A Comparison of Bayesian/Sampling Global Optimization Techniques

Bruce E. Stuckman, *Senior Member, IEEE*, and Eric E. Easom

Abstract—**A survey of current global optimization techniques for continuous variables is presented, inspired by recent publications of computer coding of several popular Bayesian/sampling methods. The methods of Perttunen, Stuckman, Mockus, Zilinskas, and Shaltenis are compared with other global optimization algorithms, specifically, a clustering algorithm, a simulated annealing algorithm and the Monte Carlo method. Results are given for these methods based upon the experimental rate of convergence on a series of standard test functions. A new test function is presented that has a global solution within an area that is small in comparison with the search space.**

I. Introduction

GLOBAL OPTIMIZATION techniques have been developed to solve nonlinear problems that are nonconvex. The advantage of global optimization techniques over local optimization techniques is the ability to find the overall or global" maximum or minimum. Local optimization techniques are limited to finding local minima whereas global optimization techniques seek out the overall minimum solution or global minimum over a bounded space.

One consideration when choosing a global optimization technique is the cost of evaluation. Many problems in the engineering and science disciplines have a high cost of evaluation. Thus, one may be willing to utilize some computation time in order to minimize the number of function evaluations. Applications of global optimization include designs of a communication system [1], a nonlinear control system [2], an electronic circuit [3] and an optical filter [4].

Since several global optimization techniques exist, a comparison study might determine that techniques produce the best results. There are two methods of comparing global optimization techniques: 1) convergence and 2) performance on a series of functions. When testing for convergence, optimization techniques are tested on the ability to converge on the optimal solution. A proof of convergence is theoretically satisfying; however, it implies an infinite number of guesses in order to obtain the solution. For practical reasons, this would be impossible to complete. Furthermore, convergence can only be proven, in general, for a limited class of functions.

A second method of comparing global optimization techniques is to implement different methods on several n-parameter test functions. Since these test functions are chosen to simulate many of the possible attributes of real-world applications, these results provide an indication of how the global optimization method will work on actual problems. In one such comparison published in 1978 [5], Dixon and Szego compiled a set of standard differentiable test functions and reported the results of several optimization techniques on these test functions. The research reported the number of function evaluations and the normalized computation time each method took to converge.

Several new global optimization techniques have since been developed. One particular class of global optimization techniques that have obtained promising results on functions of continuous variables are Bayesian/sampling techniques. Examples include methods developed by Stuckman [6], Mockus [7], Perttunen [8], Zilinskas [9] and Shaltenis [10]. The purpose of this research is to compare Bayesian/sampling techniques and some other popular methods on a number of test functions. The other methods are chosen to be Monte Carlo optimization, the clustering method developed by Torn [11], and a simulated annealing algorithm [12], specifically a commonly used method that utilizes Boltzmann's distribution with a fixed search radius and a logarithmic annealing schedule. Summarizing, the following methods of global optimization will be used in the survey.

1) Stuckman's Method—Bayesian/Sampling
2) Mockus's Method—Bayesian/Sampling
3) Perttunen's Method—Bayesian/Sampling
4) Torn's Method—Clustering
5) Monte Carlo—Uniform Random
6) Simulated Annealing—Random method
7) Zilinskas—Extrapolation—Bayesian/Sampling
8) Shaltenis and Dzemyda—Bayesian/Uniform Deterministic

To make a fair comparison of global optimization techniques, the computer code has been obtained from each of the respective authors except in the case of the simulated annealing algorithm and the Monte Carlo method, since both methods are relatively simple to implement. Secondly, a fair comparison would include some termination criterion, such as termination after N function evaluations.

A standard set of differentiable test functions compiled by Dixon and Szego [5] will be used to standardize this survey. This set includes the following test functions:

1) Goldstein—Price [13]
2) Branin's RCOS [14]
3) Shekel SQRN5 [15]
4) Shekel SQRN7 [15]

Manuscript received August 11, 1990; revised February 16, 1991, August 13, 1991, and February 9, 1992.

The authors are with the Department of Electrical Engineering, University of Louisville, Louisville, KY 40292.

IEEE Log Number 9201407.

5) Shekel SQRN10 [15]. Additional test functions that will be used include:
6) six-hump camel back [16]
7) Shubert 2-D [17]
8) Dual Square Sinc Stuckman test function [18].

Also, the authors will present and test a new test function whose global minimum has a small area with respect to the search space.

This research will compare each method after 20, 50, 100, 200, 500 and 1000 iterations on each of the test functions listed above. A comparison of methods will be made based upon which methods of global optimization yield the best results in the fewest number of function evaluations.

II. Methods of Global Optimization

A. Stuckman's Method:

Classified as a Bayesian/Sampling technique, Stuckman's n-dimensional method of global optimization is based on Kushner's one-dimensional (1-D) search. Both methods model an unknown objective function, $f(X)$ where $X = \{x_1, x_2, \ldots, x_n\}$, as a Brownian motion process, or more specifically, they calculate the point of the next guess, $f(X')$, which has the maximum probability of exceeding the previous point (for the maximization case), $f(X)$, by some positive constant ϵ_n, as shown in expression (2):

$$P[f(X') > f(X) + \epsilon_n]. \tag{2}$$

Stuckman's method searches for the point of maximum probability along a line segment drawn between two points where the function has been evaluated. New line segments are formed and the process is continued, searching all the line segments that have been formed.

Unlike Kushner's 1-D model, Stuckman's method has the ability to select dynamically the constant ϵ_n. Initially, ϵ_n is large, but decreases such that the optimization search becomes more local in nature, thus speeding up convergence [6].

B. Mockus's Method:

This technique of global optimization is also a Bayesian/ Sampling technique. The particular global optimization technique considered is the Bayesian one-step method [7], where the average deviation from the maximum functional value is minimized. Therefore, after each stage, it is assumed that the next observation will result in the minimum function value. This method is applicable to any n-dimensional test function.

The one-step model developed by Mockus [7] is given by expression (3):

$$x_{n+1} J \arg\min_{x\in A}(\frac{1}{\delta}) \int_{\pm\infty} \min(y, c) \exp[(-1/2)((y-\mu)\sigma^2]\ dy \tag{3}$$

where $y_1 = f(x_1), \ldots, y_n = f(x_n)$, and $c = \min_{x\in A}(\mu - \epsilon), \epsilon \geq 0$.

The next point to be chosen is x_{n+1}, which is the point where the risk function described by expression 3 is minimized. Note, the risk function described by expression 3 is based on a Gaussian distribution, thus indicating the assumptions for this model. With the knowledge of *a priori* information, the distribution could be changed to one that will yield better results. The constant ϵ determines the nature of the search. When ϵ is large, the search will be nearly resemble a uniform random approach, whereas when ϵ is small, the search is strictly a one-step Bayesian approach.

C. Perttunen's Method:

Like Stuckman's method of global optimization, Perttunen's Bayesian/Sampling method tries to maximize the probability of exceeding the previous function value by some positive constant, ϵ_n. There are two major differences between Perttunen's and Stuckman's methods of global optimization. First, Perttunen's method transforms objective function evaluations into statistical ranks, to eliminate the dependence of the model on normally distributed data [8]. The second difference is Stuckman's method searched for the point of maximum probability along line segments between two points where the objective function had been evaluated, whereas Perttunen's method searches along a region defined by the dimension of the objective function. For example, if given a two-dimensional (2-D) test function, the search for the point of maximum probability will be over a triangular region and for a three-dimensional (3-D) objective function, the search will be over a tetrahedron, and so on. The positive constant ϵ_n is chosen after each iteration and depends on the number of function evaluations remaining in the search.

D. Torn's Method:

This method is considered a Clustering technique of global optimization. Torn's method chooses L initial points. A local optimizer is used to search for a local minimum at each of the L initial starting points. The particular local minimizer uses both a random and linear search. First, a starting direction is randomly chosen, based upon the initial point. A line is projected from the original starting point through the randomly chosen point. The search looks for better points down the line segment. If a better point is found, the search continues with the next step being twice the distance of the previous step, otherwise, a new random direction is picked and the search is continued. However, the new search direction will have half the original step size. The local search continues until the original step size is reduced to a value less than some set tolerance [7].

Clusters are picked from the local optimization search according to a particular point density requirement. A sample is picked from each cluster and the entire process is repeated.

E. Monte Carlo Method:

The Monte Carlo method of global optimization is strictly a uniform random search technique. Points are chosen randomly within the desired search boundaries. This method was implemented using the uniform random number generator in Turbo Pascal Version 4.0 [19]. This method will converge only in a probabilistic sense [7].

F. Simulated Annealing Method:

There are many different simulated annealing algorithms for various purposes ([12] provides an overview of the various methods and their applications). The method chosen as a basis of comparison is the particular algorithm that utilizes the Boltzmann distribution, a fixed search radius and a logarithmic annealing schedule. This method can be explained in two parts,

1) A random starting point is chosen by the user. Next, a uniform random point is selected from a square centered about the initial starting point. Note, the fixed radius" of the square (the length of any side) r is a free parameter to be chosen by the user. If the point yields a function evaluation, $f(X)$, which is less than the current minimum function value, $f_{\min}$ (which is initially picked to be $+\infty$), the process is repeated, otherwise, step 2 is performed.

2) If the new function evaluation, $f(X)$, is greater than the current function minimum, $f_{\min}$, then expression 4, the annealing expression, is used as the decision function to determine if the new point will be accepted.

$$rn \leq \exp[k(t) * (f_{\min} - f)] \tag{4}$$

where rn = a uniform random variable $[0,1]$ and $k(t) \equiv$ annealing variable. If expression 4 is satisfied, the new point is accepted and step one is repeated, otherwise, the old point is kept and step one is repeated. The annealing variable $k(t)$ is inversely proportional to the "temperature,"$T(t)$, where t is the iteration number—representative of time in an annealing process. The variation of $k(t)$ (or its inverse, the temperature) as the search progresses is called the "annealing" or "cooling" schedule. Many possible annealing schedules have been investigated [12]. Examples include:

1) *Constant Annealing:* $T(t) = C$, (see for example [20]),
2) *Arithmetic Annealing:* $T(t) = T(t-1) - C$, (see for example [21], [22]),
3) *Geometric Annealing:* $T(t) = \alpha(t)T(t-1)$, (see for example [23], [24]),
4) *Inverse Annealing:* $T(t) = C/(1+t\delta)$, (see for example [25]),
5) *Logarithmic Annealing:* $T(t) = C/(\ln(1+t))$, (see for example [26], [27]).

For the purposes of this work, an logarithmic annealing schedule was chosen with the following modification:

$$k(t) = \frac{(k_2 - k_1)}{\ln(N)} \ln(t) + k_1.$$

where N is the number of iterations in the search. On iteration 1 ($t = 1$), the annealing constant is k_1 and on the last iteration ($t = n$), the annealing constant is k_2. Notice that this has the same effect as annealing schedule in [26], [27]. However, for the sake of a fair comparison, it will "anneal" the search a constant amount for any number of iterations, N, the point of termination of the search. After evaluation of several sets of user-defined parameters, the radius was chosen to be 0.15, and the initial and final annealing constants were chosen to be $k_1 = 0.01$ and $k_2 = 10.0$ for the purposes of comparison.

TABLE I
CONVERGENCE CONDITIONS AND USER DEFINED INPUTS FOR VARIOUS METHODS OF GLOBAL OPTIMIZATION

Method	Type	Proof of Convergence	User Defined Input Parameters
Stuckman	Bayesian/ Sampling	No	Bounds on search space, number of function evaluations.
Mockus	Bayesian/ Sampling	Yes, continuous functions	Bounds on search space, number of function evaluations, number of initial points.
Perttunen	Bayesian/ Sampling	No	Bounds on search space, number of function evaluations.
Torn	Clustering	No	Bounds on search space, number of function evaluations, number of initial points.
Monte Carlo	Uniform Random	Yes	Bounds on search space, number of function evaluations.
Simulated Annealing	Sequential Random	No	Starting point, radius of next search point, initial and final annealing constant and number of function evaluations.
Zilinskas	Bayesian/ Sampling	Yes	Bounds on search space, number of function evaluations, number of initial points.
Shaltenis	Bayesian/ Sampling	Yes	Bounds on search space, number of function evaluations, number of initial points.

G. Zilinskas's Method

Similar to Mockus's technique, Zilinskas's Bayesian/ Sampling method provides the minimal average deviation, but makes five assumptions about the choice of the conditional mean as outlined by Mockus [7]. This method is an adaptive version of Mockus's one-step approach and is based on expression 5,

$$x_{n+1} \in \arg\max_{x \in A} \min \sigma_x^i / (\mu_x^i - c) \tag{5}$$

where $1 \leq i \leq n$. The method uses a heuristic approach of choosing weights that determine the conditional mean and variance, such that the distribution function changes in accordance with new data.

H. Shaltenis's Method

Shaltenis's method is based on the Bayesian one—step method of Mockus except that rather than viewing the minimum of expression 3, as the point of the next observation, the technique uses it as a decision function to choose the best search strategy that minimizes the risk function.

There are $n+1$ search strategies available; one is the n-dimensional Monte Carlo technique discussed previously, the other n strategies correspond to a 1-D uniform random search along each coordinate [7]. The search strategy is picked according to the following two rules:

1) If the cost of evaluation is not expensive then use the Monte Carlo technique.
2) If the interdependence of variables is nonexistent, then use the 1-D uniform random search along a coordinate.

Table I summarizes the ability of a particular method to converge and lists the user defined inputs required for each of the methods of global optimization.

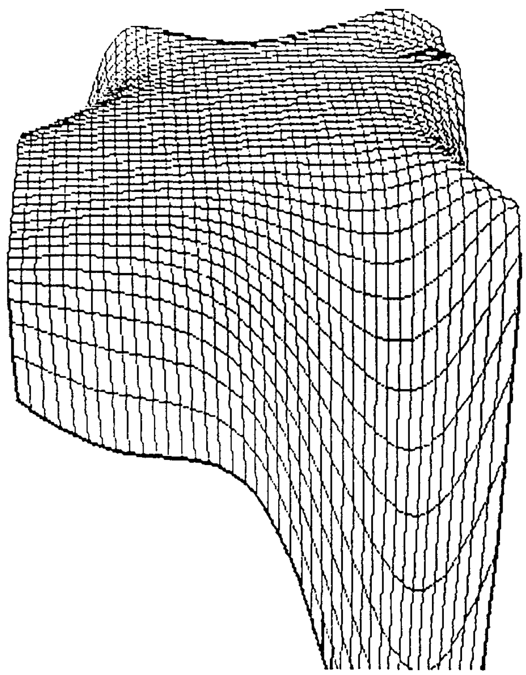


Fig. 1. The Goldstein-Price function.

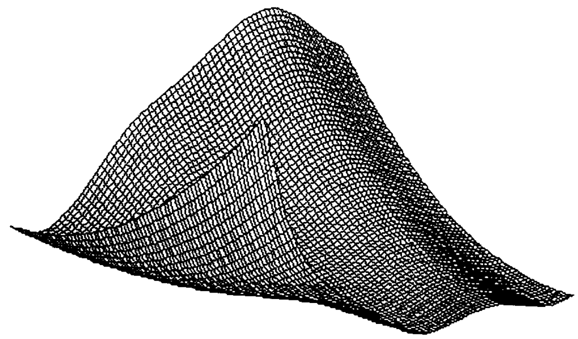


Fig. 2. Branin's RCOS function.

TABLE II
DATA FOR SHEKEL SQRN5, SHEKEL SQRN7 AND SQRN10 TEST FUNCTIONS

i	A_i				c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.6
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

III. TEST FUNCTIONS USED FOR COMPARISON

A. Introduction:

The following section contains a list of test functions that will be used as a basis of comparing different global optimization methods. The first five are a standard set of differentiable test functions compiled by Dixon and Szego [5] in their survey of global optimization techniques.

B. Test Functions:

The first test function to be considered is the Goldstein–Price test function, which is

$$\begin{aligned} f(x_1,x_2) =& [1+(x_1+x_2+1)^2 \\ & * (19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)] \\ & * [30+(2x_1-3x_2)^2 \\ & * (18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)] \end{aligned} \tag{6}$$

where $-2 \le x_i \le 2, x_i \in [x_1, x_2]$.

The function has a global minimum value of $f_{\min} = 3$ at the point $(x_1, x_2) = (0, -1)$. Fig. 1 shows a graph of the Goldstein–Price test function over the region of interest. The second test function used in the survey is the Branin RCOS function. The function is

$$f(x_1,x_2) = a(x_2-bx_1^2+cx_1-d)^2+e(1-f)\cos(x_1)+e \tag{7}$$

given the boundaries $-5 \le x_1 \le 10$ and $0 \le x_2 \le 15$ where $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $e = 10$, $f = 1/(8\pi)$.

The function has three global minima at the points $(x_1, x_2) = (-\pi, 12.275)$, $(\pi, 2.275)$ and $(9.42478, 2.475)$ with $f_{\min} = 0.397887$. Fig. 2 shows a graph of the Branin RCOS test function over the region of interest.

The next three test functions, Shekel SQRN5, Shekel SQRN7 and Shekel SQRN10 are a family of four-dimensional (4-D) functions. They are described as

$$f(X) = \sum_{i=1}^{m} \frac{1}{[(X-A_i)^T(X-A_i)+c_i]} \tag{8}$$

given the bounds: $0 \le X \le 10, X \in (x_1,x_2,x_3,x_4)$ and the A_i's and c_i's are given in Table II. For Shekel SQRN5, SQRN7 and SQRN10 test functions, $m = 5, 7, 10$ respectively.

The global minimum occurs at the point $(x_1,x_2,x_3,x_4) = (4,4,4,4)$ for all three Shekel functions, with the minimum function values for each as follows:

Shekel SQRN5: $f_{\min} = -10.15320$ Shekel SQRN7: $f_{\min} = -10.40282$ Shekel SQRN10: $f_{\min} = -10.53628$

The next test function to be considered is the 2-D six-hump camel back function. The function is shown in (9),

$$f(x_1,x_2) = [4-2.1x_1^2+\frac{x_1^4}{3}]x_1^2+x_1x_2+[-4+4x_2^2]x_2^2 \tag{9}$$

where $-3 \le x_1 \le 3$ and $-2 \le x_2 \le 2$.

Within the bounded region are six local minimum, with two global minimum located at the points $(x_1, x_2) = (-0.0898, 0.7126)$ and $(0.0898, -0.7126)$ where $f_{\min} = -1.0316$. Fig. 3 shows a graph of the six—hump camel back test function over the region of interest.

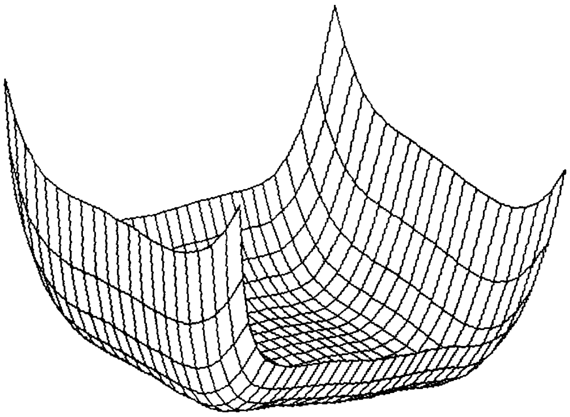


Fig. 3. The six-hump camelback function.

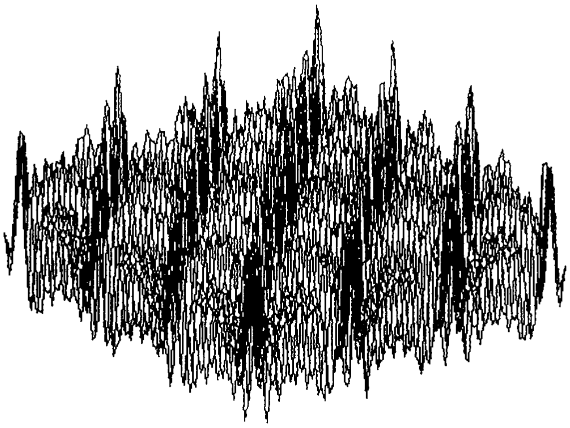


Fig. 4. Shubert's function.

The next test function to be considered is the 2-D Shubert function:

$$f(x_1, x_2) = \sum_{i=1}^{5} i\cos[(i+1)x_1 + i] * \sum_{i=1}^{5} icos[(i+1)x_2 + i] \tag{10}$$

where $-10 \leq x_1 \leq 10$ and $-10 \leq x_2 \leq 10$. The 2-D Shubert function has 760 local minima, 18 of which are global minima where $f_{\min} = 186.73$. Fig. 4 shows a graph of the 2-D Shubert test function over the region of interest.

Rather than testing a single test function for the global minimum, Stuckman introduces the randomly generated test function for which a large set of functions can be produced. One hundred randomly generated test functions were reported by Stuckman [6], and will be the set used in this survey. The form of each function is

$$f(X) = \begin{cases} \rfloor(\rfloor m_1 \lfloor + 1/2) * [\sin(a_1)/a_1]\lfloor, 0 \leq x_1 \leq b \\ \rfloor(\rfloor m_2 \lfloor + 1/2) * [\sin(a_2)/a_2]\lfloor, b \leq x_2 \leq 10 \end{cases} \tag{11}$$

given the bounds: $0 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 10$ where

$$a_i = \rfloor(x_1 - xr_{1i})\lfloor + \rfloor(x_2 - xr_{2i})\lfloor$$

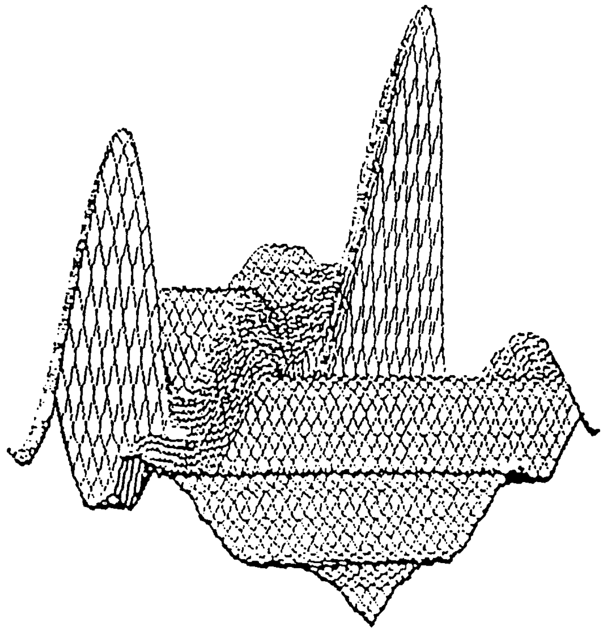


Fig. 5. A sample Stuckman's function.

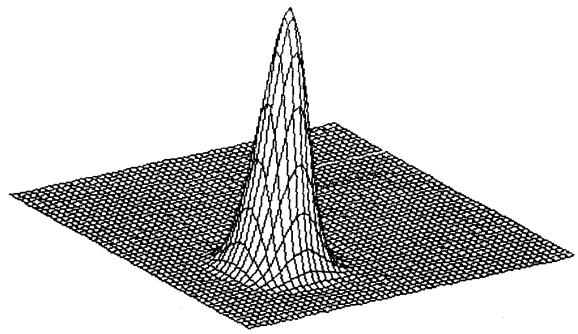


Fig. 6. Easom's function.

where b is a uniform random variable between 0 and 10 m_j is a uniform random variable between 0 and 100 xr_{11} is a uniform random variable between 0 and b xr_{12} is a uniform random variable between b and 10 xr_{21} is a uniform random variable between 0 and 10 xr_{22} is a uniform random variable between 0 and 10.

The global minimum is located at

$$(x_1, x_2) = \begin{cases} (xr_{11}, xr_{21}), & \text{if } m_1 > m_2 \\ (xr_{12}, xr_{22}), & \text{if } m_2 > m_1. \end{cases}$$

It should be noted that the due to the use of the floor functions ($\rfloor\ \lfloor$), the gradient of the function is either zero or does not exist for all points in the search space. Fig. 5 shows a graph of a typical Stuckman test function [18].

When considering real-world applications, a global optimization technique should be capable of finding a minimum whose area is small in comparison to the search space, because for most problems, little or no *a priori* information is known. In

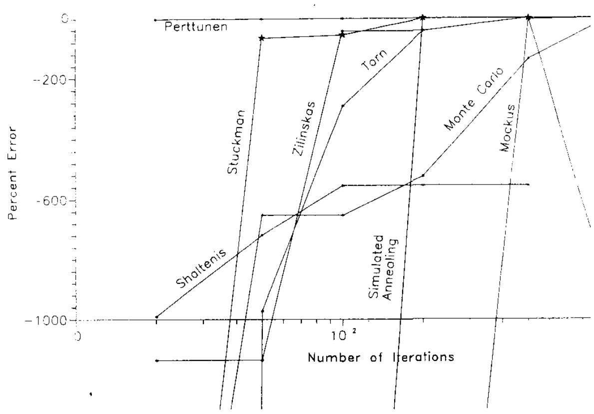


Fig. 7. Convergence on the Goldstein-Price function.

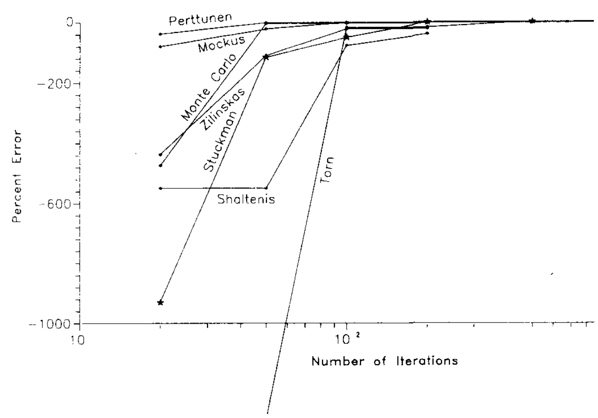


Fig. 8. Convergence on Branin's RCOS function.

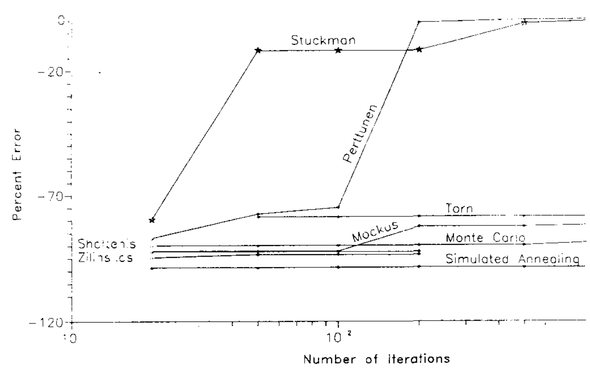


Fig. 9. Convergence on Shekel's SQRN5 function.

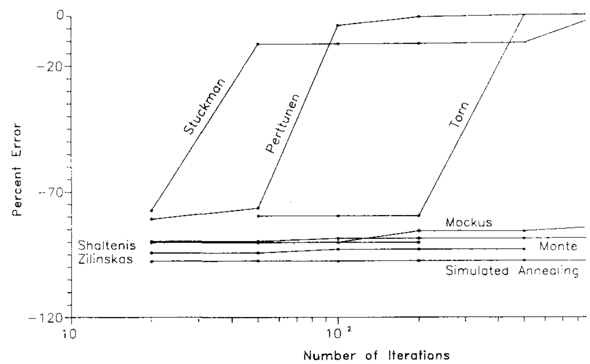


Fig. 10. Convergence on Shekel's SQRN7 function.

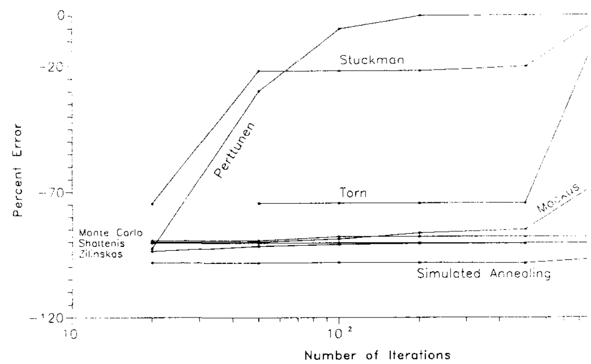


Fig. 11. Convergence on Shekel's SQRN10 function.

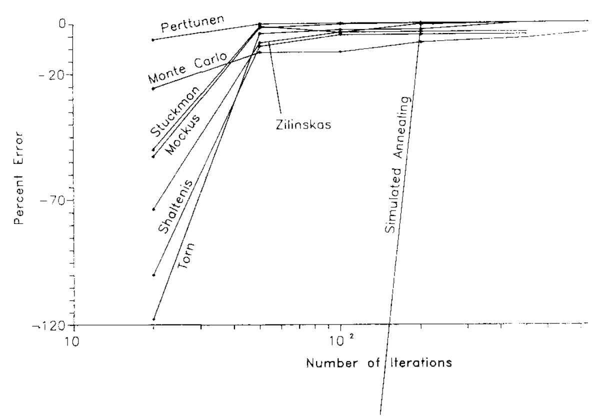


Fig. 12. Convergence on the six-hump camelback function.

order to explore the possibility of having a problem where the global minimum has a small area relative to the search space, the test function shown in (12) was developed by Easom [28]. Fig. 6 shows a 3-D view of the new test function, with the bounded region reduced to $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5$.

$$f(x_1, x_2) = \cos(x_1) * \cos(x_2) * \exp -[(x_1 - \pi)^2 + (x_2 - \pi)^2] \quad (12)$$

where $-100 \leq x_i \leq 100$. The global minimum was found to be $f(x_1, x_2)_{\min} = -1$, where $x_1 = x_2 = \pi$.

Results were obtained by running each global optimization technique on the various test functions. Due to the extreme amount of CPU time required (>1 h on a VAX 8650), results were not obtained for Zilinskas' method and Shaltenis' method for 1000 iterations. In addition, Zilinskas' technique yielded a run-time error (division by zero) when used on the Stuckman test functions because of a programming limitation in the implementation of the method. After some additional

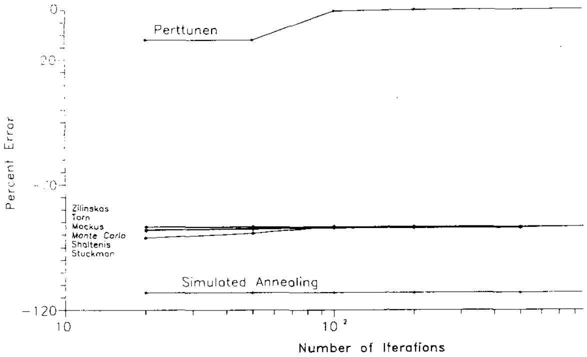


Fig. 13. Convergence on the 2-D Shubert function.

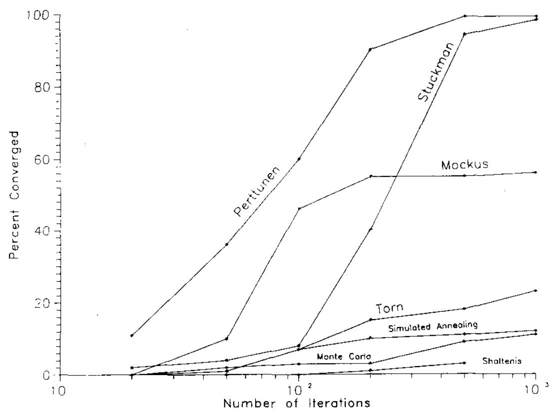


Fig. 14. Percentage convergence on the 100 Stuckman functions.

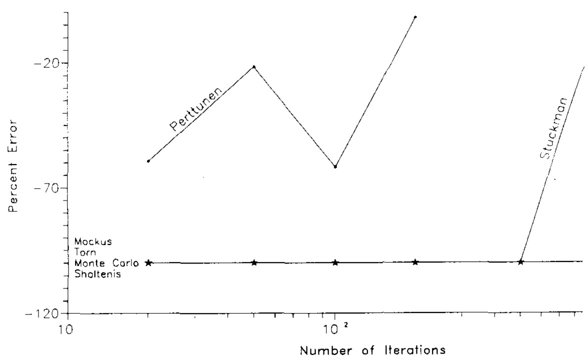


Fig. 15. Convergence on the Easom function.

TABLE III
EXPERIMENTAL RATES OF CONVERGENCE ON THE GOLDSTEIN–PRICE FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	3127	65.67	56.30	1.02	1.12	1.12
Mockus	3259	3259	3259	3259	4.90	0.39
Perttunen	2.26	1.44	0.01	0.00	0.00	0.00
Torn	*	971	290	43.50	0.00	0.00
Monte Carlo	2804	651	651	523	135	2.90
Sim. Annealing	*	5890	3530	0.04	0.60	900
Zilinskas	1134	1134	42.70	42.70	42.70	**
Shaltenis	990	718	554	554	554	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

TABLE IV
EXPERIMENTAL RATES OF CONVERGENCE ON THE BRANIN'S RCOS FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	930	116	50.04	1.00	0.00	0.00
Mockus	0.10	21.80	0.73	0.19	0.19	0.19
Perttunen	38.13	0.57	0.01	0.00	0.00	0.00
Torn	*	1316	17.96	17.96	0.00	0.00
Monte Carlo	473	3.84	3.84	3.84	3.84	3.84
Sim. Annealing	*	*	*	3315	1.77	0.06
Zilinskas	437	110	22.05	22.05	**	**
Shaltenis	549	549	76.86	39.78	39.78	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

TABLE V
EXPERIMENTAL RATES OF CONVERGENCE ON THE SHEKEL'S SQRN5 FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	79.64	11.95	12.03	12.03	1.52	0.40
Mockus	92.02	92.02	92.02	82.33	82.33	81.75
Perttunen	86.96	77.34	74.79	0.97	0.01	0.00
Torn	*	78.48	78.48	78.48	78.48	78.48
Monte Carlo	89.85	89.95	89.74	89.74	89.74	88.46
Sim. Annealing	98.59	98.66	98.48	98.39	98.44	98.56
Zilinskas	94.57	93.41	93.41	93.41	93.41	**
Shaltenis	92.20	92.20	92.20	92.20	92.20	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

experimentation, it was found that this error can occur when the method is used on a function that has a plateau," an area of constant value.

The percent error for each method for various number of function evaluations and the various test functions are presented in Figs. 7–15 and Tables III–XI.

IV. CONCLUSION

The main objective of this research was to conduct a comparison of Bayesian/sampling global optimization techniques for functions of continuous variables. From the results of each method on a series of test functions, it appears that the these methods consistently out-perform other commonly used methods of global optimization, specifically the Monte Carlo and simulated annealing methods. These methods would certainly perform well on a CPU time expended basis due to their simplicity, yet, they appear consistently at or near the bottom of the rate of convergence comparisons.

Stuckman's method and simulated annealing did not converge monotonically in all cases since both methods adapted their search strategy to the number of function evaluations. However, Stuckman's method, Perttunen's method and the Monte Carlo method were the only techniques that did not require a user-defined input such as the number of initial points or some other constant that effects the search procedure.

TABLE VI
EXPERIMENTAL RATES OF CONVERGENCE ON THE SHEKEL'S SQRN7 FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	77.56	11.27	11.36	11.29	10.93	0.00
Mockus	90.21	90.21	90.21	85.66	85.66	84.00
Perttunen	80.93	76.42	3.98	0.66	0.01	80.00
Torn	*	79.68	79.68	0.00	80.00	79.68
Monte Carlo	89.75	89.75	88.60	88.60	88.60	88.49
Sim. Annealing	97.78	97.51	97.71	97.53	97.40	97.73
Zilinskas	94.34	94.34	92.98	92.98	92.98	**
Shaltenis	90.21	90.21	90.21	90.21	90.21	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

TABLE VII
EXPERIMENTAL RATES OF CONVERGENCE ON THE SHEKEL'S SQRN10 FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	74.64	21.80	21.91	21.85	20.17	0.00
Mockus	90.02	90.02	88.69	86.30	84.88	64.80
Perttunen	92.57	29.81	5.21	0.10	0.01	0.00
Torn	74.39	74.39	74.39	74.39	74.39	0.00
Monte Carlo	89.43	89.43	87.69	87.69	87.69	87.69
Sim. Annealing	98.30	98.42	98.15	98.298	98.44	96.53
Zilinskas	93.70	91.66	90.93	90.58	90.58	**
Shaltenis	90.33	90.33	90.33	90.33	90.33	90.33

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

TABLE VIII
EXPERIMENTAL RATES OF CONVERGENCE ON THE SIX-HUMP CAMELBACK FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	50.16	1.07	3.86	0.44	0.00	0.00
Mockus	53.01	1.64	0.42	0.15	0.00	0.00
Perttunen	6.48	0.04	0.00	0.00	0.00	0.00
Torn	117.57	4.01	2.67	2.67	0.00	0.00
Monte Carlo	25.78	11.45	11.45	7.78	6.12	3.41
Sim. Annealing	1235	871	304	0.71	0.11	0.01
Zilinskas	73.83	7.67	3.65	3.65	3.65	**
Shaltenis	100.00	9.02	4.72	4.72	4.72	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

TABLE IX
EXPERIMENTAL RATES OF CONVERGENCE ON THE SHUBERT FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	91.22	89.32	86.62	86.61	86.63	86.61
Mockus	87.97	87.70	87.15	87.15	86.65	86.65
Perttunen	12.01	12.01	0.39	0.01	0.00	0.00
Torn	88.32	87.37	87.01	87.01	87.01	86.61
Monte Carlo	86.88	86.88	86.88	86.88	86.83	86.66
Sim. Annealing	113	113	113	113	113	113
Zilinskas	86.67	86.67	86.67	86.67	86.67	**
Shaltenis	88.01	87.23	87.23	87.23	87.23	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

TABLE X
EXPERIMENTAL RATES OF CONVERGENCE ON STUCKMAN'S FUNCTIONS

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	2	4	8	40	94	98
Mockus	0	10	46	55	55	56
Perttunen	11	36	60	90	99	99
Torn	0	1	7	15	18	23
Monte Carlo	0	2	3	3	9	11
Sim. Annealing	0	1	7	10	11	12
Zilinskas	†	†	†	†	†	†
Shaltenis	0	0	0	1	3	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.
† Would not run.

TABLE XI
EXPERIMENTAL RATES OF CONVERGENCE ON THE EASOM FUNCTION

	% error/Number of Iterations					
Method	20	50	100	200	500	1000
Stuckman	100	100	100	100	100	0.00
Mockus	100	100	100	100	100	100
Perttunen	59.40	21.66	61.94	2.43	2.15	**
Torn	100	100	100	100	100	100
Monte Carlo	100	100	100	100	100	100
Sim. Annealing	100	100	100	100	100	100
Zilinskas	100	100	100	100	100	**
Shaltenis	100	100	100	100	100	**

* More than 10 000% error.
** Results not obtained due to extreme CPU requirements.

Perttunen's method, Zilinskas' method and Shaltenis method required an excessive amount of CPU time to run (in some cases exceeding 1 h on a VAX 8650). Even though this is not an explicit criteria for comparison, it might be important in some applications.

Perttunen's method appears to perform the best on the 2-D test functions. The other Bayesian/sampling methods and Torn's method also appear to give satisfactory results for 1000 iterations or less. Perttunen's method also seems to perform the best at finding a global solution that is localized within a small area of the overall search space as in Easom's function. Perttunen's method was also the only method to converge on the global solution for Shubert's function—a function with many local minima.

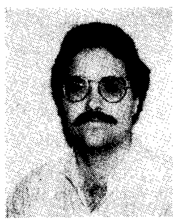
Perttunen's method also appears to perform the best on the 4-D test functions presented by Shekel. Stuckman's method and Torn's method also perform well on these functions, much better than the other methods that would require more than 1000 points for convergence.

Perttunen's, Stuckman's and Mockus's method seem to perform the best on the discrete test functions presented by Stuckman. Due to a programming limitation, Zilinskas's method would not perform at all on this set of test functions, and should be avoided in applications where the function to be optimized has an area of constant value.

REFERENCES

[1] B. E. Stuckman and N. W. Laursen, "Modern control design using global

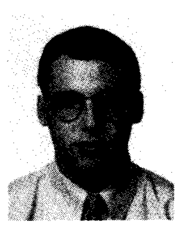
search," in *Proc. 8th Meeting Coordinating Group on Modern Control Theory*, U.S. Army, Feb. 1987
[2] B. E. Stuckman and J. C. Hill, "Pattern steering in HF phased array antennas," in *Proc. 29th Midwest Symp. Circuits and Syst.*, Aug. 1986
[3] A. Groch, L. M. Vidigal, and S. W. Director,A new global optimization method for electronic circuit design," *IEEE Trans. Circuits and Syst.*, vol. CAS-32, no. 2, Feb. 1985.
[4] J. J. McKeown and A. Nag, "An application of optimization techniques to the design of an optical filter," presented at the IMA Conf., Bristol, England, 1974.
[5] L. C. W. Dixon and G. P. Szego, "The optimization problem: An introduction," *Toward Global Optimization II*, L. C. W. Dixon and G. P. Szego, Eds., New York: North Holland, 1978.
[6] B. E. Stuckman, "A search method for optimizing nonlinear systems," *IEEE Trans. Syst. Man Cybern.*, vol. 18, no. 6, pp. 965–977, Dec. 1988.
[7] J. B. Mockus, *Bayesian Approach to Global Optimization.* New York: Kluwer Academic, 1989.
[8] C. D. Perttunen, "Global optimization using nonparametric statistics," Ph.D. dissertation, the Graduate School, Univ. Louisville, Louisville, KY, Dec. 1990.
[9] A. Zilinskas, The use of statistical models for construction of multimodal optimization algorithms," in *Proc. Third Czechoslovak-Soviet-Hungarian Seminar Inform. Theory*, Czechoslovak Acad. Sci., Prague, 1980, pp. 219–224.
[10] V. R. Shaltenis and G. Dzemyda, "The structure analysis of extremal problems using some approximation of characteristics," in *Optimal Decision Theory*, Inst. of Math. and Cybernetics, Vilnius, Lithuania, vol. 8, pp. 115–123, 1982.
[11] A. A. Torn, "A search-clustering approach to global optimization," *Toward Global Optimization II*, L. C. W. Dixon and G. P. Szego, Eds. Amsterdam: North-Holland, 1978.
[12] N. E. Collins, R. W. Eglese and B. L. Golden, "Simulated annealing bibliography," *Amer. J. Math. Management Sci.*, vol. 8, nos. 3 and 4, pp. 209–307.
[13] A. A. Goldstein and I. F. Price, "On descent from local minima," Math. Comput., vol. 25, no. 115, 1971.
[14] F. K. Branin, "A widely convergent method for finding multiple solutions of simultaneous nonlinear equations," *IBM J. Res. Develop.*, pp. 504–522, Sept. 1972.
[15] L. C. W. Dixon and G. P. Szego, "The optimization problem: An introduction," in *Toward Global Optimization II*, L. C. W. Dixon and G. P. Szego, Eds. Amsterdam: North-Holland, 1978.
[16] J. Hardy, "An implemented extension of Branin's method," in *Toward Global Optimization*, L. C. W. Dixon and G. P. Szego, Eds. Amsterdam: North-Holland, 1975.
[17] B. O. Shubert, "A sequential method seeking the global maximum of a function," *SIAM J. Numer. Anal.*, vol. 9, pp. 379–388, 1972.
[18] B. E. Stuckman, "A new multipeak test function which is not everywhere differentiable," in *Proc. 30th Midwest Symp. Circuits and Syst.*, New York, pp. 1383–1386, Aug. 1987.
[19] Turbo Pascal, *Owner's Handbook*, Version 4.0, Borland International, 1987, p. 462.
[20] J. Bernasconi, "Low autocorrelation binary sequences: Statistical mechanics and configuration space analysis," *J. Physique*, vol. 48, no. 4, pp. 559–567, 1987.
[21] P. Molitor, "Layer assignment by simulated annealing," *Microporcessing and Microprogramming*, vol. 16, nos. 4–5, pp. 345–349, 1985.
[22] S. Rees and R. C. Ball, "Criteria for an optimal simulated annealing scedule for problems of the travelling salesman type," *J. Physics*, vol. 20, no. 5, pp. 1239–1249, 1987.
[23] T. Boseniuk, W. Ebeling, and A. Engel, J. Boltzmann and R. Darwin "Strategies in Complex optimization," *Phys. Lett.*, vol. 125, no. 6–7, pp. 307–310, 1987.
[24] R. E. Burkard and F. Rendl, "A thermodynamically motivated simulation procedure for combinatorial optimization problems," *Eur. J. Operational Res.*, vol. 17, no. 2, pp. 169–174, 1984.
[25] M. Lundy and A. Mees, "Convergence of an annealing algorithm," *Mathematical Programming*, vol. 34, pp. 111–124.
[26] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operational Research*, vol. 13, no. 2, pp. 311–329, 1988.
[27] J. B. Lasserre, P. P. Varaiya, and J. Walrand, "Simulated annealing, random search," *Multistart or SAD, Systems & Control Lett.*, vol. 8, no. 4, pp. 297–301, 1987.
[28] E. E. Easom, "A survey of global optimization techniques," M. Eng. thesis, Univ. Louisville, Louisville, KY, 1990.



Bruce E. Stuckman (S'80–M'81–SM'90) received the B.S. degree in engineering in 1981, the M.S. degree in electrical and computer engineering in 1983, the Ph.D. degree in systems engineering in 1987 from Oakland University, Rochester, MI, and the J.D. degree in law in 1992 from the University of Louisville, Louisville, KY, .

He is an Associate Professor of Electrical Engineering at the University of Louisville, Director of Electrical and Systems Engineering for the OASYS Research Group, a private consulting company, an an associate in the patent and computer law department of the law office of Brooks and Kushman, Southfield, MI. He has received the University of Louisville's President's Young Award. His research interests include numerical optimization, simulation and modeling, telecommunication and communication systems, instrumentation and measurement error, and patent and computer law.

Dr. Stuckman is a member of Tau Beta Pi, Eta Kappa Nu, Sigma Xi, and ASNE.



Eric E. Easom received the B.S. degree in 1989 and the M.Eng. degree in electrical engineering in 1990 from the University of Louisville, Louisville, KY.

He is currently a Process Control Engineer in the Corporate Process Automation Department of Eli Lilly and Company, Indianapolis, IN. His research interests include process control, instrumentation, measurement, and numerical optimization.