JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 76, No. 3, MARCH 1993

Shuffled Complex Evolution Approach for Effective and Efficient Global Minimization[1]

Q. Y. DUAN,[2] V. K. GUPTA,[3] AND S. SOROOSHIAN[4]

Communicated by L. C. W. Dixon

Abstract. The degree of difficulty in solving a global optimization problem is in general dependent on the dimensionality of the problem and certain characteristics of the objective function. This paper discusses five of these characteristics and presents a strategy for function optimization called the shuffled complex evolution (SCE) method, which promises to be robust, effective, and efficient for a broad class of problems. The SCE method is based on a synthesis of four concepts that have proved successful for global optimization: (a) combination of probabilistic and deterministic approaches; (b) clustering; (c) systematic evolution of a complex of points spanning the space, in the direction of global improvement; and (d) competitive evolution. Two algorithms based on the SCE method are presented. These algorithms are tested by running 100 randomly initiated trials on eight test problems of differing difficulty. The performance of the two algorithms is compared to that of the controlled random search CRS2 method presented by Price (1983, 1987) and to a multistart algorithm based on the simplex method presented by Nelder and Mead (1965).

Key Words. Numerical optimization, global search, efficiency, effectiveness, complex of points, random search, competitive evolution.

[1]This work was partially funded by the National Science Foundation, Grant ECE-86-10584, and by the National Weather Service, Grant NA85AA-H-HY088.

[2]Doctoral Student, Department of Hydrology and Water Resources, The University of Arizona, Tucson, Arizona.

[3]Adjunct Associate Professor, Department of Hydrology and Water Resources, The University of Arizona, Tucson, Arizona.

[4]Professor, Department of Systems and Industrial Engineering, and Department of Hydrology and Water Resources, The University of Arizona, Tucson, Arizona.

0022-3239/93/0300-0501$07.00/0

1. Introduction

The degree of difficulty in solving a global optimization problem is dependent on certain characteristics, related in general to the dimensionality of the problem and the characteristics of the objective function [see, for example, Dixon and Szegö (Refs. 1 and 2) and Törn and Zilinskas (Ref. 3)]. Our experience with practical global optimization problems, particularly in the field of hydrologic model calibration [Sorooshian et al. (Ref. 4), Gupta and Sorooshian (Refs. 5 and 6), and Duan, Sorooshian, and Gupta (Ref. 7)] indicates that there are five major characteristics that complicate the solution process:

(1) There may be several major regions of attraction into which a search strategy may converge.
(2) Each major region of attraction may contain numerous (possibly uncountable) local minima (stationary points where the first derivatives are zero and the Hessian matrices are positive definite or positive semidefinite). These local optima may occur both close to and at various distances from the best solution.
(3) The objective function surface in the multiparameter space may not be smooth and may not even be continuous. The derivatives may be discontinuous and may vary in an unpredictable manner through the parameter space.
(4) The parameters may exhibit varying degrees of sensitivity and a great deal of interaction and compensation. Much of the interaction can be highly nonlinear.
(5) The response surface near the true solution is often nonconvex.

An optimization problem containing all five of the features mentioned above can be particularly difficult to solve. The task, therefore, is to construct an algorithm that is capable of dealing with these various difficulties. We present here a new global optimization strategy, entitled the shuffled complex evolution (SCE) method, which promises to be effective and efficient for a broad class of problems.

The SCE method is based on a synthesis of four concepts that have proved successful for global optimization:

(a) combination of random and deterministic approaches; see Dixon and Szegö (Ref. 2), Gomulka (Ref. 8), Törn and Zilinskas (Ref. 3);
(b) the concept of clustering; see Becker and Lago (Ref. 9), Törn (Ref. 10), and Rinnooy-Kan and Timmer (Ref. 11);

(c) the concept of a systematic evolution of a complex of points spanning the space, in the direction of global improvement; see Price (Refs. 12–14);
(d) the concept of competitive evolution; see Holland (Ref. 15).

In the interest of brevity, only a brief discussion of these concepts is presented here. The first concept is very important. The use of deterministic strategies allows the SCE algorithm to make effective use of response surface information to guide the search, while the inclusion of random elements helps to make the algorithm flexible and robust. The search begins with a randomly selected complex of points spanning the entire feasible space Ω. A large enough number of points will help ensure that the complex contains information regarding the number, location, and size of the major regions of attraction. The implementation of an implicit clustering strategy helps to concentrate the search in the most promising of the regions identified by the initial complex. The use of a systematic complex evolution strategy helps to ensure that the search is relatively robust and is guided by the structure of the objective function. The robustness is a result of the fact that the complex structure is able to cope very well with rough, insensitive, and highly nonconvex objective function surfaces and is relatively unaffected by the small local minima that are encountered enroute to the global solution. Further, no derivative information is required. The implementation of a strategy of competitive evolution has been shown by Holland (Ref. 15) to be useful in improving global convergence efficiency.

The controlled random search (CRS) method of Price (Refs. 12–14) was used as a starting point for the SCE method, since it incorporates some of the concepts mentioned above, while being easy to implement and modify. Price proposed and tested three versions of the CRS strategy (CRS1, CRS2, CRS3). The CRS methods possess several properties that are desirable for effective and efficient global optimization. However, they also have certain weaknesses. The CRS1 strategy treats each region of the sampled area in a nonpreferential manner, and convergence can be slow. On the other hand, the CRS2 and CRS3 strategies place their major emphasis on the best point in the current sample, so that the search can easily become biased toward the region of a local minimum. Also, the strategy of always replacing the current worst point in the population by each newly generated point causes the population to shrink into a small region very quickly. Tests that we have conducted show that there is a good possibility that the population will begin to contain repeated points, therefore becoming degenerate and causing the search to terminate prematurely. To minimize this possibility, the sample size has to be sufficiently large. In the SCE method, this latter problem does not arise.

2. Shuffled Complex Evolution Method

The shuffled complex evolution (SCE) strategy combines the strengths of the CRS algorithms with the concept of competitive evolution (Holland, Ref. 15) and the newly developed concept of complex shuffling. The SCE strategy is presented below and is illustrated in Fig. 1.

Step 0. Initialize. Select $p \geq 1$ and $m \geq n+1$, where p = number of complexes and m = number of points in each complex. Compute the sample size $s = p \times m$.

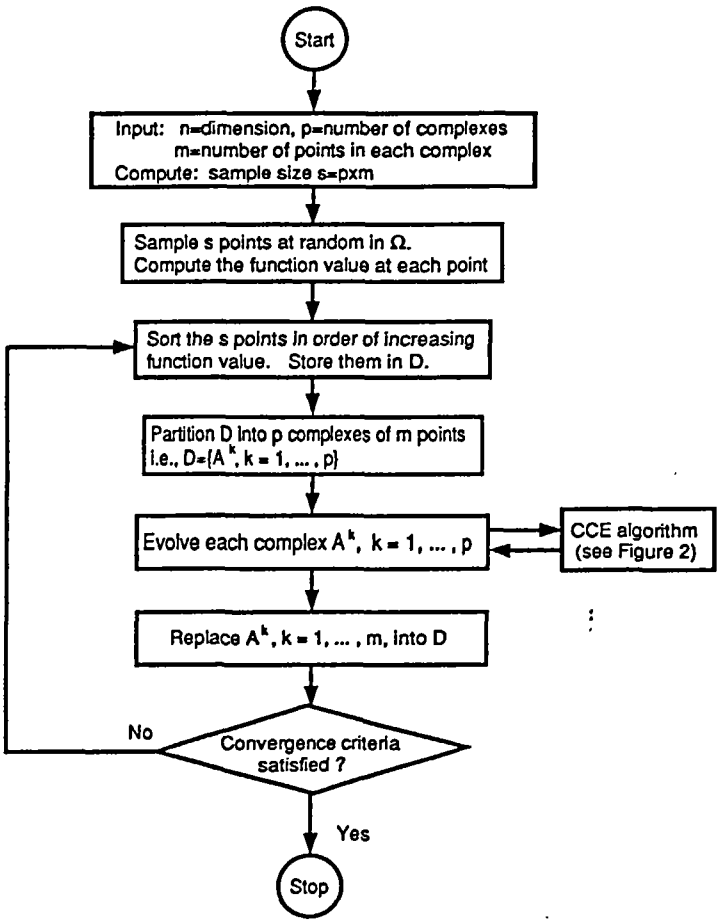


Fig. 1. Flow chart of the shuffled complex evolution (SCE) method.

Step 1. Generate Sample. Sample s points $x_1, \ldots, x_s$ in the feasible space $\Omega \subset \mathbb{R}^n$. Compute the function value f_i at each point x_i. In the absence of prior information, use a uniform sampling distribution.

Step 2. Rank Points. Sort the s points in order of increasing function value. Store them in an array $D = \{x_i, f_i, i = 1, \ldots, s\}$, so that $i = 1$ represents the point with the smallest function value.

Step 3. Partition into Complexes. Partition D into p complexes $A^1, \ldots, A^p$, each containing m points, such that:

$$A^k = \{x_j^k, f_j^k \mid x_j^k = x_{k+p(j-1)}, f_j^k = f_{k+p(j-1)}, j = 1, \ldots, m\}.$$

Step 4. Evolve Each Complex. Evolve each complex A^k, $k = 1, \ldots, p$ according to the competitive complex evolution (CCE) algorithm outlined separately.

Step 5. Shuffle Complexes. Replace $A^1, \ldots, A^p$ into D, such that $D = \{A^k, k = 1, \ldots, p\}$. Sort D in order of increasing function value.

Step 6. Check Convergence. If the convergence criteria are satisfied, stop; otherwise, return to Step 3.

The competitive complex evolution (CCE) algorithm required for the evolution of each complex in Step 4 of the SCE method is presented below and is illustrated in Fig. 2.

Step 0. Initialize. Select q, α, and β, where $2 \leq q \leq m$, $\alpha \geq 1$, and $\beta \geq 1$.

Step 1. Assign Weights. Assign a triangular probability distribution to A^k; i.e.,

$$p_i = 2(m+1-i)/m(m+1), \qquad i = 1, \ldots, m.$$

The point x_1^k has the highest probability, $\rho_1 = 2/m + 1$. The point x_m^k has the lowest probability, $\rho_m = 2/m(m+1)$.

Step 2. Select Parents. Randomly choose q distinct points $u_1, \ldots, u_q$ from A^k according to the probability distribution specified above (the q points define a subcomplex). Store them in array $B = \{u_i, v_i, i = 1, \ldots, q\}$, where v_j is the function value associated with point u_j. Store in L the locations of A^k which are used to construct B.

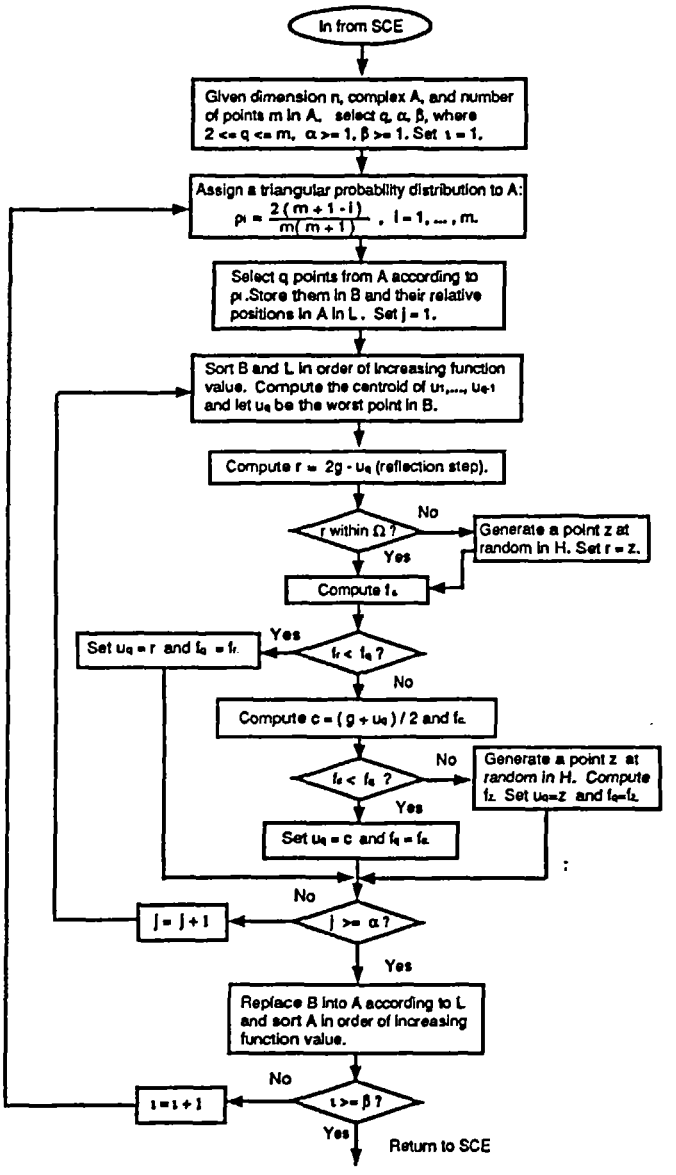


Fig. 2. Flow chart of the competitive complex evolution (CCE) strategy.

Step 3. Generate Offspring.

(a) Sort B and L so that the q points are arranged in order of increasing function value. Compute the centroid g using the expression:

$$g=[1/(q-1)]\sum_{j=1}^{q-1} u_j.$$

(b) Compute the new point $r=2g-u_q$ (reflection step).

(c) If r is within Ω, compute the function value f, and go to Step (d); else, compute the smallest hypercube $H\subset\mathbb{R}^n$ that contains A^k, randomly generate a point z within H, compute f_z, set $r=z$, and set $f_r=f_z$ (mutation step).

(d) If $f_r<f_q$, replace u_q by r, go to Step (f); else, compute $c=(g+u_q)/2$ and f_c (contraction step).

(e) If $f_c<f_q$, replace u_q by c, go to Step (f); else, randomly generate a point z within H and compute f_z (mutation step). Replace u_q by z.

(f) Repeat Steps (a) through (e) α times, where $\alpha\geq 1$ is a user-specified parameter.

Step 4. Replace Parents by Offspring. Replace B into A^k using the original locations stored in L. Sort A^k in order of increasing function value.

Step 5. Iterate. Repeat Steps (1) through (4) β times, where $\beta\geq 1$ is a user-specified parameter which determines how many offspring should be generated (how far each complex should evolve).

The philosophy behind the SCE approach is to treat the global search as a process of natural evolution. The sampled points (s in number) constitute a population. The population is partitioned into several communities (complexes), each of which is permitted to evolve independently (i.e., search the space in different directions). After a certain number of generations, the communities are forced to mix, and new communities are formed through a process of shuffling. This procedure enhances survivability by a sharing of the information (about the search space) gained independently by each community.

Each member of a community (complex) is a potential parent with the ability to participate in a process of reproduction. A subcomplex selected from the complex is like a pair of parents, except that a subcomplex may consist of more than two members. To ensure that the evolution process is competitive, we require that the probability that better parents contribute to the generation of offspring is higher than that of worse parents. The use of a triangular probability distribution ensures this competitiveness. Nelder

and Mead's (Ref. 16) procedure is applied to each subcomplex to generate most of the offspring. This strategy uses the information contained in the subcomplex to direct the evolution in an improvement direction. In addition, offspring are introduced at random locations of the feasible space under certain conditions in order to ensure that the process of evolution does not get trapped by unpromising regions. This is somewhat analogous to mutation in response to stress that can occur in biological evolution. Each mutation also helps to increase the amount of information stored in the sample. Finally, each new offspring replaces the worst point of the current subcomplex, rather than the worst point of the entire population. This ensures that every parent gets at least one chance to contribute to the reproduction process before being replaced or discarded. Thus, none of the information contained in the sample is ignored.

The SCE method is designed to improve on the best features of the CRS method (i.e., global sampling, complex evolution), by incorporating within it the powerful concepts of competitive evolution and complex shuffling. Both competitive evolution and complex shuffling help to ensure that the information contained in the sample is efficiently and thoroughly exploited. They also help to ensure that the information set does not become degenerate. These properties lead us to expect that the SCE method will have better global convergence properties over a broader range of problems. In other words, given a prespecified number of function evaluations (fixed level of efficiency), the SCE method should have a higher probability of succeeding in its objective of finding the global optimum.

3. SCE1 Algorithm

The SCE1 algorithm is a version of the SCE strategy in which the entire sample s is allocated to only one complex ($m=s$). It contains no shuffling procedure. The number of offspring β that can be generated before checking for convergence is the same as the complex size m. The procedure differs from CRS2 primarily in the manner in which the complex is evolved (described above). The size of each subcomplex selected for generation of an offspring is $n+1$, the standard size for a simplex specified by Nelder and Mead (Ref. 16) and also used in the CRS procedures. The value of the parameter α was set equal to 1 (only one step evolution of each simplex was permitted) to match the CRS procedure. The primary variable to be selected in this method is the sample size s.

4. SCE2 Algorithm

In the SCE2 algorithm, the size of a complex m is chosen to be equal to $2n+1$, where n is the dimension of the problem. The number of offspring

β that can be generated by each independently evolving complex between two consecutive shuffles is the same as the complex size $(2n+1)$. The size of each subcomplex selected for generation of an offspring is $n+1$, the standard size for a simplex specified by Nelder and Mead (Ref. 16) and also used in the CRS procedure. The value of the parameter α was set equal to 1 (only one step evolution of each simplex was permitted) to match the CRS procedure. The primary variable to be selected in this method is the number of complexes p.

5. Design of the Comparative Performance Study

A set of eight test problems was selected (see Table 1). The first seven of these are well-established test problems from the literature. The eighth is a research version of the National Weather Service flood forecasting model [Gupta and Sorooshian (Ref. 5)], which is known to contain all five of the problem characteristics described in Section 1 [Duan, Sorooshian, and Gupta (Ref. 7)]. Each function was adjusted so that the globally optimal function value is 0.0, and the function value elsewhere in the feasible space is positive. In each case, any point with a function value less than 10^{-3} is guaranteed to lie in a very small region around the global optimum. The mathematical representations of these test problems are presented in the Appendix.

Four global optimization algorithms were tested and compared on each of the eight test problems. These are CRS2, SCE1, SCE2, and MSX, a multistart algorithm using the local-search simplex procedure of Nelder and Mead (Ref. 16). Each optimization algorithm was tested by running 100 trials of the procedure on each test function. Each trial began with an independent randomly generated sample s of points selected uniformly from the feasible space. The stopping criteria are as follows. In each case, a trial was deemed a success as soon as the best function value in the sample became

Table 1. Test problems used in this study.

Problem number	Function name	Dimension	Number of optima
1	Goldstein-Price function	2	4
2	Rosenbrock function	2	1
3	Six-hump camelback function	2	6
4	Rastrigin function	2	>50
5	Shekel function	4	10
6	Hartman function	6	4
7	Griewank function	10	>1000
8	SIXPAR hydrologic model	6	Uncountable

less than 10^{-3}. However, if the trial reached 25,000 function evaluations, or if sample parameter convergence occurred (the region spanned by the sample converged down to within 10^{-12} of the parameter range in each direction) without reducing the best function value below 10^{-3}, the trial was deemed a failure. Two statistics were recorded. The first statistic is the number of failures (NF) out of the 100 trials. Dividing NF by 100 gives an approximation of the probability of failure and, therefore, measures the effectiveness (robustness) of the algorithm. The second statistic is the average number of function evaluations (AFE) for the successes (i.e., the failures are not included). This gives an approximate idea of the efficiency of the algorithm. An *xy*-plot of one statistic against the other gives an idea of the trade-off between efficiency and effectiveness inherent in probabilistic algorithms. Low values of both statistics indicate an effective and efficient algorithm.

6. Results of the Comparative Performance Study

The two statistics NF and AFE were computed for increasing sample sizes s for the CRS2 and SCE1 algorithms, for increasing numbers of complexes p for the SCE2 algorithm, and for increasing numbers of restarts for the MSX algorithm. The results are presented in Tables 2a–2h and plotted in Figs. 3a–3h. Each plot contains four curves. The dash-dot line indicates the MSX algorithm; the dotted line indicates the CRS2 algorithm; the

Table 2a. Results for the Goldstein-Price function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	25	10	10	17	32	10	1	159	2	1	163
2	10	116	15	7	167	15	1	159	3	1	231
3	2	180	20	5	207	20	1	278	4	0	311
4	1	243	25	0	250	25	0	332	⋮		
5	1	307									

Table 2b. Results for the Rosenbrock function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	0	102	10	36	207	10	0	287	2	0	281
			15	8	252						
			20	3	278						
			25	0	324						

Table 2c. Results for the six-hump camelback function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	5	32	10	10	78	10	0	95	2	0	96
2	0	75	15	2	102						
			20	0	124						

Table 2d. Results for the Rastrigin function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	89	40	10	92	184	10	50	179	2	51	163
2	80	118	15	57	248	15	36	267	3	29	263
3	66	189	20	31	315	20	21	342	4	25	378
4	61	260	25	12	387	25	12	432	5	10	475
5	56	333	30	5	466	30	5	530	6	3	545
6	50	408	40	1	564	40	0	697	7	1	644
7	45	483	50	3	699	50	2	864	8	1	752
8	38	552									
9	36	623									
10	32	693									
11	31	760									
12	28	830									
13	24	904									
14	22	972									
15	21	1042									

Table 2e. Results for the Shekel function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	59	154	10	100	—	10	38	309	2	23	486
2	39	359	20	64	489	20	18	526	3	6	714
3	19	584	30	22	651	30	5	739	4	8	956
4	7	786	40	16	767	40	4	962	5	1	1150
5	4	994	50	12	920	50	4	1183	6	1	1403
			60	10	1082	60	0	1385	7	0	1600
			70	6	1207						
			80	4	1376						
			90	5	1518						
			100	4	1677						

dashed line indicates the SCE1 algorithm; and the solid line indicates the SCE2 algorithm. The MSX algorithm curve represents its performance for increasing numbers of restarts. The CRS2 and SCE1 algorithm curves represent their performance for increasing sample sizes. The SCE2 algorithm

Table 2f. Results for the Hartman function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	25	307	10	100	—	10	54	334	1	32	329
2	4	812	20	98	404	20	36	354	2	45	415
3	3	1258	30	81	646	30	44	433	3	41	608
4	2	1730	40	45	901	40	50	525	4	40	756
5	0	2204	50	31	999	50	45	612	5	41	971
			60	31	940	60	41	693	6	43	1125
			70	29	1041	70	48	801	7	26	1329
			80	23	1157	80	44	879	8	20	1603
			90	21	1212	90	46	994	10	22	1982
			100	21	1332	100	51	1088	12	16	2306
			110	22	1520	110	40	1186	15	16	2946
			120	22	1627	120	40	1300	20	8	3984
			150	26	1950	150	50	1587	25	4	4989
			200	28	2634	200	46	2126			
			350	31	4608	350	30	3979			
			500	31	6944	500	18	6173			

Table 2g. Results for the Griewank function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	100	—	15	11	684	15	100	—	2	14	1977
			20	16	1029	20	91	1484	3	1	2465
			30	30	1684	30	45	2242	4	0	3070
			40	33	2430	40	11	2465			
			50	27	3099	50	31	2601			
			60	32	3698	60	1	2940			
			70	44	4211	70	1	3230			
			80	34	4564	80	0	3569			
			90	24	5261						
			100	22	5580						
			110	15	6006						
			120	28	6652						
			130	15	6861						
			140	21	7376						
			150	8	7732						
			200	10	9131						

curve represents its performance for increasing numbers of complexes. Note that, in most of the plots, we see the expected behavior that effectiveness increases (failures decrease) and efficiency decreases (average function evaluations increase) as we increase the sample size s (or number of complexes

Table 2h. Results for the SIXPAR function.

MSX			CRS2			SCE1			SCE2		
Starts	NF	AFE	Points	NF	AFE	Points	NF	AFE	Complexes	NF	AFE
1	65	903	—	—	—	10	98	628	1	91	629
2	56	1507	30	93	1179	30	29	1107	2	21	1104
3	51	1764	40	51	1673	40	22	1341	3	15	1359
4	46	1999	50	11	2208	50	12	1611	4	5	1697
5	40	2292	60	5	2000	60	12	1826	6	2	2397
6	29	3282	90	4	2269	90	4	2570	8	1	3133
7	19	4259	120	4	2745	120	1	3293			
8	11	5249									
9	8	6263									
10	7	7189									

or restarts), but that the trade-off is not linear. In some cases, a zero failure rate was obtained with the initial set of trials and, therefore, further trials were not necessary. These results appear as isolated points in Figs. 3b and 3c. Also, for the Griewank function, the MSX algorithm failed completely, with a 100% failure rate on the initial start and, therefore, does not appear on the plot. The important features of the results may be summarized as follows:

(a) None of the algorithms is clearly superior on all the test problems.

(b) The MSX procedure is the least robust, with its performance varying dramatically over the different functions. It performs extremely well on the Rosenbrock, camelback, and Hartman functions, fails on the Griewank function, and performs from average to poorly on the remaining functions. As might be expected, the conditions under which it performs well are those in which the region (volume) of attraction of the global optimum is relatively large.

(c) The CRS2 procedure performs variably. On the Rosenbrock, camelback, and Shekel functions, its performance is the worst among the algorithms. On the Griewank and Hartman functions, its performance does not improve in a consistent manner as the sample size is increased. Strangely, these functions represent two completely different surface characteristics. The Hartman function has four major regions of attraction with no small local optima in the major regions. The Griewank function has one major region of attraction containing more than 1000 small local optima.

(d) The SCE1 and SCE2 procedures perform consistently well on all eight test functions.

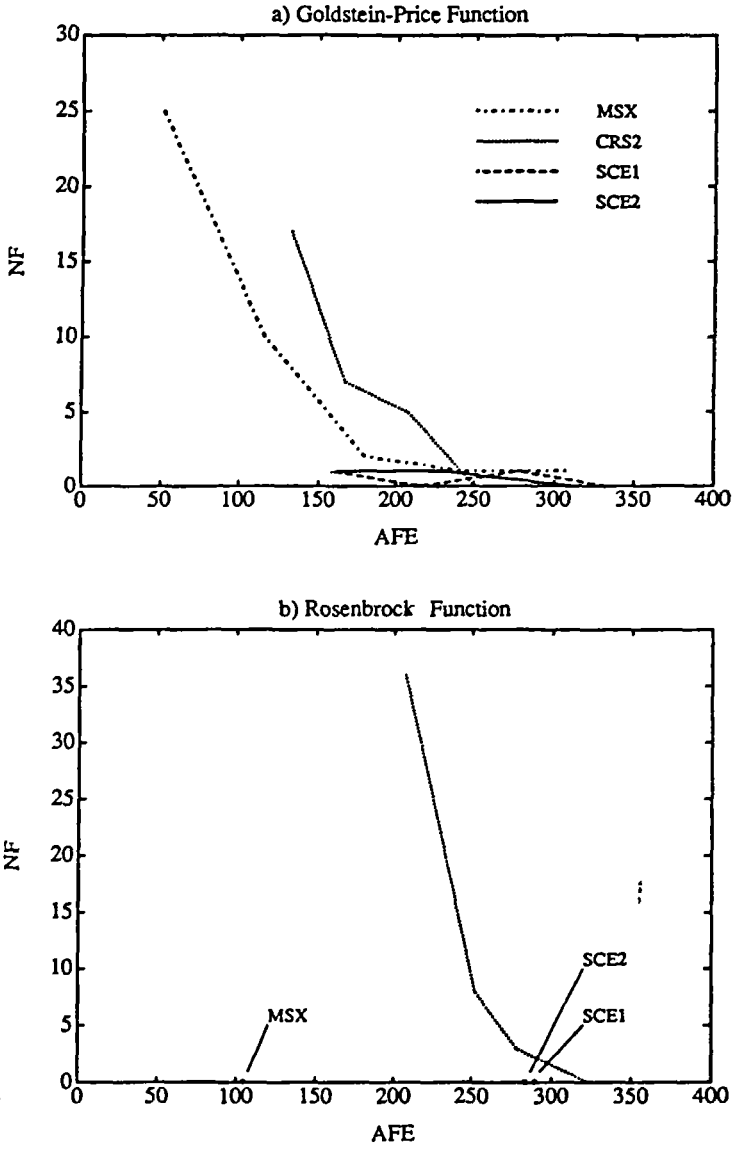
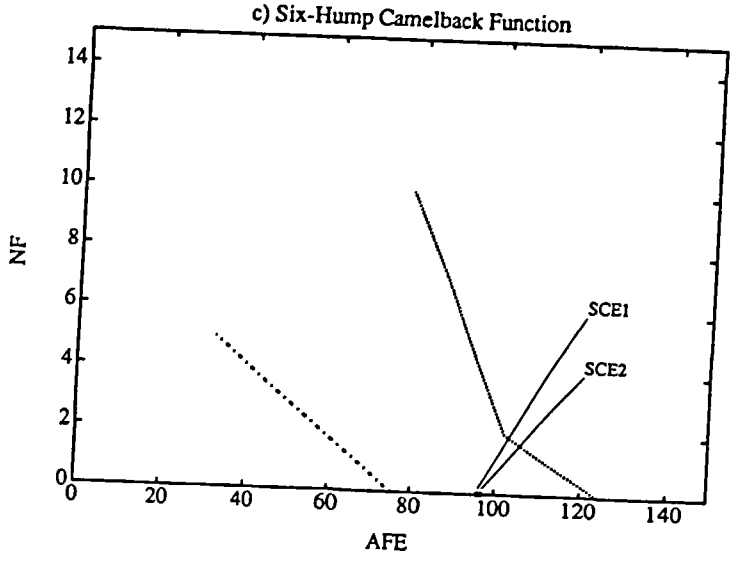


Fig. 3. Effectiveness (NF) vs efficiency (AFE) plots for the test functions.



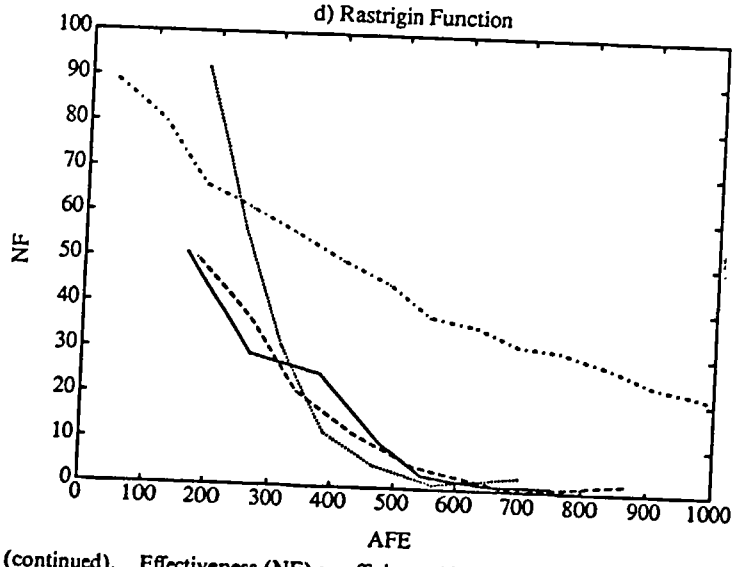
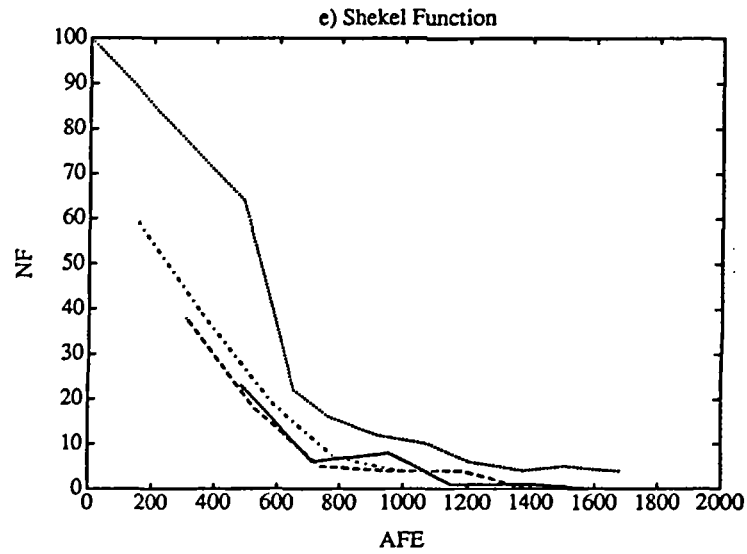


Fig. 3 (continued). Effectiveness (NF) vs efficiency (AFE) plots for the test functions.



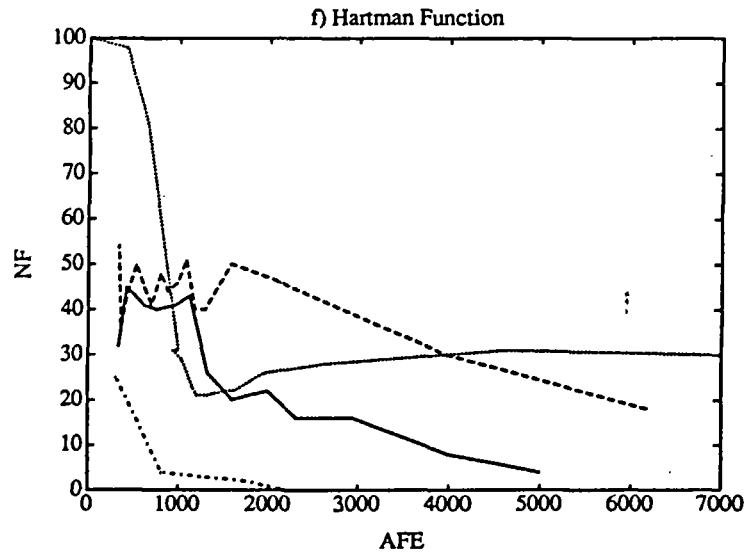


Fig. 3 (continued). Effectiveness (NF) vs efficiency (AFE) plots for the test functions.

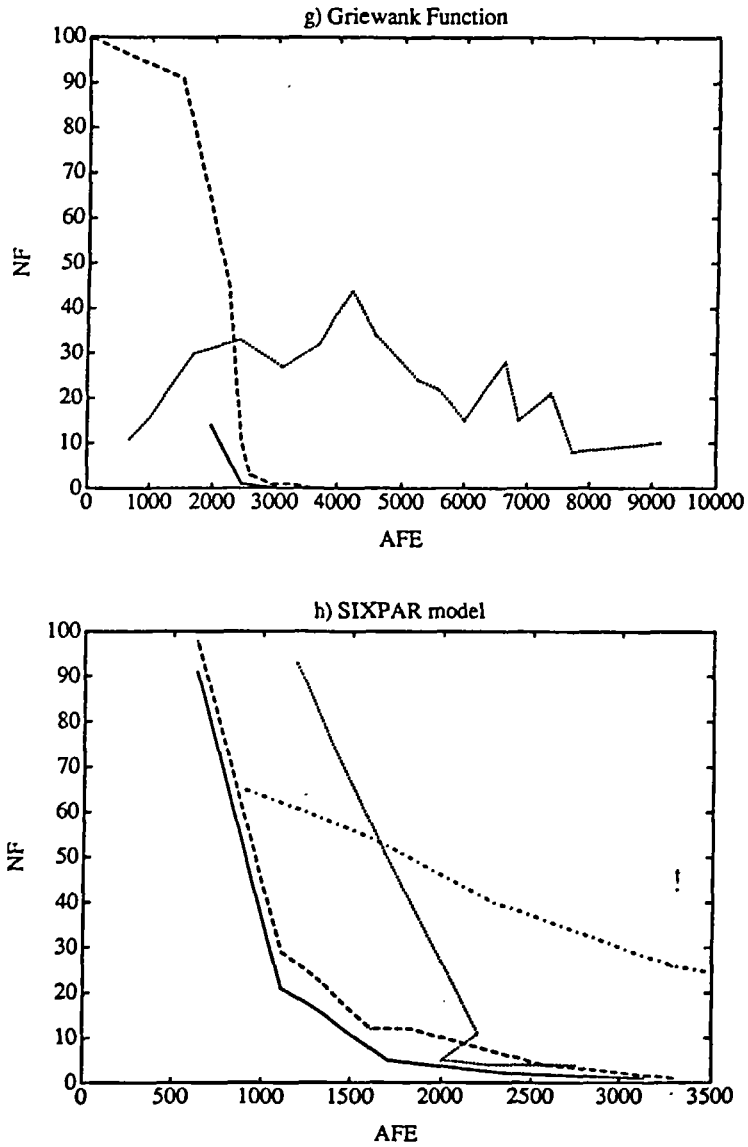


Fig. 3 (continued). Effectiveness (NF) vs efficiency (AFE) plots for the test functions.

(e) The SCE2 procedure is on the average the best algorithm. It performs the best on the SIXPAR (watershed model) and Griewank problems, is the only one of the complex evolution procedures (CRS2, SCE1, and SCE2) that appears to be reasonably effective on the Hartman function, and performs relatively well on all the others.

7. Discussion

Given a specific problem, there is likely to be a particular procedure that is most effective and efficient at solving it. In general, however, one often has little a priori knowledge about the structure of the response surface and, therefore, a procedure that is robust and efficient is desirable (Dixon and Szegö, Ref. 2). Viewed in this light, the SCE2 procedure clearly shows promise, displaying consistent performance over the range of problems tested. It should be pointed out that none of these test problems, except perhaps the SIXPAR problem (where the SCE2 procedure is clearly the best), rigorously tests the capabilities of the SCE strategy. Further studies of the algorithm should include more difficult test problems, i.e., problems which have all five of the characteristics that complicate global optimization that are described in Section 1.

8. Appendix: Test Functions

Goldstein-Price Function ($n=2$):

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

$$-2 \le x_1, \quad x_2 \le 2.$$

The global minimum is equal to 0 and the minimum point is $(0, -1)$. There are four local minima in the region of interest.

Rosenbrock Function ($n=2$):

$$f(x_1, x_2) = 100(x_2-x_1^2)+(1-x_1^2)^2,$$

$$-5 \le x_1 \le 5, \quad -2 \le x_2 < 8.$$

The minimum value is 0 and the minimum point is at $(1, 1)$. There is only one minimum in the region of interest.

Six-Hump Camelback Function ($n=2$):

$$f(x_1, x_2) = 1.036285+4x_1^2-2.1x_1^4+(1/3)x_1^6+x_1x_2-4x_2^2+4x_2^4$$

$$-2 \le x_1 \le 2, \quad -1 \le x_2 \le 1.$$

The function is symmetric about the origin and has three pairs of local minima. The global minimum is 0 at (0.08983, −0.7126) and (−0.08983, 0.7126).

Rastrigin Function ($n=2$):

$$f(x_1, x_2) = 2 + x_1^2 + x_1^2 - \cos(18x_1) - \cos(18x_2),$$
$$-1 \le x_1, \quad x_2 \le 1.$$

The global minimum is 0 at (0, 0). There are more than 50 local minima in the region of interest, arranged in a lattice configuration.

Shekel-Function ($n=4$):

$$f(x) = 10.5364 - \sum_{i=1}^{m} \frac{1}{(x-a_i)^T(x-a_i)+c_i}$$
$$x = (x_1, x_2, x_3, x_4)^T, \quad a_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4})^T,$$
$$c_i > 0, \quad i = 1, \ldots, m,$$
$$0 \le x_j \le 10, \quad j = 1, \ldots, 4,$$

where a_i and c_i, $i=1, \ldots, m$ and $m=10$, are specified in Table 2. The global minimum is 0 at a point close to (4, 4, 4, 4). There are 10 local minima in the region of interest.

Hartman Function ($n=6$):

$$f(x) = 3.32 - \sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{n} \alpha_{ij}(x_j - p_{ij})^2\right],$$
$$x = (x_1, \ldots, x_6)^T, \quad p_i = (p_{i1}, \ldots, p_{i6})^T,$$
$$\alpha_i = (\alpha_{i1}, \ldots, \alpha_{i6})^T, \quad c_i > 0, \quad i = 1, \ldots, 4,$$
$$0 \le x_j \le 1, \quad j = 1, \ldots, n,$$

where $n=6$, and α_{ij}, c_i, p_{ij}, $i=1, \ldots, 4$ and $j=1, \ldots, 6$, are specified in Tables 3 and 4. The global minimum is 0 at (0.201, 0.150, 0.477, 0.275, 0.311, 0.657). There are four local minima in the region of interest.

Table 3. Values of a_i and c_i for the Shekel function.

i	a_{1i}	a_{2i}	a_{3i}	a_{4i}	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	3	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
10	7	3.6	7	3.6	0.5

Table 4. Values of a_{ij} and c_i for the Hartman function.

i	a_{1i}	a_{2i}	a_{3i}	a_{4i}	a_{5i}	a_{6i}	c_i
1	10.00	3.00	17.00	3.50	1.70	8.00	1.0
2	0.05	10.00	17.00	0.10	8.00	14.00	1.2
3	3.00	3.50	1.70	10.00	17.00	8.00	3.0
4	17.00	8.00	0.05	10.00	0.10	14.00	3.2

Table 5. Values of p_{ij} for the Hartman function.

i	p_{1i}	p_{2i}	p_{3i}	p_{4i}	p_{5i}	p_{6i}
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

Griewank Function ($n=10$):

$$f(x)=\sum_{i=1}^{n}\frac{x_i^2}{d}-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})+1,$$

$$-600\leq x_i\leq 600, \qquad i=1,\ldots,10, \qquad d=600.$$

The global minimum is 0 and is at the origin. There are several thousand local minima in the region of interest.

SIXPAR Hydrologic Model ($n=6$). The SIXPAR model is a research version of the SMA-NWSRFS model used by the U.S. National Weather Service for flood forecasting. Rainfall data measured over a watershed are telemetered to a central processing computer, where the SMA-NWSRFS model is used to generate estimates of riverflow volume at specific downstream points. The model is highly nonlinear in the parameters, having several threshold-type parameters and reservoir depletion-rate parameters that must be estimated using historical data. A description of the model appears in Gupta and Sorooshian (Ref. 5). A discussion of the difficulties involved in the optimization of the model parameters appears in Duan, Sorooshian, and Gupta (Ref. 7). The computer code and data are available from the authors on request.

References

1. Dixon, L. C. W., and Szegö, G. P., Editors, *Toward Global Optimization 2*, North-Holland, Amsterdam, Holland, 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, Amsterdam, Holland, pp. 1–15, 1978.
3. TORN, A., and ZILINSKAS, A., *Global Optimization*, Springer-Verlag, Berlin, Germany, 1989.
4. SOROOSHIAN, S., GUPTA, V. K., and FULTON, J. L., *Evaluation of Maximum-Likelihood Parameter Estimation Techniques for Conceptual Rainfall-Runoff Models: Influence of Calibration Data Variability and Length on Model Credibility*, Water Resources Research, Vol. 19, pp. 251–259, 1983.
5. GUPTA, V. K., and SOROOSHIAN, S., *Uniqueness and Observability of Conceptual Rainfall-Runoff Model Parameters: The Percolation Process Examined*, Water Resources Research, Vol. 19, pp. 269–276, 1983.
6. GUPTA, V. K., and SOROOSHIAN, S., *The Automatic Calibration of Conceptual Catchment Models Using Derivative-Based Optimization Algorithms*, Water Resources Research, Vol. 21, pp. 473–486, 1985.
7. DUAN, Q., SOROOSHIAN, S., and GUPTA, V. K., *Effective and Efficient Global Optimization for Conceptual Rainfall-Runoff Models*, Water Resources Research, Vol. 28, pp. 1015–1031, 1992.
8. GOMULKA, J., *Deterministic Versus Probabilistic Approaches to Global Optimization*, Toward Global Optimization 2, Edited by L. C. W. Dixon and G. P. Szegö, North-Holland, Amsterdam, Holland, pp. 19–29, 1978.
9. BECKER, R. W., and LAGO, G. V., *A Global Optimization Algorithm*, Proceedings of the 8th Allerton Conference on Circuits and Systems Theory, Monticello, Illinois, pp. 3–12, 1970.
10. TÖRN, A., *A Search Clustering Approach to Global Optimization*, Toward Global Optimization 2, Edited by L. C. W. Dixon and G. P. Szegö, North-Holland, Amsterdam, Holland, pp. 49–62, 1978.
11. RINNOOY-KAN, A. H. G., and TIMMER, G. T., *Stochastic Global Optimization Methods, Part 1: Clustering Methods*, Mathematical Programming, Vol. 39, pp. 27–56, 1987.
12. 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, Amsterdam, Holland, pp. 71–84, 1978.
13. PRICE, W. L., *Global Optimization by Controlled Random Search*, Journal of Optimization Theory and Applications, Vol. 40, pp. 333–348, 1983.
14. PRICE, W. L., *Global Optimization Algorithms for a CAD Workstation*, Journal of Optimization Theory and Applications, Vol. 55, pp. 133–146, 1987.
15. HOLLAND, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.
16. NELDER, J. A., and MEAD, R., *A Simplex Method for Function Minimization*, Computer Journal, Vol. 7, pp. 308–313, 1965.