Journal of Global Optimization **8:** 413–427, 1996.

Global Optimization with a Limited Solution Time

ELIGIUS M. T. HENDRIX and JAAP ROOSMA
Department of Mathematics, Agricultural University Wageningen Dreijenlaan 4, 6703 HA Wageningen, The Netherlands

(Received: 18 November 1994; accepted: 9 November 1995)

Abstract. In this paper the box constrained global optimization problem in presence of a limited solution time is considered. A method is studied based on a combination of multistart and singlestart which implies a decision sequence on the number of random points to be generated. Search strategies are numerically illustrated. Criteria are introduced to measure the performance of solution methods for the problem class. Moreover, the performance of search strategies, specifically the efficiency of generating random points is analyzed.

Key words: Global optimization, random methods, heuristic solution strategy, local search, limited solution time.

1. Introduction

The box constrained global optimization problem, which is to find the global optimum of a real valued, in general multimodal objective function over a hyperrectangle $S \subset R^n$, has been studied by many researchers. An efficient and often applied approach is to perform a local search from points derived from a random sample from a probability distribution over S. Attention has been paid to the derivation of stopping rules to determine the sample size. In many practical situations the function evaluations necessary for the optimization can be rather time consuming, e.g. it may need minutes, as a special subroutine or program has to be run to determine the function value. In this case it is not uncommon that with a given amount of calculation time, e.g. a night or a weekend, one wants to reach a point in S with a function value as low as possible. The limited time to find a good solution can also be found in cases such as power station decisions, where the decision time is limited. In this paper we introduce for this case the term *b*ox *c*onstrained global optimization problem with a given *b*udget of function evaluations, BCB problem for short. In Section 3 the multi singlestart method is presented and possible strategies within this method are discussed. In Section 4 criteria are introduced to measure the performance of solution methods for the problem class. The criteria are numerically illustrated for various methods and instances of the BCB problem. In Section 5 an analysis can be found on the performance of random search methods for the BCB problem. This is followed by a discussion of the results and conclusions in Section 6.

2. The Box Constrained Global Optimization Problem with a Limited Solution Time

In many applications in engineering one wants to find the minimum of a real valued function *f* over a region *S* defined by lower and upper bounds on the variables. (*S* can be called a closed hyperrectangle or a box.)

$$\min f(x) \qquad \frac{(BCP)}{x \in S}.$$

It will be assumed that *f*(*x*) is a real valued possibly multiextremal continuous oracle function for which no derivatives or other global information such as a Lipschitz constant or concavity properties are available. Evaluating the function is similar to presenting parameter values to a black-box, possibly implemented in a subprogram, which calculates the function as a criterion on the parameters. Zhigljavsky (1991) introduced a classification on global optimization problems based on the prior information on the problem. In this classification the BCP problem is classified as type a), it is only known that *f* is continuous.

In technical oriented literature, pragmatic approaches can be found to tackle the problem. See, e.g. Pronzato *et al.* (1984), Bohachevsky *et al.* (1986), Brazil and Krajevski (1987). In Mathematical Programming literature, analyses on various global optimization methods are presented. An overview on global optimization methods can be found in Törn and Zilinskas (1989). Most methods are based on the idea of globally exploring the feasible area (global search), e.g. by generating points in the feasible area, and performing local searches to arrive possibly (hopefully) at the global optimum. In this approach the following elements can be distinguished:

A. GENERATING RANDOM POINTS

If the target is to discover all optima, the purpose is to try to get starting points for the local search in every region of attraction. Following the idea that the region of attraction close to the global optimum contains the lowest function values, one can also only start a local search from the best point found during the global search. This idea will specifically be explored in this paper.

B. LOCAL SEARCH

The purpose is to get a local optimum from a "good" starting point.

C. CLUSTERING

By clustering, a part of the local minimizer structure can be discovered and one can save a number of function evaluations.

Focus of this paper is on the practical case, where there exists a budget for the computer time. Given this budget the search method should result in a function value (and corresponding point in S) as low as possible. It is assumed that the budget in computer time translates directly to a budget B on the number of function evaluations during the search process. This problem will be called the box constrained global optimization problem with a given budget on function evaluations, the BCB problem. We concentrate on the combination of generating random points (global search) and local search, because these elements exist in the core of many methods. The solution method for this problem allocates budgets to local searches and to generating random points. A framework of this allocation is presented in the next section.

3. Multi Singlestart

In the literature on random search based methods the following approaches have been analyzed.

–Pure Random Search (PRS) (see, e.g. Zabinsky and Smith, 1992): Generate a number N of random points from a probability distribution over S and evaluate them. The lowest point is an approximation of the global optimum.

```
PRS
  1. Generate and evaluate random points in $S$
  2. Determine the best value $y_r$ and incumbent minimizer $x_r$
```

Random search methods have been studied, among others, by Zabinsky and Smith (1992), Zhigljavsky (1991), Romeijn (1992) and Klepper and Hendrix (1994), focusing mostly on adaptation of the distribution function over S.

–Singlestart (SIS): Generate and evaluate random points over S and start one local search from the lowest point found.

```
SIS
  1. Generate and evaluate random points in $S$
  2. Determine the best value $y_r$ and incumbent minimizer $x_r$
  3. Start a local search with starting point $x_r$
```

–Multistart (MUS): At every iteration, a random point is generated in S as a starting point for a local search.

```
MUS
  Do for $t$=1 to $N$
  1. Generate a random point $x_t$ in $S$
  2. Start a local search with starting point $x_t$
```

In Boender and Rinnooy Kan (1987) and Betro and Schoen (1987) studies can be found on when to stop the multistart process given some criteria on the tradeoff between reliability and computational effort.

When we have a budget on the number of function evaluations, as in the BCB problem class, the application of the SIS method would consume as many function evaluations as possible for the global search, whereas the MUS method allocates the budget towards local searches. The success of both methods depends on the instance of the BCB problem, which *a priori* is unknown. For an instance of the BCB problem with a few local optima and sufficient budget to perform some local searches, it may happen that multistart proves to be the best strategy. For a problem with many local optima which are much different from the global optimum, it may be better to perform one local search from the best of a long list of random points, than to spend all "ammunition" on identifying local optima. The existence of many local optima may occur in practical cases due to numerical effects, e.g. when an evaluation involves the numerical integration (fitting of continuous models) or inversion of a matrix (optimal design of experiments).

The general message is that, if the surface of *f* is "rough", so that there are many local optima, then more effort should be put into global search, conversely if there are only a few optima, then more of the budget can be allocated to local searches. Due to the character of the BCB problem, this function structure is of course unknown when the search starts. During the search process more of the structure is revealed and the allocation of budget to local searches and global search can be adapted. Here we get to the idea of multi singlestart (MSIS), where the number of random points to be evaluated depends on the structure revealed during the search.

```
MSIS
0. $t$=1
1. Generate and evaluate $N_t$ random points on $S$.
2. Identify the best point $x_{rt}$ out of the $N_t$ points.
3. Perform a local search with starting point $x_{rt}$ .
4. If budget is left, $t$=$t$+1 and go to 1.
```

At every stage t in step 1, the number of random points N_t is chosen before the next local search is performed from the best of those points (if enough budget B_t is left). So more or less effort can be put into the global search.

The number of function evaluations F_t necessary to perform one local search does not only depend on tolerances, but also on the starting point and the function under consideration. Thus, F_t can be regarded as a random variable. During the search, estimates of (the expected value of) F_t become available. Note that F_t tends to decrease when N_t increases, due to the fact that part of the local search work is taken over by the random search.

As a variant of the decision parameter N_t we introduce the parameter K_t.

K_t: number of local searches (iterations) intended to be performed before the budget is exhausted.

If the part of the budget which is not used for local searches is equally divided over the intended iterations the number of random points N_t can be derived from :

$$N_t = (B_t - F_t - F_{t+1} - \cdots - F_{t+K})/K_t.$$

A uniform estimate F for the expected number of function evaluations for a local search reduces the formula to:

$$N_t = B_t/K_t - F.$$

The maximum number of local searches that can be performed is estimated by $K\mathrm{max}_t = [B_t/F]$. At every decision stage t, K_t is chosen between 1 and $K\mathrm{max}_t$. The SIS method corresponds to K_t=1 and MUS can be approximated by choosing K_t=$K\mathrm{max}_t$.

We now define a MSIS strategy as a choice rule to determine N_t (or K_t alternatively) at every iteration from the information generated by the previous iterations. At step 1 of the algorithm among others the following information is available:

B_t : budget left

$N\mathrm{loc}_t$: number of different local optima found

t-1 : number of local searches performed

F : expected (estimate) number of function evaluations necessary for one local search

$K\mathrm{max}_t$: $[B_t/F]$

Now, various strategies can be constructed. If we follow the general idea described above, then if many optima are found, K_t should be tending to 1, which means the pure SIS strategy. If the prior expectation is the existence of many local optima, the corresponding strategy is to have K_1=1. One problem is that there is no good estimate for F_t when no local searches have been performed. Another approach is to start with the hypothesis that there exists only one local optimum. At the first and second iteration a local search with one random starting point (N_1=N_2=1) can be performed. If the local optima found are equal, the hypothesis still holds and one can proceed with multistart in an attempt to check the hypothesis by discovering other local optima. If they are not equal, there are apparently multiple optima and, in an attempt to detect the region of attraction of the global optimum, N_t can be increased. Another possibility is to base the choice of N_t on an estimate of the number of undetected local optima, see, e.g. Boender and Rinnooy Kan (1987).

To illustrate the idea of MSIS strategies we introduce the following rule which will be called β heuristic (as depending on the parameter $\beta \geq 0$) :

$$K_t = K\mathrm{max}_t - \frac{(K\mathrm{max}_t - 1)(N\mathrm{loc}_t - 1)}{(N\mathrm{loc}_t - 1) + \beta \frac{t-1}{N\mathrm{loc}_t}}$$

The parameter β weights the relative number of different optima found with respect to the number of local searches performed, in the formula. Moreover, if β is very large, the rule approximates pure multistart ($K_t \rightarrow K\mathrm{max}_t$) and if β is taken to be zero the rule becomes the singlestart strategy.

After the introduction of the BCB problem and possible solution methods there is a need to establish performance criteria.

4. Performance Criteria for the BCB Problem

In the OR literature it is common to use the number of function evaluations used and the indicator whether the global optimum has been found, as a performance criterion for global optimization methods. For the BCB problem the objective is to reach a point as good as possible given budget B, hopefully it is the global optimum. So the best function value found is the criterion. The score on these classical criteria does not only depend on the local search method, tolerances and stopping criteria, but in random search techniques also on the random series used. To filter out this random effect, the expected values for those classical criteria are suggested in this paper to be applied as criteria for the BCB problem.

$PG(B)$	: the probability of a search method reaching the global optimum within budget B.
$ER(B)$	: expectation of the record value found with budget B.

The possibility to analyze the behaviour of algorithms with respect to the criteria is limited. In general, estimates for these criteria for various search strategies on test functions can be determined by Monte Carlo simulation. Only for multistart the probability of reaching the global optimum can also be approximated analytically by the following idea. Let D^* be the region of attraction of the global optimum and let ν be its relative size. Let F be the average number of function evaluations necessary to perform one local search (which is itself stochastic) and B the budget. The number of local searches that can be executed is $\lceil B/F \rceil$. This makes the probability of reaching the global minimum at least

$$PG(B) = 1 - (1 - \nu)^{\lceil B/F \rceil}. \tag{1}$$

The $PG(B)$ and $ER(B)$ criteria are illustrated here for various instances of the BCB problem. Monte Carlo simulations are done to estimate the score on the two criteria for the β-heuristic for some values of β and for the MUS and SIS strategy. For the β-heuristic we choose N_1=N_2=1. This implies that for low budgets the heuristic performs exactly the same as MUS. For SIS an estimate should be available for the number of function evaluations necessary for one local search.

Test functions for which the number of local optima varies can be found, e.g. in Törn and Zilinskas (1989). To see any difference, test functions with many optima are of interest. Therefore the Rastrigin function (50 optima), the Shekel functions (5, 7 or 10 optima) and the Goldstein–Price function are taken from this reference.

For the local optimization a variant of Powell's method (Powell, 1964), adapted for the box constraints, is used. In the linesearch initially small steps are taken in an attempt not to miss the nearest optimum. The stopping criterion is defined on the progress in function value. Moreover, after every linesearch it is checked whether the iterate is close to an optimum already found. The tolerance of being close to an

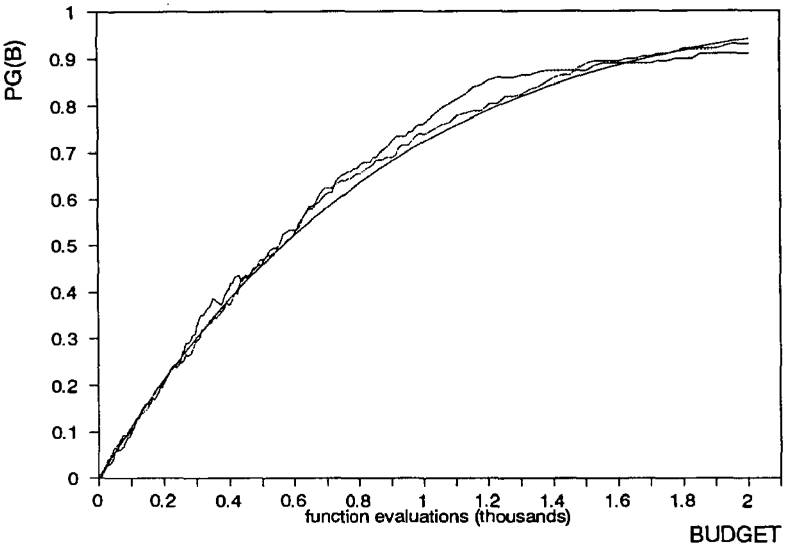

Fig. 1. Probability to reach the optimum for multistart.

optimum is taken as 1 percent of the componentwise range of the variables. This check speeds up the search process, and causes F_t to decrease when the iterations proceed.

The estimation of the criteria *PG*(*B*) and *ER*(*B*) is done by running the random search many times with various random series for fixed values of the budget B. For lower values of B, the fluctuation of the criteria is larger, does more depend on the random series, than for higher values of the budget where the probability *PG*(*B*) approaches 1. Therefore more replications were done (10,000) for small values of B than for large values of B (200).

As a numerical illustration we applied MUS to the Rastrigin test function (see Törn and Zilinskas 1989). The relative size of the region of attraction D^* for the local optimizer used was ν=0.0346. In Figure 1 the theoretical smooth curve of (1) is confronted with two curves that were found by Monte Carlo simulation. The performance of MUS, SIS and the β-heuristic have been estimated by Monte Carlo simulation for the Rastrigin test function. The results are given by Figures 2 and 3.

The results illustrate the idea of the criteria. Given an instance of the BCB problem, every solution method has its *PG*(*B*) and *ER*(*B*) curve. For the five solution methods (MSIS strategies) for which the curve has been approximated, the β-heuristic with a value of β = 10 performs the best for this test function. This illustrates how the criteria introduced can be used to judge on search strategies for the BCB problem: A particular method is better than another for a certain instance of the BCB problem, if its curve *PG*(*B*) is higher or its *ER*(*B*) curve is lower. In Figure 3 we see that from the five strategies, the β=10 rule has the best expected

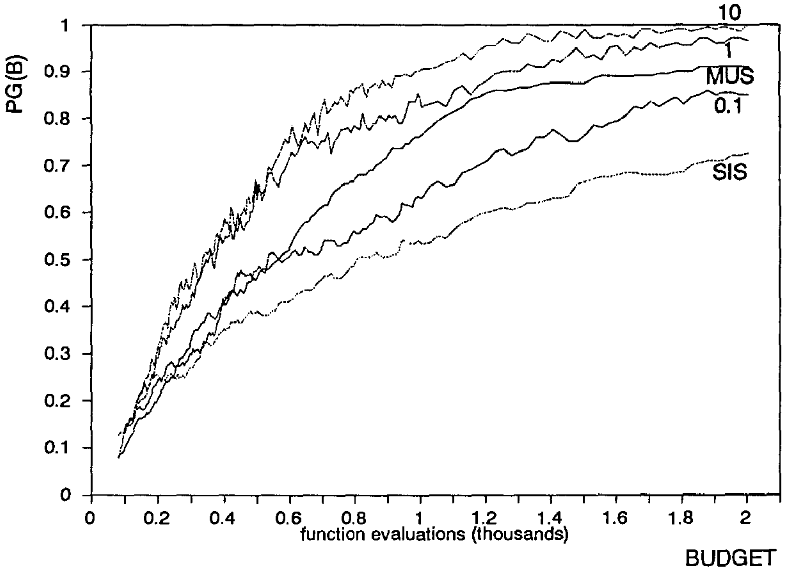

Fig. 2. Results of criterion *PG*(*B*) for the Rastrigin function for various strategies.

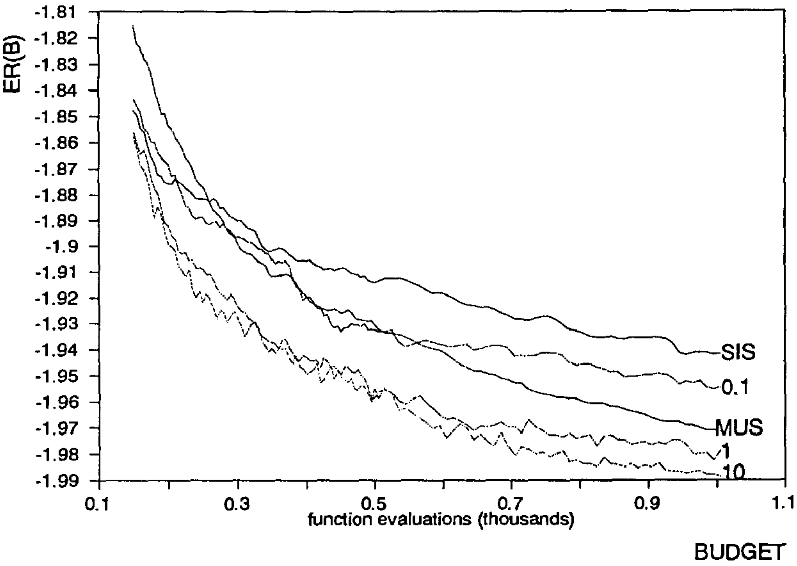

Fig. 3. Results of criterion *ER*(*B*) for the Rastrigin function for various strategies.

value for the best function value found. The global minimum has an objective function value of −2.

What determines the success of generating random points in the context of increasing the probability that the global optimum is detected? We first illustrate

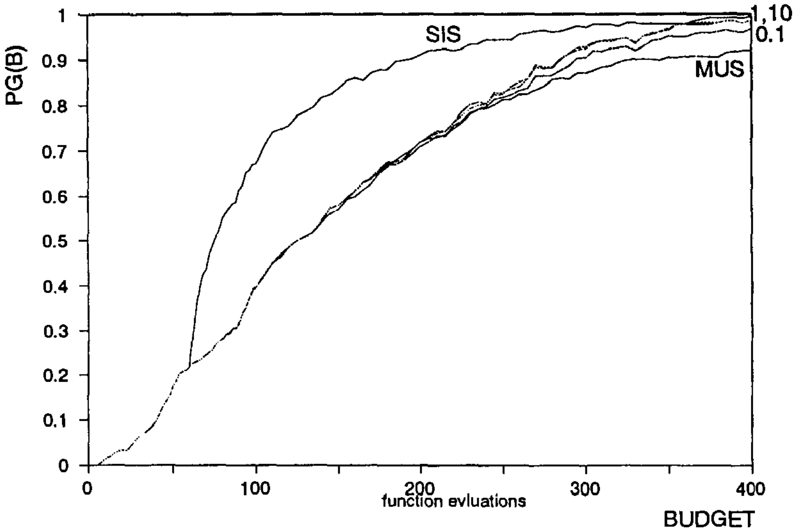

Fig. 4. Results of criterion *PG*(*B*) for the Goldstein–Price function for various strategies.

the difference in efficiency of generating points by two extreme numerical examples.

In Section 5 we will try to analyse this extreme difference. Figures 4 and 5 give the performance, according to the *PG*(*B*) criterion, of the five strategies for the Shekel-5 function and the Goldstein–Price function respectively. The results show that generating many random points is efficient for the Goldstein–Price function, whereas it apparently is not efficient for the Shekel-5 test function.

This illustration leads to the question whether there exists a MSIS strategy, or more generally a method, which performs better for all instances of the BCB problem for all values of the budget. To formalise this question we define the concept of dominating methods. A method is called PG-Dominating if for all instances of the BCB problem, *PG*(*B*) is higher than (or equals) *PG*(*B*) of all other methods for all values of budget *B*.

5. Analysis of Random Search Methods for the BCB Problem

Can a dominating method exist for the BCB problem? To answer this question we first analyze for which cases it is profitable to put more effort in global search than is done by multistart. As mentioned in Section 2, the objective of generating random points (or increasing N) is to increase the probability that the starting point of the local search is situated in D^*. We use the following notation. The relative size of a level set is defined as

$$\mu(y) = P\{f(x) \leq y \mid x \in S\}.$$

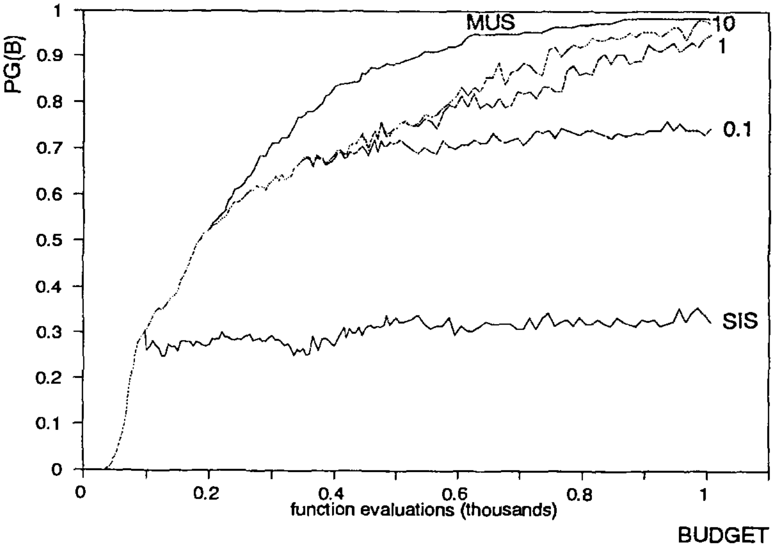

Fig. 5. Results of criterion *PG*(*B*) for the Shekel-5 function for various strategies.

When **x** is uniformly distributed over S, **y** $= f(\mathbf{x})$ is a random variable with cumulative density function $\mu(y)$ and probability density function $\mu'(y)$. By performing a random search with N points the probability density function of the record value $\mathbf{y}_r$ (lowest function value found) is

$$M'_N(y_r) = N\mu'(y_r)(1 - \mu(y_r))^{N-1}.$$

The success of a global search depends on the probability that the point x_r corresponding to y_r is in the right region of attraction, D^*. We define

$$\phi(y) = P\{x \in D^* \mid f(x) = y\}$$

as the probability that a point x is in the right region of attraction given that x is situated at a contour with height y. The efficiency of going deeper in the level sets by generating random points depends on the shape of $\phi(y)$. If one random starting point is used then the probability to reach the global optimum equals the relative size ν of $D*$:

$$\nu = \int_{y_*}^{y^*} \phi(y)\mathrm{d}\mu(y) \qquad (2)$$

in which $y_* = \min_{x\in S} f(x)$ and $y^* = \max_{x\in S} f(x)$.

By first generating N points and then starting a local search from the lowest of these points, the probability of reaching the global optimum is

$$PS_N = \int_{y_*}^{y^*} \phi(y)M'_N(y)\mathrm{d}y. \qquad (3)$$

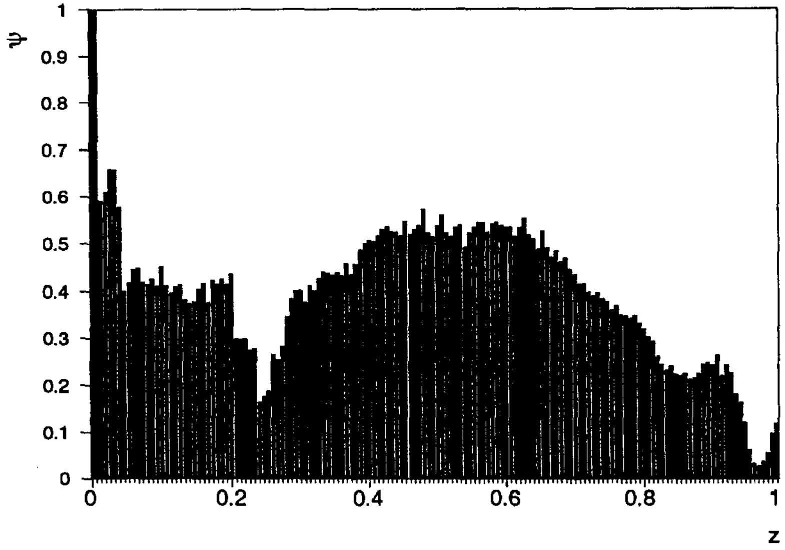

Fig. 6. Characteristic function for the Goldstein–Price problem.

Essential in the analysis is that PS_N may be worse (lower) than ν. This occurs when there is a wide relatively deep level set of which a large part does not belong to D^*. As an example from the standard test functions (see Törn and Zilinskas, 1989) the Shekel functions have this characteristic, as we have seen from Figure 5. This implies that SIS performs very bad versus MUS. The Goldstein–Price function gives the opposite result, see Figure 4.

The function $\phi(y)$ is apparently very different for those two examples. However $\mu(y)$ also differs for every problem. To make $\phi(y)$ more comparable, we introduce the following transformation. Let **z** be a uniformly distributed random variable defined as $\mathbf{z} = \mu(\mathbf{y})$. In other words $\mu^{-1}(z)$ defines the quantiles of **y**. Now equation (2) can be written as

$$\nu = \int_{y_*}^{y^*} \phi(y) \mathrm{d}\mu(y) = \int_0^1 \phi(\mu^{-1}(z)) \mathrm{d}z. \tag{4}$$

Every value of z is "as probable" . We will call the function $\psi(z) = \phi(\mu^{-1}(z))$ a characteristic function (not to be confused with the probabilistic meaning), as it contains all information to calculate (3) and consequently gives the exact information on the efficiency of generating random points. Note that $\psi(z)$ approaches 1 when z goes to 0. Equation (3) can be replaced by

$$PS_N = \int_0^1 \Psi(z) N (1-z)^{(N-1)} \mathrm{d}z. \tag{5}$$

The characteristic function $\Psi(z)$ determines the success of generating random points and contains much more information than, e.g. the number of optima. It should be

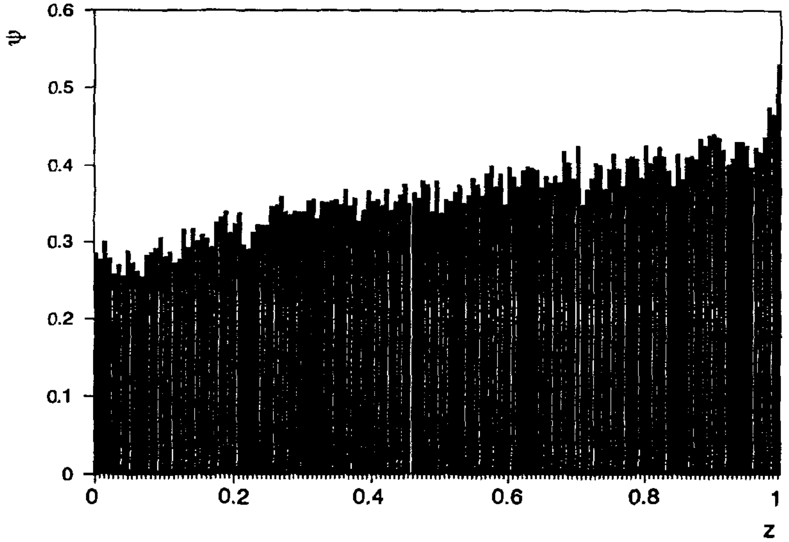

Fig. 7. Characteristic function for the Shekel-5 problem.

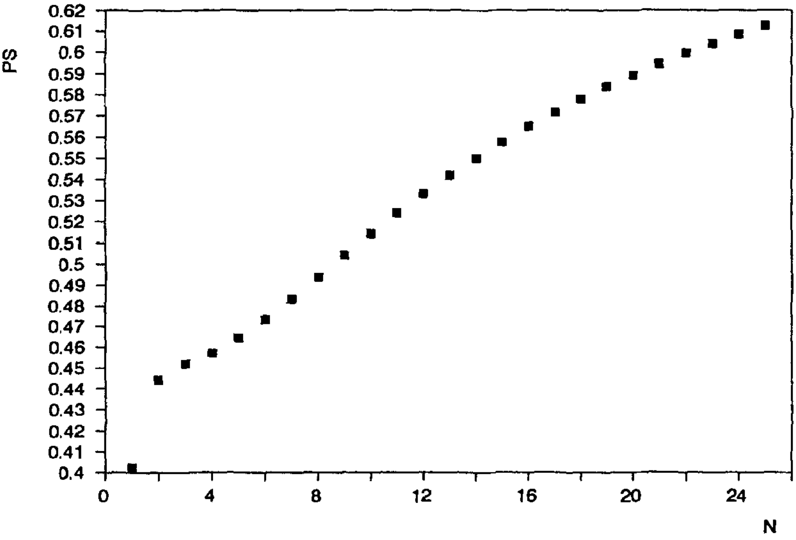

Fig. 8. Probability to reach the optimum depending on N for the Goldstein–Price problem.

mentioned that the information of $\Psi(z)$ is in general not available, so it cannot be applied in an algorithmic framework. It has been introduced for analytic reasons here. For illustrative purposes the function $\Psi(z)$ is approximated numerically by Monte Carlo simulation for the two example functions. In Figures 6 and 7 numerical approximations can be found.

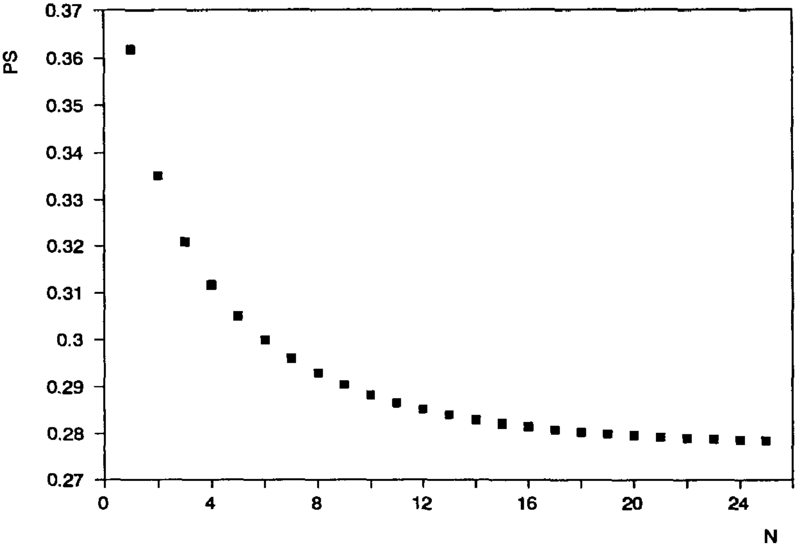

Fig. 9. Probability to reach the optimum depending on N for the Shekel-5 problem.

In limit $\Psi(z)$ approaches 1 when z goes to zero. However for the Shekel-5 function this limit is that distant that it cannot be observed in the numerical estimates. Increasing the number of random points leads the point x_r away from the right region of attraction D^*. For the Goldstein–Price function generating random points improves the probability that the global optimum is reached. This shows that given the information which becomes available during the search, it is impossible to determine a search strategy which performs better than all other strategies for all instances of the BCB problem; a PG-dominating method does not exist. We used the concept of the characteristic function to show that MUS has to be the optimal method over all possible MSIS strategies for the Shekel-5 problem. This strategy is not optimal for other instances of the BCB problem.

Furthermore, we have seen that knowledge of the characteristic function, which for practical problem solving will be out of the question (given the budget on function evaluations), gives by calculating (5) how the probability PS_N of reaching the global optimum changes, when the lowest point x_r of a random search with N points is used as a starting point for a local search. This is illustrated by Figures 8 and 9. In those figures, PS_1 equals ν, the relative size of the region of attraction of the global optimum. The well known rule from stochastic methods that $PS \rightarrow 1$ when $N \rightarrow \infty$, applies here, but cannot be derived from Figure 9. The analysis shows that the knowledge of the characteristic function is sufficient to determine the optimal sample size before performing a random search.

Under the assumption that the number of function evaluations F for one local search is known, it is possible to determine the sample size which maximizes the probability to reach the global optimum with a budget of B function evaluations.

Let K be the number of local searches to be performed in budget B and F be the number of function evaluations in one local search. As shown in Section 3, N can be taken as:

$$N = B/K - F$$

Given PS_N, the probability to reach the optimum when K local searches are performed is:

$$PG(B) = 1 - (1 - PS_{B/K-F})^K. \tag{6}$$

This analysis shows that knowledge of $\Psi(z)$ and F is sufficient to determine the optimal sample size for criterion $PG(B)$ by maximizing (6) over K. It is a new question how estimates of Ψ can be used in a practical algorithmic construction and whether it is worthwhile to do so.

6. Conclusions

In many practical situations one wants to find the minimum of a multiextremal function with a limited amount of calculation time. The name box constrained global optimization problem with a budget on the number of function evaluations (BCB) is introduced. Solving this problem with a method based on generating random points in the feasible area and local searches, implies a decision sequence on the size of the random sample at every iteration. The success of a search strategy for this decision sequence is defined as the probability that the global optimum has been detected within the budget of calculation time. Another criterion is the expected record value obtained within the budget. The success of increasing the sample is determined by the form of a so called characteristic function which depends on the problem to be solved and the local optimizer applied. It appears to be impossible to construct a dominating method, i.e. a method which performs better than all other methods for all instances of the BCB problem.

Acknowledgement

Special thanks are due to Theo Hendriks, Janos Pintér and the anonymous referees, for reading and commenting on earlier versions of the manuscript.

References

1. Betro, B. and Schoen, F. (1987), Sequential Stopping Rules for the Multistart Algorithm in Global Optimization, *Mathematical Programming* **38**, 271–286.
2. Boender, C.G.E. and Rinnooy Kan, A.H.G. (1987), Bayasian Stopping Rules for Multistart Global Optimization Methods, *Mathematical Programming* **37**, 59–80.
3. Bohachevsky, I.O., Johnson, M.E., and Stein, M.L. (1986), Generalized Simulated Annealing for Function Optimization, *Technometrics* **28**, 209–217.
4. Brazil, L.E. and Krajevski, W.F. (1987), Optimization of Complex Hydrologic Models Using Random Search Methods, in *Engineering Hydrology Proceedings*, Williamsburg, ASCE, 726–731.

5. Klepper, O. and Hendrix, E.M.T. (1994), A Method for Robust Calibration of Ecological Models under Different Types of Uncertainty, *Ecological Modelling* **74**, 161–182.
6. Powell, M.J.D. (1964), An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives, *The Computer Journal* **7**, 155–162.
7. Pronzato, L., Walter, E., Venot, A., and Lebruchec, J.F. (1984), A General Purpose Global Optimizer: Implementation and Applications, *Mathematics and Computers in Simulation* **24**, 412–422.
8. Romeijn, H.E. (1992), Global Optimization by Random Walk Sampling Methods, PhD dissertation, Erasmus University Rotterdam.
9. Törn, A. and Zilinskas, A. (1989), *Global Optimization*, Springer, Berlin.
10. Zabinsky, Z.B. and Smith, R.L. (1992), Pure Adaptive Search in Global Optimization, *Mathematical Programming* **53**, 323–338.
11. Zhigljavsky, A.A. (editor J. Pintér) (1991), *Theory of Global Random Search*, Kluwer, Dordrecht.