Global Optimization by Controlled Random Search

W. L. PRICE[1]

Communicated by L. C. W. Dixon

Abstract. The paper describes a new version, known as CRS2, of the author's controlled random search procedure for global optimization (CRS). The new procedure is simpler and requires less computer storage than the original version, yet it has a comparable performance. The results of comparative trials of the two procedures, using a set of standard test problems, are given. These test problems are examples of unconstrained optimization. The controlled random search procedure can also be effective in the presence of constraints. The technique of constrained optimization using CRS is illustrated by means of examples taken from the field of electrical engineering.

Key Words. Numerical optimization, global search, nonlinear programming.

1. Introduction

In an earlier paper (Price, Ref. 1), the author has described a controlled random search procedure (CRS) for global optimization. The procedure is direct (it does not involve gradients) and is applicable to constrained optimization. The principal features of the algorithm are described below; for further details, the reader is referred to the earlier paper.

Given a function of n variables, an initial search domain V is defined by specifying limits to each variable. A predetermined number N of trial points are chosen at random over V, consistent with the additional constraints (if any). The function is evaluated at each trial and the position and function value corresponding to each point are stored in an array A. At each iteration, a new trial point P is selected randomly from a certain set of possible trial points. Provided that the position of P satisfies the constraints, the function is evaluated at P and the function value f_P is

[1] Senior Lecturer, Department of Engineering, University of Leicester, Leicester, England. Currently at the School of Mathematics and Physics, University of East Anglia.

compared with f_M, M being the point which has the greatest function value of the N points presently stored. If $f_P < f_M$, then M is replaced (in A) by P. If either P fails to satisfy the constraints or $f_P > f_M$, then the trial is discarded, and a fresh point is chosen from the potential trial set. As the algorithm proceeds, the current set of N stored points tend to cluster around minima which are lower than the current value of f_M.

The CRS procedure achieves a reasonable compromise between the conflicting requirements of search and convergence by defining the set of possible trial points in terms of the configuration of the N points currently stored. At each iteration, $n+1$ distinct points R_1–R_{n+1} are chosen at random from the N, $N \gg n$, in store, and these constitute a simplex in n-space. The point R_{n+1} is arbitrarily taken as the pole (designated vertex) of the simplex, and the next trial point P is defined as the image point of the pole with respect to the centroid G of the remaining n points. Thus,

$$\bar{P} = 2\bar{G} - \bar{R}_{n+1},$$

where $\bar{P}$, $\bar{G}$, $\bar{R}_{n+1}$ represent the position vectors, in n-space, of the corresponding points. The points generated by this procedure are called primary trial points. Without significantly reducing the effectiveness of the primary search, the efficiency of the procedure is increased by making use also of secondary trial points defined by

$$\bar{Q} = (\bar{G} + \bar{R}_{n+1})/2.$$

Whereas the primary trial points are search oriented (P lies outside the chosen simplex), the secondary points are conducive to convergence (Q lies within the simplex). At any stage in the optimization procedure, if the cululative success rate [i.e., the percentage of successes ($f_P < f_M$) in the total number of trials so far] is below 50%, then, whenever a primary trial fails, the corresponding secondary point is chosen for the next trial. Thus, the cumulative success rate tends to converge on a value around 50%, maintaining a reasonable balance between search and convergence.

While retaining the essential principle of CRS (i.e., the defining of the next trial point in relation to a simplex chosen randomly from a stored configuration of points), the details of the procedure may be modified in a variety of ways. By using only primary trial points, for example, the CRS procedure is highly search-oriented but, for some applications, the convergence is unacceptably slow. The author has explored many variants which attempt to speed up convergence without significantly reducing the global search capability. The most successful of the early investigations was the procedure described above, which uses secondary trial points to assist convergence. More recently, however, the author has devised a new version of CRS which has a performance which is comparable with, and often superior to, the earlier version. This new version—to be known as CRS2—is the subject of the present paper. For ease of comparison, the originally published version will be called CRS1.

2. CRS2 Procedure

The CRS2 procedure differs from the CRS1 procedure in the following respects: (a) only primary trial points are used, these being defined as in CRS1 by

$$\bar{P} = 2\bar{G} - \bar{R}_{n+1};$$

and (b) the simplex point R_1 is not chosen randomly but is always that point L which has the *least* function value of the N points in store.

The flow diagram of the algorithm is shown in Fig. 1. It is apparent that CRS2 is even simpler than CRS1.

In CRS1, $n+1$ discrete points are chosen randomly from N; and, because the choice of R_{n+1} as the pole of the simplex is arbitrary (the $n+1$ points can be generated in any random order), the total number of primary trial points is $(n+1) \times {}^{N}C_{n+1}$. In CRS2, however, because R_1 is always the point L, n points are chosen randomly from $N-1$ points. Furthermore, L can never be the pole of the simplex. Thus, the number of potential trial points in CRS2 is $n \times {}^{N-1}C_n$, and these points form a subset of the corresponding primary trial points of CRS1.

As a simple illustration of the CRS2 principle, Fig. 2 shows an assumed configuration of six points at some stage in the search for global minima of a function of two variables. With $N = 6$ and $n = 2$, the number of primary CRS1 points generated by this configuration is

$$3 \times {}^{6}C_3 = 60.$$

Given that the point L, identified in Fig. 2, has the lowest function value of the configuration, the corresponding trial set for CRS2 is that subset of

$$2 \times {}^{5}C_2 = 20$$

points indicated in the diagram. In practice, N would be chosen much larger than 6 (typically, $N = 30$ for two-dimensional search using CRS2). It is apparent that the CRS2 trial set tends to bias the search to the region around L. If L is in the neighborhood of a global minimum, then clearly the effect of this bias is to achieve more rapid convergence than is attained in CRS1 by primary trials alone. Whenever a new *best point* is generated by the procedure, the CRS2 trial set changes accordingly. The danger, of course, as with all global search procedures, is that the search might

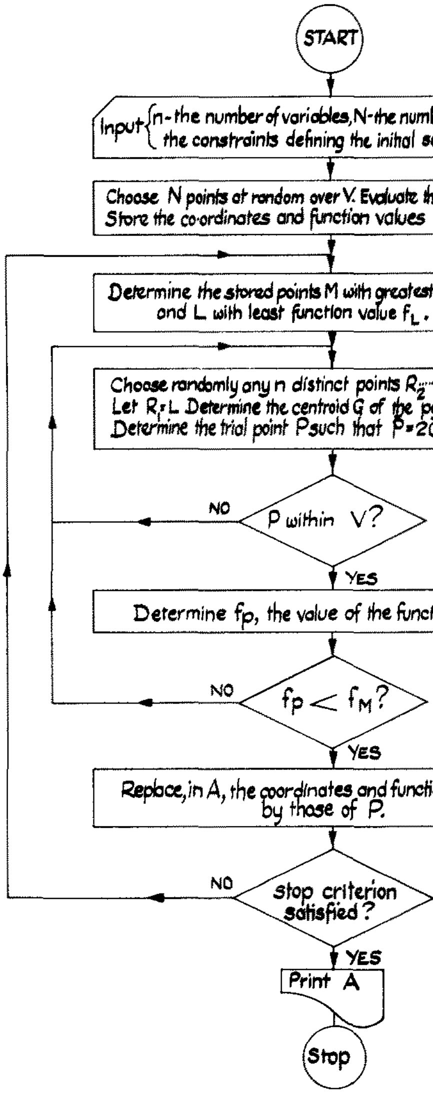

Fig. 1. Flow diagram for CRS2.

converge on a local minimum and fail to find a global minimum. The procedure must ultimately be judged on its performance over a range of appropriate test problems. Before proceeding to discuss these trials, however, it is necessary to comment on the choice of N.

Clearly, the greater value of N, the more thorough the search and the higher the probability of discovering a global minimum. On the other hand,

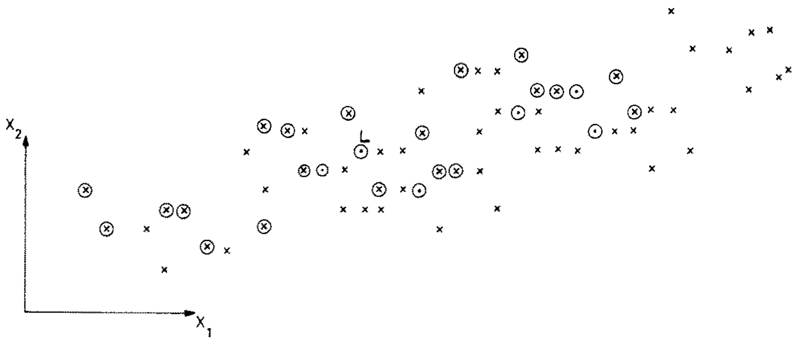

Fig. 2. An illustration of the algorithm for generating trial points in two dimensions. Symbol $\odot$ denotes configuration of six stored points, L having the lowest function value; symbol $\times$ denotes set of 60 possible trial points for CRS1; symbol $\otimes$ denotes subset of 20 trial points for CRS2.

the larger the value of N, the greater is the demand on computer storage and the slower the convergence of the algorithm. The appropriate choice of N is a matter of experience; and, for CRS1, the empirical rule adopted by the author is to take

$$N = 25n,$$

unless there is good reason to choose otherwise. For many test problems, the global minima can be located with a smaller value of N but, in some circumstances (an example is given in the next section), failure to use the recommended value of N results in convergence to a local, rather than global, minimum. Experience with CRS2, however, suggests that this procedure will tolerate a smaller value of N. The empirical rule recomended for CRS2 is

$$N = 10(n+1).$$

Of course, it is always possible to devise test problems which defeat either algorithm, and the user should err on the side of a generous value of N when optimizing complicated functions, especially when constraints are imposed.

3. Trials

Both CRS1 and CRS2 have been programmed (in BASIC) so as to run either on a small computer or on a mainframe. The trials described below were conducted on Leicester University's Cyber 73 computer. The test problems chosen are those described in Ref. 2. These problems are defined in the Appendix (Section 7).

The performances of CRS1 and CRS2 are compared in terms of the number of function evaluations required, the starting conditions, stop criteria, and random sequence used being the same for each procedure. In every case, a run was terminated when the function values of all points in store were identical to an accuracy of six decimal places, i.e.,

$$f_M - f_L < 10^{-6}.$$

Because the recommended values of N are different for the two algorithms, it is fair to compare CRS2 run with

$$N = 10(n+1)$$

not only with CRS1 run with

$$N = 25n,$$

but also with CRS1 run with

$$N = 10(n+1).$$

The results of the comparison are shown in Table 1.

In comparison with CRS1 ($N = 25n$), it will be observed that CRS2 performs about equally well on the Shekel problem, but converges more rapidly on the other problems. When the two procedures have the same value of N, CRS1 converges more rapidly on the Shekel problem, but it fails to locate the global minimum on the Hartman ($n = 6$) problem. It is also of interest to note (not shown in the table) that, while the cumulative success rate (as defined in Section 1) of CRS1 automatically adapts to approximately 50%, the success rate of CRS2 varies between 40% (Shekel problems) and 60% (Branin problems).

Table 1. Number of function evaluations for each trial.

Test problem	CRS1 $N = 25n$	CRS1 $N = 10(n+1)$	CRS2 $N = 10(n+1)$
Example 7.1, $n = 4$, $m = 5$	3800	2486	3979
Example 7.1, $n = 4$, $m = 7$	4900	2984	3824
Example 7.1, $n = 4$, $m = 10$	4400	3095	4213
Example 7.2, $n = 3$	2400	1448	1297
Example 7.2, $n = 6$	7600	(*)	4705
Example 7.3, $n = 2$	1600	1039	670
Example 7.4, $n = 2$	2500	945	914

(*) Convergence to local minimum.

A somewhat more difficult problem and the method used to achieve a global minimum are described in the earlier paper (Ref. 1, pp. 81–83). A definition of this nine-variable problem is also given in Section 5 (Example 5.3) of the present paper. For this earlier approach, the primary search mode only of CRS1 was used. A storage value of $N = 200$ was taken. Using CRS2 with $N = 100$, the same global solution is obtained, but the number of function evaluations required is reduced by about 70%. Again, the success rate is in the region of 40–60%.

4. Constrained Optimization

The problems described above are all examples of unconstrained optimization. The CRS procedure can often be effective in constrained optimization.

In general terms, a constrained optimization problem may be specified in the following form. An objective function $f(x)$ is to be minimized, subject to a set of equality constraints

$$g_i(x) = 0, \qquad i = 1, \ldots, p,$$

and a set of inequality constraints

$$h_j(x) \geq 0 \qquad j = 1, \ldots, q.$$

One of the most widely used methods for solving problems of this type is the penalty function technique.

A function $F(x)$ is defined by

$$F = f + \sum_{i=1}^{p} \lambda_i g_1^2 + \sum_{j=1}^{q} \mu_j \, [\min(0, h_j)]^2,$$

where λ_i and μ_j ($\lambda_i, \mu_j > 0$) are appropriately chosen weighting parameters. The problem is thus transformed into one of unconstrained minimization of F. Two optimization procedures based on this technique are compared by Bartholomew-Biggs (Ref. 3) who applies the methods to several engineering problems. Two of these examples, from the field of electrical engineering, have been chosen to test the CRS approach.

5. Trials

Three test problems are described. The trials were conducted on Leicester University's Cyber 73 computer using the CRS2 version of the procedure. In each case, the value of N was chosen to be $10(n + 1)$, where

n is the number of variables. A run was terminated when the function values of the N points in store were identical to within six places of decimals.

Example 5.1. *Transformer Design.* This six-variable problem arises in connection with transformer design (Ballard, Jelinek, and Schinzinger, Ref. 4). It is required to minimize

$$f = 0.0204x_1x_4(x_1+x_2+x_3)+0.0187x_2x_3(x_1+1.57x_2+x_4)$$
$$+0.0607x_1x_4x_5^2(x_1+x_2+x_3)+0.0437x_2x_3x_6^2(x_1+1.57x_2+x_4),$$

subject to the inequality constraints

$$x_i \geqslant 0, \qquad i=1,\ldots,6,$$
$$x_1x_2x_3x_4x_5x_6 \geqslant 2.07\times 10^3,$$
$$1-0.00062x_1x_4x_5^2(x_1+x_2+x_3)-0.00058x_2x_3x_6^2(x_1+1.57x_2+x_4)\geqslant 0.$$

The variables x_1, x_2, x_3, x_4 represent physical dimensions of winding and core of the transformer. The variables x_5 and x_6 correspond to magnetic flux density and current density, respectively. The objective function represents the worth of the transformer, including operating costs. The constraints, other than the simple bounds, refer to the rating of the transformer and the allowable transmission loss.

The approach used by Bartholomew-Biggs (BB) begins with a feasible starting point

$$x_1=5.54 \qquad x_2=4.4, \qquad x_3=12.02,$$
$$x_4=11.82, \qquad x_5=0.702, \qquad x_6=0.852,$$

and leads to a solution at

$$x_1=5.33, \qquad x_2=4.66, \qquad x_3=10.43,$$
$$x_4=12.08, \qquad x_5=0.752, \qquad x_6=0.878.$$

In order to test the capabilities of the CRS procedure, no such starting point is assumed. An initial search volume must be defined (e.g., by setting up the initial configuration of N points in the region $0<x_i<10$). The penalty function method could then be used for handling the constraints. However, it is advisable, when using CRS, to avoid penalty functions whenever possible. Depending on the values of the weighting parameters, the modified objective function can have a local minimum, to which CRS might converge, in the neighborhood of the global minimum. Indeed, this sometimes happens when the penalty function technique is applied to the present example, because the global solution lies on a constraint boundary. The procedure described below achieves some preliminary simplification of Example 5.1 and does not involve penalty functions.

Consider some point x which lies within, but not on, the constraint boundaries. If one or more of the x_i is reduced in value (while remaining positive) so that x moves toward the constraint boundary

$$x_1x_2x_3x_4x_5x_6 = 2.07 \times 10^3,$$

then the other constraints remain satisfied while the objective function decreases. Hence, the solution lies on

$$x_1x_2x_3x_4x_5x_6 = 2.07 \times 10^3,$$

and the number of variables can be reduced to five by the elimination of, say, x_6. The other major constraint can now be written in the form

$$1 - Ax_5^2 - B/x_5^2 > 0,$$

where

$$A = 0.00062x_1x_4(x_1 + x_2 + x_3) > 0,$$

$$B = 0.00058(x_1 + 1.57x_2 + x_4) \times (2.07)^2 \times 10^6/x_1^2x_2x_3x_4^2 > 0.$$

Therefore

$$1/2A + C > x_5^2 > 1/2A - C,$$

where

$$C = (1/2A)\sqrt{(1 - 4AB)}.$$

This implies also that

$$AB < 1/4.$$

Hence,

$$x_4 > (x_1 + x_2 + x_3)(x_1 + 1.57x_2)/[Kx_1x_2x_3 - (x_1 + x_2 + x_3)],$$

where

$$K = 1/[4 \times (2.07)^2 \times 0.62 \times 0.58] = 0.16225.$$

Because

$$x_4 > 0,$$

this also implies that

$$x_3 > (x_1 + x_2)/(Kx_1x_2 - 1),$$

which similarly implies that

$$x_2 > 1/Kx_1.$$

It is now easy to set up an initial configuration of

$$10(5+1)=60$$

points which satisfy the constraints. Taking a range of 10 units for x_1, x_2, x_3, x_4, the x_i were chosen randomly in sequence,

$$x_1 = 10R_1,$$
$$x_2 = 1/Kx_1 + 10R_2,$$
$$x_3 = (x_1+x_2)/(Kx_1x_2-1)+10R_3,$$

etc., where R_1, R_2, R_3 are random numbers in the range (0, 1). Having established a valid initial configuration, the CRS algorithm now proceeds by discarding subsequent trial points which fail to satisfy the constraints. The solution, which agrees with that given by BB, is obtained in about 2500 function evaluations (40 seconds of computer time). The BB procedures are an order of magnitude faster than this, but it will be appreciated that CRS, having no prior knowledge of a feasible starting point, is required to perform a global search over the viable domain.

Example 5.2. *Static Power Scheduling.* The second example concerns static power scheduling (Fiacco and McCormick, Ref. 5). It is required to minimize

$$f = 3000x_1 + 1000x_1^3 + 2000x_2 + 666.667x_2^3,$$

subject to the constraints

$$\begin{aligned}
0.4 - x_1 + 2Cx_5^2 + x_5x_6[D\sin(-x_8) - C\cos(-x_8)]&\\
+x_5x_7[D\sin(-x_9) - C\cos(-x_9)] &= 0,\\
0.4 - x_2 + 2Cx_6^2 + x_6x_5[D\sin(x_8) - C\cos(x_8)]&\\
+x_6x_7[D\sin(x_8-x_9) - C\cos(x_8-x_9)] &= 0,\\
0.8 + 2Cx_7^2 + x_7x_5[D\sin(x_9) - C\cos(x_9)]&\\
+x_7x_6[D\sin(x_9-x_8) - C\cos(x_9-x_8)] &= 0,\\
0.2 - x_3 + 2Dx_5^2 + x_5x_6[C\sin(-x_8) + D\cos(-x_8)]&\\
-x_5x_7[C\sin(-x_9) + D\cos(-x_9)] &= 0,\\
0.2 - x_4 + 2Dx_6^2 - x_6x_5[C\sin(x_8) + D\cos(x_8)]&\\
-x_6x_7[C\sin(x_8-x_9) + D\cos(x_8-x_9)] &= 0,
\end{aligned}$$

$$-0.337+2Dx_7^2-x_7x_5[C\sin(x_9)+D\cos(x_9)]$$
$$-x_7x_6[C\sin(x_9-x_8)+D\cos(x_9-x_8)]=0,$$
$$x_i\geq 0, \qquad i=1,2,$$
$$1.0909\geq x_i\geq 0.90909, \qquad i=5,6,7,$$

where

$$C=\sin(0.25)48.4/50.176,$$
$$D=\cos(0.25)48.4/50.176.$$

In this problem, x_1, x_2 are the real power outputs from two generators; x_3, x_4 are the reactive power outputs; x_5, x_6, x_7 are voltage magnitudes at three nodes of an electrical network; and x_8, x_9 are voltage phase angles at two of these nodes. The equality constraints are the real and reactive power balance equations, stating that the power flowing into a node must balance the power flowing out. The remaining constraints are simple limits on the real power and the voltage magnitudes.

The starting point used by BB is

$$x_1=x_2=0.8, \qquad x_3=x_4=0.2,$$
$$x_5=x_6=x_7=1.0454, \qquad x_8=x_9=0.$$

The solution is

$$x_1=0.667, \qquad x_2=1.0224, \qquad x_3=0.2283, \qquad x_4=0.1848,$$
$$x_5=x_6=1.0909, \qquad x_7=1.0691, \qquad x_8=0.1066, \qquad x_9=-0.3388.$$

Again, preliminary simplification of the problem is possible. The dimensionality is reduced from nine to three independent variables (x_7, x_8, x_9) by first solving the third and sixth equality constraints to obtain x_5, x_6, and then solving the remaining equality constraints to obtain x_1, x_2, x_3, x_4. The initial configuration of 40 points was established by choosing x_7 within the stated range and by assigning a range of one unit for each of the free variables x_8, x_9. After calculation of the other six variables, an initial point was discarded if the constraints on these six were not satisfied. The CRS procedure achieved the solution (agreeing with BB) in about 1000 function evaluations (30 seconds).

Example 5.3. *Transistor Modelling.* It is required to minimize

$$f(x_1,\ldots,x_9)=\gamma^2+\sum_{k=1}^{4}(\alpha_k^2+\beta_k^2),$$

where

$$\alpha_k = (1 - x_1x_2)x_3\{\exp[x_5(g_{1k} - g_{3k}x_7 \times 10^{-3} - g_{5k}x_8 \times 10^{-3})] - 1\} - g_{5k} + g_{4k}x_2,$$

$$\beta_k = (1 - x_1x_2)x_4\{\exp[x_6(g_{1k} - g_{2k} - g_{3k}x_7 \times 10^{-3} + g_{4k}x_9 \times 10^{-3})] - 1\} - g_{5k}x_1 + g_{4k},$$

and the numerical constants g_{ik} are given by the matrix

$$\begin{bmatrix} 0.485 & 0.752 & 0.869 & 0.982 \\ 0.369 & 1.254 & 0.703 & 1.455 \\ 5.2095 & 10.0677 & 22.9274 & 20.2153 \\ 23.3037 & 101.779 & 111.461 & 191.267 \\ 28.5132 & 111.8467 & 134.3884 & 211.4823 \end{bmatrix}$$

This function provides a least-sum-of-squares approach to the solution of a set of nine simultaneous nonlinear equations, which arise in the context of transistor modelling.

This problem, stated without constraints, has been used by the author as an example of unconstrained global optimization (Price, Ref. 1). The solution so obtained lies outside the positive orthant. For the purposes of the present investigation, the constraints

$$x_1 \geq 0, \qquad i = 1, \ldots, 9,$$

will be imposed, thus invalidating the solution found previously. As was shown in the earlier paper, a solution does exist close to the point

$$x_1 = 0.9, \qquad x_2 = 0.45, \qquad x_3 = 1,$$
$$x_4 = 2, \qquad x_5 = 8, \qquad x_6 = 8,$$
$$x_7 = 5, \qquad x_8 = 1, \qquad x_9 = 2,$$

but the function is very sensitive in this region, and the solution point is easily missed in global search if no prior knowledge of the location of the viable region is assumed.

A study of the problem reveals that it reduces from nine to three independent variables x_1, x_2, x_3. In this form, the objective function becomes

$$f = f_1^2 + f_2^2 + f_3^2,$$

where f_1, f_2, f_3 are functions of the eight quantities defined by

$$F_i = \log_e[1 + (g_{5i} - g_{4i}x_2)/(1 - x_1x_2)x_3], \qquad i = 1, 2, 3, 4,$$

$$F_i = \log_e[1 + (g_{5i}x_1x_2 - g_{4i}x_2)/(1 - x_1x_2)x_1x_3], \qquad i = 5, 6, 7, 8.$$

It is now apparent that the F_i imply eight additional inequality constraints, because the arguments of the logarithms must be positive. The viable search region is thus much more restricted than appears from the original statement of the problem. The initial configuration of 40 points was established in a manner similar to that described in Example 5.1, and the solution (quoted above) was obtained in about 27,000 function evaluations (4.5 minutes of computing time).

6. Conclusions

The results of trials suggest that, in terms of performance, the CRS2 procedure is a viable alternative to the CRS1 procedure. The CRS2 procedure has the advantage of being simpler and it requires less computer storage.

Provided that an initial configuration of viable points can readily be established, the CRS procedure is capable of global optimization, subject to inequality constraints. Whenever possible, equality constraints should be removed by elimination of variables: otherwise, these constraints must be handled by the penalty function method. The CRS algorithm is in general much slower than the localized procedures devised for nonlinear programming, but it is much less sophisticated, and it does not assume the availability of a feasible starting point close to a solution.

7. Appendix: Test Functions for Global Optimization

7.1. Shekel Function (SQRIN)

$$f(x) = -\sum_{i=1}^{m} \left\{ \frac{1}{[(x-a_i)^T(x-a_i)+c_i]} \right\},$$

$$x = (x_1, \ldots, x_n)^T, \qquad a_i = (x_{i1}, \ldots, a_{in})^T, \qquad c_i > 0.$$

The region of interest is given by

$$0 \le x_j \le 10, \qquad j = 1, \ldots, n.$$

Consider three cases from Table 2 with data $n = 4$ and $n = 5, 7, 10$.

7.2. Hartman Function

$$f(x) = -\sum_{i=1}^{m} c_i \exp\left(-\sum_{j=1}^{n} \alpha_{ij}(x_j - p_{ij})^2 \right),$$

$$x = (x_1, \ldots, x_n)^T, \qquad p_i = (p_{i1}, \ldots, p_{in})^T, \qquad \alpha_i = (\alpha_{i1}, \ldots, \alpha_{in})^T.$$

Table 2. Data for Example 7.1.

i	a_i				c_i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	7	7	0.4
6	2	9	2	9	0.6
7	5	5	3	3	0.3
8	8	1	8	1	0.7
9	6	2	6	2	0.5

Here, p_i is an approximate location of the ith local minimum, α_i is proportional to eigenvalues of the Hessian at the ith local minimum, and $c_i > 0$ is the height (depth?) of the ith local minimum (assuming that the interference of different local minima is not too strong). The region of interest is

$$0 \leq x_i \leq 1.$$

See Tables 3–5.

7.3. Branin Function (RCOS)

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

$$a = 1, \qquad b = 5.1/(4\pi^2), \qquad c = 5/\pi, \qquad d = 6, \qquad e = 10, \qquad f = 1/(8\pi).$$

The region of interest is given by

$$-5 \leq x_1 \leq 10, \qquad 0 \leq x_2 \leq 15.$$

There are three minima, all global, in this region.

Table 3. Example 7.2, $m = 4$, $n = 3$.

i	α_i			c_i	p_i		
1	3	10	30	1	0.3689	0.117	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.747
3	3	10	30	3	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.03815	0.5743	0.8828

Table 4. Example 7.2, $m = 4$, $n = 6$.

i	α_i						c_1
1	10	3	17	3.5	1.7	8	1
2	0.05	10	17	0.1	8	14	1.2
3	3	3.5	1.7	10	17	8	3
4	17	8	0.05	10	0.1	14	3.2

Table 5. Example 7.2, $m = 4$, $n = 6$.

i	p_i					
1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

7.4. Goldstein and Price Function (GOLDPR)

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

The region of interest is given by

$$-2 \le x_{1,2} \le 2.$$

There are four local minima. The global minimum occurs at $(0, 1)$ with value $f = 3$.

References

1. PRICE, W. L., *A Controlled Random Search Procedure for Global Optimization*, Toward Global Optimization 2, Edited by L. C. W. Dixon and G. P. Szegö, North-Holland Publishing Company, Amsterdam, Holland, pp. 71–84, 1978.
2. DIXON, L. C. W., and SZEGÖ, G. P., *The Global Optimization Problem: An Introduction*, Toward Global Optimization 2, Edited by L. C. W. Dixon and G. P. Szegö, North-Holland Publishing Company, Amsterdam, Holland, pp. 1–15, 1978.
3. BARTHOLOMEW-BIGGS, M. C., *A Numerical Comparison between Two Approaches to the Nonlinear Programming Problem*, Toward Global Optimization 2, Edited by L. C. W. Dixon and G. P. Szegö, North-Holland Publishing Company, Amsterdam, Holland, pp. 293–312, 1978.

4. BALLARD, D. H., JELINEK, C. A., and SCHINZINGER, R., *An Algorithm for the Solution of Constrained Generalized Polynomial Programming Problems*, Computer Journal, Vol. 17, pp. 261–266, 1974.
5. FIACCO, A. V., and MCCORMICK, G. P., *Nonlinear Programming—Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, New York, 1968.