http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

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

Information Technology Journal 9 (8): 1629-1634, 2010
ISSN 1812-5638

Differential Evolution using Uniform-Quasi-Opposition for Initializing the Population

[1,2]Lei Peng and [1]Yuanzhen Wang
[1]College of Computer Science, Huazhong University of Science and Technology, Wuhan, China
[2]School of Computer, China University of Geosciences, Wuhan, China

Abstract: 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. According to our earlier study, uniform design generation can enhance the quality of initial population. In this study, a Uniform-Quasi-Opposition Differential Evolution (UQODE) algorithm is proposed. It uses a two-population mechanism and incorporates uniform design and quasi-opposition initialization method into differential evolution to accelerate its convergence speed and improve the stability. At the same time, an adaptive parameter control technology is adopted to avoid tuning the parameters of DE. The UQODE is compared with other three algorithms of standard Differential Evolution (DE), Opposition-based Differential Evolution (ODE) and Quasi-Oppositional Differential Evolution (QODE). Experiments have been conducted on 14 benchmark problems of diverse complexities. The results indicate that our approach has the stronger ability to find better solutions than other three algorithms especially for higher dimensional problems, in terms of the quality and stability of the final solutions.

Key words: Evolutionary algorithms, opposition-based learning, uniform design method, adaptive parameter control, optimization

INTRODUCTION

Differential Evolution is a branch of evolutionary algorithms developed by Storn and Price (1997) for global continuous optimization problem. It has won the third place at the 1st International Contest on Evolutionary Computation. The algorithm 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 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. Fan and Lampinen (2003) proposed a new version of DE which a new local search operation, called trigonometric mutation. Sun *et al*. (2005) 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 and Iba (2006) 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. Gong *et al*. (2008) incorporated the orthogonal design method into DE to accelerate its convergence rate and the self-adaptive parameter control is employed to avoid tuning the parameters of DE. The experiment results indicate that ODE is able to find the optimal or close-to-optimal solutions in all cases. Changsheng Zhang *et al*. (2009) 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.

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

Corresponding Author: Lei Peng, College of Computer Science, Huazhong University of Science and Technology, Wuhan, China

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 (2002) proposed to transform F into a Gaussian random variable. Liu and Lampinen (2005) proposed a Fuzzy Adaptive Differential Evolution (FADE) which uses fuzzy logic controllers to adapt the mutation and crossover control parameters. Das *et al.* (2005) proposed two schemes which are named DERSF and DETVSF to adapt the scaling factor F. Brest *et al.* (2006) 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 and Wang (2008) 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 and Suganthan (2005) proposed self-adaptive DE (SaDE) which the trial vector generation strategies and two control parameters are dynamically adjusted based on their performance. Zhang and Sanderson (2009) proposed a new Differential Evolution (DE) algorithm (JADE) which the optional archive operation utilizes historical data to provide information of progress direction. The most recent successful parameters are used to guide the setting of new ones.

The third direction for improvement is population initialization. Before solving an optimization problem, we usually have 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 and Wang (2001) designed a GA called the orthogonal GA with quantization (OGA/Q) for global numerical optimization with continuous variables. Gong *et al.* (2008) used orthogonal design method to improve the initial population of DE (ODE). Rahnamayan *et al.* (2007a) proposed two novel initialization approaches which employ opposition-based learning and quasi-opposition to generate initial population. Xu *et al.* (2008) used chaos initialization to get rapid convergence of DE as the region of global minimum. Pant *et al.* (2009) proposed a novel initialization scheme called quadratic interpolation to DE with suitable mechanisms to improve its generation of initial population. Peng *et al.* (2010) used uniform design to generate initial population of DE.

In this study, an improvement version of DE, namely Uniform-Quasi-Opposition Differential Evolution (UQODE) is presented to solve unconstrained optimization problem. UQODE uses a two-population mechanism. According to our previous study (Peng *et al.*, 2010), uniform design generation can enhance the quality of initial population. So,in the first step, uniform design in Peng *et al.* (2010) is used to generate one population UPop. And then, we use the UPop to obtain another population QOPop by utilizing quasi-oppositional learning. Last, Select the Np fittest individuals from {UPop∪QOPop} as the initial population. We prove that the two-population mechanism and two-step generation of UQODE can increase the percentage of success and the speed of convergence. The experimental results show that UQODE outperforms DE,ODE and QODE.

DIFFERENTIAL EVOLUTION

Unlike GA that uses binary coding for representation, DE uses floating point encoding and combines simple arithmetic operators with the classical events of mutation, crossover and selection to evolve from a randomly generated initial population to a satisfactory one.

Algorithm 1: DE with rand/1/bin

Step 1: Construct a random initial population pop, define $x_i(t)$ as the i-th individual of the t-th generation:

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

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

Step 2: Evaluate the fitness $f(x_i(t))$ for the each individual

Step 3: Mutation: Randomly select three different individuals x_{p1}, x_{p2} and x_{p3} from population where $i \neq p1 \neq p2 \neq p3$.

$$h_{ij}(t+1) = x_{p1j}(t) + F \times (x_{p2j}(t) - x_{p3j}(t)) \tag{2}$$

where, $x_{p2j}(t)$-$x_{p3j}(t)$ is the differential vector, F is the scaling factor.

Step 4: Crossover: It is used to increase the diversity , which is defined as Eq. 3:

$$v_{ij}(t+1) = \begin{cases} h_{ij}(t+1), rand_{ij} < CR \text{ or } j = jr \\ x_{ij}(t), \quad \text{otherwise} \end{cases} \tag{3}$$

where, $rand_{ij}$ is a random number in the interval [0,1], CR is crossover factor, $jr \in \{1,2,...,n\}$ is a random parameter's index.

Step 5: Selection: Compare $v_i(t+1)$ with $x_i(t)$, select the vector which have a better fitness as the individual in the new generation:

$$x_i(t+1) = \begin{cases} v_i(t+1), f(v_i(t+1)) < f(x_i(t)) \\ x_i(t), \quad f(v_i(t+1)) \geq f(x_i(t)) \end{cases} \tag{4}$$

OUR PROPOSED APPROACH

Quasi-oppositional optimization: Opposition-Based Learning (OBL) was first proposed by Tizhoosh (2005) and was successfully applied to several problems (Rahnamayan *et al*., 2007b). The basic concept of OBL is the consideration of an estimate and its corresponding opposite estimate simultaneously to approximate the current candidate solution. Rahnamayan *et al*. (2007b) proposed quasi-oppositional method based on opposition point. Quasi-opposite numbers are defined as follows:

The quasi-opposite number x'_q is defined as:

$$P_r(|x'_q - x| < |x' - x|) > 1/2$$

P_r is a given the probability function, x' is the opposite number, x is the solution for an optimization problem. The x'_q definition can be extended to higher dimensions.

UNIFORM DESIGN METHOD

Experimental design method is a sophisticated branch of statistics. The uniform design , proposed by Wang and Fang (1981) 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.

We define 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

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

Step 2: The j-th column of the uniform array is constructed according to Eq. 5:

$$U_{ij} = ih_j[\text{mod} \;\; M] \tag{5}$$

where, $i = 1,2 \cdots 1,M$; $j = 1,2,...,s$

Step 3: Suppose n(n<s) is the number of the variables, randomly choose $h_{j1},...,h_{jn}$ from the vector $h = (h_1,h_2,...,h_s)$. A uniform matrix of $U'_{M\times n}$ is constructed.

Step 4: Generation of initial population

After constructing the uniform array, we can generate the uniform population which scatter uniformly over the feasible solution space according to Eq. 6:

$$pop(i, j) = U'_{ij} \times (u_j - l_j)/M + l_j \quad i = 1, 2, \cdots, M;\; j = 1, 2, \cdots, n \tag{6}$$

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

UNIFORM-QUASI-OPPOSITION DIFFERENTIAL EVOLUTION

In the quasi-oppositional differential evolution, there are two population. The population initialization steps are as follows:

- Generate the first population randomly
- Calculate the second quasi-opposite population
- Choose the M better individuals from the combination of two population as initial population

In our earlier study, the uniform design population initialization is a very effective method to obtain fitter initial candidate individual and increase the speed of convergence of algorithm. So, the basic idea of Uniform-Quasi-Opposition Differential Evolution (UQODE) is that the first population P_U are constructed by uniform design. And then, the second quasi-opposite population are calculated based on P_U.

The performance of DE is sensitive to the choice of control parameters. Storn suggested 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 self-adaptive para- meter control technology is adopted according to the following scheme:

$$F = N(0.5, 0.05),\; CR = N(0.9, 0.05) \tag{7}$$

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

Algorithm 3. Main procedure of UQODE

Step 1: Construct initial population P_U using algorithm 2. Uniform design initialization

```
for i=1 to NP
    for j=1 to D
        OP_{0i,j} = a_j+b_j-P_{Uij},
        M_{i,j} = (a_j+b_j)/2
if OP_{0i,j} is better than M_{i,j} then
    QOP_{0i,j} = M_{i,j} + (OP_{0i,j} - M_{i,j}) × rand(0,1)
else
    QOP_{0i,j} = OP_{0i,j} + (M_{i,j} - OP_{0i,j}) × rand(0,1)
```

Step 2: Select N_p fittest individuals from the set $\{P_U \cup QOP\}$ as initial population pop

Step 3: while, $|f(X_{best}) - f(X_{optimal})| > \varepsilon$ and $NFC < MAX_{NFC}$

```
for i=1 to NP
    Randomly select three individuals i≠p1≠p2≠p3 according to rand/1/bin strategy
    Set CR and F using Eq. 7
    Generate offspring using mutation, crossover and repair operators.
    Evaluate offspring using the benchmark function.
    If offspring is better than pop then:
        pop = offspring
```

Table 1: Comparison of DE, ODE, QODE and UQODE

Problems		DE			ODE			QODE			UQODE		
F	D	NFC	SR	SP	NFC	SR	SP	NFC	SR	SP	NFC	SR	SP
F1	30	86072	1	86072	50844	1	50844	42896	1	42896	**23316**	1	**23316**
	60	154864	1	154864	101832	1	101832	94016	1	94016	**68245**	1	**68245**
F2	30	95080	1	95080	56944	1	56944	47072	1	47072	**26116**	1	**26116**
	60	176344	1	176344	117756	1	117756	105992	1	105992	**77133**	1	**77133**
F3	20	174580	1	174580	177300	1	177300	116192	1	116192	**33370**	1	**33370**
	40	816092	1	816092	834668	1	834668	539608	1	539608	**170508**	1	**170508**
F4	10	323770	0.96	337260	75278	0.92	81823	181100	1	181100	**72362**	**1**	**72362**
	20	811370	0.08	10142125	421300	0.16	2633125	615280	0.16	3845500	**154897**	**1**	**154897**
F5	30	111440	0.96	116083	**74717**	0.92	**81214**	100540	0.80	125675	105176	**1**	105176
	60	193960	1	193960	128340	0.68	188735	**115280**	0.68	**169529**	204985	**1**	204985
F6	30	18760	1	18760	10152	1	10152	**9452**	1	**9452**	11050	1	11050
	60	33128	1	33128	**11452**	1	**11452**	14667	0.84	17461	18666	1	18666
F7	30	168372	1	168372	100280	1	100280	**82448**	1	**82448**	151290	1	151290
	60	294500	1	294500	202010	0.96	210427	221850	0.72	308125	**127272**	**1**	**127272**
F8	30	101460	1	101460	70408	1	70408	**50576**	1	**50576**	81988	1	81988
	60	180260	0.84	215000	121750	0.60	202900	**98300**	0.40	245800	172639	**1**	**172639**
F9	10	191340	0.76	252000	213330	0.56	380900	247640	0.48	515900	**63568**	**1**	**63568**
	20	288300	0.35	824000	253910	0.55	461700	**193330**	0.68	**284300**	276348	**0.96**	287863
F10	30	385192	1	385192	369104	1	369104	239832	1	239832	**120278**	1	**120278**
	60	---	0	---	---	0	---	---	0	---	---	0	---
F11	30	183408	1	183408	167580	1	167580	108852	1	108852	**47208**	1	**47208**
	60	318112	1	318112	274716	1	274716	183132	1	183132	**126302**	1	**126302**
F12	30	40240	1	40240	26400	1	26400	21076	1	21076	**13682**	1	**13682**
	60	73616	1	73616	64780	1	64780	**64205**	1	**64205**	75440	1	75440
F13	30	386920	1	386920	361884	1	361884	291448	1	291448	**52492**	1	**52492**
	60	432516	1	432516	425700	0.96	443438	295084	1	295084	**157248**	1	**157248**
F14	10	19324	1	19324	16112	1	16112	13972	1	13972	**4420**	1	**4420**
	20	45788	1	45788	31720	1	31720	23776	1	23776	**10689**	1	**10689**
Srave			0.89			0.87			0.85			**0.96**	

D: Dimension, NFC: No. of function calls (average over 50 trials), SR: Success rate, SP: Success performance. The last row of the table presents the average success rates. The best NFC and the success performance for each case are highlighted in boldface. DE, ODE, QODE and UQODE are unable to solve f 10 (D = 60)

EXPERIMENTS

In order to assess the performance of our proposed algorithm UQODE. We choose a set of 14 benchmark problems(14 test problems having two different dimensions) from f_1-f_{14} (Rahnamayan *et al*., 2008). The UQODE has been compared with three algorithms: DE,ODE and QODE. The performance metrics has four categories using by Rahnamayan *et al*. (2008): Number of Function Calls (NFC), Success Rate (SR), average Success Rate (SR_{ave}) and Success Perfor-mance (SP).A smaller NFC means higher convergence speed. A larger SR, SR_{ave} and SP mean higher stability. In order to minimize the effect of the stochastic nature of the algorithms on metric ,we perform 50 independent runs for each algorithm on the benchmark problems.

The parameters of all algorithms are as follows:

- Population size: NP=100
- Maximum number of function calls: $MAX_{NFC} = 1\times10^6$
- The scaling factor F and probability of crossover CR using self-adaptive parameter control scheme as Eq. 7
- Halting precision: $\varepsilon = 1\times10^{-8}$

The mean results of 50 independent runs are summarized in Table 1. Results for DE, ODE and QODE are taken from (Rahnamayan *et al*., 2007a, b). From Table 1, it can be seen that UQODE outperforms DE,ODE and QODE on 19 functions. The UQODE can provide better results with smaller NFC than DE,ODE and QODE for 18 functions. The SR of UQODE is larger than other three algorithms for $f4_{10,20}$, $f5_{30,60}$, $f7_{60}$, $f8_{60}$, $f9_{10,20}$. UQODE performs marginally better than DE,ODE and QODE in terms of average Success Rate (SR_{ave}) (0.96, 0.89,0.87 and 0.85, respectively). The UQODE has the stronger ability to find better solutions than other three algorithms especially for higher dimensional problems. These results indicate the combination of uniform design initialization and quasi-opposition initialization can effectively accelerate convergence and improve the performance of differential evolution.

CONCLUSION

In this study, we have presented a new variant of basic DE algorithm (UQODE) in which the initial population is selected using the uniform-quasi-opposition initialization method. The UQODE has compared with other three algorithms of DE,ODE and QODE. According to the experiment results, we can conclude that the combination of uniform design initialization and quasi-opposition initialization method can enhance the capability of our algorithm and UQODE is better and more stable than other three algorithms on most benchmark problems.

Future work consists on extending the present version for solving some real life optimization problems such as the Earth-Moon low energy transfer problem and researching uniform-quasi-opposition initialization method to multiobjective optimization algorithm.

ACKNOWLEDGMENTS

The authors would like to acknowledge the anonymous reviewers and the Editors for their useful comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China under Grant No.60873107(20090110), the National High-Tech Research and Development Plan of China under Grant No.2008AA12A201 (20090610), the Fundamental Research Funds for the Central Universities under Grant No. CUGL090238.(20091101), the Research Foundation of Science and Technology China University of Geosciences (Wuhan) under Grant No. CUGXGF0901 (20081120), the Research Foundation for Outstanding Young Teachers China University of Geosciences (Wuhan) under Grant No. CUGQNL0831 (20080120).

REFERENCES

Brest, J., S. Greiner, B. Boskovic, M. Mernik and V. Zumer, 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. Evolutionary Computation, 10: 646-657.

Das, S., A. Konar and U.K. Chakraborty, 2005. Two improved differential evolution schemes for faster global search. Proceedings of the Genetic Evolution Computing Conference (GECC O), June 2005, Washington DC, USA., pp: 991-998.

Fan, H.Y. and J. Lampinen, 2003. A trigonometric mutation operation to differential evolution. J. Global Optimization, 27: 105-129.

Gong, W.Y., Z.H. Cai and L.X. Ling, 2008. Enhancing the performance of differential evolution using orthogonal design method. Applied Mathematics Computation, 206: 56-69.

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

Liu, J. and J. Lampinen, 2005. A fuzzy adaptive differential evolution algorithm, soft computing--a fusion of foundations. Methodologies Appl., 9: 448-642.

Nobakhti, A. and H. Wang, 2008. A simple self- adaptive differential evolution algorithm with application on the ALSTOM gasifier. Applied Soft Computing, 8: 350-370.

Noman, N. and H. Iba, 2006. A new generation alternation model for differential evolution. Proceedings of the 8th Annual Conference on Genetic Evolution Computer Conference (GECCO 2006), July 2006, Seattle, Washington, USA., pp: 1265-1272.

Pant, M., M. Ali and V.P. Singh, 2009. Differential evolution using quadratic interpolation for initializing the population. Proceedings of the 2009 IEEE International Advance Computing Conference (IACC 2009) , March 6-7, Patiala, India, pp: 375-380.

Peng, L., Y.Z. Wang, G.M. Dai, Y.M. Chang and F.J. Chen, 2010. Optimization of the earth-moon low energy transfer with differential evolution based on uniform design. Proceedings of the IEEE Congress on Evolutionary Computation, July 18-23, Spain.

Qin, A.K. and P.N. Suganthan, 2005. Self- adaptive differential evolution algorithm for numerical optimization. Proc. IEEE Cong. Evol. Comput., 2: 1785-1791.

Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2007a. A novel population initialization method for accelerating evolutionary algorithms. Comput. Mathematics Appl., 53: 1605-1614.

Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2007b. Quasi-oppositional differential evolution. Proceedings of the IEEE Congress Evolutionary Computation, Sept. 25-28, Singapore, pp: 2229-2236.

Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2008. Opposition-based differential evolution. IEEE Trans. Evolutionary Computation, 12: 64-79.

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.

Sun, J., Q. Zhang and E.P. Tsang, 2005. DE/EDE: A new evolutionary algorithm for global optimization. Inform. Sci., 169: 249-262.

Tizhoosh, H., 2005. Opposition-based learning: A new scheme for machine intelligence. Proc. Int. Conf. Comput. Intel. Modeling Control Autom, I: 695-701.

Wang, Y. and K.T. Fang, 1981. A note on uniform distribution and experimental design. KEXUE TONGBAO, 26: 485-489.

Xu, X.J., X.P. Huang and D.L. Qain, 2008. Adaptive accelerating differential evolution. Complex Syst. Complexity Sci., 5: 87-92.

Zaharie, D., 2002. Critical values for the control parameters of differential evolution algorithms. Proceedings of the 8th International Conference on Soft Computing Mendel 2002, June 2002, Brno, Czech Republic, pp: 62-67.

Zhang, J.Q. and A.C. Sanderson, 2009. JADE: Adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput., 13: 945-958.

Zhang, C.S., J.X. Ning, S. Lu, D.T. Ouyang and T.N. Ding, 2009. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. Operations Res. Lett., 37: 117-122.