SDO: A Statistical Method for Global Optimization *

Dennis D. Cox† Susan John‡

Abstract

An algorithm for finding global optima using statistical prediction is presented. Assuming a random function model, lower confidence bounds on predicted values are used for sequential selection of evaluation points and as a convergence criterion. Performance comparision with published results on several test functions indicate that the procedure is very efficient in finding the global optimum of a multimodal function, and in terminating with relatively few evaluations. The statistical computations involve linear algebra operations which are readily vectorized and is easy to adapt to parallel processing environments.

1 Introduction

There has been considerable interest in the problem of global optimization of nonlinear functions, since it has proved to be an invaluable tool in numerous applications. Basically, the problem involves finding the global optimal point $\mathbf{x}^*$ of a general function $f(\mathbf{x})$. There is no dearth of optimization procedures that have been developed for functions that are known to be unimodal within the domain of interest. However, in practice, the problem of optimizing a function possibly having many local extrema arises frequently. In this case, there are only a few good algorithms available. The procedures developed for unimodal functions do not work well in extending to multimodal functions since they would terminate at the first extremum encountered, which may or may not be the global. A thorough, systematic search in the function domain would be impractical since it would require far too many function evaluations. Therefore, one has to rely on a limited search of the domain of interest, either random or systematic, to locate the global extremum.

The simple random search is a widely used approach which uses the classical procedures to locate a local extremum from a number of random starting points, and then retains the best result as an approximation to the global. However, there are several drawbacks to this method – an obvious one is that it can be extremely inefficient to implement; also, there is no guarantee that the global extremum would be discovered, and one has to accept the uncertainty as to whether or not it has been found, especially if not much is known about the properties of the function.

*This research was supported by the National Science Foundation and the Air Force Office of Scientific Research under Grant Number DMS 86-09819. Computational support was also provided by Cray Research, Inc. The first author's research was partially supported by the National Security Administration under contract MDA 904-89-H-2011, and the National Science Foundation under grant number DMS-9317464.

†Department of Statistics, Rice University, Houston, TX.

‡National Center for Supercomputer Applications, University of Illinois, Champaign, IL.

A variety of methods have been proposed in the literature (see [10, 11] for descriptions of many of these methods). One technique is that of simulated annealing [2, 4, 6, 31] – which is a stochastic search procedure based on the concept of annealing from chemical physics. The main advantages of this method is its ability to avoid getting trapped in local extrema in its search for a global solution. It also does not require calculation of derivatives, so it can be used in the optimization of functions which may not be smooth. However, it appears to wander around the domain region aimlessly before it is able to home in on the global, making it inefficient and expensive to implement.

The search-clustering approach to global optimization [15, 29] attempts to reduce the number of function evaluations over a simple random search by determining the domains of attraction of the local minima. The method identifies "clusters", groups of points which lead to the same local minimum, and thus avoids unnecessary searches.

There are also various types of random-search procedures in the literature [10, 21, 24, 26] – in particular, in the adaptive random search procedure, the search direction and step size are based on some appropriate probability density function whose parameters are adapted to the local behavior of the function to be minimized. The major advantages are that they are easy to implement and they lead to the global (rather than local) minimum. Convergence can be very slow, and the performance of the method is highly problem dependent.

Trajectory-description methods [3, 16, 30] can be used to deal with the minimization of multimodal functions by generating trajectories in the domain of interest which will pass through all the minima of the function, and thus ensure locating the global minumum. This method requires the evaluation of the Hessian and its inverse, which is restrictive, and for large matrices, expensive to compute.

In this paper, we introduce a statistical method for global optimization which we have found to be successful on a number of test-case problems. We refer to the method as *Sequential Design for Optimization* (SDO). In the SDO approach, the function to be minimized is modelled as a realization of a stochastic process with mean function from a given class of functions (which is a linear model) and a given covariance function. Currently, the algorithm consists of selecting a grid of points in the domain of interest of the function. A small fraction of these points will form a set of observation sites at which the function will be evaluated. The point at which the function has its minimum value among this set of observation sites is noted. The grid points which are not observation sites will form the set of prediction sites; at these points the best linear unbiased predictor (BLUP) of the function as well as its mean squared error (MSE) are computed based on the given linear model, covariance parameters, and set of observation sites. The values of BLUP and MSE are used to compute a lower confidence bound at each of the prediction sites. Assuming that the stochastic process models the function well, we select the prediction site which has the smallest value of the lower confidence bound for the next evaluation. This point is then added to the set of observation sites, and the function is evaluated at this point. We compare this function value with the current minimum function value. If it is smaller, it becomes the new function minimum. New values of the BLUP and MSE are computed based on the updated set of observation sites, and the algorithm proceeds as above, until a user-specified maximum number of function evaluations, or a convergence criterion has been reached. This criterion compares the current minimum function value with the minimum of the lower confidence bounds at the prediction sites. The criterion is met if the current minimum function value is the lesser.

Our approach is quite similar to the Bayesian approach of Mockus [22] and to the response surface model used by Elder [12] [13]. Both are motivated by earlier results of Kushner's [18] [19]. Mockus's method was motivated by considering the dynamic programming solution of a decision theoretic formulation of the global optimization problem (pages 4-7 of [22]). The exact solution is impractical to compute, so he proposed a heuristic algorithm (section 2.5 of [22]). Elder's GROPE algorithm is also based on stochastic process model of the objective function. For his model, the predictor takes a piecewise linear form over a triangulation of the domain of the objective function. Further discussion and comparison with the work of these authors is given below.

There are many advantages to SDO over other approaches to global optimization in the literature. It does not require the calculation of derivatives, and the explicit form of the function need not be known. The method easily extends to the situation in which only noisy, inexact evaluations of the objective function are available. Most importantly, we have found it highly efficient in terms of the number of function evaluations for many of the examples considered when compared to other methods reported in the literature. Furthermore, the convergence criterion provides a natural stopping rule, whereas most other procedures are open-ended, requiring an arbitrary stopping rule. However, there is still some work needed to remove the restriction to a grid, to make the approach feasible in high dimensions, and to determine appropriate values of the parameters in the stochastic process. Nonetheless, we believe the results presented here are promising enough to justify this preliminary report.

The rest of the paper is organized as follows: Section 2 introduces stochastic processes and BLUP's, and their application in the optimization algorithm. It also describes an updating algorithm due to Cox and Chang [7] to add new points to the set of observation sites efficiently. Section 3 presents our results along with comparisions with other algorithms for several test-cases. The last section talks about future research to be done on this algorithm.

Acknowledgement: The basic idea for this procedure was proposed to the authors by Professors Jerome Sacks and William Welch. We would also like to thank John Elder, IV, for providing a copy of his dissertation [12].

2 Description of the Algorithm

2.1 Basic BLUP formulation

In this section, we introduce the statistical model and the Best Linear Unbiased Predictor (BLUP) with its mean square error (MSE) based on the model which will be used in the optimization algorithm. BLUP's are used to predict values of a random function from observations or evaluations of the function.

Let $\chi \subseteq \mathbf{R^d}$ be the index set of a real valued stochastic process $Y(x)$ with mean function

$$EY(x) = \beta^t g(x), \tag{1}$$

which is a linear model, and a covariance function

$$Cov[Y(t), Y(u)] = \sigma^2 v(t, u) \tag{2}$$

where g: $\chi \rightarrow \mathbf{R^p}$ is known, $\beta \in \mathbf{R^p}$ is unknown. The process variance parameter σ^2 may either be known or unknown, but we treat it as unknown. Some possible choices of the covariance function are from the "power exponential family" that has been used in [28]. This is given by

$$v(t, u) = exp[-\sum_{i=1}^{d} \theta_i \mid t_i - u_i \mid^r] \tag{3}$$

where the $\theta > 0$ and $0 < r \leq 2$. The parameter θ defines the correlation structure of Y – large θ implies small correlations between observations, and small θ implies large correlations between observations. The parameter r also controls the smoothness of the random function, but in a more subtle way. When $r = 2$, the random function has infinitely many mean square derivatives, and when $1 < r < 2$, it has one derivative in a mean square sense.

For a given set of observation sites, $\{s_1, s_2, \ldots, s_n\} \subseteq \chi$, let y_n be an n-vector of observations given by

$$y_n[i] = Y(s_i) \tag{4}$$

Define the $n \times n$ matrix V_n and the $n \times p$ matrix F_n by

$$V_n[i,j] = \sigma^{-2} Cov[Y(s_i), Y(s_j)] = v(s_i, s_j) \tag{5}$$

$$F_n[i,.] = f(s_i)^t$$

$F_n[i,.]$ denotes the i th row of F_n. For any prediction site $x \in \chi$, the n-vector $v_n(x)$ has ith component

$$v_n(x)[i] = v(x, s_i)$$

The best linear unbiased predictor of $Y(x)$ based on the observation y is given by (see [27])

$$\hat{Y}(x) = (f(x) - F_n^t V_n^{-1} v_n(x))^t \hat{\beta} + v_n(x)^t V_n^{-1} y_n \tag{6}$$

where

$$\hat{\beta} = (F_n^t V_n^{-1} F_n)^{-1} F_n^t V_n^{-1} y_n$$

is the generalized least squares estimator of β. The normalized mean squared error of the prediction is

$$\begin{aligned} mse(x) &= \sigma^{-2} E[\hat{Y}(x) - Y(x)]^2 \\ &= v(x,x) - v_n(x)^t V_n^{-1} v_n(x) + \\ &\quad [F_n^t V_n^{-1} v_n(x) - f(x)]^t [F_n^t V_n^{-1} F_n]^{-1} [F_n^t V_n^{-1} v_n(x) - f(x)] \end{aligned} \tag{7}$$

The maximum likelihood estimate of σ^2 is

$$\hat{\sigma^2} = \frac{1}{n}(y_n - F_n\hat{\beta})^t V_n^{-1}(y_n - F_n\hat{\beta})$$

If the observations are noisy, then we may replace equation (4) with $y_n[i] = Y(s_i) + \epsilon_i$where the ϵ_i are uncorrelated, mean zero random variables with constant variance $\rho\sigma^2$. It is only necessary to add in ρI_n in the definition of V in equation (5).

2.2 Updatable Computations for BLUP

Following [5], assume that there are n observation sites $\{s_1, s_2, \ldots, s_n\}$, and that one wishes to make predictions at a set of sites $\{x_1, x_2, \ldots, x_m\} \subseteq \chi$. Define matrices $U(n \times m), L(p \times m)$, and $W(m \times m)$, where $U[.,j] = v_n(x_j), L[.,j] = f(x_j),$, and $W[i,j] = v(x_i, x_j)$

Let T_{11} be the upper triangular Cholesky factor of V_n and put

$$[T_{12}\ T_{13}\ T_{14}] = T_{11}^{-1}\,[F_n\ y_n\ U_{nm}]$$

$$T_{44} = W - U^t V_n^{-1} U$$

Let $Q \times R$ be the Q-R Decomposition of T_{12}, where Q is an $n \times p$ orthogonal matrix and R is a $p \times p$ triangular matrix. The predictions and the mean square error of the predictions can be written as:

$$\hat{\beta} = R^{-1} Q^t T_{13}$$

$$\hat{Y} = (L^t R^{-1} - T_{14}^t Q) Q^t T_{13} + T_{14}^t T_{13}$$

$$mse = diag[T_{44} + (R^{-t} L - Q^t T_{14})^t (R^{-t} L - Q^t T_{14})]$$

where $\hat{Y} = [\hat{Y}(x_1), \ldots, \hat{Y}(x_m)]^t$ and $mse = [mse(x_1), \ldots, mse(x_m)]^t$

$$\hat{\sigma}^2 = \frac{1}{n} [\|T_{13}\|^2 - \|Q^t T_{13}\|^2]$$

The updating algorithm of [5] can be used to add a new observation site. This is done by switching a site from the set of prediction sites to the set of observation sites, and then updating the predictions and the mean squared error of the predictions. This method of updating has been found to save significant computing time in our application.

The matrices T_{12}, T_{13}, T_{14}, and T_{44} can be updated with Cholesky operations. The Q-R decomposition factors of the new matrix T_{12} are updated by applying Given's rotation transformations. See [5] for details.

There are some modifications of the prediction/updating algorithm to ensure numerical stability that we need to discuss. The algorithm should not allow the addition of an observation site if it causes the Cholesky covariance factor T_{11} to become nearly singular. It is easy to check for such ill conditioning by comparing an estimate of computational error with the magnitude of the next diagonal entry to be used in the Cholesky sweep (the corresponding diagonal entry of T_{44} when a prediction site is switched into the observation sites). See Chapter 10, section 1, of [5] for details. One can object that the new data is not being used, but as argued in [5], such observations contain essentially no new information, provided they are consistent with the random function model. In fact, the algorithm checks for this consistency and proceeds if there is no problem, and otherwise stops with an error message. When this error condition occurs, it means that the stochastic process model is inadequate for predicting the given function

There is also an efficient updating algorithm for adding new prediction sites [5], which we anticipate will be useful in extending the optimization algorithm described in this paper to higher dimensions where it is not practical to use a fixed grid.

2.3 Description of the Optimization Algorithm

In SDO, the function $f(x)$ to be minimized is modeled as a realization of a stochastic process $Y(x)$, as described in Section 2.1, with mean function and covariance function given by equations (1) and (2). We used the $v(t, u)$ from the "power exponential family" defined in equation (3). We used one of the following two polynomial models for the linear model for the mean function:

1. Constant linear model: $p = 1$
   $g(t) = 1$

2. Linear or First order linear model: $p = d + 1$
   $g(t) = (1, t_1, t_2, \ldots, t_d)$

Numerous other mean functions are available, of course.

Currently, the algorithm consists of selecting a grid of points in the domain of interest of the function. A small number n of these points $\{s_1, s_2, \ldots, s_n\}$will form a set of *observation sites*, where the objective function is evaluated, and the rest, m, of these points $\{x_1, x_2, \ldots, x_m\}$will form the set of *prediction sites*. The total number of grid points as well as the initial number of observation sites are determined by the user. The function $f(x)$ is evaluated at each of the observation sites and the point at which the function has its minimum value among this set of observation sites is retained as the current minimum function value (y_{min}).

The user decides on the linear model for the mean function of the stochastic process as well as parameter values for the θ_i and r. Based on these values, and a given initial set of observation sites, the BLUP, $\hat{y}$, of the function and MSE of the predictor are computed at each of the prediction sites using the formulae given in equations (6) and (7) respectively. An estimate of σ^2, denoted by $\hat{\sigma^2}$, is also obtained and used in the MSE. These values of the BLUP and MSE are used to compute a lower confidence bound at each of the prediction sites given by

$$lcb(x_i) = \hat{y}(x_i) - b\hat{\sigma}\sqrt{mse(x_i)} \tag{8}$$

where $i = 1, 2, \ldots, m$, and b is a factor determined by the user. We used $b = 2$ and $b = 2.5$ in all of the test runs that we made.

Assuming that the stochastic process models the function well, we select the prediction site which has the smallest value of the lower confidence bound as a possible minimum to the function $f(x)$. The function is then evaluated at this point. We compare this function value with the current minimum function value, and if it is smaller, it becomes the new function minimum, denoted y_{min}. The point is added to the set of observation sites, and using the updating algorithm described in Section 2.2, the $\hat{y}$, mse, and lcb arrays are updated, as well as $\hat{\sigma^2}$.

The algorithm proceeds until a user-specified maximum number of function evaluations or the convergence criterion is reached. The convergence criterion is given by the condition

$$y_{min} < \min_i(lcb(x_i)) \tag{9}$$

where the minimum is taken over grid sites which remain as possible candidates to be selected for evaluation. Since $lcb(x_i)$ is highly likely to be less than $f(x_i)$, one can have high confidence that y_{min} is the global minimum. Setting larger values of b increases the confidence that the global minimum has been found, but at this point in the investigation we do not attach any numerical value to the actual probability of having found the minimum
.

The algorithm due to Mockus [22] is quite similar to the one proposed above in that selection of the next observation site is accomplished by minimization of a "surrogate" to the objective function. Our surrogate objective is given in (8) and is fairly easy to compute. Mockus's surrogate also requires calculation of $\hat{y}$ and mse as well, but calculation of his surrogate first requires the minimization of $\hat{y}$ (equation (5.1.2) of [22]) and then minimizing an integral expression involving the minimum of $\hat{y}$ and mse (equation (5.1.1) of [22]). There is also a user adjustable "tuning" constant (the ϵ in (5.1.2) of [22]) similar to b in (8).

3 Comparision with Other Methods

3.1 Description of the Test Cases

We considered 8 different test functions taken from the literature, denotes f_1 through f_8. All of the test functions are 2-dimensional and multi-modal in the domain of interest. Table 4 gives the name, reference, domain, and global minimum of each of the test functions. The formulae for the functions are:

$$f_1(x,y) = 2x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7;$$

$$f_2(x,y) = 2x^2 + 2y^2 - 0.3\cos(3\pi x)\cos(4\pi y) + 0.3;$$

$$f_3(x,y) = 2x^2 + 2y^2 - 0.3\cos(3\pi x + 4\pi y) + 0.3;$$

$$f_4(x,y) = \sin^2 x + \sin^2 y - 0.1\exp(-x^2 - y^2);$$

$$f_5(x,y) = 2x^2 - 1.05x^4 + x^6/6 - xy + y^2;$$

$$f_6(x,y) = (1 - 8x + 7x^2 - 7x^3/3 + x^4/4)y^2\exp(-y);$$

$$f_7(x,y) = (1 + (x+y+1)^2 g_1(x,y))(30 + (2x-3y)^2 g_2(x,y))$$
where $g_1(x,y) = 19 - 14x + 3x^2 - 14y + 6xy + 3y^2$ and
$g_2(x,y) = 18 - 32x + 12x^2 + 48y - 36xy + 27y^2$;

$$f_8(x,y) = (y - (5/4)\pi^2 x^2 + (5/\pi)x - 6)^2 + 10(1 - 1/(8\pi))\cos x + 10.$$

3.2 Results from the literature

Available results from the literature for the eight test functions are presented in Tables 2, 3, and 4. The tables contains the method used and the number of function evaluations to termination for that method. The final column of each of these tables contains information on the termination rule. An "f" in that column means the algorithm was stopped when the global minimum was found, which of course is not a practically useful stopping criterion since it requires knowing the global minimum. A "c" in that column means there was a criterion for convergence that was used to terminate the algorithm. Finally, a "u" means we could not determine the reason for terminating the algorithm.

3.3 Results from New Algorithm

The algorithm begins with transforming the domain of interest of the function to a unit square. A uniform grid of points is then selected in the unit square as follows: The user specifies the number of grid points, ngp, in each coordinate. The user also specifies the initial number of observation sites in each coordinate, $nobs$. In two dimensions there will be a total of ngp^2 grid points, and $n = nobs^2$ initial observation sites. These observation sites are uniformly spaced in the grid. Thus there will be a total of $m = (ngp^2 - nobs^2)$ sites that will form the initial set of prediction sites.

The values of the parameters used in the test runs were:

(i) Initial number of observation sites, $n = 16$;

(ii) $\theta_i = \theta\ \forall i$, where θ is given;

(iii) $r = 1$;

(iv) The linear model for the mean function of the stochastic process was either constant or first-order: $p = 1$ or $p = 3$.

Table 5 gives the results from the SDO algorithm.

The comparision of Tables 2 through 4 with Table 5 for the eight test functions is not entirely fair because we restricted ourselves to the grid described above. Because of the grid spacing, we can only claim to have found the global minimum within $\pm 0.5\%$ for the ranges of the x components. Nonetheless. for four of the test functions, f_1, f_4, f_5, and f_6, SDO performed considerably better than the best results given in Table 2. For two others (f_2 and f_3), SDO's performance, measured in terms of the number of function evaluations to convergence, was comparable with the best in Table 2. In the first five test functions, the first point that was added to the original set of observation sites was the global minimum.

The method had problems with f_7, the Goldstein and Price function. This function is a trough with a very flat bottom and steep sides in the domain of interest, and our simple stochastic process model didn't handle such a function well. However, with $\theta = 2.5$, linear model, and $b = 2.5$ as the choice of parameters, it found the minimum on the grid before running into problems. We had no success with the other case, namely, $\theta = 1$, constant model, and $b = 2.0$. For function f_7, the algorithm terminated before convergence because there was an attempt to add an evaluation (observation site) which caused instability in T_{11}, and for which the consistency checks failed. This means that the observations were not consistent with the random function model. We believe this resulted from the flat bottom and steep sides. Possibly increasing the covariance parameter θ still further would take care of this problem, but in general we think some fundamental modification is required. In any event, it took relatively few evaluations (48 and 99 with $\theta = 1.0$ and $\theta = 2.5$ respectively) for the algorithm to decide it could not handle this algorithm.

The Branin function, f_8, has three global minima: (-π,12.25), (9.4248, 2.25) and (π,2,25). It should be noted that the algorithm found the global minimum on the grid (-3.125,12.225) and this was closest to the point (-π,12.25). We found that the algorithm stepped through the other two points on the grid (9.475, 2.335) and (3.175, 2.175) that were closest to the other two global minima before it converged to the global minimum on the grid. The number of evaluations to find the minimum of f_8 for our algorithm is about the same as the best results appearing in Table 4. Also, the number of evaluations for convergence is impressively low, although there are no direct comparisons possible with the results given in Table 4 (except for results of Fagiouli [14]) because there were not any cases where a convergence criterion was apparently used.

4 Future Research

Future research will focus on developing the algorithm. Firstly, we would like to expand the algorithm to higher dimensions. We wish to eliminate the grid structure since this limits the search for the global minimum and also would be expensive to implement in more than two dimensions. One thought would be to start with a small set of observation sites and a random set of prediction sites, and then use a random sequential procedure to add points that may be possible candidates for the global minimum from the random set of prediction sites. It would be necessary to update the random sites from time to time, and perhaps there is an intelligent way to specify a probability density for the random sites. Another thought would be to start with an extremely coarse grid, and use this to identify subdomain(s) in the domain of interest which may contain the global minimum, and then focus on these subdomains to find the global minimum using a random procedure. This could be especially useful when there are multiple global minima and the aim is to locate all of them.

Another restriction that we forsee removing is the fixed values of the parameters θ and r; instead we would like to estimate parameters using, say, the maximum likelihood method. We would also like to find modifications which will handle functions like the Goldstein-Price. There is also some work to be done investigating the choice of b. It may be desirable to use different values of b in the selection of the next evaluation point and the convergence criterion. It will take some experimentation to develop the procedure, and of course theoretical results would be desirable as well. However, the results obtained so far look promising enough that we are optimistic about future success.

Note: Some preliminary versions of the results presented here appear in a previous paper by the authors [8].

References

[1] G. A. Bekey and M. T. Ung. A comparitive evaluation of two global search algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 4:112–116, 1974.

[2] I. O. Bohachevsky, M. E. Johnson, and M. L. Stein. Generalized simulated annealing for function optimization. *Technometrics*, 28:209–217, 1986.

[3] F. H. Branin. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM J. Res. Develop*, 16:504–522, 1972.

[4] D. G. Brooks and W. A. Verdini. Computational experience with generalized simulated annealing. In E. J. Wegman, editor, *Proceedings of the 20th Symposium on the Interface*, pages 135–143, Alexandria, VA, 1988. American Statistical Association.

[5] Y. F. Chang. *Recursive Methods for statistical prediction with applications.* PhD thesis, University of Illinois, Urbana-Champaign, IL, 1990.

[6] O. Corana, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Transactions on Mathematical Software*, 13:262–280, 1987.

[7] D. D. Cox and Y. F. Chang. Updatable computations for best linear unbiased predictors. Technical Report 34, Department of Statistics, University of Illinois, Urbana-Champaign, IL, September 1989.

[8] D. D. Cox and S. John. A statistical method for global optimization. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, Oct. 18 – 21, 1992, Chicago, IL, Institute for Electrical and Electronic Engineers, Inc., 1241 – 1246, 1992.

[9] L. DeBiase and F. Frontini. A stochastic method for global optimization: its structure and numerical performance. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation 2*, pages 85–102. North-Holland, Amsterdam, 1978.

[10] L. C. W. Dixon and G. P. Szego. *Towards Global Optimisation.* North-Holland, Amsterdam, 1975.

[11] L. C. W. Dixon and G. P. Szego. *Towards Global Optimisation 2.* North-Holland, Amsterdam, 1978.

[12] J. F. Elder, IV. *Efficient Optimization through Response Surface Modeling: A GROPE Algorithm.* Unpublished Ph.D. dissertation, University of Virginia, Charlottesville, VA, 1993.

[13] J. F. Elder, IV. Global R^d Optimization when Probes are Expensive: the GROPE Algorithm. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, Oct. 18 – 21, 1992, Chicago, IL, Institute for Electrical and Electronic Engineers, Inc., 1992.

[14] E. Fagiuoli, P. Pianca, and M. Zecchin. A mixed stochastic-deterministic technique for global optimization. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation 2*, pages 103–116. North-Holland, Amsterdam, 1978.

[15] J. Gomulka. A user's experience with Törn's clustering algorithm. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation 2*, pages 63–70. North-Holland, Amsterdam, 1978.

[16] J. W. Hardy. An implemented extension of Branin's method. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation*, pages 117–142. North-Holland, Amsterdam, 1975.

[17] A. R. Kan and G. T. Timmer. Stochastic methods for global optimization. *American Journal of Mathematical and Management Sciences*, 4:7–40, 1984.

[18] H. J. Kushner. Stochastic model of an unknown function. *J. Math. Analysis and Applications*, 5:150–167, 1962.

[19] H. J. Kushner. A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise. *J. of Basic Engineering*, March:97–106, 1964.

[20] P. Laud, L. M. Berliner, and P. K. Goel. A stochastic probing algorithm for global optimization. In K. Berk and L. Malone, editors, *Proceedings of the 21st Symposium on the Interface Statistical Software and the Human Interface*, pages 220–229, Alexandria, VA, 1989. American Statistical Association.

[21] S. F. Masri, G. A. Bekey, and F. B. Safford. A global optimization algorithm using adaptive random search. *Applied Math. and Computation*, 7:353–375, 1980.

[22] J. Mockus. *Bayesian Approach to Global Optimization Theory and Applications.* Kluwer Academic, Dordrecht, 1989.

[23] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation 2*, pages 117–129. North-Holland, Amsterdam, 1978.

[24] W. L. Price. A controlled random search procedure for global optimization. *Computer Journal*, 20:367–370, 1977.

[25] W. L. Price. A controlled random search procedure for global optimization. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation 2*, pages 71–84. North-Holland, Amsterdam, 1978.

[26] L. Pronzato, E. Walter, A. Venot, and J. F. Lebruchec. A general purpose global optimizer: Implementation and applications. *Mathematics and Computers in Simulation*, 26:412–422, 1984.

[27] J. Sacks, S. B. Schiller, and W. J. Welch. Designs for computer experiments. *Technometrics*, 31:41–47, 1989.

[28] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–435, 1989.

[29] A. A. Törn. A search-clustering approach to global optimization. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation 2*, pages 49–62. North-Holland, Amsterdam, 1978.

[30] G. Treccani. A new strategy for global minimization. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation*, pages 143–148. North-Holland, Amsterdam, 1975.

[31] D. Vanderbilt and S. Louie. Monte Carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 56:259–271, 1984.

TABLE 1
Summary of the Test Functions

Name & Reference	Fn	Domain	Global Minima
Simulated Annealing I [2]	f_1	$[-1,1]^2$	0 at (0,0)
Simulated Annealing II [2]	f_2	$[-1,1]^2$	0 at (0,0)
Simulated Annealing III [2]	f_3	$[-1,1]^2$	0 at (0,0)
Sine function [24]	f_4	$[-10,10]^2$	0.9 at (0,0)
Three Hump Camel-back [16]	f_5	$[-3,3]\times[-1.5,1.5]$	0 at (0,0)
Hosaki [1]	f_6	$[0,5]\times[0,6]$	-2.345 at (4,2)
Goldstein & Price [26]	f_7	$[-2,2]^2$	3 at (0,-1)
Branin [3]	f_8	$[-5,10]\times[0,15]$	0.3979 at $(-\pi,12.25)$, $(\pi,2.25)$, & $(9.4248,2.25)$

Table 2

Results from the literature: Test functions f_1 through f_6

Fn	Method	Reference	No. of Eval.	Termination
f_1	Simulated Annealing	[2]	150	u
	Stochastic Probing	[20]	66[1]	c
	GROPE	[12]	26	f
f_2	Simulated Annealing	[2]	400	u
	Stochastic Probing	[20]	66[1]	c
	GROPE	[12]	26	f
f_3	Simulated Annealing	[2]	400	u
	Stochastic Probing	[20]	100[2]	c
	GROPE	[12]	26	f
f_4	Controlled Random Search	[24]	700	f
	GROPE	[12]	352	f
f_5	Trajectory	[16]	1136	f
	Adaptive Random Search	[26]	838	f
	GROPE	[12]	24	f
f_6	Modified Random Creep	[1]	451	f
	Adaptive Random Search	[26]	830	f
	GROPE	[12]	27	f

1 – median of 75 trials

2 – median of 74 trials

Table 3
Results from the literature: Test function f_7

Fn	Method	Reference	No. of Eval.	Termination
f_7	Density Clustering	[29]	2499	u
	Density Clustering	[15]	1495	u
	Controlled Random Search	[25]	2500	u
	Space-covering	[14]	158 (1076)	f(c^3)
	Density Clustering	[9]	378	c
	Stochastic Optimization	[22]	362	f
	Trajectory	[16]	4242	f
	Random Direction	[11]	300	u
	Stochastic Optimization	[23]	520	c
	Adaptive Random Search	[26]	657	f
	Multi-start	[17]	4400	u
	Single linkage clustering	[17]	398	u
	Multi level single linkage	[17]	294	u
	Self-Regulating Simulated Annealing	[31]	1186^4	u
	Generalized Simulated Annealing	[4]	292	f
	GROPE	[12]	21	f

3 – number of function evaluations to achieve 0.9 convergence in probability

4 – mean of 100 trials

TABLE 4

Results from the literature: Test function f_8

Fn	Method	Reference	No. of Eval.	Termination
f_8	Density Clustering	[29]	1558	u
	Density Clustering	[15]	1318	u
	Controlled Random Search (2 trials)	[25]	1600	u
			1800	u
	Space-covering	[14]	1600 (6012)	f(c^3)
	Density Clustering	[9]	597[5]	c
	Stochastic Optimization	[22]	189	f
	Trajectory	[16]	6005	f
	Random Direction	[11]	160	u
	Stochastic Optimization	[23]	1021	f
	Multi-start	[17]	1600	u
	Single linkage clustering	[17]	235	u
	Multi level single linkage	[17]	219	u
	Self-Regulating Simulated Annealing	[31]	557[4]	u
	Generalized Simulated Annealing	[4]	292	f
	GROPE	[12]	29	f

3 – number of function evaluations to achieve 0.9 convergence in probability

4 – mean of 100 trials

5 – terminated at f = 1.25 (global minima at f = 0.3979)

TABLE 5
Results from SDO Algorithm

	Parameter Values					No. of Function evaluations	
Function	n	m	θ	Mean fn	b	to find minimum	to converge
f_1	16	9785	1	Constant	2.0	17	38
	16	9785	2.5	Linear	2.5	17	48
f_2	16	9785	1	Constant	2.0	17	62
	16	9785	2.5	Linear	2.5	17	68
f_3	16	9785	1	Constant	2.0	17	121
	16	9785	2.5	Linear	2.5	17	113
f_4	16	9785	1	Constant	2.0	17	97
	16	9785	2.5	Linear	2.5	17	95
f_5	16	9785	1	Constant	2.0	17	162
	16	9785	2.5	Linear	2.5	17	212
f_6	16	9785	1	Constant	2.0	55[1]	79
	16	9785	2.5	Linear	2.5	36[1]	107
f_7	16	9785	1	Constant	2.0	§	§
	16	9785	2.5	Linear	2.5	21[2]	§
f_8	16	9984	1	Constant	2.0	172[3]	488
	16	9984	2.5	Linear	2.5	223[3]	589

1 – the minimum found was -2.3449 at (4.01515, 2.0303) which was the point on the grid closest to the global minimum.

§ – no results; the algorithm stopped before finding the minimum and/or converging.

2 – the minimum found was 3.0448 at (0, -1.0101) which was the point on the grid closest to the global minimum.

3 – the minimum found was 0.3994 at (-3.125, 12.225) which was the point on the grid closest to one of the global minima and having the smallest function value.