05JEI00044
1726-2135/1684-8799

www.iseis.org/jei

Journal of Environmental Informatics 5 (1) 36-41 (2005)

GHHAGA for Environmental Systems Optimization

X. H. Yang[1*], Z. F. Yang[1, 2] and Z. Y. Shen[1]

[1]State Key Laboratory of Water Environment Simulation, School of Environment, Beijing Normal University, Beijing 100875, China
[2]Key Laboratory for Water and Sediment Sciences Ministry of Education, School of Environment, Beijing Normal University, Beijing 100875, China

ABSTRACT. The global optimization of complicated nonlinear systems is mathematically intractable and such an optimization extensively exists in science and engineering. Once an objective function has many local extreme points, the traditional optimization methods may not obtain the global optimization efficiently. A genetic algorithm (GA) based on the genetic evolution of a species provides a robust procedure to explore broad and promising regions of solutions and to avoid being trapped at the local optimization. However, the computational amount is very large. To reduce computations and to improve the computational accuracy, a method based on the two-point crossover and two-point mutation of the hybrid accelerating genetic algorithm with Hooke-Jeeves searching operator is developed for systems optimization. With the shrinking of searching range, the method gradually directs to optimal result by the excellent individuals obtained by Gray code genetic algorithm embedding with Hooke-Jeeves searching operator and Hooke-Jeeves algorithm. The efficiency of the new algorithm is verified by application of several test functions. The comparison of our GA with six existing other algorithms is presented. This algorithm overcomes the Hamming-cliff phenomena in other existing genetic methods, and is proved to be very efficient for the given environmental systems optimization.

Keywords: Environmental systems, genetic algorithm, global optimization, gray code, Hooke-Jeeves algorithm

1. Introduction

The global optimization of complicated nonlinear systems, which extensively exists in scientific and engineering fields, is mathematically intractable. Finding the minimum of a nonlinear function with a single variable, even without any constraints, can still be challenging. A severe difficulty arises if the function in question has many pure local minima, as what one really wants is a global minimum point (David et al., 2003). This is due to the fact that, once an objective function has many local extremes, the traditional optimization methods may not reach the global solution efficiently. The genetic algorithm (GA), which is based on the genetic evolution of a species, was proposed by Holland in 1975 (Holland, 1975). This method is globally oriented in searching and thus potentially useful in solving optimization problems where the objective function contains multiple optima and other irregularities. The detailed algorithms and implementation procedures were given by Goldberg (Goldberg, 1989). De Jong (1975) showed that the standard binary-encoded GA (SGA) could constitute an interesting alternative to perform the global optimization of a function depending on several continuous variables (Andre et al., 2001). This algorithm provides a robust procedure to explore broad and promising regions of solutions, and to avoid being trapped at the local optima (e.g., Yang et al., 2004; Leung et al., 2001). Experimental results indicated that through an appropriate selection of representation patterns of elements of the search space and operators, GA is capable in solving a number of "difficult" problems (Bessaou et al., 2001). However, in SGA, the computational amount is significant and the phenomena of premature convergence are generally inevitable. To reduce computational efforts and improve the computing precision, the binary-encoded accelerating genetic algorithm, real-encoded genetic algorithm and integer-encoded genetic algorithm were developed in the past decades (Jin et al., 2000; Janikow et al., 1991; Renders et al., 1996). However, these genetic algorithms are normally weak in dealing with global optimizations for complicated nonlinear systems with continuous variables. In fact, the Hamming distance between two closest integers in binary code may become considerably large. For instance, integers 63 and 64 are expressed by the 00111111 and 01000000 in binary code, respectively. All of the codes must be changed if we turn 63 into 64 in binary code. This operation reduces the efficiency of the genetic algorithms. The phenomenon is termed the 'Hamming cliff'. To solve the above 'Hamming cliff' problem, SGA could be improved with gray encodings of parameters (e.g., Andre et al., 2001; Yang et al., 2003). For example, the integers 63 and 64 can be expressed by 01000000 and 01100000 in Gray code, respectively. The Gray-encoded genetic algorithm (GGA) can partly overcome the 'Hamming cliff' of binary code (Zhong et al., 2001). However, realization of this algorithm requires appropriate operators, and its mathematical theory has not been well developed yet. It was also found that this algorithm still needs a large amount of computational efforts (Yang,

* Corresponding author: y1x1h1@sohu.com

2002). Thereofore, GGA is desired to be improved.

Thus, in this paper, a Gray-encoded, Hooke-Jeeves, Hybrid Accelerating Genetic Algorithm (GHHAGA) will be presented to reduce computational amounts and to improve the solution precisions for nonlinear optimization problems (Hooke et al., 1961). Five nonlinear functions and two environmental optimization problems will be used for verifycation of the proposed algrothm.

2. Descriptions of the GHHAGA

The GHHAGA starts with an initial population of n "individuals": each individual is composed of Gray code, individually associated with the variables of the objective function at hand. Then evolution starts and genetic operations are constituted. The reproduction operators are applied to this population; and the offspring are created from parents. The new population is constituted in selecting the best individuals. After two-point crossover and two-point mutation, the new individuals are created. And the output of the best point is further indentified through the Hooke-Jeeves algorithm around the above best individual (point) (Hooke et al., 1961). The new best point inside the offspring will be inserted to replace the worst one in the previous phase. Repeat the above genetic operations until the evolution times Q is met. The new intervals are obtained from the variable ranges of n_s-excellent individuals by Q-times of the Hooke-Jeeves evolution, and then the whole process steps back to the Gray-encoding. Finally, the most excellent chromosome becomes the optimum solution of GHHAGA.

2.1. Steps of GHHAGA

Consider the following nonlinear optimization problem:

$$\min f(x_1, x_2, \cdots, x_{n_p})$$

$$\text{s.t.} \quad a_j \le x_j \le b_j \quad \text{, for} \quad j = 1,2, ..., n_p \tag{1}$$

where $x = \{x_j, j = 1, 2, ..., n_p\}$, x_j is a variable to be optimized, f is an objective function and $f \ge 0$. Steps of GHHAGA are given as follows.

Step 1. *Gray encoding.*

An e-bit Gray variable is used to represent one variable x_j. The integer of the decoded Gray variable ranges from 0 to 2^e - 1 and can be mapped linearly to the variable range $[a_j, b_j]$. The j^{th} variable range is the interval $[a_j, b_j]$, and then each interval is divided into 2^e - 1 sub-intervals:

$$x_j = a_j + I_j \cdot \Delta x_j \tag{2}$$

where $\Delta x_j = (b_j - a_j)/(2^e - 1)$. The Gray code array of the j^{th} variable is denoted by the grid points of $\{d(j,k) \mid k = 1,2, ..., e\}$ (Yang et al., 2003):

$$I_j = \sum_{m=1}^{e} (\bigoplus_{k=m}^{e} d(j, k)) \cdot 2^{m-1} \tag{3}$$

Step 2. *Generating initial father population.*

Initially, the chromosomes are generated at random in Gray-encoded genetic algorithm, and n-chromosomes in father population are:

$$I_j(i) = \text{int}(u(j, i) \cdot 2^e) \quad \text{for} \quad j = 1, 2, ..., n_p; i = 1, 2, ..., n \tag{4}$$

where $u(j, i)$ is uniformity random number, $u(j, i) \in [0, 1]$, $I_j(i)$ is searching location, int() is an integer function. From Equation (3), the n-corresponding chromosomes are $d(j, k, i)$ for j = 1, 2, …, n_p; k = 1, 2, …, e; i = 1, 2, …, n. To cover homogeneously the whole solution space and to avoid the risk of having too much individuals in the same region, a large uniformity random population is selected in this algorithm.

Step 3. *fitness evaluation.*

The fitness function $F(i)$ of i^{th} chromosome for the optimization is defined:

$$F(i) = \frac{1}{[f(i)]^2 + 0.1} \tag{5}$$

Step 4. *Selection.*

The chromosomes in the initial father population are selected by a known probability $p_s(i)$ as:

$$p_s(i) = F(i) / \sum_{j=1}^{n} F(j) \tag{6}$$

Such two groups of n-chromosomes are selected by the above probabilities.

Step 5. *Two-point crossover.*

For two-point crossover, two crossing points are randomly chosen, and two individuals $d_1(j, k, i)$, $d_2(j, k, i)$ are gotten by the crossing probability p_c. In order to enhance the diversity of population, the crossing probability is set as p_c = 1.

Step 6. *Two-point mutation.*

For two-point mutation, two mutating points are randomly chosen, and a new offspring $d_3(j, k, i)$ can be computed by a mutating probability p_m (Yang et al., 2003).

Step 7. *Hooke-Jeeves evolution*.

The Hooke-Jeeves algorithm is a useful, local descent algorithm, which does not make use of the objective function derivatives (Hooke et al., 1961). The best point in the previous phase becomes a new initial solution in the Hooke-Jeeves algorithm, and then a new best point is obtained by this Hooke-Jeeves algorithm. The new best point inside the offspring will be inserted to replace the worst one in the previous phase. Repeat step 3 to step 7 until the evolution times Q or

termination criteria is met.

Step 8. *Accelerating cycle.*

The variable ranges of n_s-excellent individuals obtained by Q-times of the Hooke-Jeeves evolution become the new ranges of the variables, and then the whole process back to the Gray-encoding. The computation process is over until the objective function value gets to an expected value, or algorithm running times gets to the design T times. Herein, the most excellent chromosome currently is the optimum solution of GHHAGA.

A pseudo code of GHHAGA is given in Figure 1.

```
Begin
For t=0 to T (acceleration cycle times t)
Give variable interval [a_j^t, b_j^t]
Gray encoding
q=0
Initialize population Pop(q,t)
While (q≤Q) do (evolution times q)
For i=1 to n do
Evaluate fitness of Pop (q,t)
Endfor
For i=1 to n do
Select operation to Pop(q,t)
Endfor
For i=1 to n do
Two-point crossover operation to Pop(q,t)
Endfor
For i=1 to n do
Two-point Mutation operation to Pop(q,t)
Endfor
Hooke-Jeeves evolution
Endwhile
Get new variable interval [a_j^{t+1}, b_j^{t+1}] from the
  variable ranges of n_s -excellent individuals in
  Pop (1,t), Pop (2,t),…, Pop (Q,t)
Endfor
```

Figure 1. Pseudo code for GHHAGA.

2.2. GHHAGA Theory

We now turn to the analysis of the behavior of our algorithm when it is applied to problem (1). The GHHAGA is convergent (Yang, 2002). The global optimization of the GHHAGA is not only accurate but also stable. Let the Hooke-Jeeves evolution times be Q, the number of excellent individuals be n_s, the number of optimized variable be p and the times of accelerating evolution be T, the probability p_0 of excellent individuals surround the optimum point is $p_0 = \left(1-2^{-Qn_s}\right)^{pT}$. The GHHAGA is global convergence with probability $p_0 = 1.000\ 000$ when $n_p = 20$, $t = 5$, $n_s = 10$, $Q = 5$; $n_p = 30$, $t = 10$, $n_s = 10$, $Q = 5$, etc.

3. Experimentation

3.1. Criteria

Three main criteria are very important when trying to determine the performances of an algorithm: convergence, speed and robustness (Andre et al., 2001). The parameters of the GHHAGA are selected as follows: The length $e = 10$, population size $n = 300$, the number of excellent individuals $n_s = 10$, the times of Hooke-Jeeves evolution $Q = 5$, the crossover probability $p_c = 1.0$, the mutation probability $p_m = 0.5$, and the times of Hooke-Jeeves searching $m \leq 300$.

The global optimization of five test functions (Andre et al., 2001) is accomplished by using the following methods: Standard Binary-encoded GA (SGA) (Andre et al., 2001) and GHHAGA. To compare with the global optimization ability of the above algorithms objectively, the less than or equal 18,000 computations of the objective functions are done, and one of the three termination criteria is used for ensuring the optimization precision and avoiding algorithm invalidation.

Criteria one: The relative error E_{rel} between the result f_{algo} given by the algorithm and the optimum value f_{exact} of each test function is used each time if it is possible:

$$E_{rel} = \frac{| f_{algo} - f_{exact} |}{| f_{exact} |} \tag{7}$$

Criteria two: When the optimum is 0, it is no longer possible to use this expression. So we calculate the absolute error E_{abs} as follows:

$$E_{abs} = | f_{algo} - f_{exact} | \tag{8}$$

where we let the absolute error or relative error in neighbor generations be less than or equal 10^{-2}.

Criteria three: The total computation number for the functions is less than or equal 18,000.

3.2. Experimentation and Result

To test our GHHAGA, the following five analytical test functions were used.

Goldprice:

$$\begin{aligned} f(x,y) = &[1+(x+y+1)^2(19-14x+3x^2-14y+6xy+3y^2)] \\ &\times[30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2)] \end{aligned} \tag{9}$$

Table 1. Parameters of a_{ij} and p_{ij}

i	a_{ij}						c_i	p_{ij}					
1	10.0	3.0	17.0	3.5	1.7	8.0	1.0	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.05	10.0	17.0	0.1	8.0	14.0	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3.00	3.50	1.7	10.0	17.0	8.0	3.0	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	17.0	8.0	0.05	10.0	0.01	14.0	32	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

where $-2 \le x \le 2, -2 \le y \le 2$.

Hartman 2:

$$f(x) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2] \quad (10)$$

where $0 \le x_i \le 1$ for $j = 1, \ldots, 6$ with $x = (x_1, \ldots, x_3)$, $p_i = (p_{i1}, \ldots, p_{i6})$, and $a_i = (a_{i1}, \ldots, a_{i6})$ as shown in Table 1.

Hosc 45:

$$f(x) = 2 - \prod_{i=1}^{10} \frac{x_i}{n!} \quad (11)$$

where $x = (x_1, \ldots, x_{10})$, and $0 \le x_i \le i$, $n = 10$.

Brown 1:

$$f(x) = [\sum_{i \in J}(x_i - 3)]^2 + \sum_{i \in J}[10^{-3}(x_i - 3)^2 - (x_i - x_{i+1}) + e^{20(x_i - x_{i+1})}] \quad (12)$$

where $J = \{1, 2, \ldots, 19\}$, $-1 \le x_i \le 4$, for $1 \le i \le 20$, and $x = [x_1, \ldots, x_{20}]^T$.

F15n:

$$f(x) = 0.1\left\{\begin{array}{l} \sin^2(3\pi x_1) + \sum_{i=1}^{19}[(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1}))] \\ +0.1(x_{20} - 1)^2[1 + \sin^2(2\pi x_{20})] \end{array}\right\} \quad (13)$$

where $-10 \le x_i \le 10$, for $1 \le i \le 20$ and $x = [x_1, \ldots, x_{20}]^T$.

This set of classical test functions, very often used in the literature (Andre et al., 2001; Bessaou et al., 2001), includes some functions having the following features:

- Continuous/discontinuous;
- Convex/non-convex;
- Unimodal/multimodal;
- Quadratic/non-quadratic;
- Low dimension/high dimension.
- The efficiency of the algorithm is quantified by:
- The rate of success;
- The number of evaluations of the objective function;
- The error between the obtained solution and the exact value of the global optimum of the function.

Because of the stochastic nature of GAs, the discussion of results derived from one single execution of the algorithm is meaningless (Bessaou et al., 2001). So all results reported in this section are obtained by averaging the results from 100 executions per function. The computation results of the five nonlinear test functions are given in Table 2 with the SGA (Andre et al., 2001) and GHHAGA. It is obviously observed that the GHHAGA is the best one both in efficiency (see success rate and number of evaluation of the functions in Table 2) and in accuracy (see minimum found in Table 2) compared with existing algorithms. The results given in Table 2 show that the global optimum generally is reached since the ratio of success is equal to 100% for the ten tested functions. So the 'Hamming cliff' phenomena are avoided in GHHAGA.

We have performed a comparison of our GA with six other methods of iterative improvement listed in Table 3: pure random search (PRS) (Anderssen et al., 1972), multistart (MS) (Rinnoy et al., 1987), simulated diffusion (SD) (Aluffi et al., 1985), simulated annealing (SA) (Andre et al., 2001), tabu search (TS) (Cvijovic et al., 1995) and binary-encoded genetic algorithm (GA) (Andre et al., 2001). The efficiency was qualified by use of the number of function evaluations to find the global optimum. Each program was stopped as soon as the relative error between the best point found and the known global optimum was less than 1%. The number of function evaluations used by the various algorithms to optimize test functions is listed in Table 3. The results of our GA are satisfied (see the numbers of function evaluations in Table 3). In addition our results were the average outcome of 100 independent runs; for some published methods, the times of runs was equal to 4 or unspecified (Andre et al., 2001).

4. Application

Example 1. Consider the wastewater treatment cost time series as a dynamical system optimization to satisfy:

$$y(t) = A - K/(1 + be^{-at}) \quad (14)$$

where $y(t)$ is the computed value of wastewater treatment cost in the t^{th} year, unit: 10^4 US $; A, K, b and a are optimization parameters. The observation data can be found in the literature (Jin, 2000). This objective function is as follows:

Table 2. Results with the SGA (Andre et al., 2001) and the GHHAGA

		Minimum found		Number of evaluation of the functions		Success rate %	
Name of the functions	Theoretical minimum	SGA	GHHAGA	SGA	GHHAGA	SGA	GHHAGA
Goldprice	3	3.00000	3.00085	8185	303	59	100
Hartman2	-3.32237	-3.30652	-3.31953	19452	708	23	100
Hosc45	1	1.99506	1.00000	11140	300	0	100
Brown1	2	43.62810	1.99877	6844	312	0	100
F15n	0	0.52117	0.00000	9541	786	0	100

Table 3. Number of Function Evaluations on Global Optimization of Two Functions with the Seven Different Methods

		The number of function evaluation	
Method Name		Goldprice	Hartman2
PRS	Pure random search	5125	18090
MS	Multistart	4400	6000
SD	Simulated diffusion	5439	3975
SA	Simulated annealing	563	4648
TS	Tabu search	486	2845
GA	Binary-encoded genetic algorithm	4632	53792
GHHAGA	Gray-encoded, Hooke-Jeeves, hybrid accelerating genetic algorithm	303	708

Table 4. The Calculating Result with Several Methods for the Dynamical Optimization Problem

		Parameters				
Method	Evaluation number for *h*	A	K	b	a	Least mean square sum *h*
Initial interval	0	[500, 700]	[200, 300]	[5, 8]	[0, 1]	/
GHHAGA	2400	607.740	247.395	5.974	0.860	6.690
AAGA	6000	603.667	243.487	5.665	0.838	6.934
BAGA	3000	600.700	240.700	6.069	0.853	8.130

$$h = \sum_{i=1} (y(t_i) - y_i)^2 \tag{15}$$

where y_i is the observation value of the wastewater treatment cost in the t_i^{th} year, unit: 10^4 US \$, n is the total number of observation data. The least residual square sum h is 6.690 with GHHAGA. GHHAGA runs 1 second for optimization of this model only. The computational results of the above model are given in Table 4. For the GHHAGA, the evaluation number of the objective function is 2400. For adaptive accelerating genetic algorithm (AAGA) (Yang, 2002), the evolution number is 6000. For binary-encoded accelerating genetic algorithm (BAGA) (Jin, 2000), the evolution number is 30000. The comparison of the above methods can be observed in Table 4. The results of our GHHAGA are satisfied both in efficiency and in accuracy.

Example 2. Consider the forecast problem of the sediment concentration in Gongzui reservoir as a system optimization to satisfy:

$$u_s(t) = c_0 + c_1 x(t) + c_2 x(t)^2 + c_3 x(t)^3 \tag{16}$$

where $u_s(t)$ is the simulated value of the sediment concentration in Gongzui reservoir in the t^{th} year, $x(t) = (Q_{in}(t) - Q_{up}(t))/1000$, $Q_{up}(t)$ is the runoff of upstream reservoir and $Q_{in}(t)$ is the inflow into Gongzui reservoir in the t^{th} year, c_0, c_1, c_2, c_3 are optimization parameters. The observation data can be found in the literature (Jin, 2000). The amount of the sediment concen- tration into Gongzui reservoir is one of the most important elements in deciding the running of the Gongzui reservoir. The objective function is given as follows:

$$\min \quad f = \sum_{i=1}^{n} [(u_{si} - u_i)^2 \cdot (u_i + \frac{1}{u_i})^2 + \psi(u_{si})] \tag{17}$$

$$\text{and} \quad \psi(u_{si}) = \begin{cases} 0 & u_{si} \geq 0 \\ 500 & u_{si} < 0 \end{cases} \tag{18}$$

where u_i and u_{si} are the observation value and the simulated value of the sediment concentration in Gongzui reservoir in the t_i^{th} year, respectively; n is the number of observation data; $\psi(u_{si})$ is a penalty function.

The objective function f is 317.628 with GHHAGA. For the GHHAGA, the evaluation number of the objective function is 1800. For binary-encoded accelerating genetic algorithm (BAGA) (Jin, 2000), the evaluation number of the objective function is 10500, and the objective function f is 331.21. The result of GHHAGA is given in Table 5. From Table 5, we can conclude that our GHHAGA are highly effective in optimization.

From above two examples, we find that the evaluation number of objective function through GHHAGA is less than those obtained through other GA techniques in optimization. The convergence of GHHAGA has been considerably improved by providing new initial interval information in accelerating cycle step of gray encoding. And its computational amount is significantly reduced. This GHHAGA overcomes the Hamming-cliff phenomena in existing genetic methods.

Table 5. The Calculating Result for the Sediment Concentration in Gongzui Reservoir with GHHAGA

	Parameters				
Evaluation number for f	C_0	C_1	C_2	C_3	Objective function f
0	[-5, 5]	[-5, 5]	[-5, 5]	[-5, 5]	/
1800	-0.918	4.998	-3.362	1.356	317.628

5. Conclusions

In this paper, a new method, GHHAGA, is proposed to solve nonlinear optimization problems in environmental systems. The capability of the global convergence of the corresponding algorithm was analyzed in detail. With the techniques of Gray-encoding, Hooke-Jeeves hybrid searching operator and accelerating cycle being emplyed, the efficiency and accuracy of the proposed algorithm are significantly enhanced compared with those traditional algorithms. GHHAGA has been applied to five nonlinear test functions and two cases of environmental systems optimization.

Based on comparisons, the proposed algorithm presented good performances for optimization of nonlinear systems with continuous variables. It can effectively mitigate the Hamming-cliff phenomena that were common in other genetic methods. As an extension in the future, the proposed algorithm can also be applied in tackling multi-objective optimization problems in environmental systems.

Acknowledgments. This work was supported by Foundation item: National Key Project for Basic Research, No. 2003CB415204.

References

Aluffi, P.F., Parisi, V. and Zililli, F. (1985). Global optimization and stochastic differential equations. *J. Opt. Theor. Appl.*, 47, 1-16.

Anderssen, R.S., Jennings, L.S. and Ryan, D.M. (1972). *Optimization*, University of Queensland.

Andre, J., Siarry, P. and Dognon, T. (2001). An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization. *Adv. Eng. Software*, 32, 49-60.

Bessaou, M. and Siarry, P. (2001). A new tool in electrostatics using a really-coded multipopulation genetic algorithm tuned through analytical test problem. *Adv. Eng. Software*, 32, 363-374.

Cvijovic, D. and Klinowski, J. (1995). Taboo search: An approach to the multiple minima problem. *Sci.*, 267, 664-666.

David, K. and Ward, C. (2003). *Numerical Analysis: Mathematics of Scientific Computing*, China Machine.

Goldberg, D.E. (1989). *Genetic Algorithms: Search, Optimization and Machine Learning*, Addison-Wesley.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan.

Hooke, R. and Jeeves, T.A. (1961). "Direct search" solution of numerical and statistical problems. *J. Ass. Comput. Mach.*, 8, 212-229.

Janikow, C.Z. and Michalewicz, Z. (1991). An experimental comparison of binary and floating point representation in genetic algorithms, in *Proc. of the Fourth International Conference on Genetic Algorithms*, San Francisco, pp. 31-36.

Jin, J.L. and Ding, J. (2000). *Genetic Algorithm and Its Applications to Water Science*, Sichuan, Sichuan University.

Jong, D. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Ph.D. Dissertation, University of Michigan, Ann Arbor, MI, USA.

Leung, Y.W. and Wang, Y.P. (2001). An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Trans., Evol. Comput.*, 5(1), 41-53.

Renders, J.M. and Flasse, S.P. (1996). Hybrid methods using genetic algorithms for global optimization. *IEEE Trans. Syst., Man Cybern. Part B: Cybern.*, 26(2), 243-258.

Rinnoy, K. and Timmer, G.T. (1987). Stochastic global optimization methods. *Math. Program.*, 39, 57-78.

Yang, L.N., Wei, G.M., Guo, K. and Luo, J.S. (2003). Application genetic algorithms with initial group floating for mixed integer nonlinear programming. *Comput. Tech. Geophys. Geochem. Explor*., 25(3), 253-258.

Yang, X.H. (2002). *Study on Parameter Optimization Algorithm and its Application in Hydrological Model*, Ph.D. Dissertation, School of Water Resources and Environment, Hohai University, Nanjing, China.

Yang, X.H., Lu, G.H., Yang, Z.F. and Li, J.Q. (2003). Gray coding based accelerating genetic algorithm and its theory. *Theory Pract. Syst. Eng.*, 23(3), 100-106.

Yang, X.H., Yang, Z.F., Shen, Z.Y. and Li, J.Q. (2004). An ideal interval method of multi-objective decision-making for comprehensive assessment of water resources renewability. *Sci. China Ser. E-Eng. Mater. Sci.*, 47 (Suppl.), 42-50.

Zhong, M. and Sun, S.D. (2001). *Genetic Algorithms: Theory and Applications*, Beijing Defense Industry.