Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2015, 7(3):455-459

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

A novel modified differential evolution algorithm for clustering

Zhao Hongwei and Xia Honggang*

Shenyang University, School of Information Engineering, No.21, Wanghua Street, Shenyang, China

---

ABSTRACT

Differential evolution (DE) is easy to trap into local optima. In this paper, a modified differential evolution algorithm (MDE) proposed to speed the convergence rate of DE and enhance the global search of DE. The MDE employed a new mutation operation and modified crossover operation. The former can rapidly enhance the convergence of the MDE, and the latter can prevent the MDE from being trapped into the local optimum effectively. In this work, firstly, we employed a new strategy to dynamic adjust mutation rate (MR) and crossover rate (CR), which is aimed at further improving algorithm performance. Secondly, the MDE algorithms are used for data clustering on several benchmark data sets. The performance of the algorithm based on MDE is compared with DE algorithms on clustering problem. The simulation results show that the proposed MDE outperforms the other two algorithms in terms of accuracy, robustness and convergence speed.

Key words: Differential evolution; Convergence; Stability; Data clustering

---

INTRODUCTION

Standard Many real-life optimization problems are complex and difficult to solve in an exact manner within a reasonable amount of time. The classical optimization methods applied are highly sensitive to the starting point and frequently converge to a local optimum solution or diverge altogether. Due to the computational drawbacks of existing numerical methods, researchers have to rely on meta- heuristic intelligence optimization algorithms based on simulations to solve some complex optimization problems. In 1995, a new floating point encoded an evolutionary algorithm for global optimization; called Differential Evolution (DE)[1] was proposed. Differential Evolution, inspired by the natural evolution of species, has been successfully applied to solve numerous optimization problems in diverse fields. However, when implementing the DE, users not only need to determine the appropriate encoding schemes and evolutionary operators, but also need to choose the suitable parameter settings to ensure the success of the algorithm, which may lead to demanding computational costs due to the time-consuming trial-and-error parameter and operator tuning process. To overcome such inconvenience, researchers have actively investigated the adaptation of parameters and operators in DE. Literature [2],[3] divided the parameter adaptation techniques into three categories: deterministic, adaptive, and self-adaptive control rules. Deterministic rules modify the parameters according to certain predetermined rationales without utilizing any feedback from the search process. Adaptive rules incorporate some form of the feedback from the search procedure to guide the parameter adaptation. Self-adaptive rules directly encode parameters into the individuals and evolve them together with the encoded solutions. Parameter values involved in individuals with better fitness values will survive, which fully utilize the feedback from the search process. Generally speaking, self-adaptive rules can also refer to those rules that mainly utilize the feedback from the search process such as fitness values to guide the updating of parameters. Differential Evolution uses a special kind of differential operator, and recently, it has been applied in different fields of engineering and science [4],[5].

The performance of the conventional DE algorithm highly depends on the chosen trial vector generation strategy and associated parameter values used. Inappropriate choice of strategies and parameters may lead to premature convergence or stagnation, which have been extensively demonstrated in some paper[7],[8].

In this paper, we propose a modified differential evolution (MDE) algorithm to avoid the expensive computational costs spent on searching for the most appropriate trial vector generation strategy, as well as its associated parameter values by a trial-and-error procedure. Instead, both strategies and their associated parameters are adjusted adaptively to prevent algorithm trapped into local optima.[9]

The paper is organized as follows. Differential evolution algorithm would be reviewed in Section 2. In Section 3, modified differential evolution algorithm (MDE) algorithm is described in detail. Afterward, several standard benchmark optimization problems are carried out to test and compare the performance of the MDE and the other three algorithms in section 4. Finally, conclusions were given in Section 5.

DIFFERENTIAL EVOLUTION ALGORITHM

Many In this section, we describe the basic operations of differential evolution and introduce necessary notations and terminologies which facilitate the explanation of different adaptive DE algorithms later. The differential evolution algorithm works as follows: [10], [11]

(1) Initialize the optimization problem and algorithm parameters.
(2) Mutation operation

At each iteration k, this operation creates mutation vectors v_i based on the current parent population $x_i=(x^1{}_i, x^2{}_i, x^3{}_i, \cdots, x^D{}_i)$ $i=1,2,\cdots,NP$ .The following are different mutation strategies frequently used in the literature:

$$v_i = x_{r0} + F \times rand \times (x_{r1} - x_{r2}) \tag{1}$$

(3)Crossover operation

After mutation, a binomial crossover operation forms the final trial/offspring vector $u_i = (u_i{}^j, j=1,2,3,\cdots,D)$, the procedure of Crossover works as follow:[12]

$$u_i{}^j = \begin{cases} u_i{}^j \text{ , if } rand\text{<CR or } j = jrand \\ x_i{}^j \text{ , otherwise} \end{cases} \tag{2}$$

Where CR is the crossover rate; rand belongs to a uniform distribution in the ranges [0,1]; jrand=randint(1,D) is an integer randomly chosen from 1 to D.

(4)Selection operation
The selection operation selects the better on from the parent vector x_i and the trial vector u_i according to their fitness values $f(x)$ .For example, if we have minimization problem, the selected vector is given by

$$x_{i+1} = \begin{cases} x_i \text{ , if } f(x_i) < f(u_i) \\ u_i \text{ , otherwise} \end{cases} \tag{3}$$

Where x_{i+1} used as a parent vector in the next generation. [13]
Check the termination criteria

A MODIFIED DIFFERENTIAL EVOLUTION ALGORITHM

A new variant of differential evolution algorithm is proposed in this paper. Similar to all population-based optimization algorithms, two main steps are distinguishable for DE, namely, population initialization and crossover. We will modify these two steps using the MDE scheme. The original DE is chosen as a parent algorithm and the proposed opposition-based ideas are embedded in DE to accelerate its convergence speed.

Although a large number of simulations showed that various metaheuristic methods are insensitive to population initialization value, reasonable and comprehensive initial population is benefit to accelerate convergence speed. According to our review of optimization literature, random number generation, in absence of a priori knowledge, is the common choice to create an initial population. Therefore, by utilizing MDE, we can obtain fitter starting candidate solutions even when there is no a priori knowledge about the solution(s). Of course, in absence of any a priori knowledge, it is not possible that we can make the best initial guess. Logically, we should be looking in all directions simultaneously, or more concretely, in the opposite direction. If we are searching for x, and if we agree that searching in opposite direction could be beneficial, then calculating the opposite number x is the first step.

In DE, the parameter mutation rate (MR) and crossover rate (CR) are both influence the optimization performance of DE. The parameter MR plays an important role in the amplification of the differential variation and the increase of difference between two individuals in the search space. Large MR value may lead to premature convergence, whereas low MR value may lead the convergence too. To the best of our knowledge, no optimal choice of the scaling parameter MR has been suggested in the literature of DE. This means MR is problem-dependent and the user should choose MR carefully after some trial and error tests. For taking the best values for CR, there are certain basic rules. Large values are effective for all problems, but they are not always the fastest. The problems with heavy interaction between design variables generally require a high CR. But, if interaction between design variables is lower, a lower CR can be used, which results in obtaining a satisfactory solution with a smaller number of iterations. In original DE, the mentioned CR and MR values are constant. These parameters never change since they are initialized at the beginning of DE. In this paper, in order to improve the balance between the exploration and exploitation in DE algorithm we propose a new strategy to adjust the parameters MR and CR. In our proposition, the parameter MR present a dynamic adaptation using a decreasing linear of1 to 0.1 during the optimization cycle, and the parameter CR is generated using a decreasing linear of 0.99 to 0.4.

RESULTS AND DISCUSSION

MDE is compared with classic differential evolution algorithm DE/best/1, PSO, as well as recent adaptive DE algorithms JADE. For fair comparison, we set the parameters of MDE to be fixed, Fmin=0.1, Fmax=0.9 and CRmin=0.1, CRmax=0.9 in all simulations. We follow the parameter settings in the original paper of JADE, except that the parameters of DE/rand/1 are set to be F=0.5 and CR=0.9. The results reported in this section are best, worst, mean and standard deviation (SD) over 30 independent simulations. For each simulation, all the procedure to be run in computer Inter(R) Pentium(R) 4, CPU 2.93GHz, and the numerical results using differential evolution algorithm are report in Table 1.

1. Maintenance-free benchmark functions

Ten well-knownbenchmark functions are used in the test. These functions contain three functions.
The first function is Sphere function whose global minimum value is 0 at (0, 0,…, 0). Initialization range for the function is [−5.12, 5.12]. It is a unimodal function with non-separable variables.

$$f_1(x)=\sum_{i=1}^{n}x_i^2 \qquad x\in[-5.12,5.12]^D \tag{4}$$

The second function is Rosenbrock function whose global minimum value is 0 at (1,1,…,1). Initialization range for the function is [−15,15]. It is a unimodal function with non-separable variables. Its global optimumis inside a long, narrow, parabolic shaped flat valley. So it is difficult to converge to the global optimum.

$$f_2(x)=\sum_{i=1}^{n}100\times(x_{i+1}-x_i^2)^2+(1-x_i)^2 \quad x\in[-3,3]^D \tag{5}$$

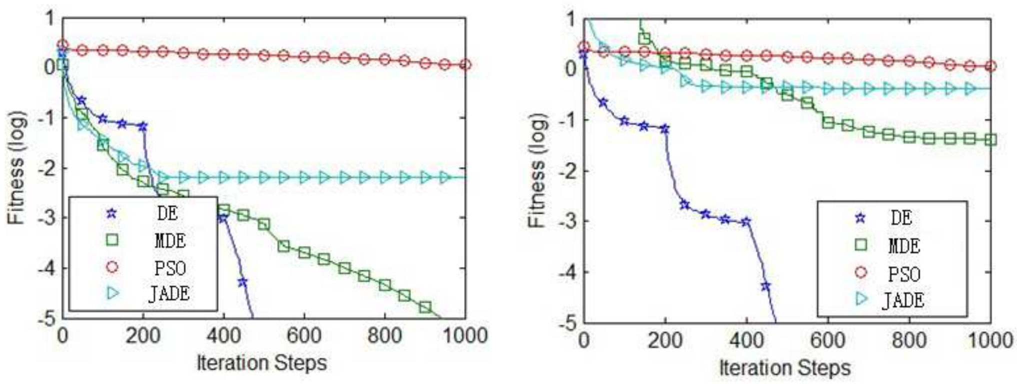
The third function is Rastrigin function whose global minimum value is 0 at (0,0…0). Initialization range for the function is [−15,15]. It is a multimodal function with separable variables.

$$f_3(x)=\sum_{i=1}^{D}(\ x_i^2-10\cos(2\pi x_i)\ +10) \tag{6}$$

Table.1Comparison among PSO, DE, MDE and JADE on 20D problems.

Function		PSO	DE	MDE	JADE
f_1	Mean	4.90E-10	1.24E-06	**3.23E-10**	2.46E-03
	Std	3.23E-10	1.44E-06	**3.23E-10**	4.07E-03
	Min	3.23E-10	2.36E-07	**3.23E-10**	8.28E-05
	Max	1.62E-09	5.54E-05	**3.23E-10**	1.22E-02
f_2	Mean	4.49E-08	4.94E-03	**3.23E-10**	1.88E+00
	Std	6.02E-08	5.09E-03	**3.23E-10**	2.11E+00
	Min	1.67E-09	5.34E-04	**3.23E-10**	6.94E-03
	Max	2.88E-07	2.16E-02	**3.23E-10**	7.53E+00
f_3	Mean	1.91E-04	1.35E-01	**1.10E-08**	3.63E-01
	Std	1.12E-04	6.32E-02	**1.71E-08**	2.58E-01
	Min	3.35E-05	3.21E-02	**5.67E-10**	5.13E-02
	Max	5.88E-04	2.10E-01	**6.57E-08**	9.27E-01

To analyses the performanceof the proposed MDE approach for clustering algorithm, the results of PSO and DEwith differentdatasets have been compared in this paper. The algorithm base on MDE algorithms is used for data clustering on Iris data sets, which is able to provide the same partition of the data points in all runs. Cluttering result of which sets by DE and the MDE clustering algorithm .From the result, for allreal data sets, the basic clustering algorithm with MDE outperforms the other methods.



(a) (b)

Fig.1. Convergence results of MDE, DE, PSO and JADE on 20-D benchmark function (a) Sphere ;(b) Rosen rock.

CONCLUSION

In this paper, A new simple but effective and efficient MDE algorithm was proposed for clustering, we compared it with those of DE, PSO,JADE optimization algorithms on several benchmark functions. Comparison of experimental results show, that firstly, the clustering algorithm based on MDE makes similar data gather obviously; secondly, the model is more stable and accurate than the old one; thirdly, it distinguishes samples precisely while also improving the cluster quality and obtaining better centers with clear division which represents reducing computation amount . However, the convergence speed issue remains to be modified and researched.

Acknowledgments
This paperwas supported by the international cooperation project(Project No.S2012ZR0191) and the Natural Science Foundation of Liaoning Province (Project No.2013020011) and and the Social Science Foundation of Liaoning Province (Project No.L14ASH001) and this work was supported in part by the International S&T Cooperation Program of China (ISTCP) under Grant 2011DFA91810-5 and Program for New Century Excellent Talents in University of Ministry of Education of China under Grant NCET-12-1012.

REFERENCES

[1] Badamchizadeh, M.A., Nikdel, A., Kouzehgar, M., *International Journal of Artificial Intelligence*, **2010**; 5 (10).
[2] D. H. Kim, J. H. Cho, Adaptive Tuning of PID Controller for Multivariable System Using Bacterial Foraging Based Optimization, Proceedings of 3rd Web Intelligence Conference, 231-235, **2005**.
[3] S. Mishra, *IEEE Trans. Evolutionary Computation*. Vol. 9, No. 1, 61-73, **2005**.
[4] Kirkpatrick S, Gelatt C and Vecchi M. *Science*, **1983**, 220: 671-680.
[5] LI Xiaolei, SHAO Zhijiang, QIAN Jixin. *Systems Engineering Theory & Practice*, **2002**, 22(11): 32-38.
[6] D. Dasgupta, N. Attoh-okine. Immunity Based Systems: A Survey[C]. *Proc IEEE International Conference on Systems, Man,and Cybernetics*,Orlando,Florida,**1997**:369-374.
[7] D.Zaharie, R.Matousek and P.Osmera. Control of population diversity and adaptation in differential evolution algorithms [C]. In Proc: Mendel 9th International Conference Soft Computer, Eds. Brno, Czech Republic, **2003**(1): 41–46.
[8] S.Das,A.Konar, U.K.Chakraborty. Two improved differential evolution schemes for faster global search [C]. In ACM-SIGEVO *Proc. Genetic Evolut. Comput. Conf*, Washington, DC, pp: 991–998.
[9] H R Tizhoosh. Opposition-based Learning: A New Scheme for Machine Intelligence[C]//Proc. of International Conference on Computational Intelligence. Modeling Control and Automation. Vienna, Austria, **2005**: 695-701.
[10] I. E. Evangelou, D. G. Hadjimitsis, A. A. Lazakidou, C. Clayton, Data mining and knowledge discovery in complex image data using artificial neural networks, in Proceedings of the Workshop on Complex Reasoning an Geographical Data, Paphos, Cyprus, **2001**.
[11] M. S. Kamel, S. Z. Selim, *Pattern Recognition*, 27 (3) (**1994**), pp. 421-428.
[12] Sato,T.,Hagiwara,M. *IEEE international conference on computational cybernetics and simulation*, vol 4. Orlando, FL, USA, pp 3954~3959.
[13] Tao,X.M, Xu,J.and Yang,L.B. *Journal of Electronics &Information Technology*. **2010**;32(1): 92-94.