S5–06

Global Optimization for Discrete and Continuous Variables by Updated Random Tunneling Algorithm

Pak Myong Guk[1*], *Pak Sin Myong*[2]

[1]School of Science and engineering, Kim Chaek University of Technology, Pyongyang, DPRK

[2]Faculty of Power Engineering, Kim Chaek University of Technology, Pyongyang, DPRK

[*]Corresponding author: Email: pmg6819@star–co. net. kp

Summary

Recently new global optimization procedure called Random Tunneling Algorithm has been proposed for the problems which have different objective functions. There exist two phases in the tunneling algorithm for global optimization problems, 1) minimization phase and 2) tunneling phase. The local minimum is searched in the minimization phase and the point in a lower valley is searched in the tunneling phase. Branching Generalized Random Tunneling Algorithm (BGRTA) has been applied to the global optimum for the discrete and continuous design variables. By treating the discrete design variables as penalty function, the augmented objective function is constructed. As a result, all design variables can be treated as the continuous design variables.

In this paper, we propose Updated Random Tunneling Algorithm (URTA) for the global optimum for the discrete and continuous design variables. The proposed algorithm enables to find new point reasonably in minimization and tunneling phase. We use the random search which generates the increments according to Cauchy distribution in the minimization phase. Likewise, in the tunneling phase, we generate the increments from the local minimum with different weights according to Cauchy distribution and search the points in the lower valley. The wide range of search becomes possible owing to the property of Cauchy distribution and weight control. We apply the proposed method to some benchmark problems in order to show its effectiveness. The proposed method works very well for these application problems.

Being universal and simple, URTA can also solve the problems which have non–differentiable objective functions.

Keywords: RTA, Global optimization, Discrete variables, Hybrid variables, Tunneling phase

1. Introduction

We often face up to global optimization in the scientific, technological and economical field. Optimization problem could be either constrained or unconstrained according to the existence of constraints. Optimization can also be distinguished by either discrete or continuous variables. Practical relationship between variables is often non–linear, and it is difficult to get the differential information for the objective function and constraints in practice. And the objective function may have not only one valley but also multi–valleys.

We can not apply the public optimization methods such as gradient method or conjugate gradient method to these optimization problems. So Genetic Algorithm [4, 7, 12, 13], Simulated Annealing [15] and Random Tunneling Algorithm [8, 11, 16] are applied to solve these problems.

The study on RTA for global optimization of the multi–valleys object function has long history. There exist two phases in the tunneling algorithm for global minimization problems, 1) minimization phase and 2) tunneling phase. The local minimum is searched in the minimization phase and the point in a lower valley is searched in the tunneling phase. Global optimization is realized by repeating these phases.

Main focus of most studies has been on updating the tunneling phase. For example, new point in a lower valley is searched from the local minimum $\boldsymbol{x}^*$ by finding the solution of tunneling function $h(x)$[11]. It is difficult for this method to find the solution of tunneling function $h(x)$ if the number of local minimums increases. A particular instance is to choose at random the new point approximately according to a Boltzmann distribution, whose temperature T is updated during the algorithm. As $T\rightarrow 0$, such distribution peaks around the global minima of the cost function, producing a kind of random tunneling effect. The motivation for such an approach comes from recent works on the simulated annealing approach in global optimization. In addition, Dynamic Tunneling Algorithm (DTA) may search new point x in a lower valley by solving a differential equation [16]. And Multi–trajectory Dynamic Tunneling Algorithm was proposed to cancel the numerical instability of DTA [6]. There are two approaches in treatment of constraints [8, 9]. The first is to consider the constraints as a penalty function. The other is to adopt new branching phase which is called a constraints–phase in tunneling phase.

All studies which discussed the handling method for hybrid variable used the penalty function for discrete design variables [8~10]. For example, Branching Generalized Random Tunneling Algorithm (BGRTA) converted discrete variables to continuous variables by adopting skillful function and treated that as a penalty function [8, 9]. By treating the discrete design variables as penalty function, the augmented objective function is constructed. As a result, all design variables can be treated as the continuous design variables.

In this paper, we propose a Updated Random Tunneling Algorithm (URTA) to solve the constrained global optimization for hybrid variables.

Section 2 discusses the details of the implementation of the

Updated Random Tunneling Algorithms for constrained optimization of hybrid variables.

Section 3 examines the validity of the proposed method through typical benchmark problems. Here presents the experimental results on 5 benchmark problems. The results are also compared with best–known solutions obtained using earlier Random Tunneling Algorithms implementations.

2. Updated Random Tunneling Algorithm

Main features of this paper are as follows:

1) We proposed new sampling algorithm for variables which exists in minimization and tunneling phase. This enables all constraints to be treated effectively.

2) We proposed new algorithm for treating the hybrid variables.

2.1 Problem specification

The general nonlinear optimization problem which include constraints and uses the hybrid variables can be formulated as follows:

$$\min f(x) = \begin{cases} g_j(x) \le 0, \ j = \overline{1, p} \\ h_j(x) = 0, \ j = \overline{p+1, m} \end{cases}$$

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in R_n \qquad (1)$$

$$a_i \le x_i \le b_i, i = \overline{1, q}$$

$$x_{q+k} \in D_k, D_k = \{d_{k,1}, d_{k,2}, \cdots, d_{k,z(k)}\}, k = \overline{1, n-q}$$

where n–the number of all variables, q–the number of continuous variables, $(n–q)$– the number of discrete variables.

Where, the continuous variables are bounded by the box constraints, and the discrete variables can take only values which are limited in number.

For example, the k^{th} discrete variable can take only a value among the set $D_k=\{d_{k,1}, d_{k,2}, \cdots, d_{k,z(k)}\}$ where $z(k)$ is the number of values for the kth discrete variable. And, g_j, $j\in\{1, \cdots, q\}$ are inequality constraints and h_j, $j\in\{q+1, \cdots, m\}$ are equality constraints which can be transformed to inequality constraints using Eq. (2).

$$\left|h_j(x)\right| - \varepsilon \le 0, j = \overline{p+1,\ m} \qquad (2)$$

Where ε is a small tolerance

2.2 Description of URTA

There exist two phases in the tunneling algorithm for global optimization problems in URTA, 1) minimization phase and 2) tunneling phase. This is as same as the phases in earlier RTA.

The basic difference between URTA and earlier RTA is that we have proposed new sampling algorithm for variables which exists in not only minimization phase and but also tunneling phase. This enables all constraints to be treated effectively.

2.2.1 New sampling algorithm for variables Select_X

Figure 1 shows new sampling algorithm for variables Select_X which can generate a new point $\boldsymbol{x}^{new}$ at random. This new point $\boldsymbol{x}^{new}$ means a variables vector which can satisfy all constraints such as g_j and h_j, and also satisfy the box constraints for the continuous variables. In this algorithm, input data are weight ω and current variables vector $\boldsymbol{x}^{current}$; output data are newly generated variables vector $\boldsymbol{x}^{new}$.

```
Procedure Select_X(ω, x^current; x^new)
0  BEGIN
1      n_itr←0;
2      repeat
3          α←[2×random(0,1)-1]×π/2;
4          x^new←x^current+ω×tan(α);
5      until x^new is feasible
6      if n_itr>20 then
7          x^new←x^current;   goto (15)
8      end if
9      if isDiscrete then
10       ContinuousToDiscrete(x^new)
11     end if
12     if not SatisfyLimit(x^new) then
13         n_itr←n_itr+1;   goto (2)
14     end if
15 END
```

Figure 1. New sampling algorithm for variables Select_X

We set the repeat number n_{itr} to zero (Line 1).

Starting from the current point $\boldsymbol{x}^{current}$, we repeat the search of new point $\boldsymbol{x}^{new}$, until the point can satisfy the box constraints (Line 2–5).

If the discrete values exist in vector $\boldsymbol{x}^{new}$, we must treat the discrete variables. In vector $\boldsymbol{x}^{new}$, the discrete variables take not discrete values, but continuous values.

The function ContinuousToDiscrete($\boldsymbol{x}$) converts the continuous values which are set for all discrete variables in vector $\boldsymbol{x}^{new}$ into the correponding discrete values (Line 9–11).

Then, we examine the hybrid variables $\boldsymbol{x}^{new}$ by SatisfyLimit($\boldsymbol{x}$) whether those can satisfy all constraints.

If all constraints will not be satisfied at new point $\boldsymbol{x}^{new}$, searching a new point is repeated by repeat number n_{itr}(Line 12–14).

If the number of n_{itr} is larger than 20, the search is stoped substituting the given current point $\boldsymbol{x}^{current}$ for new point $\boldsymbol{x}^{new}$ (Line 6–8).

2.2.2 Minimization phase

In the minimization phase, input data are weight ω, minimization number n_{min} and current vector $\boldsymbol{x}^{current}$; output data are the newly selected vector $\boldsymbol{x}^{new}$. Figure 2 shows the algorithm for minimization phase, Minimize.

```
Procedure Minimize(n_min, ω, x^current; x^new)
0  BEGIN
1    for i=1 to n_min
2        Select_X(ω, x^current,x^new)
3        if f(x^new)<f(x^current) then
4            x^current←x^new
5        end if
6    end for
7    x^new←x^current
8  END
```

Figure 2. Algorithm for minimization phase, Minimize.

The point determined through the algorithm Select_X is just the starting point $\boldsymbol{x}^{\text{current}}$ for the minimization phase. The local minimum is searched around the point $\boldsymbol{x}^{\text{current}}$ (Line 2).

If the objective function value $f(\boldsymbol{x}^{\text{new}})$ is less than the current value $f(\boldsymbol{x}^{\text{current}})$, we consider that the point $\boldsymbol{x}^{\text{new}}$ is the new local minimum (Line 3–5). After repeating search of the local minimum by minimization number $n_{\min}$, we consider the final point as the local minimum (Line 1–7).

2.2.3 Tunneling phase

In the tunneling phase, we generate the increments according to Cauchy distribution from the local minimum $\boldsymbol{x}^{\text{current}}$ which is given by minimization phase.

If the point determined by the increments is not lower than $\boldsymbol{x}^{\text{current}}$, search of new points is repeated in a lower valley with decreased weight which corresponds to the decreased radius of search.

Figure 3 shows algorithm for tunneling phase, Tunneling.

```
Procedure Tunneling(n_tnl, x^current; x^new)
0  BEGIN
1      for i=1 to n_step
2         for j=1 to n_tnl
3            Select_X(ω0^i, x^current, x^new)
4            if f(x^new)<f(x^current) then
5               goto (10)
6            end if
7         end for
8      end for
9      x^new ← x^current
10 END
```

Figure 3. Algorithm for tunneling phase, Tunneling

In the tunneling phase, input data are tunneling number n_{tnl} and current variables vector $\boldsymbol{x}^{\text{current}}$; output data are the newly searched variables vector $\boldsymbol{x}^{\text{new}}$. And it needs the weight number n_{step}, and weight values ω_0^i. These data are given as consts.

At first, we search the points in a lower valley around the current point xcurrent with the weight ω_0^i, applying algorithm Select_X (Line 3).

If the objective function value $f(\boldsymbol{x}^{\text{new}})$ is less than the current value $f(\boldsymbol{x}^{\text{current}})$, we consider the point $\boldsymbol{x}^{\text{new}}$ as new point in a lower valley (Line 4–6).

If we can't find the new point in any lower valley, we repeat the tunneling by the tunneling number n_{tnl} (Line 2–7).

If we can't still find the new point in any lower valley after repeating ntnl times, we repeat the upper tunneling process with changed weight value ω_{0i} by the weight number n_{step} (Line 1–8). If we can not find a new point in a lower valley even though we repeated search of the point by tunneling repeat number $n_{step}\times n_{min}$, we consider the current point $\boldsymbol{x}^{\text{current}}$ as the new point $\boldsymbol{x}^{\text{new}}$(Line 1–9).

Finally, URTA combines the minimization phase with the tunneling phase to search the global optimum.

3. Numerical experiments

This section examines the performance of the URTA algorithm on 5 test problems. In fact, these benchmark problems have already been studied by numerous researchers.

We use the minimization weight ω=0.01, minimization number $n_{\min}$=20 and tunneling number n_{tnl}=5 in all examinations. The algorithm is tested on a Core(TM)2 Duo CPU 2.20GHz computer.

3.1 Problem 1(Unconstrained Optimization) [14]

To test the generality of this algorithm, we have solved the unconstrained global optimization problem [14] which contains Bessel function and have two continuous variables. This problem is specified as follows:

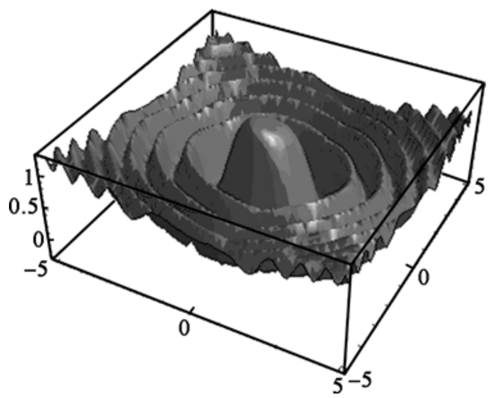
$$\min:\quad f(x,y) = J_0(x^2+y^2) + 0.1|1-x| + 0.1|1-y| \tag{3}$$

where $-\infty\le x\le\infty$, $-\infty\le y\le\infty$, J0 means Bessel function of the first kind.

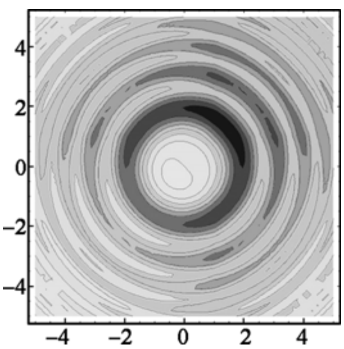
The visual figure of this function is shown in Fig. 4 to look for the global optimum. Fig. 4 a) is three dimensional surface and Fig. 4 b) is its contour. This function is well known as a model function which has a lot of minima and a global minimum (Fig. 4). The results are given in Table 1.

Table 1. Results of the first problem.

Method	x	y	$f(x, y)$	Call number
Randy[14]	1.0	1.660 6	–0.335 6	–
URTA	1	1.660 58	–0.335 59	1 000



a)



b)

Figure 4. Shape of object function [14]
a) 3D surface, b) contour

As shown in the Table 1, URTA finds out the global minimum by function calls of 1 000.

3.2 Problem 2(Constrained Optimization [10])

We solved the global minimum problem [10] of 3 variables with one linear constraint and 3 constraints to test the new constraint–handling technique of this algorithm. This is a practical problem which finds the minimum of mass of coil spring.

The design variables are wire diameter $d(=x_1)$, average diameter of coil $D(=x_2)$ and round number of coil $N(=x_3)$, and theses are continuous in all.

The problem is specified as follows.

$$\min f(x) = (2 + x_3)x_1^2 x_2 \tag{4}$$

Subject to

$$\begin{cases} g_1(x) = 1 - x_2^3 x_3 \big/ (71785 x_1^4) \le 0 \\ g_2(x) = \dfrac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \dfrac{1}{5108 x_1^2} - 1 \le 0 \\ g_3(x) = 1 - 140.45 x_1 \big/ (x_2^2 x_3) \le 0 \\ g_4(x) = (x_1 + x_2)\big/1.5 - 1 \le 0 \end{cases} \tag{5}$$

$$0.05 \le x_1 \le 2.00,\quad 0.25 \le x_2 \le 1.30,\quad 2.00 \le x_2 \le 15.0 \tag{6}$$

This is used as a benchmark problem to verify the constrained optimization as having many minima. The results are given in Table 2.

Table 2. Results of the second problem

Item	Arora[2]	Coello[3]	Hu[5]	URTA
$x_1(d)$	0.053 396	0.051 480	0.051 466	0.051799
$x_2(D)$	0.399 180	0.351 661	0.351 384	0.359 338
$x_3(N)$	9.185 400	11.632 201	11.608 659	11.137 588
$g_1(x)$	0.000 019	–0.002 080	–0.003 336	–0.000 048 1
$g_2(x)$	–0.000 018	–0.000 110	–0.000 110	–0.000 003 4
$g_3(x)$	–4.123 832	–4.026 318	–4.026 318	–4.058 669 2
$g_4(x)$	–0.698 283	–0.731 239	–0.731 324	–0.725 909 7
f_{min}	0.012 730	0.012 705	0.012 667	0.012 666 04
Function Call	900 000	1 291	none	2 000

As shown in Table 2, the global minimum found by the proposed algorithm with function calls of 2000 is lower than earlier global minimum.

3.3 Global optimization for Hybrid variables [9]

We solved three global minimum problems [9] for hybrid variables to test the new discrete variables–handling technique of this algorithm.

3.3.1 Problem 3

The first problem for discrete variables is as follows:

$$\begin{aligned} &\text{Min} \quad f(\boldsymbol{x}) = -x_1 - 1.8x_2 \\ &\text{Subject to} \\ &g_1(\boldsymbol{x}) = x_1^2 + (x_2 + 6)^2 - 85 \le 0 \\ &x_1 \ge 1, x_2 \ge 0; (x_1, x_2 : \text{integer}) \end{aligned} \tag{7}$$

The results are given in Table 3.

Table 3. Results of the third problem

Item	Continuous		Discrete	
	Kitayama[9]	URTA	Kitayama[9]	URTA
x_1	4.477	4.481 36	6	6
x_2	2.059	2.057 13	1	1
$g_1(x)$	–	–9.439e–05	0	0
f_{min}	–8.183 2	–8.184 204	–7.8	–7.8

As shown in Table 3, the continuous global minimum found by the proposed algorithm is lower than earlier global minimum [9] and the discrete global minimum is as same as earlier global minimum [9].

3.3.2 Problem 4

The second problem for discrete variables is as follows:

$$\begin{aligned} &\text{Min} \quad f(\boldsymbol{x}) = -1.1x_1 + x_2 \\ &\text{Subject to} \\ &g_1(\boldsymbol{x}) = x_1 - x_2 + 1 \le 0 \\ &g_2(\boldsymbol{x}) = -4x_1^2 + 28x_1 - x_2 - 40 \le 0 \\ &0 \le x_1 \le 5,\ 1 \le x_2 \le 8 \end{aligned} \tag{8}$$

The discrete values of every design variables are at intervals of 0.5.

We can find the global minimum 0.5 at the point (x_1=5, x_2=6) as like as earlier global minimum [9].

3.3.3 Problem 5(Pressure vessel problem [9])

The global optimization of pressure vessel in Fig. 5 is widely known in global optimization for hybrid variables.

The design variables in pressure vessel are the radius of pressure vessel R(continuous), length L(continuous), thickness T_s, T_h(both are discrete). The object function is the total cost to manufacture and the geometrical meaning of these variables are shown in Fig. 5.

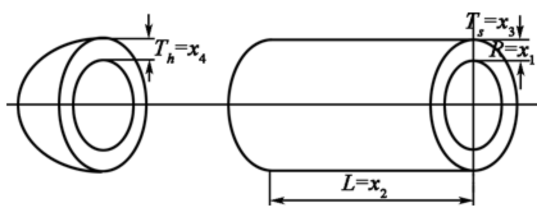


Figure 5. Design variables in pressure vessel

This problem is specified as follows:

$$f(x) = 0.622\,4x_1x_2x_3 + 1.778\,1x_1^2x_4 + 3.166\,1x_2x_3^2 + 19.84x_1x_3^2 \tag{9}$$

Subject to

$$\left.\begin{aligned} g_1(x) &= 0.019\,3x_1/x_3 - 1 \le 0 \\ g_2(x) &= 0.009\,54x_1/x_4 - 1 \le 0 \\ g_3(x) &= x_2/240 - 1 \le 0 \\ g_4(x) &= \left(1\,296\,000 - \frac{4}{3}\pi x_1^3\right)\Big/\left(\pi x_1^2 x_2\right) - 1 \le 0 \end{aligned}\right\} \quad (10)$$

$$25 \le x_1 \le 150,\ \ 25 \le x_2 \le 240,\ \ 0.0625 \le x_3, x_4 \le 1.25 \quad (11)$$

The discrete variables should have the values at intervals of 0.0625 inch based on ASME.

Table 4 shows the global minimum found by proposed algorithm compared with that by others. Table 4 uses inch as unit to compare directly with earlier results.

Table 4. Results of the fifth problem

Item	Arakawa[1]	URTA*	Kitayama[9]	URTA**
R, in	38.858	38.860 1	38.880	38.879 2
L, in	221.402	221.365	220.893	220.956
T_s, in	0.750	0.750	0.750	0.750
T_h, in	0.375	0.375	0.375	0.375
$g_1(x)$	−0.000 05	−1.6E−08	0.000 51	0.000 49
$g_2(x)$	−0.011 45	−1.1E−02	−0.010 89	−0.010 90
$g_3(x)$	−0.077 49	−7.7E−02	−0.079 61	−0.079 35
$g_4(x)$	−0.000 02	−4.0E−08	0.000 75	0.000 49
$f_{\min}$	5 850.77	5 850.38	5 846.30	5 845.45

(*−ε=0, ** −ε=5×10^{-4})

As shown in Table 4, the global minimum found by the proposed algorithm is lower than earlier global minimum [1] in case of ε=0.

We can also find that the proposed algorithm found a new updated global minimum lower than earlier global minimum [9] in case of ε=5×10^{-4}.

4. Conclusions

In this paper, we proposed Updated Random Tunneling Algorithm (URTA) for the global optimum of the discrete and continuous design variables and applied the proposed method to some benchmark problems in order to show its effectiveness.

The research results show that:

(1) The proposed algorithm works very well for all benchmark problems. Overall, the results of these 5 numerical experiments show that the proposed algorithm is efficient and reliable.

(2) Being universal and simple, Updated Random Tunneling Algorithm (URTA) can also solve the global optimization problems which have non−differentiable objective functions and constraints.

Acknowledgements

The author wish to acknowledge Professor Kang Il Yong of Kim Chaek University of Technology for his contribution during the numerical experiments.

References

1. Arakawa, et al., 2003. Genetic Range Genetic Algorithm to Obtain Quasi−Optimum Solutions, ASME/DETC/DAC, Paper No. 48800, in CD−ROM.

2. Arora J. S. 1989. Introduction to Optimum Design, pp 23−56, McGraw−Hill, New York,

3. Coello Coello C.A., 2000, Use of a Self−Adaptive Penalty Approach for *Engineering Optimization Problems*, *Computers in Industry*, 41, 3−127.

4. Jorng Tzong Horng, YingPing Chen, ChengYan Kao, 2001, A Theoretical Aspect of a Stochastic Sketching Method for Global Optimization, *J. of Information Science and Engineering*, 17, 47−71.

5. Hu X. H. et. al. 2003, Engineering Optimization with Particle Swarm, *IEEE Swarm Intelligence Symposium*, 53− 57.

6. Keiichiro Yasuda, Takahiko Kanazawa, 2004, Multi−Trajectory Dynamic Tunneling Algorithm for Global Optimization, *IEEJ Trans. EIS*, 124(7), 1 483−1 493. (in Japanese)

7. Kento Uemura et al., 2013. Big−valley Explorer: A Framework of Real−coded Genetic Algorithms for Multi−funnel Function Optimization, *Transactions of the Society of Evolutionary computation*, 4(1), 1−12 (in Japanese)

8. Kitayama S., Yamazaki K., 2003. Global Optimization by Generalized Random Tunneling Algorithm (1st report: Proposalof Algorithm and Numerical Examples), Nihon Kikai Gakkai Ronbunshu A (*Transactions of the Japan Society of Mechanical Engineers*, *Series A*), 694(69), 1 250−1 256.(in Japanese).

9. Kitayama S., Yamazaki K., 2006, Global Optimization by Generalized Random Tunneling Algorithm (4th report), *Trans. of the Japan Society of Mechanical Engineers*, *Series A*, 695(70), 970–977. (in Japanese)

10. Kitayama S., Arakawa M., Yamazaki K., 2007, Global Optimization by Generalized Random Tunneling Algorithm (5th Report, Approximate Optimization Using RBF Network), *Trans. of the Japan Society of Mechanical Engineers*, *Series C*, 729(73), 1 299–1 306. (in Japanese)

11. Levy A. V. and Montalvo A., 1985, The Tunneling Algorithm for the Global Minimization of Functions; *SIAM J. Sci. Stat. Comput.*, 6(1), 15−29.

12. Omar Al Jadaan, Lakshmi Rajamani, C. R. Rao, 2005. Adaptive penalty function for solving constrained evolutionary optimization, *Journal of Theoretical and Applied Information Tech*nology, 339−351., see www.jatit.org

13. Rahila Patel et al., 2010. Multi−Objective Optimization Using Multi Parent Crossover Operators, *J. of Emerging Trends in Computing and Information Sciences*, 2(2), 99−105

14. Randy L. Haupt, Sue Ellen Haupt, 2004. Practical genetic algorithms, A John Wiley & Sons, Inc., pp 205−210

15. Renpu GE, 1990. A filled fuction method for finding a global minimizer of a function of several variables, *Mathematical Programming*, 46, 191−204.

16. Yao Y., 1989, Dynamic Tunneling Algorithm for Global Optimization; *IEEE Trans. on Systems*, *Man & Cybernetics*, 37, 19(5), 1 222−1 230.