

Nonlinear Continuous Global Optimization by Modified Differential Evolution

Md. Abul Kalam Azad*, Edite M.G.P. Fernandes[†] and Ana M.A.C. Rocha**

**Assistant Researcher, Algoritmi Center*

[†]Professor, Dept. of Production and Systems

***Assistant Professor, Dept. of Production and Systems*

School of Engineering, University of Minho, 4710-057 Braga, Portugal

Abstract. The task of global optimization is to find a point where the objective function obtains its most extreme value. Differential evolution (DE) is a population-based heuristic approach that creates new candidate solutions by combining several points of the same population. The algorithm has three parameters: amplification factor of the differential variation, crossover control parameter and population size. It is reported that DE is sensitive to the choice of these parameters. To improve the quality of the solution, in this paper, we propose a modified differential evolution introducing self-adaptive parameters, modified mutation and the inversion operator. We test our method with a set of nonlinear continuous optimization problems with simple bounds.

Keywords: Nonlinear optimization; Simple bounds; Global optimization; Differential evolution

PACS: 02.60.Pn

INTRODUCTION

Problems involving global optimization over continuous spaces are ubiquitous throughout the scientific community. In optimization, the task is to optimize certain properties of a system by pertinently choosing the system parameters called variables. In general, a variety of actual optimization problems are formulated as large-scale mathematical programming problems involving continuous variables with nonlinear objective function.

Generally, the nonlinear continuous optimization problems with simple bounds on variables are formulated as

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with $\Omega = \{\mathbf{x} \in \mathbb{R}^n : lb_j \leq x_j \leq ub_j, j = 1, \dots, n\}$. The task of global optimization in (1) is to find a solution where the objective function obtains its most extreme value, the global minimum. When the objective function has a huge number of local minima, local optimization techniques are likely to get stuck before the global minimum is reached, and some kind of global search is needed to find the global minimum with some reliability.

In the last decades, depending on the nature of objective function involved many solution methods have been proposed to solve (1). In these methods, algorithms for solving global minimization problem can be classified into deterministic methods that guarantee to find a global optimum with a required accuracy and stochastic methods that find the global minimum only with high probability. In deterministic methods, the values of the objective function are assumed to be exact, and the computation is completely determined by the values sampled so far. A number of deterministic methods [5, 8, 9] are present to solve global optimization problems of kind (1). Stochastic methods involve function evaluations at a suitably chosen random sample of solutions and subsequent manipulation of the sample to find good local (and hopefully global) minima. The stochastic methods can be classified as point-to-point search technique and population based search technique. For solving those problems, a number of stochastic methods [1, 4, 6, 7, 10, 12, 13] are present.

A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution. Typically, every candidate solution has more than one neighbor solutions and the choice of movement depends only on information about the solutions in the neighborhood of the current one. Differential evolution (DE) proposed by Storn and Price [13] is a population-based heuristic approach that is very efficient when solving derivative-free global optimization problems. DE has only three parameters and according to Storn and Price is sensitive to these parameters. To improve the local search and quality of the solution, in this paper, we propose a modified differential evolution

(mDE) introducing self-adaptive parameters, modified mutation and the inversion operator for nonlinear continuous optimization problems with simple bounds.

DIFFERENTIAL EVOLUTION

Differential evolution is a simple yet powerful evolutionary algorithm for global optimization problems. It is a floating-point encoding for global optimization over continuous spaces. It creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better or equal fitness. DE's three parameters are amplification factor of the difference vector F , crossover control parameter CR , and population size NP . DE is a direct search method which utilizes NP n -dimensional component points. The target points are defined by $\mathbf{x}_{i,t} = (x_{i1,t}, x_{i2,t}, \dots, x_{in,t})$ where t is the index of generation and $i = 1, 2, \dots, NP$. NP does not change during the optimization process. The initial population is chosen randomly and should cover the entire solution space. As a rule, it is assumed a uniform probability distribution for all random decisions unless otherwise stated. DE's operations are outlined briefly in the following.

Mutation: For each target point $\mathbf{x}_{i,t}$, $i = 1, 2, 3, \dots, NP$, a mutant point is created according to

$$\mathbf{v}_{i,t+1} = \mathbf{x}_{r_1,t} + F(\mathbf{x}_{r_2,t} - \mathbf{x}_{r_3,t}) \quad (2)$$

with uniformly chosen random indices r_1, r_2, r_3 from the set $\{1, 2, \dots, NP\}$, mutually different and $F > 0$. The indices r_1, r_2 and r_3 are also chosen to be different from the running index i , so that NP must be greater or equal to four to allow for this condition. F is a real and constant parameter $\in [0, 2]$ which controls the amplification of the differential variation $(\mathbf{x}_{r_2,t} - \mathbf{x}_{r_3,t})$. Here, $\mathbf{x}_{r_1,t}$ is called the base point. Differential mutation is DE's exploration operator.

Crossover: In order to increase the diversity of the perturbed component points, crossover is introduced. To this end, the trial point $\mathbf{u}_{i,t+1}$ is formed, where

$$\begin{aligned} u_{ij,t+1} &= \begin{cases} v_{ij,t+1} & \text{if } (r_j \leq CR) \quad \text{or} \quad j = z_i \\ x_{ij,t} & \text{if } (r_j > CR) \quad \text{and} \quad j \neq z_i \end{cases}, \\ j &= 1, 2, \dots, n. \end{aligned} \quad (3)$$

In (3), the random number $r_j \sim U[0, 1]$ performs the mixing of j th component of points, $CR \in [0, 1]$ is a constant parameter for crossover which has to be determined by the user and the uniformly chosen integer random index z_i from the set $\{1, 2, \dots, n\}$ ensures that $\mathbf{u}_{i,t+1}$ gets at least one component from $\mathbf{v}_{i,t+1}$.

Selection: To decide whether or not it should become a member of generation $t + 1$, the trial point $\mathbf{u}_{i,t+1}$ is compared to the target point $\mathbf{x}_{i,t}$ using the greedy criterion in the following way

$$\mathbf{x}_{i,t+1} = \begin{cases} \mathbf{u}_{i,t+1} & \text{if } f(\mathbf{u}_{i,t+1}) \leq f(\mathbf{x}_{i,t}) \\ \mathbf{x}_{i,t} & \text{otherwise.} \end{cases} \quad (4)$$

OUR MODIFIED DIFFERENTIAL EVOLUTION

According to Storn and Price [13], DE is much more sensitive to the choice of F than it is to the choice of CR . The suggested choices for the three parameters are: (i) $F \in [0.5, 1]$; (ii) $CR \in [0.8, 1]$; and (iii) $NP = 10 \times n$. Recall that n is the dimensionality of the problem. The parameters in DE are kept constant throughout the entire evolutionary process. However, it is not an easy task to set appropriate parameters since these depend on the nature and size of the optimization problems. In original DE, three points are chosen randomly for mutation and the base point is then chosen at random within the three. This has an exploratory effect but it slows down the convergence of DE. In the following the modified differential evolution (mDE) is outlined.

Modification of F and CR : Since the choice of F and CR mainly depend on the nature and size of the problems, the modified differential evolution (mDE), proposed in this paper, includes the modifications proposed by Brest et al. [2] for the control parameters F and CR . New control parameters for next generation $F_{i,t+1}$ and $CR_{i,t+1}$ are calculated as

$$\begin{aligned} F_{i,t+1} &= \begin{cases} F_l + \lambda_1 \times F_u, & \text{if } \lambda_2 < \tau_1 \\ F_{i,t}, & \text{otherwise} \end{cases} \\ CR_{i,t+1} &= \begin{cases} \lambda_3, & \text{if } \lambda_4 < \tau_2 \\ CR_{i,t}, & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

where $\lambda_k \sim U[0, 1], k = 1, \dots, 4$ and $\tau_1 = \tau_2 = 0.1$ represent probabilities to adjust parameters F_i and CR_i , respectively. $F_l = 0.1$ and $F_u = 0.9$, so the new $F_{i,t+1}$ takes a value from $[0.1, 1.0]$ in a random manner and the new $CR_{i,t+1}$ takes a value from $[0, 1]$. $F_{i,t+1}$ and $CR_{i,t+1}$ are obtained before the mutation is performed. So, they influence the mutation, crossover and selection operations of the new point $\mathbf{x}_{i,t+1}$.

Modification of Mutation: Sometimes it is required to improve local search in DE, the mDE includes the modification proposed by Kaelo and Ali [11] for selecting the best point after choosing three random points as a base point in (2). Further, we introduce modification in the selection of the base point in the mutation operation. After every B generations the best point found so far is used as the base point and two randomly chosen points are used in differential variation in (2). These modifications allow mDE to maintain its exploratory feature as well as explore the region around each best point for each mutant point and at the same time expedite the convergence.

Inversion: We also implement inversion in mDE. In biology, an inversion is a chromosome rearrangement in which a segment of a chromosome is reversed end to end. Since in DE, a point has n -dimensional real number components, inversion can easily be applicable. An illustrative example of inversion is shown in Figure 1.

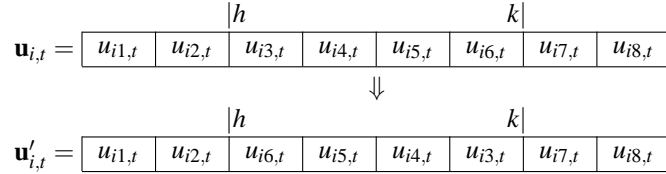


FIGURE 1. Inversion used in mDE

Bounds Check: To generate the mutant point by using (2) and to perform the inversion, some components of point can be generated outside of the domain Ω . So, in this paper, for mDE after inversion the bounds of each component are checked with the following projection of bounds:

$$u'_{ij,t+1} = \begin{cases} lb_j & \text{if } u'_{ij,t+1} < lb_j \\ ub_j & \text{if } u'_{ij,t+1} > ub_j \\ u'_{ij,t+1} & \text{otherwise.} \end{cases} \quad (6)$$

Termination Condition: Suppose t and G_{\max} are the current and maximum generations, respectively, and $f_{\max, t}$ and $f_{\min, t}$ are the maximum and minimum objective function values at generation t . The mDE terminates if $t > G_{\max}$ or $(f_{\max, t} - f_{\min, t}) \leq \varepsilon$, where ε is a very small positive number.

EXPERIMENTAL RESULTS

We code some variants including our modified variant of DE in C. The name of the solvers are: (i) DE_Original, (ii) DE_Kaelo, (iii) DE_Brest, (iv) mDE1 (without inversion) and (v) mDE2. We set $NP = \min(100, 10n)$ and $\varepsilon = 10^{-6}$ for all solvers. We also set $F = 0.5$ and $CR = 0.9$ for solvers DE_Original and DE_Kaelo and $p_{\text{inv}} = 0.05$ for solver mDE2. We set other parameter values discussed so far for solvers DE_Brest, mDE1 and mDE2. We set $B = 10$ for solvers mDE1 and mDE2.

For a fair comparison among different solvers of DE, we test all solvers on a set of 64 nonlinear continuous optimization problems with simple bounds and use same termination condition. We run all solvers 30 times for each problem. We use performance profile proposed by Dolan and Moré [3] to evaluate and compare the performance of different solvers. Figure 2(a) shows the performance profile for different solvers after 30 runs. In this figure it is shown that solvers mDE1 and mDE2 win over the other DE. Again, for larger value of τ our mDE have greater values of $\rho_s(\tau)$ meaning that with respect to robustness of solvers our mDE also win over others DE.

Finally here, we compare mDE2 with other solution methods. We use two deterministic solvers DIRECT [5] and MCS [9] and three stochastic solvers ASA [10], PSwarm [14] and CMA-ES [6]. We think that for a fair comparison among different solvers, the number of objective function evaluations should be used. A comparison based on the number of iteration does not seem to give a reliable measure because the amount of work done in each iteration is completely different among solvers, since some are population based and other are single point based. In Figure 2(b), it is shown that when $\tau = 1$ MCS wins over the other solvers, followed by DIRECT and mDE2. Our mDE2 outperforms the other stochastic solvers.

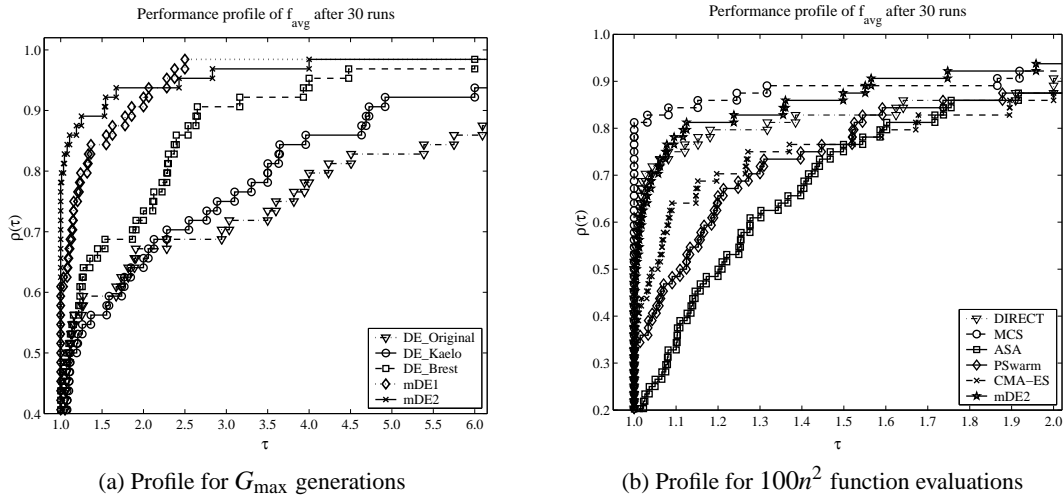


FIGURE 2. Performance profile of f_{avg} for different solvers

CONCLUSIONS

In this paper, to make the DE more efficient, a modified differential evolution algorithm has been proposed. The modifications focus on self adaptive control parameters and a modified mutation. Inversion operator has also been implemented in the proposed mDE. A set of 64 test problems was solved and compared. It has been shown that mDE is rather competitive and effective in solving nonlinear continuous optimization problems with simple bounds.

ACKNOWLEDGMENTS

This work is supported by FCT (Fundação para a Ciência e a Tecnologia) and Ciência 2007, Portugal.

REFERENCES

1. S. I. Birbil and S. C. Fang, "Electromagnetism-like mechanism for global optimization", *J. Glob. Optim.* **25**, 263–282 (2003).
2. J. Brest, S. Greiner, B. Bošković, M. Mernik and V. Žumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems", *IEEE Trans. Evol. Comput.* **10**, 646–657 (2006).
3. E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles", *Math. Program.* **91(A)**, 201–213 (2002).
4. M. Dorigo, V. Maniezzo and A. Colnari, "The ant system: optimization by a colony of cooperating agents", *IEEE Trans. Syst. Man Cybern.* **26(B)**, 29–41 (1996).
5. D. E. Finkel and C. T. Kelley, "Additive scaling and the DIRECT algorithm", *J. Glob. Optim.* **36** 597–608 (2006).
6. N. Hansen, "The CMA evolution strategy: a comparing review", in: *Towards a new evolutionary computation. Advances in estimation of distribution algorithms* edited by J. A. Lozano, P. Larrañga, I. Inza and E. Bengoetxea, Springer, 2006, pp. 75–102.
7. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
8. R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems", *J. ACM.* **8**, 212–229 (1961).
9. W. Huyer and A. Neumaier, "Global optimization by multilevel coordinate search", *J. Glob. Optim.* **14**, 331–355 (1999).
10. L. Ingber, "Adaptive Simulated Annealing (ASA): Lessons Learned", *Control. Cybern.* **25**, 33–54 (1996).
11. P. Kaelo and M. M. Ali, "A numerical study of some modified differential evolution algorithms", *Eur. J. Oper. Res.* **169**, 1176–1184 (2006).
12. J. Kennedy, R. C. Eberhart and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, 2001.
13. R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", *J. Glob. Optim.* **11**, 341–359 (1997).
14. A. I. F. Vaz and L. N. Vicente, "A particle swarm pattern search method for bound constrained global optimization", *J. Glob. Optim.* **39**, 197–219 (2007).