

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Chaotic Differential Evolution Algorithm for Solving Constrained Optimization Problems

Zhiyong Li and Xiang Wang
Zhengzhou Institute of Aeronautical Industry Management, Countrs, China

Abstract: A novel chaotic differential evolution algorithm is proposed to solve constrained optimization problems by combining differential evolution algorithm, chaos optimization and DEB comparing rules. Chaos initialization function and chaotic local search function are designed in terms of the new algorithm by random, ergodic and no-period behavior of chaos. The two functions which makes the global optimization of DE algorithm be improved, are imbedded into the framework of DE algorithm. Meanwhile, DEB comparing rules are applied to chaotic differential evolution algorithm to solve the constraints. Simulation results show that the new algorithm works well and robustly in solving bump problem.

Key words: Constrained optimization problems, differential evolution algorithm, chaos optimization, chaotic differential evolution algorithm

INTRODUCTION

Constrained Optimization Problems (COPs) (Yong *et al.*, 2009) is a basic research problem in the field of engineering and operations research. When solving COPs with the traditional mathematical method, it often falls into local optimum. Therefore, the usage of evolutionary algorithms for solving COPs becomes a research hotspot. In order to validate the algorithm, we usually use 13 Benchmark function (hereinafter “13 functions”) which was summarized by Runarsson and Yao (2000). However, it doesn't work well when we solve the second function, Bump, with most of algorithms.

Differential Evolution (DE) algorithm (Storn and Price, 1997) is a new evolutionary algorithm which has comprehensive ability for optimization and rapid convergence properties. As a global optimization algorithm because of lack of constraint handling mechanism, it can not be directly applied to COPs. Scholars use different ways to improve it in order to make it apply to COPs, (Mezura-Montes *et al.*, 2005) and (Lampinen, 2002) has tried to introduce constraint handling mechanism in DE algorithm, to solve COPs's standard 13 functions but the result shows that the solving quality of Bump problem is poor. Yichao and Xizhao (2008) and Bingyu *et al.* (2004) use modified DE algorithm only to solve the Bump problem get the better effect. However, it still has the problem that the standard deviation is larger and the robustness of solving quality is not strong.

In order to successfully solve optimization problem of Bump function, present study improve DE algorithm based on the following two basic ideas: Firstly, the initial population of DE algorithm is initialized by using chaos sequence, population distribution can be made more uniform, hence it is more likely to find the global optimum solution; secondly, chaotic local search approach can improve DE's search performance. Based on the above ideas, chaotic differential evolution algorithm is proposed for solving bump problem. This algorithm makes improvement in four areas in the framework of basic DE algorithm: first, the initial population of DE algorithm is initialized by using chaos vector sequence; Secondly, the introduction of DEB comparison criteria enables the algorithm to have constraint handling capacity; Thirdly, When the optimal solution of DE algorithm is not updated after several generations, using chaotic local search algorithm helps DE algorithm to jump out the local optimal solution; Finally, at regular intervals of DE algorithm generation, using chaos initialization vector sequence to replace half worse solutions of the original population. The experimental results show that chaotic differential evolution algorithm can successfully solve optimization problem of bump function and its robustness is stronger.

DEFINITION OF THE QUESION

The general form of the COPs problems are as follows:

$$\begin{aligned} \min \quad & f(X) \quad X \in D \quad (1) \\ \text{s.t.} \quad & \begin{cases} g_i(X) \leq 0 & i = 1, 2, \dots, q \\ h_k(X) = 0 & k = q+1, q+2, \dots, m \end{cases} \end{aligned}$$

where,

$$D = \prod_{i=1}^{Dim} [a_i, b_i] \subseteq \mathbb{R}^{Dim}, X = (x_1, x_2, \dots, x_{Dim}) \text{ and } x_i \in [a_i, b_i]$$

Dim is scale of problem; $S = \{X | X \in D \wedge g_i(X) \leq 0 \wedge h_k(X) = 0\}$ is the feasible region; $f : D \rightarrow \mathbb{R}$ is called objective function, inequality $g_i(X)$ is inequality constraint conditions; $h_k(X)$ is equality constraint conditions.

Bump problem is a specific COPs in this study, it is 2nd function in the 13 Benchmark what Runarsson (2) summarized, the concrete form as the following:

$$\begin{aligned} \min f(X) = & \left| \frac{\sum_{i=1}^n \cos^4(x_i) - \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}} \right| \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^n x_i \leq 7.5n \\ \prod_{i=1}^n x_i \geq 0.75 \\ 0 \leq x_i \leq 10 \quad i = 1, 2, \dots, n \end{cases} \end{aligned}$$

where, X expresses a feasible solution $X = (x_1, x_2, \dots, x_n)$, x_i is a component of feasible solution.

CHAOTIC DIFFERENTIAL EVOLUTION OF SOLVING CONSTRAINT OPTIMIZATION PROBLEM

The basic thought of chaotic differential evolution algorithm is to use the three major features of chaos sequences-the randomness, ergodicity and regularity (Gong and Shaoqian, 2000), to improve differential evolution algorithm and to enhance its search capabilities; at the same time, introducing DEB comparison criteria to the modified algorithm can make it have constraint handling capacity (Fig. 1). Chaos optimization improve the performance of differential evolution in three aspects: Firstly, according to ergodicity and randomness of chaos sequences, the initial population of DE algorithm which replaces the initial population of the random generation is generated to make distribution of initial population more uniform; Secondly when the optimal solution of populations in DE algorithm is not updated after several generations' continuous evolution, the chaotic local search is used around the optimal solution to enhance the optimal solution's quality of population according to the ergodicity of chaos sequence; Thirdly, assuming

Criterion 1: When two solutions both are feasible solutions, the smaller solution of objective function wins.
 Criterion 2: When a solution is feasible solution, the other solution is non-feasible solution, the feasible solution wins.
 Criterion 3: When two solutions both are non-feasible solutions, the smaller sum of restraint violates wins.

Fig. 1: Three comparison criterions of DEB

population size of DE algorithm is NP, at regular intervals algebra in DE algorithm, the chaos sequence generate NP/2 new individual; Then the half individuals of current population which the fitness function is poor can be replaced by those new generation. In this case, we can maintain the diversity of population.

DEB comparison criterion: The goal of DE algorithm is to solve global optimization problems, because of lack of constraint handling capability; it can not be directly used to solve COPs. In order to make genetic algorithm have constraint capability, Deb (2000) invented DEB comparison criterion, it is a method that the two individuals compare advantages and disadvantages in the constraint condition. This criteria is used to improve the chaotic DE algorithm and is applied to algorithm with the two individuals' comparison all related operation which make the algorithm have constraint handling capability.

The basic implementation ideas of DEB comparison criterion are as follows: first, all of constraints violation degree is added which is defined as the sum of constraint violation degree; second when designing two individuals comparison, a feasible solution priority strategies is used. This specific strategy can be expressed as the following three comparison criterions:

According to COPs, functional forms of the sum of constraint violation degree and DEB comparison criteria are as follows.

Definition 1 (sum of constraint violation degree): $G(X)$ is the sum of constraint violation degree for an individual X in the population, the form is:

$$G(X) = \sum_{j=1}^m G_j(X) \quad (2)$$

where,

$$G_j(X) = \begin{cases} \max\{0, g_j(X)\}, & 1 \leq j \leq q \\ \max\{0, |h_j(X) - \delta_j|\}, & q+1 \leq j \leq m \end{cases}$$

where, δ is relaxation degree of equality constraint. We don't discuss the settings of δ here because that the problem about g 02 doesn't have equality constraints.

Definition 2 (function form of DEB comparison criteria): Prior(X1, X2) express the results that the two solutions X1 and X2 are compared using DEB comparison criteria, the form is:

$$\text{prior}(X_1, X_2) = \begin{cases} f(X_1) < f(X_2), & G(X_1) = G(X_2) = 0 \\ G(X_1) < G(X_2), & G(X_1) > 0 \wedge G(X_2) > 0 \\ 0, & G(X_1) > 0 \wedge G(X_2) = 0 \\ 1, & G(X_1) = 0 \wedge G(X_2) > 0 \end{cases} \quad (3)$$

Differential evolution algorithm of solving COPs problem: Differential evolution algorithm is a global optimization algorithm proposed by Storn and Price (1997). Because of his outstanding performance in the first session of IEEE evolutionary programming contest, DE algorithm has become a new research focus in the field of evolutionary computing. As same as the genetic algorithm, DE algorithm is based on population evolutionary and also contains three operators of mutation, intersect and selection. However, DE algorithm is based on real valued coding different from genetic algorithm based on binary coding. Compared with global optimization problem, mutation and intersection operator remain unchanged to solve COPs in the modified DE algorithm. The algorithm only improves the selection operator by using DEB comparison criteria to make it have the constraints handling capacity.

Assuming the form of COPs is same to, population size of DE algorithm is NP. The following is the definition of the three operators in the improved algorithm.

Definition 3 (mutation operator of DE algorithm): Select three variables from the current population randomly $\{X_i | X_i = (x_{i,1}, x_{i,2}, \dots, x_{i, \text{Dim}}), i=1, 2, \dots, NP\}$, one is as disturbance variables, the other two subtractions are as difference vector. Variance vector is gotten by the following formula.

$$V_i^{G+1} = X_{i1}^G + F * (X_{i2}^G - X_{i3}^G) \quad (4)$$

where, X_i denotes arbitrary solution vector of the population, Dim denotes problem scale, $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$ and $r_1 \neq r_2 \neq r_3 \neq$, NP is the population size, F is the scaling factor, G is the algebra of current evolution.

Definition 4 (intersection operator of DE algorithm): The ith vectors V_i^{G+1} generated by mutation and the ith vectors X_i^G of original population are executed intersection operator according to the following formula (5), to get the candidate vector U_i^{G+1} .

$$U_{i,j}^{G+1} = \begin{cases} V_{i,j}^{G+1}, & \text{if } \text{rnd} \leq \text{CR} \text{ or } j = j_{\text{rand}} \\ X_{i,j}^G, & \text{otherwise} \end{cases} \quad (5)$$

where, $j \in \{1, 2, \dots, \text{Dim}\}$, Dim said problem scale, CR is the crossover probability, rnd is the random number on [0,1] uniform distribution, j_{rand} is an integer selected randomly on the $\{1, 2, \dots, \text{Dim}\}$.

Definition 5 (selection operator of DE Algorithm): The ith candidate vector U_i^{G+1} and the ith vectors X_i^G of original population are executed selection operator according to the following formula (6), to get the individual X_i^{G+1} of next generation with better fitness.

$$X_i^{G+1} = \begin{cases} U_i^{G+1} & \text{if } \text{prior}(U_i^{G+1}, X_i^G) = 1 \\ X_i^G & \text{otherwise} \end{cases} \quad (6)$$

where, X_i^{G+1} indicates the (G+1) th generation and ith individual of populations. We should pay attention that the comparison of two individuals use DEB comparison criteria.

Chaotic Optimization of solving COPs problem: Chaos is a general phenomenon in nature and is a powerful tool to express nonlinear systems. Its main feature is that it traverses all the states without repetition according to its own "law" within a certain range. It can also be expressed as the randomness, ergodicity and regularity of chaotic motion. Present study just tries to improve DE evolutionary algorithm to enhance its global search capability based on these features.

The common chaos model mainly has the following two forms:

$$y_{k+1} = \mu y_k (1 - y_k) \quad (\text{Robert, 1976}) \quad (7)$$

$$y_{k+1} = \alpha y_k - 2 \tanh(\beta y_k) \exp(-3y_k^2) \quad (\text{Li Jiang and Tian Lun, 2002}) \quad (8)$$

where, k said iterations, $k \in \{1, 2, \dots, K\}$, K is Maximum number of iterations. Equation 7 is called the Logistic

```

Begin
For i = 1 To N Do
Initialize solution vector  $Y_{i,0} = (y_{i,1,0}, y_{i,2,0}, \dots, y_{i,Dim,0})$ , where  $y_{i,j,0}$  is random number obeying even distribution on  $[0,1]$  and  $y_{i,j,0} \neq 0, 0.25, 0.5, 0.75, 1$   $j \in \{1, 2, \dots, Dim\}$ ;
For k = 0 To kIter Do
Iteratively produce kth generation chaotic vector  $Y_{i,k}$  by using formula (9);
End For
Using formula (11), map chaotic vector  $Y_{i, kIter}$  of KIter th generation to domain vector  $Z_i$ ;
End For
Return (the set  $\{Z_i | i = 1, 2, \dots, N\}$  which contains N solution vectors
End
    
```

Fig. 2: Pseudo code of chaotic initialization

mapping, $\mu = 4$, $y_1 \in (0,1)$ and $y_1 \neq 0.25, 0.5, 0.75$ in formula (7). $y_1 \in (0,1)$, $\alpha \in (0,1)$, $\beta = 2$ in Eq. 8.

The Eq. (7) and (8) is the variable form of chaos equation; however, vector representation which is used in the DE algorithm is as follows after the transformation.

$$Y_{k+1} = \mu Y_k (1 - Y_k) \tag{9}$$

$$Y_{k+1} = \alpha Y_k - 2 \tanh(\beta Y_k) \exp(-3 Y_k^2) \tag{10}$$

where, $Y_k = (y_{1,k}, y_{2,k}, \dots, y_{Dim,k})$ said kth generation chaos vector, corresponding to the solution vectors of DE algorithm. Dim said the problem dimension. k said iterations; $y_{i,1} \in (0,1)$ and $y_{i,1} \neq 0.25, 0.5, 0.75$ in formula (9), $y_{i,1} \in (0,1)$, $i \in \{1, 2, \dots, Dim\}$ in formula (10); the parameter μ in formula (9) is same as in formula (7), $\mu = 4$, α and β in Eq. 10 is the same as in Eq. (8), $\alpha \in (0,1)$, $\beta \neq 2$.

Chaotic initialization population: Storn pointed out that if the initialization population can be the uniform distribution as far as possible, then the DE algorithm is more possible to find the global optimal solution (Storn and Price, 1997). Corresponding to it, the chaos variable has randomness and ergodicity. Therefore this study attempts to get the chaos sequence by using Eq. 9 and maps it into the initial population which satisfies the definition domain of DE algorithm.

The basic realization idea of chaotic initialization population is as follows: first, randomly initialize a vector Y_0 that dimension is Dim; second, by using Logistic mapping (Eq. 9), iterate the vector to arrive KIter generation to get vector Y_{KIter} ; third, because each element of vector Y_{KIter} which is in the range (0, 1) and not within the definition domains, is not a real solution, vector Y_{KIter}

is mapped to solution vector Z of definition domain; last, repeat the above three steps N times, to get the population containing N solution vectors.

Suppose to solve COPs described in 1st part. Giving the lowerbound of solution vector $A = (a_1, a_2, \dots, a_{Dim})$, the upperbound is $B = (b_1, b_2, \dots, b_{Dim})$. Chaotic vector Y which is generated by the formula (9), transforms into the vector Z in the definition domain by using the e Eq. 11.

$$Z = A + Y(B - A) \tag{11}$$

Based on the above ideas, the pseudocode of chaos_init function which defines population containing N elements generated by chaotic initialization is shown in Fig. 2.

Chaotic local search: Chaotic local search execute chaos variation around the optimal solution by using traversal characteristics of chaos sequences and expect to find the better solution to guide the evolution to move forward. Yue and Guanzheng (2009) pointed out that Eq. 8 is better than Eq. 7 when the chaotic mutation are executed, so Eq. 10 which is corresponding to Eq. 8 is selected as a generation formula for chaos vector.

Based on the above discussion, the implementation idea of chaotic local search algorithm is first to initialize random vector Y_0 that dimension is Dim; then to produce iteratively a sequence $\{Y_i | i=1, 2, \dots, M\}$ containing M chaos vectors by using formula (10); then to add these M vectors to the optimal solution vectors X_{best} of population to get variation vector sequence $\{Z_i | i=1, 2, \dots, M\}$; Finally, to find the optimal vector Z_{best} amount of $\{Z_i\}$ according to DEB comparison criteria and return it.

```

Begin
Zbest = Xbest;
Initialize solution vector  $Y_0 = (y_{0,1}, y_{0,1}, \dots, y_{0,Dim})$ , where  $y_{0,j}$  is random number obeying even distribution on [0,1] and  $j \in \{1,2,\dots,Dim\}$ ;
For I = 1 To M Do
Generate chaotic vector  $Y_1$  according to formula (10);
Use equation  $Z_1 = X_{best} + Y_1$  to execute chaotic mutation and to generate mutation vector  $Z_1$ ;
If prior ( $Z_1, Z_{best}$ )
 $Z_{best} = Z_1$ ;
End If
End For
Return ( $Z_{best}$ );
End
    
```

Fig. 3: Pseudo code of chaotic local search

Assumed to solve COPs problem of part 1, according to above idea, pseudo-code of chaotic local search function is as Fig. 3.

Chaotic differential evolution algorithm of solving COPs

problem: The implementation method solving chaotic differential evolution algorithm of COPs is, first, to generate the initial population of DE algorithm by using chaotic initialization function *chaos_init*; second, to execute mutation, crossover and selection operations of DE evolutionary algorithm; in the implementation process of DE algorithm, if the optimal solution of population is not updated when it evolve the certain generations, then the population is optimized by using local search function *chaos_search*; at the same time, at evolutionary intervals generation, the new initial solution set which is generated by using chaotic initialization function *chaos_init* replace the half worse solution of population.

Pseudo-code is shown in Fig. 4.

EXPERIMENTS

Benchmark function is 20 dimensions Bump problem which is introduced earlier.

Present study carries out independently 20 times improved chaotic DE algorithm to solve Bump problem (Table 1).

Where Dim expresses the dimension of Bump function; NP expresses population's scale; F expresses scaling factor; CR expresses the crossover probability;

MAX_GEN expresses the maximum times of iterations for DE algorithm; CIter expresses the algebra which the population optimal solution is not updated continuously in the condition of triggering chaotic local search criteria; FIter expresses the operation frequency that the population is replaced through carrying on chaotic initialization; μ is a parameter of chaos vector equation formula (9); α and β is parameters of chaos vector equation formula (10); KIter expresses iteration times of chaos equation which generate finally chaos vector in chaotic initialization function *chaos_init*; M expresses number of the chaotic variation vector which is generated around optimal solution in chaotic local search function *chaos_search*.

The detailed information of optimal solution after carrying out 20 times experiments is shown by histogram in Fig. 5. We can clearly see from the Fig. 5 that all 20 times tests obtain known optimal solution -0.803619.

In order to verify validity of the chaotic DE algorithm, Table 2 compares it to solve the quality of bump functions with other algorithms. Other algorithms include:

- The improved version of the Stochastic Ranking (IRY) (Runarsson and Yao, 2005)
- The Diversity Differential Evolution (Diversity-DE) (Mezura Montes *et al.*, 2005)
- Modified Differential Evolution (MDE) (Yichao and Xizhao, 2008)
- Combining Particle Swarm Optimization with Differential Evolution (PSODE) (Bingyu *et al.*, 2004)

```

Begin
Generate initial popular which size is NP by using function chaos_init; /*chaotic initialization*/

For G = 1 To MAX_GEN Do
For I = 1 To NP Do

Generate mutation vector  $V^G_i$  by using mutation operator; /*DE mutation*/

Generate intersection vector  $U^G_i$  by using intersection operator; /*DE intersection*/

Generate population vector  $X^G_i$  by using selection operator; /*DE selection*/

End For

If the optimal solution of population is not updated when it evolves CIter generation

Then execute chaotic search to G-th generation optimal solution  $X^G_{best}$  by using chaotic
search function, the result exist in ; /*chaotic local searching*/

If prior ( $Z^G_{best}$ ,  $X^G_{best}$ ) Then
 $X^G_{id} = Z^G_{best}$ , where id is a random integer in [1, NP]; /* random replacement */
End If
End If

If number G is an integral multiple of FIter /*execute chaotic initialization replacement
at FIter intervals algebra */

Then generate initial population  $\{Y^G_i | i = 1, 2, \dots, NP/2\}$  which size is NP/2 by using chaos_init;

Replace NP/2 worse solutions of G-th generation population  $\{X^G_i | i = 1, 2, \dots, NP\}$  by using solution set;

 $\{Y^G_i | i = 1, 2, \dots, NP/2\}$ ;

End If

End For

End
    
```

Fig. 4: Pseudo code of chaotic differential evolution

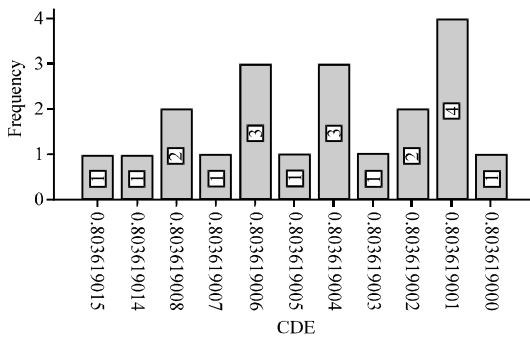


Fig. 5: Column chart of the results in 20 independent experiments for Chaotic DE

In the four algorithm participating comparison, IRY algorithm (Runarsson and Yao, 2005) makes the

Table 1: Parameter settings of Chaotic DE

Parameter setting												
Fuction	Dim	NP	F	CR	MAX GEN	CIter	FIter	μ	α	β	KIter	M
Bump	20	200	0.5	0.9	6000	100	2000	4	0.5	2.7	300	150

Table 2: The comparison results with other papers

Items of comparison	Algorithm				
	CDE	IRY	Diversity-DE	MDE	PSODE
Best	-0.803619	-0.803619	-0.803619	-0.803619	-0.803619
Mean	-0.803619	-0.782715	-0.798079	-0.803619	-0.803604
Worst	-0.803619	-0.723591	-0.751742	---	---
Std	4.2E-9	2.2E-2	1.01E-2	0.000201	0.0002

improvement to classics RY algorithm (Runarsson and Yao, 2000), it is the most competitive to solve COPs at present; In order to have the comparability, the other three are all DE algorithm, Diversity-DE are the most

effective DE algorithm to solve COPs, MDE and PSODE are DE algorithm which are designed only for bump function.

As can be seen from Table 2, chaotic difference evolution algorithm is the best in five algorithms to solve Bump function. Compared with IRY algorithm and Diversity-DE algorithm which solve the standard 13 problem of COPs, the algorithm in this study is that not only the mean of optimal solution and the worst solution is prior but also the variance of optimal solution is smaller. It indicates that this algorithm to solve Bump problems has high quality and stability. MDE algorithm and PSODE algorithm which are designed only for Bump function optimization problem is higher quality but the method adopted in the article also has advantages in comparison with them, especially in terms of standard deviation, it indicates that stability of this algorithm is stronger.

CONCLUSION

An novel difference evolution algorithm based on the chaotic optimization is proposed in this study. The new algorithm which solves successfully Bump function, uses the DEB comparison criterion to process constraint. The simulation results show that: according to the Bump function, the new algorithm is very competitive and robustness.

Two chaotic optimization operators do not depend on the specific form of DE algorithm. Therefore, we will focus on combine the two operators with other forms of DE algorithm to solve the constrained problem in the future.

REFERENCES

- Bingyu, L., X. Yunshi and W.U. Qidi, 2004. Hybrid algorithm based on particle swarm optimization for solving constrained optimization problems. *Control Decis.*, 19: 804-807.
- Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Comput. Methods Applied Mech. Eng.*, 186: 311-338.
- Gong, L. and L. Shaoqian, 2000. Chaotic spreading sequences with multiple access performance better than random sequences. *IEEE Trans. Circuits Syst. I: Fundam. Theor. Appl.*, 47: 394-397.
- Lampinen, J., 2002. A constraint handling approach for the differential evolution algorithm. *Proceedings of the Congress on Evolutionary Computation*, May 12-17, Honolulu, HI, USA., pp: 1468-1473.
- Li-Jiang, Y. and C. Tian-Lun, 2002. Application of chaos in genetic algorithms. *Commun. Theor. Phys.*, 38: 168-172.
- Mezura-Montes, E., J. Velazquez-Reyes and C.A.C. Coello, 2005. Promising infeasibility and multiple offspring incorporated to differential evolution for constrained optimization. *Proceedings of the Conference on Genetic and Evolutionary Computation, (CGEC'05)*, Washington, pp: 225-232.
- Robert, M.M., 1976. Simple mathematical models with very complicated dynamics. *Nature*, 261: 459-467.
- Runarsson, T.P. and X. Yao, 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.*, 3: 284-294.
- Runarsson, T.P. and X. Yao, 2005. Search biases in constrained evolutionary optimization. *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.*, 35: 233-243.
- Storn, R. and K. Price, 1997. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, 11: 341-359.
- Yichao, H. and W. Xizhao, 2008. Solution of hard constrained optimization problem based on modified differential evolution algorithm. *Comput. Eng.*, 34: 193-194.
- Yong, W., C. Zixing, Z. Yuren and X. Chixin, 2009. Constrained optimization evolutionary algorithms. *J. Software*, 20: 11-29.
- Yue, T. and T. Guanzheng, 2009. Differential evolution algorithm with chaotic-local-search strategy or global optimization. *Comput. Eng. Appl.*, 45: 15-17.