Experiences in using evolutionary and non-evolutionary optimization methods in models calibration

Dimitri P. Solomatine

UNESCO-IHE Institute for Water Education, Delft, The Netherlands, *sol* at *ihe.nl*

Abstract:- Modelling and decision making related to environmental problems need adequate optimization methods and tools. In case the objective function to be minimized is not known analytically and no assumption can be made about the number of its extrema, gradient-based methods are inapplicable and direct multi-extremum (global) methods must be used. Apart from the popular evolutionary and genetic algorithms, other methods appear to be at least as effective and efficient. Nine algorithms were implemented in the GLOBE global optimization system, and they are compared in terms of effectiveness (accuracy), efficiency and reliability on several benchmark and hydrologic modelling problems.

Keywords: global optimization, evolutionary and genetic algorithms, adaptive cluster covering, models calibration.

1. INTRODUCTION

Many aspects of environmental modelling and management are closely linked to the notion of optimization. Examples are problems of resources allocation, reservoir optimization [Solomatine and Torres, 1996], groundwater remediation [Maskey et. al., 2002], water distribution [Savic and Walters, 1997] and models calibration ([Wang, 1991; Solomatine, 1995, 1998, 1999; Franchini and Galeati 1997]. Models should closely mimic reality (or the modelled system), so this "closeness" should be minimized. Typically this is achieved optimizing (calibrating) both model structure and its parameters.

Traditional (linear and non-linear, gradient-based) optimization can be quite successfully applied when the problem or a model allows for analytical descriptions. However this is often not the case: problems and models are encapsulated in software and the analytical calculation of derivatives is impossible so that direct optimization methods like the methods of randomized search should be used. Also the assumption of single-extremality typically cannot be tested, so the solution methods should belong to the class of *global (multi-extremum) optimization* (GO) methods (Figure 1).

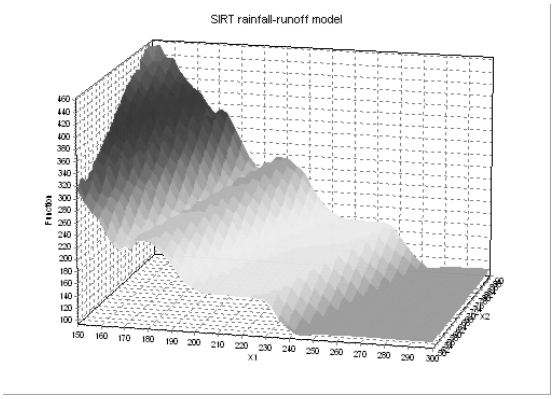


Figure 1. An example of a (multi-extremum) model error function which is to be minimized in the process of calibration

A global minimization problem can be formulated as follows: find an *optimizer* vector $\mathbf{x}^*$ such that generates a minimum of the objective function $f(\mathbf{x})$ where $\mathbf{x} \in X$ and is subject to some constraints. In calibration often a simple box constraint is used: $X=\{\mathbf{x} \in \mathrm{R}^n: \ a_i \le x_i \le b_i, \ i=1,..,n\}$. This *constrained optimization* problem can be transformed to an *unconstrained optimization* problem by introducing the *penalty* function with a high value outside the specified constraints. In this paper the optimization problem is considered to be

a single-objective one. Many problems are formulated, however, as multi-objective ones [Deb et al., 2002; Barreto et al., 2006].

During the last decade the development of evolutionary and genetic algorithms (EGA) has lead to successful solution of many optimization problems that previously were not even posed. The success of EGAs is fully deserved and can be explained by their methodological appeal, relative simplicity, robustness and the existence of a well-organized community. The recent developments in hybrid EAs, in particular, memetic algorithms, lead to improvements in the effectiveness of this class of algorithms.

Among the non-derivative methods for GO EAs and GAs seem to be the most popular ones. Publications related, for example, to water-related applications started to appear already in the 1990s [Wang 1991; Cieniawski et al., 1995; Solomatine 1995, Savic and Walters 1997; Franchini and Galeati 1997], and by now EGAs have become a widely-spread technology. One can see, however, that quite often the use of EGAs is not supported by the proper analysis and not always justified.

The motivation for this paper was sharing the experience with various GO methods that were used to solve problems related to water and environment.

2. METHODS OF GLOBAL OPTIMIZATION

The strategies employed in the most GO algorithms try to perform the two conflicting tasks:

- exploring the search space – generating points in the unexplored regions;
- exploiting the best solutions – using the best points (found so far) to find even better points.

The reader is referred, e.g., to Torn & Zilinskas 1989, and Pintér 1995 for an extensive coverage of various methods. It is possible to distinguish the following groups: set (space) covering techniques; random search methods, including EGAs; methods based on multiple local searches (multistart) based on gradients assessments; multistart based on clustering; other methods (simulated annealing, trajectory techniques, tunneling approach, etc.); hybrid methods combining several approaches. An overview of different methods was given in (Solomatine 1998, 1999, 2005). Here only EGA, multistart and ACCO will be briefly described.

Evolutionary and genetic algorithms. There are various versions of EGAs varying in the way crossover, selection and construction of the new population is performed. In EAs, for example, mutation of coordinates is performed with respect to corresponding variances of a certain *n*-dimensional normal distribution, and various versions of recombination are introduced.

Multistart and clustering. The basic idea of the family of *multistart* methods is to apply a search procedure several times, and then to choose an assessment of the global optimizer. The *region (area) of attraction* of a local minimum $\mathbf{x}^*$ is the set of points in X starting from which a given local search procedure P converges to $\mathbf{x}^*$.

For the global optimization tool GLOBE used in the present study, we developed two multistart algorithms – *Multis* and *M-Simplex*. They follow the following scheme: (1) generate a set of N random points and evaluate f at these points; (2) reduce the initial set by choosing p best points (with the lowest f_i); (3) launch local search procedures from each of p points. The best point reached is the minimizer assessment. In *Multis*, at step 3 the Powell-Brent local search (Press et al., 1991) is started. In *M-Simplex* the downhill simplex descent [Melder and Nead 1965] is used.

One of the versions of multistart used in global optimization is based on clustering, that is creating groups of mutually close points that hopefully correspond to regions of attraction [Torn & Zilinskas 1989]. The ACCO strategy developed by the author and covered below, also uses clustering as the first step, but it is followed by the global randomized search, rather than local search.

Adaptive cluster covering (ACCO) (Solomatine 1995, 1998, 1999) combines reduction, clustering and covering (Figure 2).

1. Clustering. Clustering is used to identify the most promising subdomains in which to continue the global search by active space covering.

2. Covering shrinking subdomains. Each subdomain is covered randomly. The values of the objective function are then assessed at the points drawn from some distribution. Covering is repeated and each time the subdomain is progressively reduced in size.

3. Adaptation. Adaptive algorithms update their behaviour depending on the new information revealed about the problem. In ACCO, adaptation is formed by *shifting* the subregion of search, *shrinking* it, and changing the density (number of points) of each covering – depending on the previous assessments of the global minimizer.

4. Periodic randomization. Any strategy of randomized search may miss a promising region for search. In order to reduce this danger, the

problem is solved several times with the re-randomization of the initial population.

Depending on the implementation of each of these principles, it is possible to generate a family of various algorithms, suitable for certain situations, e.g. with non-rectangular domains (hulls), non-uniform sampling and with various versions of cluster generation and stopping criteria. Figure 1 shows the example of an initial sampling, and iterations 1 and 2 for one of the clusters in a two dimensional case.

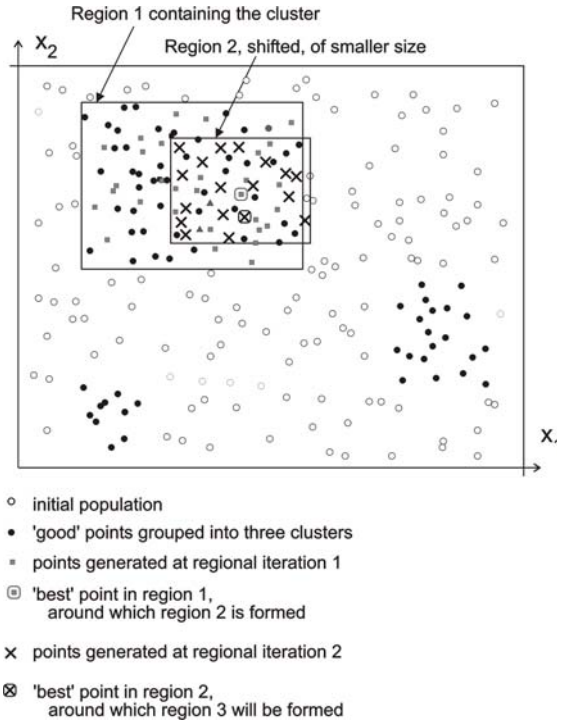


Figure 2. Adaptive cluster covering

ACCOL algorithm is the combination of ACCO with the multiple *local searches*:

1. *ACCO phase*. ACCO strategy is used to find several regions of attraction, represented by the promising points that are close ('potent' points). The potent set P_1 is formed by taking one best point from each cluster found during the progress of ACCO. After ACCO stops, the set P_1 is reduced to P_2 by leaving only several m (1...4) best points which are also distant from each other;

2. *Local search (LS) phase*. An accurate algorithm of local search is started from each point of P_2 (multistart) to find accurately the minimum; a version of the Powell-Brent search is used.

Experiments have shown, that in comparison to traditional multistart, ACCOL brings significant economy in the number of function evaluations.

ACD algorithm [Solomatine, 1998] is also a random search algorithm combining ACCO with the downhill simplex descents (DSD) of Nelder & Mead [1965]. Its basic idea is to identify the area around the possible local optimizer by using clustering, and then to apply covering and DSD.

ACDL algorithm combining ACD with the multiple local searches has been developed as well.

This author developed GLOBE software (*http://www.data-machine.com*) incorporating nine GO algorithms. GLOBE can be configured to use an external program as a supplier of the objective function values. The number of independent variables and the constraints imposed on their values are supplied by the user in the form of a simple text file. Currently, GLOBE includes the following algorithms (with variations, nine):

- controlled random search: CRS2 [Price 1983] and CRS4 [Ali & Storey 1994];
- Simple GA;
- multistart algorithms: Multis and M-Simplex;
- adaptive cluster covering (ACCO) and ACCO with local search (ACCOL);
- adaptive cluster descent (ACD) and ACD with local search (ACDL).

3. COMPARING ALGORITHMS

Our experience of using GO algorithms includes:

- traditional benchmark functions used in GO with known global optima [Dixon & Szegö 1978; Duan et al. 1993];
- optimization of water-related problems: dynamic programming for reservoir optimization [Lee, 1997]; groundwater remediation [Maskey et al 2002]; optimization of pipe networks [Abebe & Solomatine 1998];
- calibration of various models: lumped hydrological model [Solomatine 1995], 2D free-surface hydrodynamic model [Constantinescu 1996]; hydrologic model [Solomatine, 1998]; distributed groundwater model (Solomatine et al., 1999); electrostatic mirror model; ecological model of plant growth.

GLOBE uses the implementation of a Simple GA [Michalewicz, 1999]. Fitness rank elitist selection is used together with a complex stopping rule preventing premature termination. Several versions of GAs and sets of parameters were compared, and the one that performed the best was chosen for this study. This is a variant of GA with the 'fitness rank' selection, one-point crossover,

15-bit coding of variables, bit mutation, preservation of the best points discovered so far. To prevent redundant re-evaluations, in each generation checks are made for the appearance of repetitive points.

The most comprehensive experiments with all 9 algorithms were set up for the standard benchmark problems [Dixon & Szegö 1978, Duan et al. 1993]. The size of this paper does not allow to present all the results (see [Solomatine, 1998, 1999, 2005]; *www.ihe.nl/hi/sol/p_jogo/allplots.htm*). Figure 2 shows two typical examples of the process of minimization (averaged on 5 runs).

Performance indicators investigated were:

- *effectiveness* (how close the algorithm gets to the global minimum);
- *efficiency* (running time) of an algorithm measured by the number of function evaluations needed;
- *reliability* (robustness) of the algorithms measured by the number of successes in finding the global minimum, or at least approaching it sufficiently closely.

Effectiveness and efficiency. For the functions of 2 variables, *ACCOL*, *CRS4* and *M-Simplex* are the most efficient. With the functions of higher dimensions, *ACCOL* and *CRS4* again performed best, and had a similar performance, but *M-Simplex* was the worst with all Shekel 4-variable functions. With other benchmark functions it was a bit better than *ACCOL* and *CRS4*. *ACDL* was on average the third best in performance after *ACCOL* and *CRS4*, being a 'slow starter'. However, on some runs *ACDL* showed very high efficiency. *GA* is the least efficient method, and is also ineffective with all Shekel functions. *Multis* and *CRS2* are both effective, reaching the global minimum in most cases, but much slower than other algorithms.

Reliability (robustness). Reliability can be measured as the number of successes in finding the global minimum with the predefined accuracy. No random search algorithm can be 100% reliable. For the most functions of two variables most algorithms were quite reliable (with the exception of GA which often converged prematurely). With the functions of three and more variables *CRS2* and *Multis* algorithms appeared to be the most reliable but were the least efficient. *ACDL* was not always reliable even though it showed high efficiency on some runs. In most cases, the found minimizer estimate is normally quite close to the global minimum (*GA* that was failing more often than others).

4. DISCUSSION

Algorithms which are permanently oriented towards the whole function domain have to perform more function evaluations, that is, have low efficiency (*CRS2* and *Multis*). *ACDL* on some runs has shown high efficiency but not the reliability.

The observed lower efficiency of *GA* can be partly attributed to the fact that the population size and mutation probabilities were not optimized (whereas different types of crossover and selection strategies were tried.)

There are however, deeper reasons why GA is not the fastest algorithm. In comparison with the strategy used, for example, in ACCO, GA uses an operation that is not necessarily optimal: the 'crossover' (exchange of some of the parents' coordinate values) often leads to the redundant evaluations of the 'offspring' in the search space quite far from their (highly fit) parents, and hence normally with lower fitness. In fact, cutting through the bit string representing variable x_i leads to the new offspring that has effectively random value of x_i.

The question of usefulness of such crossover operator can be posed. What is the main role of crossover – exploration, or exploitation? Crossover helps in exploring the search space (since this typically leads to a random jump in direction of x_i). We claim however that using this operator is not the best way to exploit the best solutions since the high fit of the offspring and the parents are not necessarily inherited by many of their offspring. It is the selection operator, rather than the crossover, that ensures the effective exploitation of the good solutions found so far.

Evolution is a robust way of pushing the fit individuals forward, but there are no reasons to assume that it is the fastest or most efficient way of finding the fittest organism. Many processes in nature, are redundant since the "objective" of nature is a reliable reproduction and evolution, and reliability is achieved through the high redundancy which is typically in conflict with efficiency.

Many other GO algorithms do not suffer from the mentioned deficiency of Simple GA. ACCO separates the tasks of exploring the space (initial random search and covering) and exploiting the best solutions (by reduction, and increased density of covering close to the best points). The relatively higher efficiency of *ACCOL* and *CRS4* can be also explained by their orientation towards smaller search domains which is especially efficient for high dimensions. *Multis* follows a much more direct strategy of seeking the points with the lower

function value, requiring, however, more function evaluations.

The mentioned features of the Simple GAs were gradually recognized by the research community, and during the last 5-10 years a number of methods that follow an idea of combining the appealing features of GA with the best features of other methods were developed (see, e.g., Yao et al. [1999]; Oh [2004]). Among such hybrid GAs (or EAs) there is a group that is particularly relevant to the topic of this paper – that of memetic algorithms (MA), evolutionary algorithms that include local search (for an overview, see Eiben and Smith [2003]. Indeed, the local search used in MA is basically a version of covering (used in ACCO) or descent based on the assessment of gradient, like it is done in Multix described above.

One of the important differences between ACCO and hybrid GA or MA is that ACCO does not explicitly use the crossover operator. This however does not mean that ACCO cannot be seen as following an evolutionary approach. It is a population-based algorithms and the generation of points in the proximity of good points (covering) can be interpreted as *asexual* reproduction (reproduction by division), and adaptation (shifting the domains) plays the role of selection. Note that any iterative optimization scheme implements a process that can be treated as evolution: with every population the generated instances obtain higher and higher fit, the weakest perish and the fit survive.

5. CONCLUSIONS

1. Using GO techniques are often the only way to solve complex optimization problems where the objective function is calculated by software. Among the GO algorithms compared, *ACCOL* and *CRS4* showed the highest effectiveness, efficiency and reliability. *M-Simplex* performs well with the functions of low dimension but in higher dimensions it often converges prematurely to a local minimum.

Simple GA, *CRS2*, and *Multis* provide reasonable solutions as well. However, all of them require more function evaluations. Our other experiments [Abebe and Solomatine, 1998] however, show that for certain classes of problems with highly discrete variables, *GA*, due to its inherently discrete nature, can actually be more accurate than other algorithms built originally for continuous variables (being still less efficient than for example, *ACCO*).

2. The choice between the various methods of global optimization may depend on the type of problem, and more research is needed to compare the reportedly efficient methods like simulated annealing, topological multilevel linkage, shuffled simplex evolution [Duan et al., 1993] and others (see, e.g., Neumaier [2005]. The best results can probably be achieved by adaptation of an algorithm in relation to the information about the response surface (e.g., Fogel et al. [2001]), or by *structural adaptation*, that is, switching in the process of search between different algorithms;

3. It is suggested that many of GO algorithms, e.g. *ACCO* and *CRS*, can be seen as evolutionary algorithms. Such attribution can be quite fruitful; one of the possibilities is the extension of hybrid EGAs and memetic algorithms with covering, reduction and adaptation – the approaches that ensure the high efficiency of the *ACCO* algorithm.

6. REFERENCES

Abebe A.J., and Solomatine D.P. Application of global optimization to the design of pipe networks. *Proc. 3rd Int. Conf. on Hydroinformatics*, Copenhagen, 989-995, 1998.

Ali, M.M., and Storey, C. Modified controlled random search algorithms. *Intern. J. Computer Math*., 53, 229-235, 1994.

Barreto, W., Price, R.K., Solomatine, D.P., and Vojinovic, V. Approaches to Multi-Objective Multi-Tier Optimization in Urban Drainage Planning. *Proc. 7th Int. Conference on Hydroinformatics*, Nice, 2006.

Cieniawski, S.E, Eheart, J.W. and Ranjithan, S. Using genetic algorithms to solve a multiobjective groundwater monitoring problem. *Water Resour. Res*., 31(2), 399-409, 1995.

Constantinescu A. (1996). Calibration of hydrodynamic numerical models using global optimization techniques. M.Sc. thesis HH262, IHE, Delft, 85p.

Deb, K, Pratap, A., Agarwal, S, and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. evolutionary computation, 6, 182-197, 2002.

Dixon, L.C.W. and Szego, G.P. (eds.). Towards global optimization, North-Holland, Amsterdam, 1978, 472p.

Duan, Q., Gupta, V., and Sorooshian, S. Shuffled complex evolution approach for effective and efficient global minimization. *J. of Optimiz. Theory Appl.*, 76 (3), 501-521, 1993.

Eiben, A.E. and Smith, J.E. Introduction to Evolutionary Computing. Springer, 2003.

Fogel D.B., Fogel G.B., and Ohkura K. Multiple-vector self-adaptation in evolutionary algorithms, *Biosystems*, 61(2), 155-162, 2001.

Franchini, M., and Galeati, G. Comparing several genetic algorithm schemes for the calibration of conceptual rainfall-runoff models. *Hydrol. Sci. J.*, 42 (3), 357-379, 1997.

GLOBE: a global optimization tool. Web: http://www.data-machine.com, 2006.

Lee H. Optimal reservoir operation for flood control using a hybrid modelling approach. M.Sc. thesis HH319, IHE, Delft, 1997.

Maskey, S. Jonoski, A. & Solomatine, D.P. Groundwater remediation strategy using global optimization algorithms. *ASCE Journal of Water Resources Planning and Management*, 128(6), 431-440, 2002.

Michalewicz, Z. *Genetic algorithms + data structures = evolution programs*. Springer, Berlin, 1999.

Nelder, J.A, and Mead, R. A simplex method for function minimization. *Computer J.*, 7(4), 308-313, 1965.

Neumaier's web page on global optimization: solon.cma.univie.ac.at/ ~neum/glopt.html, 2006.

Oh, I.-S., Lee, J.-S., and Moon, B.-R. Hybrid Genetic Algorithms for Feature Selection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(11), 1424-1437, 2004.

Pintér, J. Global optimization in action. Kluwer, Amsterdam, 1995.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. Numerical recipes in Pascal. Cambridge University Press, Cambridge, 1990, 759p.

Price, W.L. Global optimization by controlled random search. *J. Optimiz. Theory Appl.*, 40, 333-348, 1983.

Savic, D.A. & Walters, G.A. Genetic algorithms for least-cost design of water distribution networks. *ASCE J. of Water Res. Planning and Mngt.*, 123(2), 67-77, 1997.

Solomatine, D.P. The use of global random search methods for models calibration, Proc. XXVIth Congress of the International Association for Hydraulic Research (IAHR), London, September, vol. 1, 224-229, 1995.

Solomatine, D.P., and A. Avila Torres. Neural Network Approximation of a Hydrodynamic Model in Optimizing Reservoir Operation. *Proc. 2nd Int. Conference on Hydroinformatics*, Zurich, Switzerland, 201-206, 1996.

Solomatine, D.P. Genetic and other global optimization algorithms - comparison and use in calibration problems. *Proc. 3rd Int. Conf. on Hydroinformatics*, Copenhagen, 1021-1028, 1998.

Solomatine, D.P., Y.B.Dibike and N. Kukuric. Automatic calibration of groundwater models using global optimisation, *Hydrological Sciences J.*, 44(6), 879-894, 1999.

Solomatine, D.P. Two strategies of adaptive cluster covering with descent and their comparison to other algorithms. *J. Global Optimization*, 14 (1), 55-78, 1999.

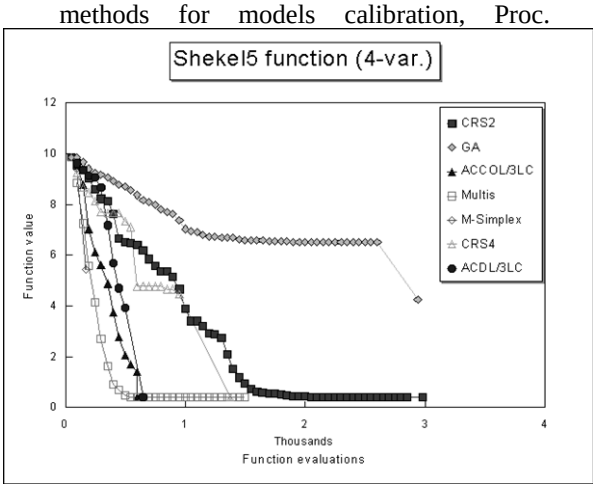
Solomatine, D.P. Adaptive cluster covering and evolutionary approach: comparison, differences and similarities. *Proc. Congress on Evolutionary Computing*, Edinburgh, 2005.

Törn, A., and Žilinskas, A. *Global optimization*. Springer-Verlag, Berlin, 1989, 255pp.

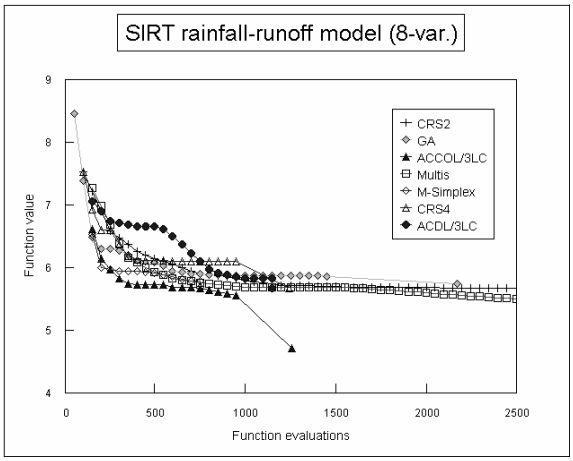
Törn, A. & Viitanen, S. Topographical Global Optimization using pre-sampled points. *J. of Global Optimization*, 5, 267-276, 1994.

Wang, Q.J. The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. *Water Resour. Res.*, 27(9), 2467-2471, 1991.

Yao, X., Liu, Y., and Lin, G. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3:2, 82-102, 1999.



(a)



(b)

Figure 3. Typical examples of the algorithms performance:
(a) Shekel5 function of (4 variables);(b) rainfall-runoff model error (8 variables).