locations of two global optima.

Properties of the MPS Method

From the construction, it is easy to see that this sampling procedure has the following two properties: first, every point in $S(f)$ has a positive probability to be drawn, so that the probability of excluding the global optimum is zero. Secondly, as the iteration continuous, more and more sample points will concentrate on the modes of function $g(x)$, which in turn pursue the modes of function $f(x)$. Thus the algorithm automatically pursues the modes of $f(x)$. It can be proved that if neglecting the stopping criteria in Step 4, the method can identify the global optimum of $f(x)$.

Suppose the stopping rule in Step 4 is not applied, so that the sampling algorithm iterates to infinity. For any integer $k > 0$, the minimum function value obtained after k-th iteration is clearly

$$f_k = \min \{f(x^{(i)}), i = 1, 2, \ldots, km\}.$$

The following theorem shows that, under fairly general conditions, the sequence f_k converges to the global minimum, as k increases to infinity.

Theorem 1. Suppose the objective function $f(x)$ is continuous in a neighborhood of the global minimum on the compact subset $S(f) \subset \Re^n$. Then, as $k \to \infty$, f_k obtained by using the mode-pursing sampling method converges to the global minimum $f_0 = \inf_{x \in S(f)} f(x)$.

Proof: Suppose the global minimum is attained at $x_0 \in S(f)$, so that $f(x_0) = f_0$. By construction, the set of all sampled points is extended by m new points after each iteration. It follows that the sequence $\{f_k\}$ is decreasing. Since $f(x)$ is continuous in a neighborhood of the global minimum, for any $\varepsilon > 0$, there exists $\delta > 0$, such that $0 < f(x) - f(x_0) < \varepsilon$, for all $\| x - x_0 \| < \delta$. Because in each iteration the density $g(x)$ is positive on $S(f)$, there exists an integer $K > 0$, such that after K-th iteration, a point satisfying $\| x_1 - x_0 \| < \delta$ can be sampled, which implies that $0 < f(x_1) - f(x_0) < \varepsilon$. It follows that $0 < f_k - f_0 < \varepsilon$ for all $k > K$. The proof is completed.

In summary, because MPS does not pre-assume any properties of the objective function, it applies to general black-box functions, which can be either continuous or discontinuous across the entire design space. If the objective function is continuous in a neighborhood of the global optimum, it can be proved that the MPS systematically converges to the global optimum.

GLOBAL OPTIMIZATION STRATEGY

From the above sections we know that the mode-pursuing sampling method has the property of statistically sampling more points near the minimum (mode) while still covering the entire design space. For the purpose of optimization, one may iterate this sampling procedure until a maximum number of function evaluations has been reached. Such an approach is a legitimate global optimization method. However such a crude approach can be slow in convergence, when $f(x)$ is relatively flat around the global optimum. In such cases, more "intelligence" needs to be built in. A natural method, as seen in some existing global optimization methods, is to use a threshold of sample density or similar measures to shrink the design space to a small area, in which a local optimization can be performed. No matter what measure is used, whether it is a sample density or a hyper-sphere with fixed diameter, it would be too difficult to decide the value of the measure, and be too rigid because the method is to be applied to all types of problems. On the other hand, the idea of integrating a global technique with a local technique has been recognized as a plausible approach in designing a global optimization method [11]. In this work, we design a special approach to guide the optimization process to identify an attractive local area without using a rigid measure.

Concept of the Strategy

As the mode-pursuing sampling progresses, more sample points will be generated around the current minimum point, with chances to search for a better minimum. If a better point is found, the mode shifts to the new point and more sample points will be generated about that point. If the current minimum point is the global optimum, more and more points are generated in the small region that contains the optimum with a slim chance that this point can be exactly reached. On the other hand, according to the Taylor's theory, we know that

Copyright © 2003 by ASME

in a continuous small area around the minimum any function can be accurately approximated by a quadratic function. Thus if we can perfectly fit the points around the current minimum to a quadratic model, it means that we have reached the "valley" that contains the global minimum. Given such a perfect quadratic model, local optimization can be performed without expensive function evaluations to locate the global minimum.

An *n*-D generic quadratic model is usually expressed by Eq.3.

$$y = b_0 + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} b_{ii} x_i^2 + \sum_{i<j} \sum_{j=1}^{n} b_{ij} x_i x_j \quad (3)$$

where b_i, b_{ii}, and b_{ij} represent regression coefficients, $x_i, (i = 1 \cdots n)$ are design variables, and y is the response. It is to be noted that the model follows the standard formula used in the response surface method (RSM) [7].

The coefficient of determination, R^2, is usually applied to evaluate the modeling accuracy. The R^2 is defined as

$$R^2 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \quad (4)$$

where, $\hat{y}_i$ are the fitted function values; y_i are the function values at the sample points; $\bar{y}$ is the mean of y_i; and $0 < R^2 \le 1$. If the number of sample points used to fit Eq. 3 is equal to the number of coefficients of Eq. 3, which is $(n+1)(n+2)/2$, then R^2 is equal to 1. In practice, the number of sample points is usually taken larger than the number of coefficients. Only based on an equal number of sample points and a fixed regression model, the R^2 value can be appropriately compared. In general, the closer the R^2 value to 1, the better the modeling accuracy [7].

Now the question is how to determine the "neighborhood" of the current minimum for model fitting. As we know to fit the quadratic model in Eq. 3, at least $(n+1)(n+2)/2$ points are needed to obtain the estimates of the same number of coefficients. But if the number of points used to fit the model is the minimum, the R^2 value will equal to its maximum value 1, making it hard to evaluate the fitting accuracy. In order to avoid this situation and at the same time to keep the number of function evaluations low, we use $(n+1)(n+2)/2+1$ points to fit the quadratic model. When there are enough number of points that are close to the minimum point, these points together will determine a sub-region, which is a hyper-box defined by the minimal lower bound and maximal upper bound of all points in each x direction. At the beginning of the sampling procedure, the size of the sub-region will almost be the same as the original design space, because the total number of points generated by far is small and sparse. If the objective function $f(x)$ is quadratic, a perfect fit will be obtained and the global optimum can be quickly secured. If the function $f(x)$ is multi-modal, a quadratic model fitting won't be perfect. Even though by chance it fits well, it won't likely pass the model validation stage, which will be discussed later. In this case, the mode-pursuing sampling process will concentrate less on the current optimum but instead spread out more in the rest of the space by tuning a speed control factor r (to be discussed later). As the optimization process iterates, more and more points are generated, small sub-regions can be obtained and the process gradually converges to the small region containing the global optimum. As one can see, in the proposed method, the size of the sub-region (though at the beginning it is almost the same as the original space) is adaptive to the complexity of the objective function. A rigid measure in identifying the local area, such as the sample density or a hyper-sphere with fixed diameter, is avoided.

Quadratic Model Detection

In this work, a quadratic function within a sub-region is detected by a two-stage approach. At the first stage, $(n+1)(n+2)/2+1$ points are fitted to a quadratic model. If the R^2 is not close to 1, it means the model is highly polynomial. The difficulty comes if the R^2 is close to 1. In this situation, it is risky to say that the function is quadratic, because in many situations, especially in high dimensional problems, R^2 is very close to 1 even though the function is not quadratic. This is due to the fact that only one additional point besides the minimum $(n+1)(n+2)/2$ points is used in model fitting. Therefore, a second stage testing is used. At the second stage, a number of new expensive points $[n/2]$ are randomly generated in the sub-region, where $[n/2]$ refers to the number $(n/2)$ rounded to the closest integer. Then these new points are combined with the previous $(n+1)(n+2)/2+1$ points, which are fitted again to a quadratic model. If this fitting gives a R_{new}^2 which is close to 1 and the maximum absolute difference between all real function values and their corresponding predicted values is close to zero, then it is most likely that the function is quadratic. That is to say, the criterion for the second stage is

$$\begin{aligned} &|R_{New}^2 - 1| < e_R \quad \text{and} \\ &Diff = |\max(f_m^{(i)} - f^{(i)})| < e_d,\ i = 1, \cdots,\ j = (n+1)(n+2)/2 + 1 + [n/2] \end{aligned} \quad (5)$$

where, f_m is the obtained quadratic model and f is the real function; e_R and e_d are small numbers for the two criteria, respectively. In practice, e_R is often taken as 10^{-5}. It is difficult, however, to select e_d because it is problem dependent. This work defines e_d as:

$$e_d = c_d [\max(f^{(i)}) - \min(f^{(i)}),\ i = 1, \cdots,\ j] \quad (6)$$

where, c_d is a coefficient in [0,1] to be specified by the user. Generally, the smaller the c_d, the more accurate result one will get and the more function evaluations are needed. The default value of c_d is 10^{-2}.

Algorithm of the Proposed Method

The flowchart of the algorithm is shown in Figure 4. Detailed description of the algorithm is given below. The speed control factor, r, is to be discussed in the next section.

Description of the Algorithm

Input: *n*, *x*, *f*(*x*) ← n: the number of design variables; *x*: desgin varaible vector; *f*(*x*): objective black-box function

Output: x^*, $f(x^*)$, *nit*, *nfe* ← x^*: the obtained optimum; $f(x^*)$: the function value at x^*; *nit*: the number of optimization iterations; *nfe*: the number of expensive function evaluations

Copyright © 2003 by ASME

BEGIN

1) Randomly sample [$(n+1)(n+2)/2+1-n_p$] number of points ← n_p: an arbitrary number; usually set as $n_p = n$;
2) Sample [n_p] points using the mode-pursuing sampling method; ←Thus so far there are in total [$(n+1)(n+2)/2+1$] points, which is just enough to fit a quadratic model;
3) Find [$(n+1)(n+2)/2+1$] points around the current mode, along with the sub-region defined by these points (the minimal lower bound and maximal upper bound of all points in each x direction);
4) In the sub-region, obtain the approximate quadratic model and the R_1^2 value by fitting the [$(n+1)(n+2)/2+1$] points;
5) **If** $|R_1^2 - 1| < e_R$, **then** generate [$n/2$] expensive points in the sub-region; obtain an updated quadratic model, the R_{new}^2 value, and the *Diff* value by fitting all the points within the sub-region; add the new points to the point set ← [$n/2$]: the number ($n/2$) rounded to the closest integer;
   Else update the speed control factor r and go back to Step 2);
6) **If** $|R_{New}^2 - 1| < e_R$ and $Diff < e_d$, **then** perform local optimization on the approximation model to find x_t^*.
7) **If** x_t^* is in the sub-region, **then** program stops with all the outputs; **Else** obtain its real function value and add this point (x_t^*, $f(x_t^*)$) and the points generated in Step 5) to the original set of expensive points, update the speed factor r and go back to Step 2).

END

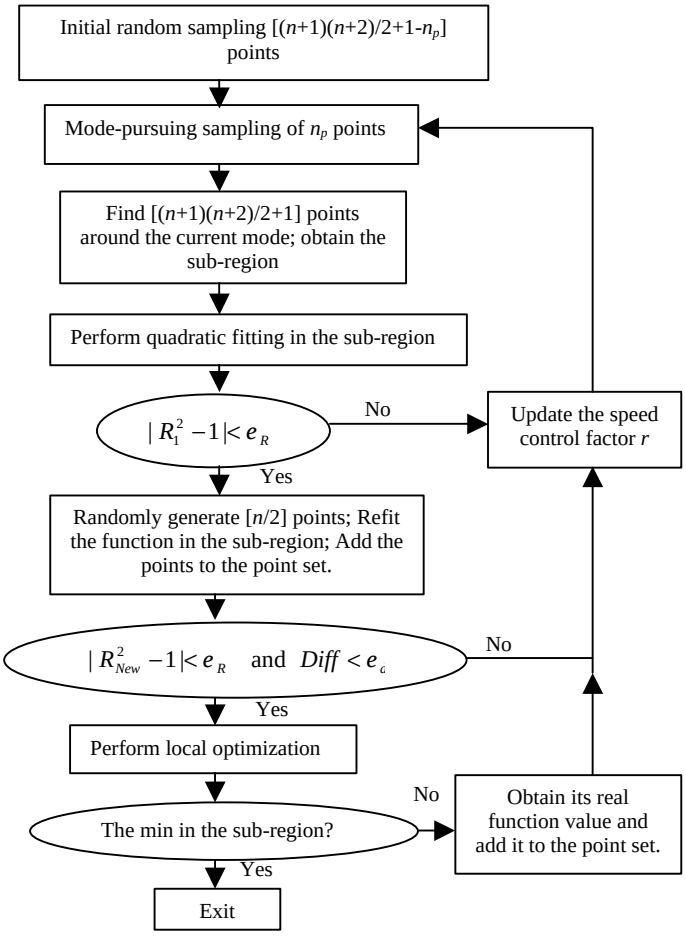

Figure 4 Flowchart of the proposed global optimization method.

Speed Control Factor

In Figure 4, the speed control factor, r, adjusts the "greediness" of the mode-pursuing sampling method. Referring to Figure 2, $\{G(i),\ i = 1, 2, ..., K\}$ denote the cumulative distribution function of the ranked $\{\tilde{g}(i),\ i = 1, 2, ..., K\}$, which can be considered as the discrete distribution $\{P_1, P_2, \ldots, P_K\}$ in the algorithm of Fu and Wang [3]. Note that $G(i) \in [0, 1]$. Define $G_{\min} = \min(G(i))$ and $G_{\max} = \max(G(i))$. It is easy to see that $G_{\min}$ is the sampling probability for the current mode and $G_{\max} = 1$. The difference $\{G(i) - G(i-1),\ i = 2, \cdots, K\}$ represents the probability of the sampling points being picked, whose function values are between $\tilde{g}(i-1)$ and $\tilde{g}(i)$. As one can see from Figure 2, if more "local" points closing to the current mode are desired, then the $G_{\min}$ is to be raised. Otherwise, $G_{\min}$ is to be lowered to allow more "exploratory" sample points. A speed control factor, r, is desired to allow the user to find the balance between "local" points and "exploratory" points, or to tune the "greediness" of the mode-pursuing sampling process. In this work, the tuning is done through the transformation $\hat{G}(i) = G(i)^{1/r}$. The next sampling is then based on $\hat{G}(i)$ curve instead on $G(i)$ curve. If $r=1$, the original discrete distribution is used. A higher $r>1$ increases the probability of sampling around the current mode with less chance of spreading out in the space. Figure 5 is a screen shot of $G(i)^{1/r}$ curves for the SC problem, where r increases as the curve becomes flatter. During the optimization process, however, r does not increase monotonically; it updates dynamically according to the previous iteration.

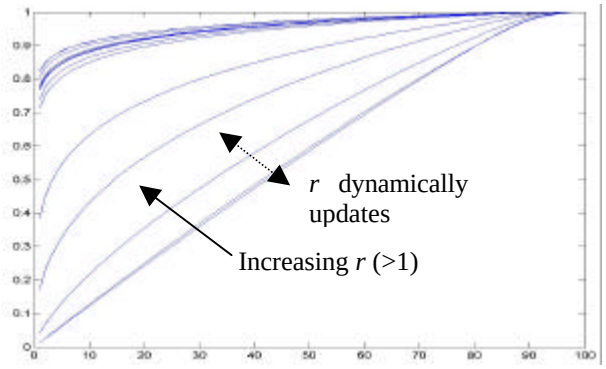

Figure 5 A screen shot of $G(i)^{1/r}$ curves for the SC problem.

How to adjust the value of the speed control factor? Generally, if a function is detected to be complex (non-quadratic), then r is reduced to generate more "exploratory" points. Otherwise, r is increased to increase the "local" points. Theoretically, $r \in [1, \infty]$. When $r = 1$, the original discrete distribution is used and the convergence speed doesn't change. The convergence speed of the mode-pursuing sampling procedure is increased as r increases. Let $G_{\min} = 0.75$ represents a practically very aggressive sampling scheme, which is chosen as a reference point to determine $r_{\max}$. For a given sample set on $g(i)$, we can obtain a $G_{\min}$. If we set $G_{\min}^{1/r_{\max}} = 0.75$, we can have $r_{\max} = \log G_{\min} / \log .75$. Therefore, $r \in [1, \quad \log G_{\min} / \log 0.75]$. In this work, the R^2 information is

Copyright © 2003 by ASME

used to guide the "greediness" of mode-pursuing sampling. Usually for most problems with only one extra sample point, $R^2 \in (.8, 1]$. When $R^2 \le 0.8$, r is set to 1 to allow more "exploratory" points. When $R^2 \in (.8, 1]$, one quarter of ellipse is chosen to describe the r-R^2 relationship as shown in Figure 6. The origin of the ellipse is (.8, $r_{\max}$) with the length of the short axis 0.2 and the long axis $r_{\max} - 1$. The ellipse's origin and long axis are automatically adjusted by the $G_{\min}$ feature calculated from $\hat{f}(x)$, which is obtained from the available expensive sample points. Figure 6 shows the r-R^2 curve of the last iteration when solving the SC problem. Therefore, the speed control factor, r, is controlled by the R^2 value from the previous iteration. It adjusts the "greediness" of the sampling process to achieve a balance between local sampling and exploratory sampling. It is to be noted that the speed control factor does not undermine the global sampling property of the mode-pursuing sampling strategy. As one can see from Figure 2(e) and Figure 5, with $r \to \infty, G(i)^{1/r} \to 1$. All of the contours can still be possibly picked.

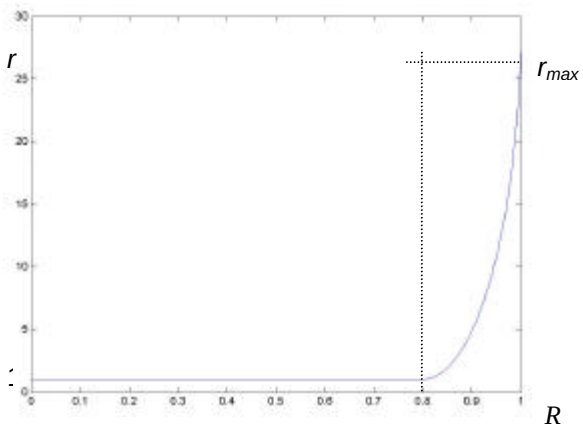

Figure 6 The last r-R^2 curve for the SC problem.

Note that there is a risk associated with using the R^2 value as a feedback for the sampling control. Since the sampling controlled by the factor, r, tends to be more aggressive, the optimization may converge to a local optimum before enough exploratory points have been generated. It would be ideal to have an indicator of whether the current mode is the global optimum or not. Such an indicator will be theoretically better than the R^2 value. However, without knowing the global optimum or properties of the expensive function *a priori*, it is difficult to develop such an indicator. The R^2 value thus only provides a necessary condition for a global optimum, yet not a sufficient condition. A similar use of the R^2 value is found in the Ref. [4]. Nevertheless, the proposed method works well for most of the test problems. It also maintains the possibility of mode shifting because even with aggressive sampling, new exploratory sample points are still being generated.

CONSTRAINED OPTIMIZATION PROBLEMS

The algorithm of the proposed method in the previous section is developed for the unconstrained optimization problems. For constrained optimization problems, we only deal with the problems with inexpensive constraints. Specifically, consider the problem

$$\textit{Minimize} \quad f(x) \tag{7}$$

$$\textit{Subject to} \quad g_k(x) \le 0, \;\; k = 1, \cdots q \tag{8}$$

where, $f(x)$ is the expensive objective function; $g_k(x)$ are the inexpensive equality or inequality constraints, including the variable bounds [1]. A few researches are seen in literature dealing with metamodeling for constrained optimization problems ([2; 9; 14]).

The basic principles for the constrained problem expressed by Eqs. 7-8 are the same as those for the unconstrained problems. The flowchart for constrained problems is the same as Figure 4. However, the interpretation for some steps is different:

1. For all sampling steps in Figure 4, "Initial random sampling [$(n+1)(n+2)/2+1-n_p$] points", "Mode-pursuing sampling of n_p point", and "Randomly generate [$n/2$] points," all points that do not satisfy constraints are discarded first before the evaluation of the objective function. This means that all samples generated from these steps should satisfy the constraints.
2. For the step "Perform local optimization" in Figure 4, the objective function remains the same as for the unconstrained optimization. However, the constraints $g_k(x)$ are included in the optimization. In this work, the Matlab™ Optimization Toolbox is used for the constrained local optimization.

TEST OF THE APPROACH

The proposed method has been tested with a number of well-known test problems. Their expressions are described as follows:

- A simple quadratic function (QF), n = 2.
$$f_{QF}(x) = (x_1 + 1)^2 + (x_2 - 1)^2, \quad x_i \in [-3 \;\; 3], \tag{9}$$
- Six-hump camel-back function (SC), n = 2, as described by Eq. 1.
- Golden-Price function (GP), n = 2.
$$f_{GP}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$
where, $x_i \in [-2 \;\; 2]$. (10)
- Hartman function (HN), n = 6.
$$f_{HN}(x) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2], \; x_i \in [0, 1], \; i = 1, \cdots, n \tag{11}$$
where

I	a_{ij}, $j = 1, \cdots, 6$						c_i
1	10	3	17	3.5	1.7	8	1
2	.05	10	17	0.1	8	14	1.2
3	3	3.5	1.7	10	17	8	3
4	17	8	.05	10	0.1	14	3.2

Copyright © 2003 by ASME

I	p_{ij}, $j=1, \cdots, 6$					
1	.1312	.1696	.5569	.0124	.8283	.5886
2	.2329	.4135	.8307	.3736	.1004	.9991
3	.2348	.1451	.3522	.2883	.3047	.6650
4	.4047	.8828	.8732	.5743	.1091	.0381

- A function of 16 variables (F16), n = 16.

$$f_{F16}(x) = \sum_{i=1}^{16}\sum_{j=1}^{16} a_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1), \quad x_i \in [-1\ 0]$$

$$[a_{ij}]_{row1-8} = \begin{bmatrix} 1\,0\,0\,1\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1 \\ 0\,1\,1\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,1\,0\,1\,1\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,1\,0 \end{bmatrix} \qquad (12)$$

$$[a_{ij}]_{row9-16} = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}$$

- Griewank function (GN), n = 2.

$$f_{GN}(x) = \sum_{i=1}^{n} x_i^2 / 200 - \prod_{i=1}^{n} \cos(x_i / \sqrt{i}) + 1, \ x_i \in [-100, 100] \qquad (13)$$

For each problem, 10 different runs are executed because random sampling is involved in the method. The average number of function evaluations *nfe*, and number of iterations, *nit*, are used as an indication of the time and resource required by the method. The median number is also given for reference. For the solution, the minimum and the maximum for the 10 runs are recorded, along with the average.

Table 1 Summary of test results on the proposed method.

Func.	n	Space	Analy. Min.	Minimum		*nfe*		*nit*	
				Range of Variation	Median	Avg.	Median	Avg.	Median
QF	2	$x_i \in$ [-3 3]	0	[0, 0]	0	9.6	8	1.4	1
SC	2	$x_i \in$ [-2 2]	-1.032	[-1.031, -1.014]	-1.030	37.8	30.5	9	7
GP	2	$x_i \in$ [-2 2]	3.000	[3.000, 3.216]	3.005	138	134	32.9	34
HN	6	$x_i \in$ [0 1]	-3.322	[-3.322, –3.148]	-3.305	592.1	576	49.6	47
F16	16	$x_i \in$ [-1 0]	25.875	[25.880, 25.915]	25.885	254.8	250	3.8	3.5
GN	2	$x_i \in$ [-100 100]	0	[0.003, 1.367]	0.1469	371	43	123.8	12

As one can see from the summary table, the proposed method successfully captures the global optimum for all test problems but GN. The total number of function evaluations is modest. Over the 10 runs for each case, the variation range for the minimum objective function value is very small. The variations of the *nfe* and *nit* numbers are also small, as shown by the average and median of these numbers. That indicates the proposed method is very robust, though being random in nature. The only exception is GN. Though for some of the runs the results are very good (obtained f=0.003 with 9 function evaluations), the performance of the method for GN is not stable, as demonstrated by the large solution variation and differences between the average and median *nfe* and *nit*'s. It is found that GN has 500 local optima in the design space. The optimization process for GN tends to be trapped in one of the 500 local optima if following the procedure shown in Figure 4. It is most likely because the mode-pursuing sampling process is terminated prematurely when there are not enough "exploratory" points to present the entire design space. To confirm the speculation and also to provide an example to Theorem 1, GN problem is solved by applying the mode-pursuing sampling method alone with a stopping criterion as $|f\text{-}0|<1e\text{-}3$, since we know the analytical minimum is zero. In another word, we will let the mode-pursuing sampling process continue until the analytical minimum is found. The results are summarized in Table 2, where x^* means the optimum point and f^* is the obtained function minimum.

Table 2 Optimization results on GN by applying the mode-pursuing sampling method alone.

Run No.	x^*	f^*	*nfe*	*nit*
1	(0.0000, -0.0070)	0	149	45
2	(0.0005, 0.00047)	0	1742	756
3	(0.0010, 0.0135)	0	1241	315
4	(0.0017, -0.0044)	0	1475	714
5	(0.0000, -0.0001)	0	390	173
6	(-0.0020, 0.0004)	0	1846	627
7	(0.0014, 0.0005)	0	2015	990
8	(0.0057, 0.0045)	0	218	93
9	(0.0058, -0.0070)	0	59	17
10	(-0.0003, -0.0005)	0	1291	627

As one can see from Table 2, the global optimum is obtained for all the runs. The number of function evaluations, however, differs dramatically. The test on GN checks the validity of Theorem 1, proving that the proposed mode-pursuing sampling method is systematic. The overall optimization strategy, which is based on the MPS method, may converge prematurely for problems having a large quantity of local optima such as GN.

Overall, from both the accuracy and efficiency perspectives, the proposed optimization method demonstrates

Copyright © 2003 by ASME

good performance. The solution is generally robust. In addition, because samples can be evaluated independently and simultaneously, assuming computers are available, the total time needed by the optimization will be represented by the number of iterations, rather than the number of total function evaluations. Thus the proposed method also supports simultaneous computation, which leads to potential significant time savings.

DESIGN OF A PRESSURE VESSEL

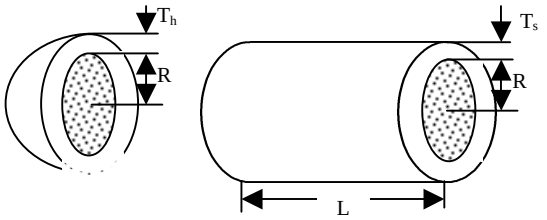

Figure 7 Pressure Vessel.

The design of a pressure vessel was first documented in [15]. The cylindrical pressure vessel is shown in Figure 7. The shell is made in two halves of rolled steel plate that are joined by two longitudinal welds. Available rolling equipment limits the length of the shell to 20 ft. The end caps are hemispherical, forged, and welded to the shell. All welds are single-welded butt joints with a backing strip. The material is carbon steel ASME SA 203 grade B. The pressure vessel should store 750 ft^3 of compressed air at a pressure of 3,000 psi. There are four design variables—radius (R) and length (L) of the cylindrical shell, shell thickness (Ts), and spherical head thickness (Th)—which have the following ranges of interest (unit: inch): $25 \le R \le 150,\ 1.0 \le T_S \le 1.375,\ 25 \le L \le 240,\ 0.625 \le T_h \le 1.0$.

The design objective is to minimize total system cost which is a combination of welding, material, and forming costs. Meanwhile, the constraints include ASME boiler and pressure code for wall thickness T_s and T_h, as well as the tank volume. The optimization model is formulated as:

$$\begin{aligned} &\min F = 0.6224 T_s R L + 1.7781 T_h R^2 + 3.1661 T_s^2 L + 19.84 T_s^2 R \\ &\text{Subject to } g_1 = T_s - 0.0193R \ge 0 \\ &\qquad g_2 = T_h - 0.00954R \ge 0 \\ &\qquad g_3 = pR^2 L + (4/3)pR^3 - 1.296E6 \ge 0 \\ &\qquad R \in [25,\ 150],\ T_s \in [1.0,\ 1.375],\ L \in [25,\ 240],\ T_h \in [0.625,\ 1.0] \end{aligned} \tag{14}$$

We assume the objective function is an expensive black-box function and thus the design optimization problem is of the form described by Eqs. 7-8. The optimum continuous solution is found as F = 7006.8, occurring at: R^* = 51.814 in., Ts^* = 1.0 in., L^* = 84.579 in., and Th^* = 0.625 in.

For the design problem, 10 different runs are executed. The minimum and the maximum for the 10 runs are recorded along with the average. The abbreviations in Table 3 have the same meaning as those in Table 1.

Table 3 Summary of test results on the pressure vessel design problem.

			Minimum		*nfe*		*nit*	
n	Space	Analy. Min.	Range of Variation	Median	Avg.	Median	Avg.	Median
4	[25 150] [25 240] [1.0 1.375] [.625 1.0]	7006.8	[7006.8, 7007.9]	7006.8	44.7	46	6.7	7

Because the obtained optima from the 10 runs are practically the same as the analytical optima, the differences between the analytical solution and the one found by the proposed method should be due to computational rounding errors. As shown in Table 3, the number of function evaluations, *nfe*, and the number of iterations, *nit*, are small.

DISCUSSIONS

As shown in Table 1, the proposed method finds the optimum of the QF function with only an average of 9.6 function evaluations. It is in fact true that for any *n*-D quadratic function, the number of the function evaluations needed is between nfe_l and nfe_u, defined as

$$nfe_l = (n+1)(n+2)/2 + 1 + [n/2]$$
$$nfe_u = nfe_l + 1 + n + [n/2] \tag{15}$$

nfe_l is required when the algorithm converges at the first iteration, where $[(n+1)(n+2)/2 + 1]$ is the number of points for the first quadratic fitting; $[n/2]$ is the number of points generated for model validation. According to the algorithm, the real function value at the model minimum is to be obtained if the model minimum is outside of the sub-region. Then the second iteration starts with n new sample points based on the mode-pursuing sampling method, and $[n/2]$ number of points for model validation. Since the function is quadratic, the model minimum at the first iteration should be the real optimum. Thus the second iteration will converge with the maximum number of function evaluations $nfe_u = nfe_l + 1 + n + [n/2]$. For n = 2, nfe_l = 8 and nfe_u = 12, which were exactly observed for the QF function.

Generally, it can be seen by construction that this algorithm requires $O(n^2)$ number of sample points. The number of points can be reduced if the quadratic model in Eq. 3 is further simplified.

CONCLUSIONS

In this work, a novel mode-pursuing sampling (MPS) method was developed, which is proven to be able to systematically converge to the global optimum if the objective

Copyright © 2003 by ASME

function is continuous in a neighborhood of the optimum. Based on MPS, a global optimization method was developed for expensive black-box functions. This optimization strategy applies local quadratic fitting to adaptively identify an attractive sub-region, in which local optimization can be performed. The MPS is then adjusted by a speed control factor to balance the "local" search and global "exploration." The developed optimization method is tested with both low (n=2) and high dimensionality (n=16) test problems. It is also applied to a real design optimization problem. Overall, the proposed method is found to have following benefits:

1. In general, it is an effective and efficient global optimization method for expensive black-box objective functions. The objective function can be either continuous or discontinuous. The MPS method is proven to converge to the global optimum if the objective is continuous in a neighborhood of the global optimum.
2. The method supports simultaneous computation. The extent of simultaneous computation can be controlled by the parameter n_p.
3. Though being random in nature, it is robust. It also requires little parameter tuning. For a new problem, the only optimization parameter might need the user specification is the c_d coefficient, which is also not mandatory.
4. It works for both unconstrained and constrained problems with expensive objective functions and inexpensive constraints.
5. Unlike other metamodeling-based global optimization methods such as in Refs. [12; 14], it does not call any existing global optimization tool. Therefore, it can work as a standalone global optimization method even for inexpensive functions.

Through the test, it is also found that for problems with a large quantity of local optima, the proposed optimization method may be trapped into a local optimum. This is due to the lack of a rigorous indicator of the global optimum. In such cases, the MPS may be applied alone as it guarantees to converge to the global optimum, as long as the function is continuous in a neighborhood of the global optimum.

ACKNOWLEDGMENTS

Financial support of this research project from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

REFERENCES

[1] Arora, J. S. (1989). *Introduction to Optimum Design*. McGraw-Hill, New York.

[2] Frits, A. P. and Mavris, D.N. (2002). A Screening Method For Customizating Designs Around Non-Convergent Region of Design Spaces. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia.

[3] Fu, J. C. and Wang, L. (2002). A random-discretization based Monte Carlo sampling method and its applications. *Methodology and Computing in Applied Probability*, 4: pp. 5-25.

[4] Hacher, K., Eddy, J. and Lewis, K. (2001). Tuning a Hybrid Optimization Algorithm by Determining the Modality of the Design Space. *ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Pittsburgh, PA.

[5] Haftka, R., Scott, E. P. and Cruz, J. R. (1998). Optimization and Experiments: A Survey. *Applied Mechanics Review*, 51(7): pp. 435-448.

[6] Jones, D. R., Schonlau, M. and Welch, W. J. (1998). Efficient Global Optimization of Expensive Black Box Functions. *Journal of Global Optimization*, 13: pp. 455-492.

[7] Montgomery, D. (1991). *Design and Analysis of Experiments*. John Wiley and Sons, New York.

[8] Neumaier, A. (2001). *Chapter 4: Constrained Global Optimization, In: Algorithms for Solving Nonlinear Constrained and Optimization Problems: The State of the Art*, COCONUT project.

[9] Sasena, M., Papalambros, P. and Goovaerts, P. (2002). Global Optimization of Problems with Disconnected Feasible Regions Via Surrogate Modeling. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia.

[10] Scholau, M. S., Welch, W. J. and Jones, D. R. (1998). Global Versus Local Search in Constrained Optimization of Computer Models, *New Development and Applications in Experimental Design, Lecture Notes-Monograph Series*.

[11] Törn, A. (1998). *Course notes on Probabilistic Algorithms*, Department of Computer Science, Åbo Akademi University, Turku, Finland.

[12] Wang, G. G. (2003). Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points. *Transactions of ASME, Journal of Mechanical Design*, 125: pp. 1-11.

[13] Wang, G. G., Dong, Z. and Aitchison, P. (2001). Adaptive Response Surface Method - A Global Optimization Scheme for Computation-intensive Design Problems. *Journal of Engineering Optimization*, 33(6): pp. 707-734.

[14] Wang, G. G. and Simpson, T. W. (2002). Fuzzy Clustering Based Hierarchical Metamodeling for Design Optimization. *9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia.

[15] Wilde, D. (1978). *Globally Optimal Design*. Wiley, New York

 Copyright © 2003 by ASME