Proc. 3rd Intern. Conference on Hydroinformatics, Copenhagen, Denmark, 1998. Balkema Publishers. pp.1021-1028

Genetic and other global optimization algorithms - comparison and use in calibration problems

D.P. Solomatine
International Institute for Infrastructural, Hydraulic and Environmental Engineering, P.O.Box 3015, 2601 DA, Delft, The Netherlands
Email: sol@ihe.nl

ABSTRACT: Many issues related to water resources require the solution of optimization problems. If the objective function is not known analytically, traditional methods are not applicable and multi-extremum (global) optimization (GO) methods must be used. In the present paper a brief overview of GO methods is given and nine of them are compared in terms of effectiveness (accuracy), efficiency (number of needed function evaluations) and reliability on several problems including two problems of model calibration. Two algorithms - adaptive cluster covering (ACCO) and controlled random search (CRS4) - show better performance than the popular genetic algorithm. The global optimization tool GLOBE used to perform the experiments can be downloaded from ***www.ihe.nl/hi***.

1. INTRODUCTION

Many issues related to water resources require the solution of optimization problems. These include reservoir optimization, problems of optimal allocation of resources and planning, calibration of models, and many others. Traditionally, optimization problems were solved using linear and non-linear optimization techniques which normally assume that the minimized function (objective function) is know in analytical form and that it has a single minimum. (Without a loss of generality we will assume that the optimization problem is minimization problem)

In practice, however there are many problems that cannot be described analytically and many objective functions have multiple extrema. In these cases it is necessary to pose multi-extremum (global) optimization problem (GOP) where the traditional optimization methods are not applicable, and other solutions must be investigated. One of these typical GOPs is that of automatic model calibration, or parameter identification. The objective function is then the discrepancy between the model output and the observed data, i.e. the model error, measured normally as the weighted RMSE. One of the approaches to solve GOPs that has become popular during the recent years is the use of the so-called genetic algorithms (GAs) (*Goldberg 1989, Michalewicz 1996*). A considerable number of publications related to water-resources are devoted to their use (*Wang 1991, Babovic et al. 1994, Cieniawski 1995, Savic & Walters 1997, Franchini & Galeati 1997*). (Evolutionary algorithms (EA) are variations of the same idea used in GAs, but were developed by a different school. It is possible to say that EAs include GAs as a particular case).

Other GO algorithms are used for solving calibration problems as well (*Duan et al., 1993, Kuczera 1997*), but GAs seem to be prefered. Our experience however, shows that many practitioners are unaware of the existence of other GO algorithms that are more efficient and effective than GAs. This serves as a motivation for writing this paper which has the following main objectives:

- to classify and briefly describe GO algorithms;
- to demonstrate the relative performance of several GO algorithms, including GAs, on a suite of problems, including model calibration;
- to give some recommendations to practitioners whose problem is formulated as a GOP.

2. APPROACHES TO SOLVING OPTIMIZATION PROBLEMS

A global minimization problem with box constraints is considered: find an *optimizer* x^* such that generates a minimum of the objective function $f(x)$ where $x \in X$ and $f(x)$ is defined in the finite interval (box) region of the n-dimensional Euclidean space: $X = \{x \in R^n: a \# x \# b\}$ (componentwise). 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 cases when the exact value of an optimizer cannot be found, we speak about its *estimate* and, correspondingly, about its *minimum estimate*.

Approaches to solving this problem depend on the properties of $f(x)$:

1. $f(x)$ is a single-extremum function expressed analytically. If its derivatives can be computed, then gradient-based methods may be used: conjugate gradient methods; quasi-Newton or variable metric methods, like DFP and BFGS methods (*Jacobs 1977*, *Press et al. 1991*). In certain particular cases, e.g. in the calibration of complex hydrodynamic models, if some assumptions are made about the model structure and/or the model error formulation, then there are several techniques available (like *inverse modelling*) that allow the speeding up of the solution (*Van den Boogaard et al., 1993*).

Many engineering applications use minimization techniques for single-extremum functions, but often without investigating wether the functions are indeed single-extremum (unimodal). They do recognize however, the problem of the Agood@ initial starting point for the search of the minimum. Partly, this can be attributed to the lack of the wide awareness of the engineering community of the developments in the area of global optimization.

2. $f(x)$ is a single-extremum function which is not analytically expressed. The derivatives cannot be computed, and *direct search* methods can be used such as *Nelder & Mead 1965*.

3. No assumptions are made about the properties of $f(x)$, so it is a multi-extremum function which is not expressed analytically, and we have to talk about *multi-extremum* or *global* optimization.

Most calibration problems belong to the third category of GO problems. At certain stages the GO techniques may use the single-extremum methods from category 2 as well.

3. MAIN APPROACHES TO GLOBAL MINIMIZATION

The reader is referred to *Torn & Žilinskas 1989*, *Pintér 1995* for an extensive coverage of various methods. It is possible to distinguish the following groups:

- set (space) covering techniques;
- random search methods;
- evolutionary and genetic algorithms (can be attributed to random search methods);
- methods based on multiple local searches (multistart) using clustering;
- other methods (simulated annealing, trajectory techniques, tunneling approach, analysis methods based on a stochastic model of the objective function).

Several representatives of these groups are covered below.

Set (space) covering methods. In these the parameter space X is covered by N subsets $X_1,...,X_N$, such that their union covers the whole of X. Then the objective function is evaluated in N representative points $\{x_1, ..., x_N\}$, each one representing a subset, and a point with the smallest

function value is taken as an approximation of the global value. If all previously chosen points $\{x_1, ..., x_k\}$ and function values $\{f(x_1), ..., f(x_k)\}$ are used when choosing the next point x_{k+1}, then the algorithm is called a *sequential (active) covering* algorithm (and *passive* if there is no such dependency). These algorithms were found to be inefficient.

The following algorithms belong to the group of random search methods.

Pure direct random search (uniform sampling). N points are drawn from a uniform distribution in X and f is evaluated in these points; the smallest function value is the minimum f^* assessment. If f is continuous then there is an asymptotic guarantee of convergence, but the number of function evaluations grows exponentially with n. An improvement is to make the generation of evaluation points in a sequential manner taking into account already known function values when the next point is chosen, producing thus an *adaptive random search* (*Pronzato et al. 1984*).

Controlled random search (CRS) is associated with the name of W.L.Price who proposed several versions of an algorithm where the new trial point in search (parameter) space is generated on the basis of a randomly chosen subset of previously generated points; the widely cited method is *CRS2 (Price 1983).* At each iteration, a simplex is formed from a sample and a new trial point is generated as a reflection of one point in the centroid of the other points in this simplex. If the worst point in the initially generated set is worse than the new one, it is replaced by the latter. The ideas of CRS algorithms have been further extended by *Ali and Storey 1994a* producing CRS4 and CRS5. In *CRS4* if a new best point is found, it is ›rewarded‹ by an additional search around it by sampling points from the beta-distribution. This method is reportedly very efficient.

Evolutionary strategies and genetic algorithms. The family of *evolutionary algorithms* is based on the idea of modelling the search process of natural evolution, though these models are crude simplifications of biological reality. Evolutionary algorithms (EA) are variants of randomized search, and use the terminology from biology and genetics. For example, given a random sample at each iteration, pairs of parent individuals (points), selected on the basis of their ›fit‹ (function value), recombine and generate new ›offspring‹. The best of these are selected for the next generation. Offspring may also ›mutate‹ that is randomly change their position in space. The idea is that fit parents are likely to produce even fitter children. In fact, any random search may be interpreted in terms of biological evolution: generating a random point is analogous to a mutation, and the step made towards the minimum after a successful trial may be treated as a selection.

Historically, evolution algorithms have been developed in three variations - evolution strategies (ES), evolutionary programming (EP), and genetic algorithms (GA). *Back & Schwefel 1993* give an overview of these approaches, which differ mainly in the types of mutation, recombination and selection operators. In GA, the binary coding of coordinates is introduced, so that an l-bit binary variable is used to represent integer code of one coordinate x_i, with the value ranging from 0 to 2^l-1 that can be mapped into the real-valued interval $[a_i,b_i]$. An overall binary string G of length nl called a *chromosome* is obtained for each point by connecting the codings of all coordinates. The *mutation operator* changes a randomly chosen bit in the string G to its negation. The recombination (or *crossover*) operator is applied as follows: select two points (parents) S and T from the population according to some rule (e.g., randomly), select a number $\tilde{n}$ (e.g., randomly) between 1 and nl, and form either one new point S', or two new points S' and T', by taking left-hand side bits of coordinate values from the first parent S, and right-hand side bits from the other parent T.

There are various versions of GA varying in the way crossover, selection and construction of the new population is performed. In *evolutionary strategies (ES)*, mutation of coordinates is performed with respect to corresponding variances of a certain n-dimensional normal distribution, and various versions of recombination are introduced. On GAs applications see,

e.g., *Wang 1991, Babovic et al. 1994, Cieniawski 1995, Savic & Walters 1997, Franchini & Galeati 1997*.

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. One of the popular versions of multistart used in global optimization is based on clustering, that is creating groups of mutually close points that hopefully correspond to relevant regions of attraction of potential starting points (*Torn & Žilinskas 1989*). The *region (area) of attraction* of a local minimum x^* is the set of points in X starting from which a given local search procedure P converges to x^*. For the global optimization tool GLOBE used in the present study, we developed two multistart algorithms - *Multis* and *M-Simplex*. They are both constructed according to the following pattern:

1. Generate a set of N random points and evaluate f at these points.

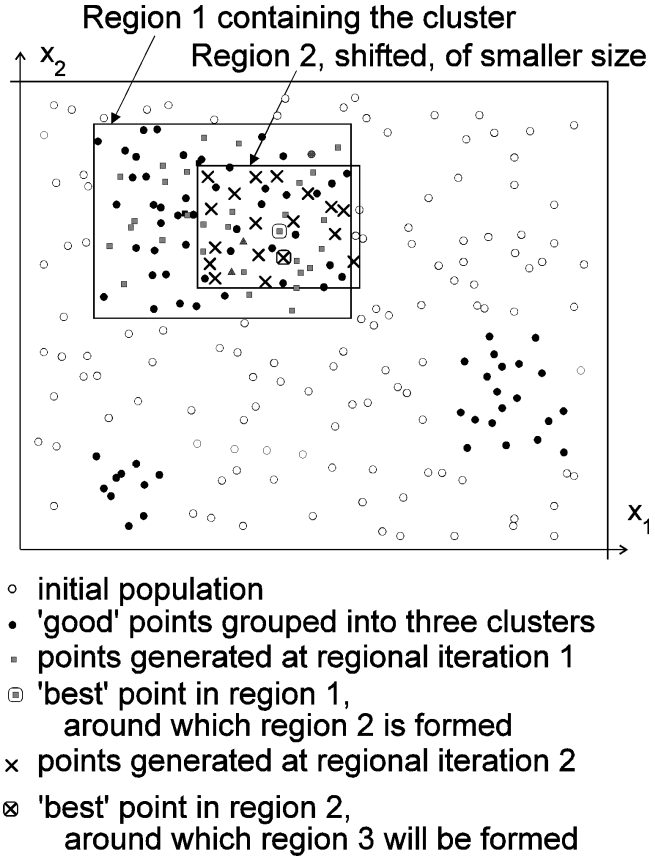
2 (*reduction*). Reduce the initial set by choosing p best points (with the lowest f_i).

3. (*local search*). Launch local search procedures starting from each of p points. The best point reached is the minimizer assessment.

In *Multis*, at step 3 the Powell-Brent local search (see *Powell 1964, Brent 1973, Press et al., 1991*) is started. In *M-Simplex* the downhill simplex descent of *Melder & Nead 1965* is used.

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*) is a workable combination of generally accepted ideas of reduction, clustering and covering (Fig.1).



1. Clustering. Clustering (identification of groups of mutually close points in search space) 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 the uniform or some other

distribution. Covering is repeated multiple times and each time the subdomain is progressively reduced in size.

3. Adaptation. Adaptive algorithms update their algorithmic 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. Due to the probabilistic character of points generation, any strategy of randomized search may simply miss a promising region for search. In order to reduce this danger, the initial population is re-randomized, i.e. the problem is solved several times.

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.

ACCOL strategy 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 (such points we will call ›potent‹). The potent set P_1 is formed by taking one best point found for each cluster during 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, with the distance at each dimension being larger than, for example, 10% of the range for this dimension;

2. *Local search (LS) phase*. An accurate algorithm of local search is started from each of the potent points 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 function evaluations.

ACD algorithm (*Solomatine 1998*) is also a random search algorithm, and it combines 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 in this area. The main steps of ACD are:

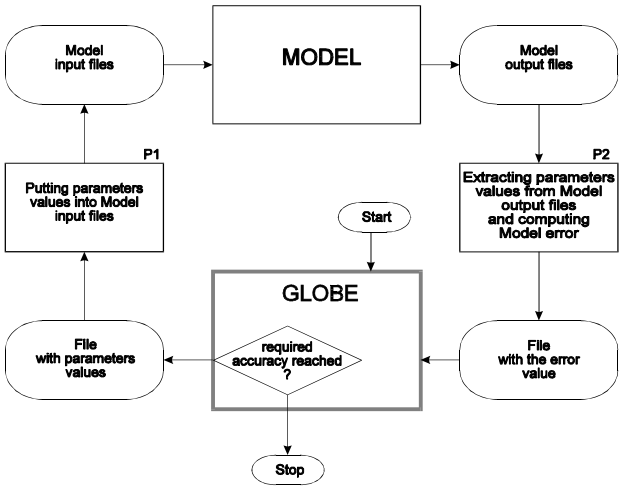
- sample points (e.g., uniformly), and reduce the sample to contain only the best points;
- cluster points, and reduce clusters to contain only the best points;
- in each cluster, apply the limited number of steps of DSD to each point, thus moving them closer to an optimizer;
- if the cluster is potentially ›good' that is contains points with low function values, cover the proximity of several best points by sampling more points, e.g. from uniform or beta distribution;
- apply local search (e.g., DSD, or some other algorithm of direct optimization) starting from the best point in ›good‹ clusters. In order to limit the number of steps, the fractional tolerance is set to be, say, 10 times greater than the final tolerance (that is, the accuracy achieved is somewhat average);
- apply the final accurate local search (again, DSD) starting from the very best point reached so far; the resulting point is the assessment of the global optimizer.

ACDL algorithm, combining ACD with the multiple local searches, has been built and tested as well.

4. GLOBAL OPTIMIZATION TOOL *GLOBE*

A PC-based system GLOBE incorporating 9 GO algorithms was built. 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. Figure 2 shows how GLOBE is used in the problems of automatic calibration. *Model* must be an executable module (program) which does not require any user input, and the user has to supply two transfer programs *P1* and *P2*. These three programs (*Model, P1, P2*) are activated from GLOBE in a loop. GLOBE runs in DOS protected mode (DPMI) providing enough memory to load the program modules. A Windows version is being developed. The user interface includes several graphical windows displaying the progress of minimization in different coordinate planes projections. The parameters of the algorithms can be easily changed by the user.



Currently, GLOBE includes the following nine algorithms described above:

- *CRS2 (controlled random search,* by *Price 1983)*;
- *CRS4* (modification of the *controlled random search* by *Ali & Storey 1994a)*;
- *GA* with a one-point crossover, and with a choice between the real-valued or binary coding (15 bits were used in our experiments); with the standard random bit mutation; between the tournament and fitness rank selection; and between elitist and non-elitist versions.
- *Multis* - multistart algorithm;

- *M-Simplex* - multistart algorithm;
- *adaptive cluster covering* (*ACCO*);
- *adaptive cluster covering with local search* (*ACCOL*).
- *adaptive cluster descent* (*ACD*);
- *adaptive cluster descent with local search* (*ACDL*).

5. COMPARING NINE 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, Solomatine 1995b*);

calibration of a lumped hydrological model (*Solomatine 1995b)*;

calibration of a 2D free-surface hydrodynamic model (*Constantinescu 1996*);

calibration of a distributed groundwater model (*Solomatine et al 1998*);

calibration of an ecological model of plant growth;

calibration of an electrostatic mirror model (*Vdovine et al., 1995*);

solution of a dynamic programming problem for reservoir optimization (*Lee 1997)*; optimization of a pipe network(*Abebe & Solomatine 1998*).

Table 1. Functions used in comparing algorithms

Function	Nmb. of vars	Nmb. of optima	Value of the global minimum
Rosenbrock	2	1	0.0
Hosaki	2	2	. -2.338
Rastrigin, shifted by 2.0	2	>50	0.0
Six-hump camelback (Branin), shifted by 1.036285	2	6	0.0
Goldstein-Price function	2	4	3.0
Flexible mirror model error	3	?	. 0.0
Hartman3, shifted by 3.32	3	4	.- 0.6
Hartman6, shifted by 3.32	6	4	. 0.0
Shekel5, shifted by 10.5364	4	5	. 0.0
Shekel7, shifted by 10.5364	4	7	. 0.0
Shekel10, shifted by 10.5364	4	10	. 0.0
Griewank function	10	>1000	0.0
ADM model error	11	?	< 23.8
SIRT model error	8	?	< 47.0

The most comprehensive experiments with all 9 algorithms included in GLOBE tool were set up for the problems listed in Table 1. The size of this paper does not allow the description of all the results; Figure 3 shows several typical examples of the process of minimization (averaged on 5 runs), including those for two hydrological conceptual rainfall-runoff models (Sugawara-type tank model SIRT, see *Solomatine 1995b* and the distributed model ADM, *Franchini & Galeati 1997*).

The number N of points in the initial sample and the number of points in the reduced sample were chosen according to the rule that these numbers must grow linearly with the dimension n, from N=50 at n=2, to N=300 at n=30. For *CRS2* and *CRS4* the formula recommended by their authors is $N=10(n+1)$. In *ACCOL*, *ACDL*, *Multis* and *M-Simplex* the fractional tolerance of 0.001 was used. In GA fitness rank elitist selection is used together with a complex stopping rule preventing premature termination.

Since GA uses discretized variables (we used the 15-bit coding, i.e. the range is 0...32767) an accurate comparison would only be possible if the values of the variables for other algorithms were discretized in the same range as well. This has been done for *ACCO, ACD* and *CRS4*. Other algorithms, including the local search stages of *ACCOL* and *ACDL*, use real-valued variables.

Three main **performance indicators** were investigated:
effectiveness (how close the algorithm gets to the global minimum);
efficiency (running time) of an algorithm measured by the number of function evaluations needed (the running time of the algorithm itself is negligible compared with the former);
reliability (robustness) of the algorithms can be measured by the number of successes in finding the global minimum, or at least approaching it sufficiently closely.

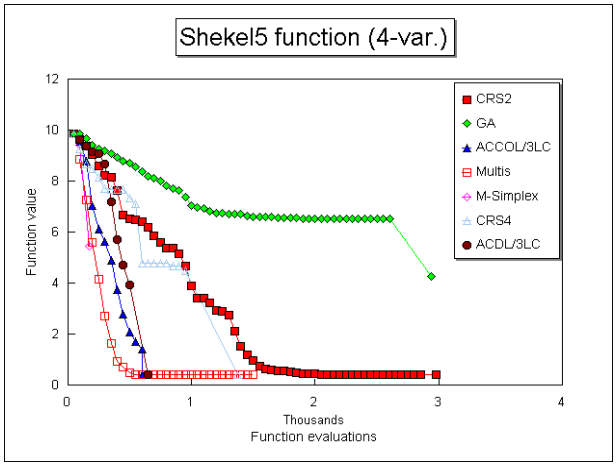
Effectiveness and efficiency. The plots on Figure 3 show the progress of minimization for some of the functions averaged across 5 runs (*the last point* represents the best function value found through all five runs). The vertical line segment between the last two points means that the best function value has been reached in one of the runs *earlier* than shown by the abscissa of the last but one point. Note that most points of the *ACCOL* plot correspond both to *ACCO* and *ACCOL*, and only some of the last points correspond to the local search phase of *ACCOL;* the same applies to *ACDL* and *ACD*.

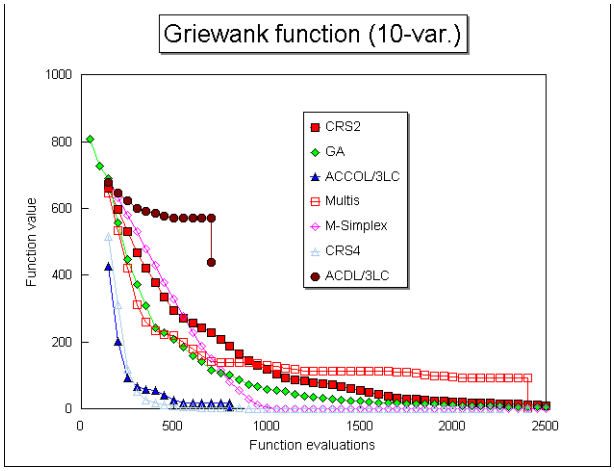
The comparison results can be summarized briefly as follows. For functions of 2 variables, *ACCOL, CRS4* and *M-Simplex* are the most efficient, that is, faster in getting to the minimum. In Hosaki, Rastrigin and six-hump camelback functions *M-Simplex* quite unexpectedly showed the best results. With functions of higher dimensions, *ACCOL* and *CRS4* again performed best, and had similar performance. *M-Simplex* was the worst with all Shekel 4-variable functions, but was even a bit better than *ACCOL* and *CRS4* with Hartman 3- and 6-variable functions. *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. Because of the randomized character of search no algorithm can be 100% reliable. For most functions of 2 variables most algorithms were quite reliable (with the exception of GA, which was often converging prematurely). Only the Rastrigin function with many equidistant local minima with almost equal values presented difficulties.

With the functions with more than two variables the situation was different. It can be seen from Fugure 3 that for most algorithms the ordinate of the last point can be considerably less than the ordinate of the previous point. This means that the least function value was found in some runs, but not in all of them. The *CRS2* and *Multis* algorithms appeared to be the most reliable for functions of higher dimensions but were by far the least efficient. *ACDL* was not always reliable even though it showed efficiency on some runs.

In most cases, except for *GA* the found minimizer estimate is normally quite close to the global minimum. Small differences could be attributed partly to the way the real-valued variables were coded. A more accurate statistical analysis of single-start failure probabilities has yet to be done.





6. 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*). The lower efficiency of *GA* can also be attributed to the type of ›crossover‹ used (exchange of some of the parents‹ coordinate values) which often leads to redundant evaluations of the ›offspring‹ in the search space quite far from their highly fit parents, and hence normally with lower fitness. So the fitness gained by the parents may not be inherited by many of their offspring. It was also found that *GA* often converges prematurely, especially in the variant with tournament selection. Whether this feature is inherent to the whole class of evolutionary algorithms following the ideas of natural evolution, which are indeed quite appealing but highly redundant, or it is a feature of the version of a GA implemented in this study, has yet to be investigated. It is worth mentioning that reportedly other types of crossover, like intermediate recombination in evolutionary strategies (*Back & Schwefel 1993*) may improve the efficiency of evolutionary algorithms.

The relatively higher efficiency of *ACCOL* and *CRS4* can be explained by their orientation towards smaller search domains which is especially efficient for high dimensions. *ACDL* on some runs has shown high efficiency but its reliability was not the best.

7. CONCLUSIONS

1. Our experience showed that GO techniques are useful in solving various classes of

optimization problems. Among the GO algorithms compared *ACCOL* and *CRS4* showed the highest effectiveness, efficiency and reliability. In many practical problems where one function evaluation is expensive (slow), and their total number is then the critical parameter, *ACCO* (without the local search phase) would be the first choice to obtain a reasonable optimizer assessment.

ACDL algorithm proved to be efficient and effective on some of the runs with functions of higher dimensions. However, accurate tuning of its parameters is needed to improve its reliability.

M-Simplex performs very well with the functions of low dimension but in higher dimensions it often converges prematurely to a local minimum.

GA, *CRS2*, and *Multis* provide reasonable solutions as well. However, all of them require considerably more function evaluations, and *GA* may also converge prematurely before it reaches the global minimum. So for problems involving ›expensive‹ functions with continuous variables there are better alternatives like *ACCOL* or *CRS4*. Our other experiments (*Abebe and Solomatine 1998*) however, show that for certain classes of problems with highly discrete variables, e.g. in water distribution network optimization, *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 various methods of global optimization may depend on the type of problem, and more research is needed to compare reportedly efficient methods like simulated annealing, evolution strategies, topological multilevel linkage, shuffled simplex evolution and others (see *Ali and Storey 1994b; Locatelli and Schoen 1996, Neumaier 1998, Duan 1993, Kuczera 1997*). The best results can probably be achieved by *structural adaptation*, that is, switching in the process of search between different algorithms.

3. Practically in all problems with continuous variables where the use of GAs was reported, other GO algorithms can be used as well.

4. GLOBE tool showed itself as an efficient engine for model calibration; it can be downloaded from ***www.ihe.nl/hi/***.

REFERENCES

Abebe A.J. & Solomatine D.P. 1998. Application of global optimization to the design of pipe networks. *Proc. Int. Conf. Hydroinformatics-98*.

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

Ali, M.M. & Storey, C. 1994b. Topographical multilevel single linkage. *J. of Global Optimization*, 5, pp. 349-358.

Babovic, V., Wu, Z. & Larsen L.C. 1994. Calibrating hydrodynamic models by means of simulated evolution, *Proc. Int. Conf. on Hydroinformatics*, Delft, The Netherlands. Balkema, Rotterdam, pp.193-200.

Back, T. & Schwefel, H.-P. 1993. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1, No. 1, pp. 1-23.

Brent, R.P. 1973. Algorithms for minimization without derivatives. Prentice-Hall, Englewood-Cliffs, N.J., 195p.

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

Constantinescu A. *1996. Calibration of hydrodynamic numerical models using global optimization techniques*. M.Sc. thesis No. HH262, IHE, Delft, 85p.

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

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

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

Goldberg, D.E. 1989. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Griewank, A.O. 1981. Generalized descent for global optimization. *J. Optimiz. Theory Appl.*, 34 (1), 11-39.

Jacobs, D.A.H. 1977. The state of the art in numerical analysis, Academic Press, London.

Kuczera, G. 1997. Efficient subspace probabilistic parameter optimization for catchment models. Water Resour.

Res, 33 (1), 177-185, January.
Lee H. 1997. *Optimal reservoir operation for flood control using a hybrid modelling approach*. M.Sc. thesis No. HH319, IHE, Delft.
Locatelli, M. & Schoen, F. 1996. Simple linkage: analysis of a threshold-accepting global optimization method. *J. of Global Optimization*, 5, pp. 95-111.
Michalewicz, Z. 1996. *Genetic algorithms + data structures = evolution programs*. Springer, Berlin.
Nelder, J.A & Mead, R. 1998. A simplex method for function minimization. *Computer J.*, vol. 7, No. 4 p. 308-313, 1965.
Neumaier WWW page. ***solon.cma.univie.ac.at/ ~neum/glopt.html***.
Pintér, J. 1995. Global optimization in action. Kluwer, Amsterdam.
Powell, M.J.D. 1964. An efficient method of finding the minimum of a function of several variables without calculating derivatives. *Computer J.*, 7, 155-162.
Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T. 1990. Numerical recipes in Pascal. The art of scientific computing. Cambridge University Press, Cambridge, 759p..
Price W.L. 1983. Global optimization by controlled random search. *J. Optimiz. Theory Appl.*, 40, 333-348.
Pronzato L., Walter E., Venot A., Lebruchec J.-F. 1984. A general purpose global optimizer: implementation and applications, in *Mathematics and Computers in Simulation*, 26, 412-422.
Savic, D.A. & Walters, G.A. 1997. Genetic algorithms for least-cost design of water distribution networks. *J. of Water Res. Planning and Mngt.*, 123 (2), 67-77.
Solomatine, D.P. 1995a. The use of global random search methods for models calibration, Proc. XXVIth Congress of the International Association for Hydraulic Research (IAHR), vol.1, pp. 224-229, London, September.
Solomatine D.P. 1995b. Application of global optimization to models calibration. Presented at the Seminar *Methods and software for estimation of large-scale spatial interaction models*, Netherlands Interdisciplinary Demographics Inst., July, *www.ihe.nl/hi/sol/sol.htm.*
Solomatine, D.P. 1998. Two strategies of adaptive cluster covering with descent and their comparison to other algorithms. *J. Global Optimiz.*, 1998.
Solomatine D.P. Gautam D., Franchini M. 1998. A conceptual rainfall-runoff model and tools for its automatic calibration, 1998.
Törn, A. & ðilinskas, 1989. A. *Global optimization*. Springer-Verlag, Berlin, 255pp.
Törn, A. & Viitanen, S. 1994. Topographical Global Optimization using pre-sampled points. *J. of Global Optimization*, 5, pp. 267-276.
Wang, Q.J. 1991. The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. Water Resour. Res., 27 (9), 2467-2471.
Van den Boogaard, H.F.P., Hoogkamer, M.J.J. & Heemink, A.W. 1993. Parameter identification in particle models. *Stochastic Hydrology and Hydraulics*, 7, 109-130.
Vdovine, G., Middelhoek, S., Bartek, M., Sarro, P.M., & Solomatine, D.P. 1995. Technology, characterization and applications of adaptive mirrors fabricated with IC-compatible micromachining. *Proc. SPIE's Int. Symposium on Optical Science, Engineering and Instrumentation*, vol. 2534/13, San-Diego, USA, July 10-14.