Roman Knobloch; Jaroslav Mlýnek; Radek Srb
The classic differential evolution algorithm and its convergence properties

Applications of Mathematics, Vol. 62 (2017), No. 2, 197–208

Persistent URL: http://dml.cz/dmlcz/146702

THE CLASSIC DIFFERENTIAL EVOLUTION ALGORITHM AND ITS CONVERGENCE PROPERTIES

Roman Knobloch, Jaroslav Mlýnek, Radek Srb, Liberec

Received September 30, 2016. First published March 6, 2017.

Abstract. Differential evolution algorithms represent an up to date and efficient way of solving complicated optimization tasks. In this article we concentrate on the ability of the differential evolution algorithms to attain the global minimum of the cost function. We demonstrate that although often declared as a global optimizer the classic differential evolution algorithm does not in general guarantee the convergence to the global minimum. To improve this weakness we design a simple modification of the classic differential evolution algorithm. This modification limits the possible premature convergence to local minima and ensures the asymptotic global convergence. We also introduce concepts that are necessary for the subsequent proof of the asymptotic global convergence of the modified algorithm. We test the classic and modified algorithm by numerical experiments and compare the efficiency of finding the global minimum for both algorithms. The tests confirm that the modified algorithm is significantly more efficient with respect to the global convergence than the classic algorithm.

Keywords: optimization; cost function; global minimum; global convergence; local convergence; differential evolution algorithm; optimal solution set; convergence in probability; numerical testing

MSC 2010: 60G20, 65K05

1. Introduction

Optimization tasks are frequent in science, engineering, and also production practice. The optimization can usually be expressed in terms of searching for the extreme value of some objective function. The variables of the objective function represent the system parameters and the extreme value corresponds to the optimized state of the system. When we look for the minimum of the objective function, we usually refer to it as a cost function.

The research was supported by grant SGS-FP-TUL 21167.

DOI: 10.21136/AM.2017.0274-16

At present, evolutionary algorithms (including differential evolution algorithms) are more and more used for optimization tasks with complicated cost functions. A detailed survey of these methods is available e.g. in [6]. The evolutionary algorithms are primarily utilized in situations when other usual methods fail to find the optimized state. For instance the commonly used gradient methods usually do not provide the global minimum when the cost function has a lot of local minima. In these cases the gradient methods are prone to converge to a local minimum and their result strongly depends on the choice of the starting point.

The evolutionary methods try to overcome this problem by generating whole populations of potential solutions. The generation process is partly random. This contributes to better exploring the space of all possible solutions (the search space). The potential solutions are then assessed according to their cost function value. On the other hand, the evolutionary algorithms are characterized by a slower convergence and longer calculation times.

Because of the random creation of individual potential solutions, the convergence analysis of differential evolution algorithms is more demanding. This is probably the reason why the published results concerning sufficient conditions for the global convergence of the differential evolution algorithms are relatively rare.

These algorithms were first introduced by Storn and Price in [4] and [7]. They now consist of a larger group of similar algorithms that differ in implementation details. We concentrate on the standard *DE/rand/1/bin* algorithm which is best known and mostly used. That is why it is termed as the classic differential evolution algorithm in [5] (further referenced to as CDEA).

In the first part of the article we concentrate on the global convergence of the CDEA and find an example of the cost function demonstrating that the CDEA does not in general guarantee the convergence to the global minimum of the cost function.

In the second part we design a simple modification improving CDEA's global convergence abilities. This modified algorithm is further referenced to as MDEA. We prove that the modified algorithm does ensure the convergence to the global minimum in asymptotic sense. The CDEA and MDEA are then tested by numerical experiments. The numerical testing confirms that MDEA converges to the global minimum with substantially higher probability than CDEA.

2. Classic differential evolution algorithm

In this section we briefly describe the functioning of CDEA. Generally, CDEA seeks for the minimum of the cost function by constructing whole populations of individuals. Each individual is an ordered set of specific values from the cost function domain. In this way each individual represents a potential solution of the optimization task. The quality of this individual is determined by the evaluation of the cost function.

The next population is formed from the existing population by means of mutation and crossover operators. Specifically, we go successively through all individuals in the population G. For each individual y_i^G (termed as the *target individual*) we select randomly three other (different) individuals y_{r1}^G, y_{r2}^G, y_{r3}^G from the current population. We form in a specific way (including randomness) a combination of these three individuals and the target individual. This combination is termed the *trial individual* and denoted y_i^{trial}. Then we evaluate the cost function for the target y_i^G and trial individual y_i^{trial} and compare the results. The individual with lower value of the cost function advances to the position of the target individual of the next population y_i^{G+1}. When this procedure is completed for all target individuals in population G, we have the new population of individuals numbered $G+1$.

The next part illustrates CDEA operation more specifically in the form of the pseudo code.

Input:

Optimization task parameters:

f denotes the cost function, D is the dimension of the cost function domain, $\langle x_i^{\min}, x_i^{\max}\rangle$ is the domain of each cost function variable x_i.

CDEA parameters:

NP denotes the population size (the number of individuals in each population), NG is the total number of populations, F stands for the mutation factor, $F \in \langle 0, 2\rangle$, and CR denotes the crossover probability, $CR \in \langle 0, 1\rangle$. The symbol G stands for the population number, index i is the number of the individual in a specific population, index j describes the jth component of a specific individual y_i.

Computation:

(1) create the initial population ($G = 1$) of NP individuals y_i^G, $1 \leqslant i \leqslant NP$, randomly or according to a prescribed scheme

(2) (a) evaluate all individuals y_i^G of the population G (calculate $f(y_i^G)$ for each individual y_i^G)

  (b) store the individuals y_i^G and their evaluations $f(y_i^G)$ into the matrix $\boldsymbol{A}$ with NP rows and $D+1$ columns

(3) **repeat until** $G \leqslant NG$

  (a) **for** $i = 1$ **to** NP **do**

    (i) randomly select three different indices $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$, $r_m \neq i$, $m \in \{1, 2, 3\}$

    (ii) randomly select an index $k_i \in \{1, \ldots, D\}$

(iii) **for** $j = 1$ **to** D **do**

  if $(\text{rand}(0,1) \leqslant CR$ **or** $j = k_i)$

    then $y_{i,j}^{\text{trial}} = y_{r3,j}^{G} + F(y_{r1,j}^{G} - y_{r2,j}^{G})$

    else $y_{i,j}^{\text{trial}} = y_{i,j}^{G}$

  endif

endfor(j)

(iv) **if** $f(y_i^{\text{trial}}) \leqslant f(y_i^G)$

  then $y_i^{G+1} = y_i^{\text{trial}}$

  else $y_i^{G+1} = y_i^G$

endif

endfor(i)

(b) store the individuals y_i^{G+1} and their evaluations $f(y_i^{G+1})$, $1 \leqslant i \leqslant NP$, of the new population $G+1$ into the matrix $\boldsymbol{A}$, $G = G+1$

endrepeat.

Output:

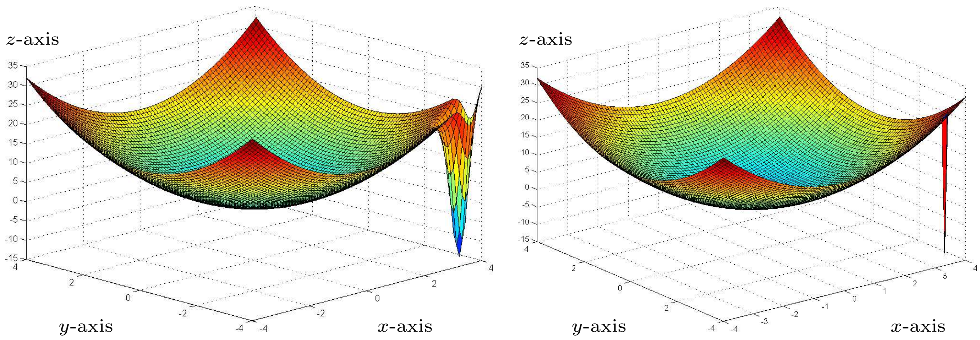
The matrix $\boldsymbol{A}$ with NP rows and $D+1$ columns contains the final population of individuals including their evaluations. The row of the matrix $\boldsymbol{A}$ that contains the lowest cost function value represents the best found individual $y_{\min}$.

3. Classic differential evolution algorithm and the global convergence

Although CDEA is commonly used for a lot of diverse optimization tasks, there are not many publications dealing theoretically with their global convergence properties. Most authors describe CDEA as a global optimizing technique (see for instance [5], [7]), but quite often there is only very little of quantitative discussion to this topic. The principle question is whether CDEA converges to the global minimum of the cost function or not.

3.1. Counterexamples to the global convergence. It is not difficult to find counterexamples to the global convergence of CDEA. Let us consider for instance the following two graphs of cost functions with the domain in 2-D.

Even for the cost function shown in Figure 1(a) the probability that CDEA finds the global minimum of the cost function is less than one. The reason is that CDEA converges relatively fast to a local minimum. It means that the individuals in subsequent populations concentrate around the local minimum. As soon as the size of the population falls under some critical value, the population is too small to generate trial individuals that could hit the global minimum (this situation is called *the premature convergence*). In this case even increasing the number of populations does not lead to increasing the chances of finding the global minimum. Moreover, the probability that CDEA finds the global minimum falls with the decreasing measure of the global minimum region. The probability of finding the global minimum for the cost function in Figure 1(b) is substantially smaller than for the cost function in Figure 1(a). By reducting the measure of the global minimum region this probability can be made as small as desired.



(a) (b)

Figure 1. Examples of cost functions in 2-D.

4. Modification ensuring the asymptotic global convergence

As mentioned in the previous part, CDEA does not in general guarantee the convergence to the global minimum of the cost function. This is caused by the too fast convergence of CDEA to a local minimum (premature convergence) resulting in rapid reduction of the population size (loss of diversity). This observation gives us a hint how to modify CDEA, so that it provides better results regarding the global convergence. The most straightforward way is to limit the premature convergence by replacing some individuals with the highest values of the cost function in each population by random individuals. Though these random individuals reduce partially the convergence speed they increase substantially the diversity of the population. The increased diversity then ensures even the asymptotic global convergence of the modified algorithm.

5. Modified differential evolution algorithm

In this section we describe the modification of CDEA called MDEA. Since there is in principle necessary to make one simple change in the algorithm, we present only the differences with respect to CDEA. See the pseudo-code description of CDEA in Section 2.

Input:

We add another parameter R that determines the ratio of random individuals in each population, $R \in \langle 0, 1\rangle$, e.g., $R = 0.1$ means that 10% of individuals in each population are generated randomly.

Computation:

We add another procedure to the part (3), specifically:

(c) determine in matrix $\boldsymbol{A}$ the quantity $\lfloor NP \cdot R\rfloor$ of individuals with the highest cost function values and replace these individuals by random individuals from the search space.

Here the symbol $\lfloor x\rfloor$ denotes the integer part of the real number x.

6. Asymptotic global convergence

In this section we present several theoretical concepts and statements that can be used to prove the asymptotic global convergence of MDEA. More specifically, we are able to show that when the number of populations satisfies $G \to \infty$ than the probability that MDEA finds the global minimum approaches 1.

6.1. Optimal solution set. We have an optimization task to find the minimum of a cost function $f(x_1, x_2, \ldots, x_n)$ defined on a bounded domain. For brevity we denote by the symbol x the set of all variables $x_1, x_2, \ldots, x_n$ of the optimized function. That is, we should find the minimum of the function $f(x)$. This function may have several minima. We would like to find the minimum with the lowest cost function value

$$\min\{f(x)\colon\ x \in S\}, \tag{6.1}$$

where S is a measurable search space of a finite measure representing all possible configurations of variables x. The solution set can be defined as

$$S^* = \{x^*\colon\ f(x^*) = \min\{f(x)\colon\ x \in S\}\},$$

where x^* represents the global minimum of the cost function. We consider an expanded solution set

$$S_\varepsilon^* = \{x \in S\colon\ |f(x) - f(x^*)| < \varepsilon\}, \tag{6.2}$$

where $\varepsilon > 0$ is a small positive real number. Denoting by μ the Lebesgue measure, we suppose that $\mu(S_\varepsilon^*) > 0$ for each ε. We call the set S_ε^* defined by relation (6.2) *the optimal solution set.*

6.2. Convergence in probability. To examine the global convergence of MDEA we need to introduce a concept of the convergence in probability defined in [1].

Definition. Let $\{Y(k), k = 1, 2, \ldots\}$ be a population sequence generated by a differential evolution algorithm for solving the optimization problem (6.1). We say that the algorithm converges to the optimal solution set in probability if and only if

$$\lim_{k\to\infty} p\{Y(k) \cap S_\varepsilon^* \neq \emptyset\} = 1, \tag{6.3}$$

where p denotes the probability of an event.

Now we can prove the following theorem.

Theorem 6.1. *Let us suppose that for each population $Y(k)$ of a differential evolution algorithm there exists at least one individual y such that*

$$p\{y \in S_\varepsilon^*\} \geqslant \alpha > 0,$$

where α is a small positive value. Then the algorithm converges to the optimal solution set S_ε^* in probability. That is, the relation (6.3) *holds.*

Here $p\{y \in S_\varepsilon^*\}$ denotes the probability that y belongs to the optimal solution set S_ε^*.

P r o o f. Let us suppose that an individual $y^{\text{rand}} \in S$ is generated randomly in each population $Y(k)$. The probability that it hits the optimal solution set is given by the relation

$$p\{y^{\text{rand}} \in S_\varepsilon^*\} = \frac{\mu(S_\varepsilon^*)}{\mu(S)} = \alpha > 0.$$

It means that the relation

$$p\{y^{\text{rand}} \notin S_\varepsilon^*\} = 1 - \alpha$$

holds for each population. We can estimate that the first k populations do not include an individual $y \in S_\varepsilon^*$ by the relation

$$\prod_{i=1}^{k} p\{Y(i) \cap S_\varepsilon^* = \emptyset\} \leqslant (1-\alpha)^k.$$

Based on the construction of individuals in the population $Y(k)$, the best individual in the population $Y(k)$ has the same or better evaluation than the best individual from all the previous populations, implying

$$\lim_{k\to\infty} p\{Y(k) \cap S_\varepsilon^* = \emptyset\} = \lim_{k\to\infty} \prod_{i=1}^{k} p\{Y(i) \cap S_\varepsilon^* = \emptyset\} \leqslant \lim_{k\to\infty} (1-\alpha)^k = 0,$$

which induces

$$\lim_{k\to\infty} p\{Y(k) \cap S_\varepsilon^* \neq \emptyset\} = 1 - \lim_{k\to\infty} p\{Y(k) \cap S_\varepsilon^* = \emptyset\} = 1 - 0 = 1$$

which was to prove. □

Since the probability that a random individual hits the optimal solution set with a positive measure is strictly positive, MDEA (in contrast to CDEA) complies with the assumptions of the theorem. This implies the global convergence of MDEA in probability.

7. Numerical testing and verification

In this section we describe the numerical testing of MDEA and CDEA and compare the results of the two algorithms.

7.1. Cost function. For the tests we designed special cost functions that contribute to quantitative considerations regarding convergence probabilities and can be easily generalized to an arbitrary dimension D of the search space. The cost function $f(x_1, x_2, \ldots, x_D)$ of D variables x_i is constructed as a sum of two simple functions $f = B + M$, where B is a base function and M is a modifier creating the global minimum of the function f. The base function is of a simple form

$$B = \sum_{i=1}^{D} (x_i - x_{iL})^2,$$

where the coordinates x_{iL} define the position of the local minimum at the point x_L. The modifier is expressed by the formula

$$M = d \cdot \left[\sum_{i=1}^{D} \frac{1}{w} (x_i - x_{iG})^D - 1 \right],$$

defined for x_i complying with the condition $\sum_{i=1}^{D} (x_i - x_{iG})^2 \leqslant 1$ (the unit ball), otherwise $M = 0$.

The parameter d defines the "depth" of the global minimum, the parameter w determines the "width" of the global minimum. The coordinates x_{iG} determine the position of the global minimum at the point x_G. Putting the two parts together we get the final expression for the cost function $f(x_1, x_2, \ldots, x_D)$ in the form

$$f(x_1, x_2, \ldots, x_D) = \sum_{i=1}^{D} (x_i - x_{iL})^2 + d \cdot \left[\sum_{i=1}^{D} \frac{1}{w} (x_i - x_{iG})^D - 1 \right]. \tag{7.1}$$

Figure 2 represents the cost function $f(x_1, x_2)$ with the two-dimensional domain $(D = 2)$ in the three-dimensional space.

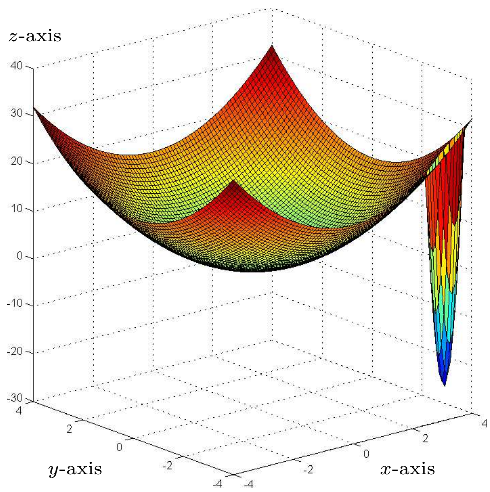


Figure 2. Example of a cost function.

7.2. Numerical testing—parameters. We performed the numerical tests and comparison between CDEA and MDEA in the Matlab environment. We took as a cost function the function defined in the previous section by relation (7.1) in an 8-dimensional Euclidean space. That is, we use the parameter $D = 8$. The domain of each variable x_i, $i = 1, 2, \ldots, 8$, is the interval $\langle -4, 4 \rangle$, the domain of the cost function

$f(x_1, x_2, \ldots, x_8)$ is then the Cartesian product $\langle -4, 4\rangle \times \langle -4, 4\rangle \times \ldots \times \langle -4, 4\rangle = \langle -4, 4\rangle^8$. The point of the global minimum is $x_G = [3.0, 3.0, \ldots, 3.0]$, the point of the local minimum is $x_L = [0.0, 0.0, \ldots, 0.0]$. The parameters are $d = 144$ and $w = 1$. Concerning the parameters of the differential evolution algorithms, the number of individuals in the population is $NP = 800$, the total number of populations is $NG = 4000$, the crossover probability is $CR = 0.9$, the mutation factor is $F = 0.8$. The ratio of random individuals in the population is $R = 0$ for CDEA (the classic differential evolution algorithm has no random individuals), for MDEA we tested the following alternatives for the value of parameter R: $R = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7$.

7.3. Numerical testing—results. We summarized the results of the numerical testing in Table 1. The algorithms CDEA and all variants of MDEA were run 80 times. All algorithm parameters (except for the parameter R defining the number of random individuals in each population) were the same for both CDEA and MDEA as stated in the previous subsection. We recorded the number of successful attempts to find the global minimum of the cost function and evaluated the corresponding success ratio.

	CDEA	MDEA $R = 0.1$	MDEA $R = 0.2$	MDEA $R = 0.3$
Percentage of random individuals	0.0%	10.0%	20.0%	30.0%
Number of random individuals	0	80	160	240
Global minimum found	0	9	20	28
Success ratio in %	0.00%	11.25%	25.00%	35.00%
	MDEA $R = 0.4$	MDEA $R = 0.5$	MDEA $R = 0.6$	MDEA $R = 0.7$
Percentage of random individuals	40.0%	50.0%	60.0%	70.0%
Number of random individuals	320	400	480	560
Global minimum found	37	50	41	52
Success ratio in %	46.25%	62.50%	51.25%	65.00%

Table 1. Results of numerical testing of CDEA and MDEA.

It follows from Table 1 that for the specific cost function the algorithm CDEA was not able to find the global minimum of the cost function in any of 80 attempts. This is caused by a premature convergence of CDEA to the local minimum. As soon as the population concentrates around the local minimum and is small with respect to the distance from the local to global minimum, there is no chance for the algorithm to converge to the global minimum even if we increase substantially the number of populations. On the other hand MDEA was able to identify the global minimum of the cost function with nonzero success ratio in all variants of the algorithm. The success ratio increases with the increasing number of random individuals in the population from 11.25% ($R = 0.1$) up to 65.00% ($R = 0.7$). It is important to point out that the higher success ratio in finding the global minimum is paid off by the decrease in the convergence speed. It implies that in practical applications it is always important to strike a balance between the success ratio and convergence speed.

7.4. A real optimization task. We also compared the performance of CDEA and MDEA in a real optimization task. The aim of optimization is to locate a set of infrared heaters above a relatively complicated shell metal mould, so that the generated heat radiation intensity incident onto the mould surface is as uniform as possible. For the detailed description of this problem see the references [2] and [3]. Here we present only the results of CDEA and MDEA and their comparison regarding the attained minimum of the cost function.

The parameters of the optimization are: $D = 96$ (dimension of the task), $NP = 192$ (number of individuals in each population), $NG = 30000$ (number of populations), $CR = 0.98$, and $F = 0.6$ (crossover and mutation constants). When using MDEA we again tried several values of the parameter R representing the ratio of random individuals in the population. Specifically, $R = 0.02; 0.04; 0.08; 0.12; 0.16; 0.20$.

The best optimized state was achieved by MDEA with parameter $R = 0.12$ and the value of the cost function was by almost 9% lower than for the optimized state located by CDEA, which is a significant difference.

8. Conclusions

We concentrated in this paper on the topic of global convergence of the classic differential algorithm (CDEA).

In the first part we demonstrated by a counterexample that CDEA is not a global optimizer in the sense of finding the global minimum of the cost function under general circumstances. This does not exclude a possibility that using CDEA provides the global minimum, but such a result is not guaranteed and depends strongly on the specific cost function and optimization parameters.

Subsequently, we succeeded in designing a suitable simple modification of CDEA improving substantially its global convergence. MDEA (in contrast to CDEA) guarantees the asymptotic convergence to the global minimum of the cost function. That is, the probability that MDEA converges to the global minimum is increasing with increasing number of populations. We also provided the concepts and theoretical conclusions confirming the asymptotic convergence of MDEA.

In the next part of the paper we summarized testing CDEA and MDEA in numerical experiments. In these tests we verified better properties of MDEA concerning the global convergence as compared with CDEA. The results of the numerical tests are in conformity with theoretical expectations.

The last part describes the test of CDEA and MDEA in a real optimization task. The results justify using MDEA for optimization problems where attaining the lowest cost function value is essential.

References

[1] *Z. Hu, S. Xiong, Q. Su, X. Zhang*: Sufficient conditions for global convergence of differential evolution algorithm. J. Appl. Math. *2013* (2013), Article ID 139196, 14 pages. MR doi

[2] *J. Mlýnek, R. Knobloch, R. Srb*: Mathematical model of the metal mould surface temperature optimization. AIP Conference Proceedings 1690. AIP Publishing, Melville, 2015, Article No. 020018, 8 pages. doi

[3] *J. Mlýnek, R. Knobloch, R. Srb*: Optimization of a heat radiation intensity and temperature field on the mould surface. ECMS 2016 Proceedings, 30th European Conference on Modelling and Simulation. Regensburg, Germany, 2016, pp. 425–431. doi

[4] *K. V. Price*: Differential evolution: a fast and simple numerical optimizer. Proceedings of North American Fuzzy Information Processing. Berkeley, 1996, pp. 524–527. doi

[5] *K. V. Price, R. M. Storn, J. A. Lampien*: Differential Evolution. A Practical Approach to Global Optimization. Natural Computing Series. Springer, Berlin, 2005. zbl MR doi

[6] *D. Simon*: Evolutionary Optimization Algorithms. Biologically Inspired and Population-Based Approaches to Computer Intelligence. John Wiley & Sons, Hoboken, 2013. zbl MR

[7] *R. M. Storn, K. V. Price*: Differential evolution—a simple and efficient heuristics for global optimization over continuous spaces. J. Glob. Optim. *11* (1997), 341–359. zbl MR doi

Authors' address: *Roman Knobloch*, *Jaroslav Mlýnek*, *Radek Srb*, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, e-mail: `roman.knobloch@tul.cz`, `jaroslav.mlynek@tul.cz`, `radek.srb@tul.cz`.