A Novel Differential Evolution with Uniform Design for Continuous Global Optimization

Lei Peng
School of Computer, China University of Geosciences, Wuhan, China
College of Computer Science, Huazhong University of Science and Technology, Wuhan,China
Email:penglei0114@gmail.com

Yuanzhen Wang
College of Computer Science, Huazhong University of Science and Technology, Wuhan,China
Email:wangyz2005@163.com

Guangming Dai
School of Computer, China University of Geosciences, Wuhan, China
Email: gmdai@cug.edu.cn

Zhongsheng Cao
College of Computer Science, Huazhong University of Science and Technology, Wuhan,China
Email: caozhongsheng@126.com

Abstract—**Differential Evolution (DE) is a simple and efficient optimizer, especially for continuous global optimization. Over the last few decades, DE has often been employed for solving various engineering problems. At the same time, the DE structure has some limitations in the complicated problems. This fact has inspired many researchers to improve on DE by proposing modifications to the original algorithm. Population initialization is very important to the performance of differential evolution. A good initialization method can help in finding better solutions and improving convergence rate. In this paper, a uniform-differential evolution algorithm (UDE) is proposed. It incorporates uniform design initialization method into differential evolution to accelerate its convergence speed and improve the stability. UDE is compared with other four algorithms of Standard Differential Evolution (SDE), Orthogonal Differential Evolution (ODE), Opposition Based Differential Evolution(OBDE) and Chaos Differential Evolution(CDE). Experiments have been conducted on 23 benchmark problems of diverse complexities. The results indicate that our approach has the stronger ability and higher calculation accuracy to find better solutions than other four algorithms.**

Index Terms—**differential evolution, global optimization, uniform design method, orthogonal design method, Opposition Based, Chaos Initialization**

I. INTRODUCTION

Global optimization is the task of finding the absolutely best set of parameters to optimize an objective function. Generally, there are solutions that are locally optimal but not globally optimal. Consequently, global optimization problems are typically quite difficult to solve exactly. Using classical determinate direct search techniques may fail to solve such problems because these problems usually contain multiple local optima.

The problem of finding a global minimum of the unconstrained optimization problem:

$$\min_{x \in R^n} f(x)$$

Where f is a generally nonconvex, real valued function defined on R^n.

In recent years, the use of alternative approaches to solve complex optimization problems is very common. Evolutionary Algorithms (EAs) such as genetic algorithm, evolutionary programming, evolution strategy and genetic programming have received many interests from researchers and practitioners due to their competitive results when solving this kind of problems.

Differential Evolution (DE) is a branch of evolutionary algorithms developed by Rainer Storn and Kenneth Price [1] for global continuous optimization problem. It has won the third place at the 1st International Contest on Evolutionary Computation. It shares similarities with previous EAs. For example, DE works with a population of solutions, called vectors, it uses recombination and mutation operators to generate new vectors and, finally, it has a replacement process to discard the less fit vectors. DE uses real encoding to represent solutions. Some of the differences with respect to other EAs are the following: DE uses a special mutation operator based on the linear combination of three individuals and a uniform crossover operator. It has several attractive features. Besides being an exceptionally simple evolutionary strategy, it is significantly faster and robust for solving numerical optimization problem and is more likely to find the functions true global optimum.

Despite having several striking features and successful applications to different fields, DE has sometimes been shown slow convergence and low accuracy of solutions

© 2012 ACADEMY PUBLISHER
doi:10.4304/jcp.7.1.3-10

when the solution space is hard to explore. Many efforts have been made to improve the performance of DE and many variants of DE have been proposed.

The first direction for improvement is hybridization. Sun et al. [2] developed DE/EDA which combines DE with EDA for the global continuous optimization problem. It combines global information extracted by EDA with differential information obtained by DE to create promising solutions. The presented experimental results demonstrated that DE/EDA outperforms DE and EDA in terms of solution quality within a given number of objective function evaluations. Noman et al.[3] proposed a DE variant which incorporated a Local Search(LS) technique to solve optimization problem by adaptively adjusting the length of the search, using a hill-climbing heuristic. Experimenting with a wide range of benchmark functions,the results show that the proposed new version of DE performs better, or at least comparably,to classic DE algorithm. He et al.[4] proposed a new binary differential evolution algorithm based on the theory of immunity in biology. The test results show the improvement of the searching ability and increment in the convergence speed in comparison with the other algorithms. Das et al.[5]introduced a stochastic selection mechanism to improve the accuracy and convergence speed of DE. The idea of a conditional acceptance function (that allows accepting inferior solutions with a gradually decaying probability) is borrowed from the realm of the Simulated Annealing (SA). The resulting hybrid algorithm has been compared with three state-of-the-art adaptive DE schemes. The experiment results indicate that the mixed algorithm is able to find better solutions on a six-function testbed and one difficult engineering optimization problem. Omran et al.[6] incorporated a hybrid of concepts from chaotic search, opposition-based learning, differential evolution and quantum mechanics, named CODEQ to solve constrained problems. The experiment results indicate that CODEQ is able to find excellent solutions in all cases. Zhang et al.[7] proposed a hybrid of DE with PSO, called DE-PSO which incorporates concepts from DE and PSO, updating particles not only by DE operators but also by mechanisms of PSO. The presented experimental results demonstrate its effectiveness and efficiency. Wang et al.[8] combined the self-adaptive mixed distribution based univariate estimation of distribution algorithm (MUEDA) and a modified DE (MDE) to form a new algorithm, named ED-DE. It solved Economic Load Dispatch (ELD) problem successfully. Coelho et al.[9]combined ant colony optimization(ACO) with a differential evolution method (MACO) for chaotic synchronization. Jia et al.[10] proposed a Chaos and Gaussian local optimization based hybrid differential evolution (CGHDE) to high-dimensional complex engineering problems. The randomicity of chaotic local search can explore in a wide search space to overcome the premature in the earlier evolution phase and Gaussian optimization can refine the optimum in the later run phase. The experiment results indicate that CGHDE is able to find excellent solutions than other algorithms.

The second direction for improvement is dynamic adaptation of the control parameters. DE is sensitive to the two crucial parameters, to a certain extent the parameter values determine whether DE is capable of finding a near-optimum solution or not. So, recently, some studies focus on adaptive control parameters. Zaharie[11] proposed to transform F into a Gaussian random variable. Liu et al.[12] proposed a fuzzy adaptive differential evolution (FADE) which uses fuzzy logic controllers to adapt the mutation and crossover control parameters. Das et al. [13] proposed two schemes which are named DERSF and DETVSF to adapt the scaling factor F. Brest et al.[14] presented a novel approach to self-adapt parameters F and Cr. In their method, these two control parameters are encoded at the individual level. Nobakhti et al.[15] proposed a Randomised Adaptive Differential Evolution (RADE) method, which a simple randomised self-adaptive scheme is proposed for the DE mutation weighting factor F. Qin et al.[16] proposed self-adaptive DE (SaDE) which the trial vector generation strategies and two control parameters are dynamically adjusted based on their performance. Zhang et al.[17] proposed a new differential evolution (DE) algorithm (JADE) which the optional archive operation utilizes historical data to provide information of progress direction.Pan et al[18]proposed a self-adaptive DE algorithm, namely SspDE. It used an associated strategy list(SL),a mutation scaling factor F list (FL),and a crossover rate CR list (CRL) to be more effective in obtaining better quality solutions.

The third direction for improvement is population initialization. Before solving an optimization problem, it usually has no information about the location of the global minimum. It is desirable that an algorithm starts to explore those points that are scattered evenly in the decision space. Population initialization is a crucial task in evolutionary algorithms because it can affect the convergence speed and also the quality of the final solution. Recently, some researchers are working some methods to improve the EAs population initialization. Leung et al.[19] designed a GA called the orthogonal GA with quantization (OGA/Q) for global numerical optimization with continuous variables. Gong et al [20] used orthogonal design method to improve the initial population of DE(ODE). Rahnamayan et al. [21-23] proposed two novel initialization approaches which employ opposition-based learning and quasi-opposition to generate initial population. Xu et al.[24] used chaos initialization to get rapid convergence of DE as the region of global minimum. Pant et al.[25] proposed a novel initialization scheme called quadratic interpolation to DE with suitable mechanisms to improve its generation of initial population. Peng et al.[26] used Uniform-Quasi-Opposition to generate initial population of DE and accelerate its convergence speed and improve the stability. Ozer[27] used chaotic maps to generate sequences from different chaotic systems to construct initial population and proposed Chaotically Initialized Differential Evolution (CIDE).

© 2012 ACADEMY PUBLISHER

In this paper, an improvement version of DE, namely Uniform-Differential Evolution (UDE) is presented to solve unconstrained optimization problem. UDE combines DE with uniform initialization. According to our previous study, uniform design generation can enhance the quality of initial population. The two experiments are designed and UDE is compared with SDE, ODE,OBDE,CDE. The experimental results show that UDE outperforms SDE, ODE, OBDE, CDE.

The paper is organized as follows: Section 2 provides an overview of differential evolution, uniform design method, orthogonal design method, opposition based method and Chaos initialization method. Our proposed approach is presented in detail in Section 3.After that, in Section 4 the experimental design, the results are included. The last section, Section 5, is devoted to conclusions and future works.

II. PRELIMINARY

A. Differential evolution

The DE algorithm in pseudo-code is shown in Algorithm 1. Each vector i in the population at generation t , x_i called target vector will generate one offspring called trial vector v_i . Trial solutions are generated by adding weighted difference vectors to the target vector. This process is referred to as the mutation operator where the target vector is mutated. A crossover step is then applied to produce an offspring which is only accepted if it improves on the fitness of the parent individual. Many variants of standard DE have been proposed, which use different learning strategies and/or recombination operations in the reproduction stage. In this paper, the DE/best/1/exp strategy is used.

Algorithm 1.Procedure of DE with best/1/exp

Generate the initial population P , define $x_i(t)$ as the i -th individual of the t -th generation:

$$x_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{in}(t))$$
$$i = 1, 2, \cdots, M; t = 1, 2, \cdots, t_{\max}$$

where n is the number of decision variable, M is the population size, $t_{\max}$ is the maximum generation.

Evaluate the fitness $f(x_i(t))$ for the each individual.

while the termination condition is not satisfied **do**

  for i=1 to M **do**

    Select x_{best} , x_{p1} , x_{p2} and $i \neq p1 \neq p2 \neq best$.

    j=randint(1,n)

    L=0

    $v_i = P_i$

    repeat

      $v_{ij} = x_{bestj} + F \times (x_{p1j} - x_{p2j})$

      j=(j+1) *mod n*

      L=L+1

    until $rand_{ij}$ [0,1)>CR *or* L>n

    Evaluate the offspring v_i

    If v_i is better than x_i **then**

      $x_i = v_i$

    end if

  end for

end while

F is the scaling factor, CR is crossover factor.

B. Uniform design method

Experimental design method is a sophisticated branch of statistics. The uniform design, proposed by Fang and Wang[29] in 1980,is one of space filling designs and has been widely used in computer and industrial experiments. The main objective of uniform design is to sample a small set of points from a given set of points, such that the sampled points are uniformly scattered.

It defines the uniform array as $U_M q^n$, where n is factors and q is levels. When n and q are given, the population can be constructed by selecting M combinations from q^n .The steps of initialization population are as Algorithm 2.

Algorithm 2. Uniform Design Initialization

1: Find all the primer numbers $h = (h_1, h_2, \cdots, hs)$ which are less than M , where M is the size of population.

2: The j -th column of the uniform array is constructed according to (1)

$$U_{ij} = ih_j[\text{mod} \quad M] \qquad (1)$$
$$\text{where } i = 1, 2, \cdots, M \text{ ; } j = 1, 2, \cdots, s$$

3: Suppose n ($n < s$) is the number of the variables, randomly choose $h_{i1}, \cdots, h_{jn}$ from the vector $h = (h_1, h_2, \cdots, hs)$. A uniform matrix of $U'_{M \times n}$ is constructed.

4: Generation of initial population

After constructing the uniform array, it can generate the uniform population which scatters uniformly over the feasible solution space according to (2).

$$P(i, j) = U'_{ij} \times (u_j - l_j) / M + l_j$$
$$i = 1, 2, \cdots, M \text{ ; } j = 1, 2, \cdots, n \qquad (2)$$

where u_j and l_j are the maximum and minimum values of the variable j .

C. Orthogonal design method

Orthogonal design method [19,20] with both orthogonal array (OA) and factor analysis (such as the statistical optimal method) is developed to sample a small, but representative set of combinations for experimentation to obtain good combinations. OA is a fractional factorial array of numbers arranged in rows and columns, where each row represents the levels of factors in each combination, and each column represents a specific factor that can be changed from each combination. It can assure a balanced comparison of levels of any factor. The array is called orthogonal

© 2012 ACADEMY PUBLISHER

because all columns can be evaluated independently of one another, and the main effect of one factor does not bother the estimation of the main effect of another factor.

Algorithm 3. Orthogonal Design Initialization

For k=1 to J do
  $j = \frac{Q^{k-1}-1}{Q-1}+1$
  For i=1 to R do
    $a_{i,j} = \left\lfloor \frac{i-1}{Q^{j-k}} \right\rfloor \bmod Q$
  End for
End for
For k=2 to J do
  $j = \frac{Q^{k-1}-1}{Q-1}+1$
  For s=1 to j-1 do
    For t=1 to j-1 do
      For i=1 to R do
        $a_{i,(j+(s-1)(Q-1)+t)} = (a_{i,s} \times t + a_{i,j}) \bmod Q$
      End for
    End for
  End for
End for
Increment $a_{i,j}$ by one for all $i \in [1, R]$ and $j \in [1, C]$
eval=0
For i=1 to R do
  For j=1 to n do
    k= $a_{i,j}$
    $P(i, j) = l_j + (k-1)(\frac{u_j - l_j}{Q-1}), 1 \le k \le Q$
    ($[l_j, u_j]$ is quantized Q-1 fractions)
  End for
Evaluate $P(i, j)$ and eval++
End for
Sort the $P(i, j)$
Select the best M solution from $P(i, j)$ to generate the first population

D. Opposition Based Initialization

The concept of opposition-based learning (OBL) [21,22], in its earlier simple form, was introduced by Tizhoosh. The main idea behind OBL is the simultaneous consideration of an estimate and its corresponding opposite estimate in order to achieve a better approximation for the current candidate solution. As an advantage of opposite versus random points, purely random resampling or selection of solutions from a given population, has a higher chance of visiting or even revisiting unproductive regions of the search space.

Algorithm 4. Opposition Based Initialization

Generate uniformly distributed random population P_0
  For i=0 to M do
    For j=0 to n do
      $OP_{0i,j} = a_j + b_j - P_{0i,j}$
Select M fittest individuals from set the $\{P_{0,} OP_0\}$ as the initial population.

E. Chaos Initialization

Chaos is a kind of characteristic of nonlinear systems and it has been extensively studied and applied in many fields[24,27]. Although it appears to be stochastic, it occurs in a deterministic nonlinear system under deterministic conditions. Chaotic sequences have been proven easy and fast to generate and store, there is no need for storage of long sequences. Merely a few functions (chaotic maps) and few parameters (initial conditions) are needed even for very long sequences. In addition, an enormous number of different sequences can be generated simply by changing its initial condition. Moreover, these sequences are deterministic and reproducible. Recently, chaotic sequences have been adopted instead of random sequences and very interesting and somewhat good results have been shown in many applications.

Algorithm 5. Chaos Initialization

Set the Maximum number if chaotic iteration, CI, according to the problem ,the population size M and i=0
  While i≤M do
    Randomly initialize chaotic variables taking into account the constrains , j=1,2,….,n and set counter k=0;
    While (k<CI) do
      Generate different chaotic variables cm_k^j, j=1,2,.,n, using Logistic map.
      k=k+1
    End While
  Map the chaotic variables cm_k^j to feasible region according to equation $X_{j,i}^0 = X_j^{\min} + cm_k^j \times (X_j^{\max} - X_j^{\min})$, j=1,2,.,n
  Set i=i+1
End While

III. UNIFORM DIFFERENTIAL EVOLUTION

The performance of DE is sensitive to the choice of control parameters. Based on our former research, the better choice of the parameters are $F = 0.5$ and $CR = 0.9$ .In order to avoid tuning the parameter F and CR , a parameter control technology is adopted according to the following scheme:

$$F = N(0.5, 0.02),\ CR = N(0.9, 0.02) \tag{3}$$

$N(\tau, \varepsilon)$ is a normal distribution that can generate values in the range of $[\tau - 3\times\varepsilon, \tau + 3\times\varepsilon]$.

© 2012 ACADEMY PUBLISHER

Algorithm 6. Main procedure of Uniform Differential Evolution

1: **Initialization:** construct the population P by uniform initialization.
2: **Optimization using DE with Best/1/Exp.**
- **Mutation**

Select the best individual x_{best} in the t -th generation and two different individuals x_{p1}, x_{p2} from population where $i \neq p1 \neq p2 \neq best$.
- **Crossover**

Crossover operation is used to increase the diversity ,
- **Selection**

Compare $v_i(t)$ with $x_i(t)$, select the vector which has a better fitness as the individual in the new generation :
3: If stop criterion is met, go to step 4,else go to step 2
4: **Terminate**

IV. EXPERIMENTS

In order to assess the performance of our proposed algorithm, a comprehensive set of benchmark functions, including 23 different global optimization problems f01~f23 [20,28],have been employed for performance verification of the proposed approach. The formal definitions of the test functions and their global optimum(s) are summarized in [30]. Generally, following characteristics are desirable to provide a comprehensive test suite:

Functions 1 ~ 5 are unimodal problems and functions 8~ 13 are multimodal. Functions 6~7 are two special problems exhibiting a step landscape and a noisy landscape respectively. Functions 14~23 are low-dimensional functions which have only a few local minima.

Two experiments are designed. For each test functions, it performs 50 independent runs for each algorithm with different random seeds.

The first test compares the convergence speed of UDE with SDE, ODE,OBDE,CDE by measuring the number of successful runs and the mean number of function calls (NFC) of successful runs which are the most commonly used metrics. The test results of SDE and ODE come from the literature [20].

In the first experiment, the parameters of UDE are as follows:

- Population Size: NP=100.
- Maximum number of NFC(MAX_{NFC}) is 500000.
- The scaling factor F and probability of crossover CR of UDE use parameter control scheme as (3) .
- Stopping criterions are

$|f(x_{best}) - f(x_{optimal})| \leq 0.005$ or MAX_{NFC} is reached,
where $f(x_{best})$ is the best solution in the current run,
$f(x_{optimal})$ is the globally minimal function value.

The results are list in Table Ⅰ. From this table it can firstly be observed that ODE,OBDE and UDE can solve 23 benchmark problems in all 50 runs, but SDE cannot solve function f05 and f07 in all runs and it traps in the local optima once and four times. CDE traps in the local optima six times on function f20.Secondly, UDE needs less mean NFEs of successful runs than SDE, ODE, OBDE,CDE in 17 test functions f02~f06,f10~f13, f15~f17 and f19~f23. Especially in functions f04,f05, f06,f20,f21,f22,f23, it can be found that UDE makes considerable reduction of the mean NFEs of successful runs .

From these discussions, it can be concluded that firstly, the performance of UDE is better than other four algorithms; secondly, the uniform design can accelerate DE's convergence speed.

The second experiment compares the stability and calculation accuracy among the five algorithms. UDE has been compared with SDE, ODE,OBDE,CDE. The performance metrics have: (1)the mean NFEs(MNFEs) (2) the mean best function value(Mean best) (3)the standard deviation of the function values(Std).It performs 50 independent runs for each algorithm on the benchmark problems.

The parameters of UDE are as follows:

- Population Size: NP=100
- Maximum number of function calls is on Table Ⅱ
- The scaling factor F and probability of crossover CR of UDE use parameter control scheme as (3)

The mean results of 50 independent runs are summarized in Table Ⅱ. Results for SDE, ODE are taken from [20]. From Table Ⅱ, it can be seen that UDE needs less function evaluations than SDE, ODE,OBDE,CDE in 6 functions(f06, f09, f11, f14, f15, f16). UDE can provide better mean best results than SDE, ODE,OBDE,CDE for 7 functions (f03, f07, f10, f12, f13, f15, f20). Furthermore, UDE obtains smaller standard deviation than other four algorithms in 10 functions (f01, f03,f04,f10,f12, f13, f20, f21,f22,f23).

The results of the mean function values indicate that UDE is able to obtain more accurate solutions .The results of the standard deviation of the function values present that UDE is more stable than other four algorithms. Also, these results demonstrate that uniform design initialization used in DE can be effectively worked and enhance the performance of DE and accelerate the convergence speed and improve the stability and calculation accuracy of differential evolution.

V. CONCLUSIONS

In this article, it has presented a new variant of differential evolution algorithm (UDE) in which the initial population is selected using the uniform design initialization method. An adaptive parameter control technology is adopted .UDE has compared with other four algorithms of SDE, ODE,OBDE,CDE. According to the experiment results, it can conclude that uniform design initialization can enhance the capability of our algorithm and UDE is better and more stable than other four algorithms on the benchmark problems.

Future work consists on extending the present version for solving some real life optimization problems and combining uniform differential evolution with other local optimizer.

© 2012 ACADEMY PUBLISHER

TABLE I. COMPARISON WITH SDE, ODE,OBDE,CDE AND UDE ON $f1-f23$. THE BETTER RESULTS OF NUMBER OF SUCCESSFUL RUNS AND MEAN NFES OF SUCCESSFUL RUNS IN **BOLDFACE**.

F	Number of successful runs					Mean NFEs of successful runs				
	SDE	ODE	OBDE	CDE	UDE	SDE	ODE	OBDE	CDE	UDE
f01	50	50	50	50	50	53548	35235	**16752**	17274	16867
f02	50	50	50	50	50	45912	36914	21344	21698	**20547**
f03	50	50	50	50	50	144076	95520	47110	47720	**46991**
f04	50	50	50	50	50	189680	126731	188408	199598	**111897**
f05	49	50	50	50	50	236808	232171	310538	315624	**227266**
f06	50	50	50	50	50	30286	20051	35816	36984	**17521**
f07	46	50	50	50	50	328491	**83072**	161828	176882	129778
f08	50	50	50	50	50	95590	**42346**	81452	79596	101567
f09	50	50	50	50	50	168732	**63763**	194664	196318	77927
f10	50	50	50	50	50	53784	36802	63386	64768	**35341**
f11	50	50	50	50	50	51602	34010	59810	61448	**32952**
f12	50	50	50	50	50	36290	23648	45312	46270	**23595**
f13	50	50	50	50	50	50236	33409	50770	51794	**21949**
f14	50	50	50	50	50	3702	3383	**412**	844	525
f15	50	50	50	50	50	946	1124	1076	1458	**868**
f16	50	50	50	50	50	998	1016	642	868	**606**
f17	50	50	50	50	50	1356	1584	562	894	**454**
f18	50	50	50	50	50	1556	1621	**800**	1054	808
f19	50	50	50	50	50	1038	946	520	780	**404**
f20	50	50	50	44	50	14504	4059	3278	4634	**1193**
f21	50	50	50	50	50	5918	5473	5650	6440	**4141**
f22	50	50	50	50	50	5066	5053	5444	6548	**1959**
f23	50	50	50	50	50	5184	4782	1614	6332	**1547**

ACKNOWLEDGMENT

The authors would like to acknowledge the anonymous reviewers for their useful comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China under Grant No.60873107, the National High-Tech Research and Development Plan of China under Grant No. 2008AA12A201,the Fundamental Research Funds for the Central Universities under Grant No. CUGL090238 and CUG100708, the Research Foundation for Outstanding Young Teachers China University of Geosciences (Wuhan) under Grant No. CUGQNL0831.

REFERENCES

[1] R. Storn and K. Price, “Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces”,Journal of Global Optimization., vol. 11, pp. 341-359, 1997.

[2] J. Sun, Q. Zhang and E. P. K. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," Information Sciences, vol. 169, pp. 249-262, 2005.

[3] N. Noman and H. Iba, "A new generation alternation model for differential evolution," in 8th Annual Genetic and Evolutionary Computation Conference 2006, July 8, 2006 - July 12, 2006, Seattle, WA, United states, 2006, pp. 1265-1272.

[4] X. He and L. Han, "A novel binary differential evolution algorithm based on artificial immune system," in 2007 IEEE Congress on Evolutionary Computation, CEC 2007, September 25, 2007 - September 28, 2007, Singapore, pp. 2267-2272.

[5] S. Das, A. Konar and U. K. Chakraborty, "Annealed differential evolution," in 2007 IEEE Congress on Evolutionary Computation, CEC 2007, September 25, 2007 - September 28, 2007, Singapore, pp. 1926-1933.

[6] M. G. H. Omran and A. Salman, "Constrained optimization using CODEQ," Chaos, Solitons and Fractals, vol. 42, pp. 662-668, 2009.

[7] C. Zhang, J. Ning, S. Lu, D. Ouyang, and T. Ding, "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization," Operations Research Letters, vol. 37, pp. 117-122, 2009.

[8] Y. Wang, B. Li and T. Weise, "Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems," Information Sciences, vol. 180, pp. 2405-2420, 2010.

© 2012 ACADEMY PUBLISHER

[9] L. D. S. Coelho and D. L. D. A. Bernert, "A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization," Expert Systems with Applications, vol. 37, pp. 4198-4203, 2010.
[10] D. Jia and G. Zheng, "Hybrid differential evolution combined with chaos and Gaussian local optimization," Kongzhi yu Juece/Control and Decision, vol. 25, pp. 899-902, 2010.in Chinese
[11] D.Zaharie, "Critical values for the control parameters of differential evolution algorithms", in Proc. MENDEL 8th Int.Conf.Soft Comput.,2002,pp.62-67
[12] J.Liu and J.Lampinen, "A fuzzy adaptive differential evolution algorithm", Soft Computing--A Fusion of Foundations, Methodologies and Applications,vol.9,no.6, pp.448-642, 2005.
[13] S. Das, A. Konar and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in GECCO 2005 - Genetic and Evolutionary Computation Conference, June 25, 2005 - June 29, 2005, Washington, D.C., United states, 2005, pp. 991-998.
[14] J. Brest, S. Greiner, B. Bokovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," IEEE Transactions on Evolutionary Computation, vol. 10, pp. 646-657, 2006.
[15] A. Nobakhti and H. Wang, "A simple self-adaptive Differential Evolution algorithm with application on the ALSTOM gasifier," Applied Soft Computing Journal, vol. 8, pp. 350-370, 2008.
[16] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in 2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005, September 2, 2005 - September 5, 2005, Edinburgh, Scotland, United kingdom, 2005, pp. 1785-1791.
[17] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," IEEE Transactions on Evolutionary Computation, vol. 13, pp. 945-958, 2009.
[18] Q. Pan, P. N. Suganthan, L. Wang, L. Gao, and R. Mallipeddi, "A differential evolution algorithm with self-adapting strategy and control parameters," Computers & Operations Research, vol. 38, pp. 394-408, 2011.
[19] Y. W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," IEEE Transactions on Evolutionary Computation, vol. 5, pp. 41-53, 2001.
[20] W. Gong, Z. Cai and L. Jiang, "Enhancing the performance of differential evolution using orthogonal design method," Applied Mathematics and Computation, vol. 206, pp. 56-69, 2008.
[21] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "A novel population initialization method for accelerating evolutionary algorithms," Computers & Mathematics with Applications, vol. 53, pp. 1605-1614, 2007.
[22] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Opposition-based differential evolution," IEEE Transactions on Evolutionary Computation, vol. 12, pp. 64-79, 2008.
[23] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Quasi-oppositional differential evolution," in 2007 IEEE Congress on Evolutionary Computation, CEC 2007, September 25, 2007 - September 28, 2007, Singapore, 2008, pp. 2229-2236.
[24] X.Xu, X.Huang, and D.Qain, "Adaptive accelerating differential evolution," Complex Systems and Complexity Science ,5(3),pp:87-92,2008. In Chinese
[25] M. Pant, M. Ali and V. P. Singh, "Differential evolution using quadratic interpolation for initializing the population," in 2009 IEEE International Advance Computing Conference, IACC 2009, March 6, 2009 - March 7, 2009, Patiala, India, 2009, pp. 375-380.
[26] L. Peng and Y. Wang, "Differential evolution using uniform-quasi-opposition for initializing the population," Information Technology Journal, vol. 9, pp. 1629-1634, 2010.
[27] A. Bedri Ozer, "CIDE: chaotically initialized differential evolution," Expert Systems with Applications, vol. 37, pp. 4632-4641, 2010.
[28] X. Yao, Y. Liu and G. Lin, "Evolutionary programming made faster," IEEE Transactions on Evolutionary Computation, vol. 3, pp. 82-102, 1999.
[29] Y. Wang and K. T. Fang, "A note on uniform distribution and experimental design", KEXUE TONGBAO, vol. 26, no. 6, pp. 485-489,1981.In Chinese.
[30] L.Peng,Y.Wang,and G.Dai, "UDE:differential evolution with uniform design",in 3rd International Symposium on Parallel Architectures,Algorithms and Programming, PAAP 2010, December 18, 2010-December 20, 2010, Dalian, China, 2010, pp.239-246

Lei Peng is a lecturer in School of Computer Science in China University of Geosciences, Wuhan, China and a Ph.D. student at College of Computer Science and Technology, Huazhong University of Science and Technology. The research interests are evolutionary algorithm and evolutionary engineering optimal design.

Yuanzhen Wang is Professor in College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. The research interest is DataBase.

Guangming Dai is Professor in School of Computer Science in China University of Geosciences, China. His main interests are in the area of evolutionary algorithm and mission analysis and design.

Zhongsheng Cao is Associate Professor in College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. The research interest is DataBase and Data Mining.

© 2012 ACADEMY PUBLISHER

TABLE II. COMPARISION WITH DE, ODE, OBDE,CDE,UDE. MAX_EVAL : THE MAX NUMBER OF FUNCTION CALLS, MEAN BEST: THE MEAN BEST FUNCTION VALUE (OVER 50 TRIALS), STD: THE STANDARD DEVIATION OF THE FUNCTION VALUES. THE BEST MNFES, MEAN BEST AND STD IN **BOLDFACE**.

F	Max_eval	MNFEs					Mean best					Std				
		SDE	ODE	OBDE	CDE	UDE	SDE	ODE	OBDE	CDE	UDE	SDE	ODE	OBDE	CDE	UDE
f01	150000	150000	150000	**127952**	128696	128653	1.64E-18	2.06E-23	**8.91E-51**	9.11E-51	8.97E-51	5.29E-18	1.83E-23	8.28E-52	8.29E-52	**6.69E-52**
f02	200000	200000	200000	200000	200000	200000	2.97E-15	1.43E-18	**1.83E-42**	2.83E-42	4.55E-42	5.78E-15	8.11E-19	**1.64E-42**	2.87E-42	4.87E-42
f03	500000	500000	500000	**362474**	365588	366015	3.53E-20	5.25E-27	9.47E-51	9.41E-51	**9.36E-51**	3.53E-20	9.66E-27	5.07E-52	5.64E-52	**4.80E-52**
f04	500000	500000	500000	500000	500000	500000	9.73E-10	**2.72E-15**	2.22E-07	2.11E-07	2.74E-15	5.00E-10	9.30E-15	8.14E-07	7.19E-07	**5.17E-15**
f05	500000	494788	**428776**	500000	500000	486784	2.55E-29	**0**	1.23E-16	7.29E-18	1.63E-27	1.15E-28	**0**	4.26E-16	1.06E-17	7.97E-27
f06	150000	30454	22640	36390	36798	**17539**	0	0	0	0	0	0	0	0	0	0
f07	300000	300000	300000	300000	300000	300000	0.00598	0.00145	0.00279	0.00288	**0.00137**	0.00125	**4.20E-04**	9.92E-04	1.38E-03	5.52E-04
f08	300000	167324	**90381**	300000	300000	131610	-12569.48662	-12569.48662	-12569.48662	-12569.48662	-12569.48662	0	0	7.35E-12	7.35E-12	**0**
f09	300000	247626	127666	286512	288890	**125567**	0	0	0	7.11E-17	0	0	0	0	5.02E-16	0
f10	150000	150000	150000	150000	150000	150000	3.19E-10	4.67E-13	1.52E-08	1.74E-08	**4.64E-15**	1.10E-10	1.86E-13	3.47E-09	3.47E-09	**1.25E-15**
f11	200000	138236	109853	162874	162866	**88541**	0	0	0	0	0	0	0	0	0	0
f12	150000	150000	150000	150000	150000	150000	4.99E-20	6.73E-26	1.89E-16	2.22E-16	**1.57E-32**	3.68E-20	9.27E-26	9.12E-17	8.83E-17	**1.66E-47**
f13	150000	150000	150000	150000	150000	150000	4.42E-18	4.37E-24	1.07E-15	1.33E-15	**1.35E-32**	4.66E-18	3.67E-24	4.29E-16	6.07E-16	**8.29E-48**
f14	10000	9796	9552	2220	2436	**2079**	0.998	0.998	0.998	0.998	0.998	7.92E-15	**0**	1.01E-15	1.01E-15	1.01E-15
f15	150000	34484	32430	150000	150000	**18150**	3.08E-04	3.08E-04	3.075E-04	3.99E-04	**3.075E-04**	0	**0**	2.14E-19	2.77E-04	6.45E-19
f16	10000	10000	10000	10000	6894	**2871**	-1.03163	-1.03163	-1.03163	-1.03163	-1.0316285	9.16E-14	**0**	3.09E-16	6.13E-16	6.73E-16
f17	10000	10000	10000	**2868**	3232	10000	0.39789	0.39789	0.39789	0.39789	0.39789	6.35E-11	2.01E-10	3.36E-16	3.36E-16	**3.36E-16**
f18	10000	10000	10000	10000	10000	10000	3	3	3	3	3	1.34E-14	**0**	3.22E-015	2.69E-15	3.56E-15
f19	10000	10000	10000	10000	10000	10000	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	2.68E-15	**2.68E-15**	2.69E-15	2.69E-15	2.69E-15
f20	20000	20000	20000	20000	20000	20000	-3.31962	**-3.322**	-3.279	-3.244	**-3.322**	0.01681	1.13E-12	5.76E-02	5.69E-02	**3.17E-16**
f21	10000	10000	10000	10000	10000	10000	-10.1532	-10.1532	-10.1532	-10.1532	-10.1532	1.29E-05	1.04E-06	3.07E-06	1.33E-06	**7.76E-15**
f22	10000	10000	10000	10000	10000	10000	-10.40294	-10.40294	-10.40294	-10.40294	-10.40294	5.84E-08	2.49E-08	5.99E-07	1.48E-05	**3.30E-10**
f23	10000	10000	10000	10000	10000	10000	-10.53641	-10.53641	-10.53641	-10.53641	-10.53641	5.80E-08	2.35E-08	2.04E-07	1.201E-05	**8.44E-11**

© 2012 ACADEMY PUBLISHER