Performance analysis of DE mutation schemes for constrained LSGO problems

A Vakhnin\textsuperscript{1}, E Sopov\textsuperscript{1,2}, D Kustov\textsuperscript{2} and I Panfilov\textsuperscript{1,2}

\textsuperscript{1}Reshetnev Siberian State University of Science and Technology
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russian Federation
\textsuperscript{2}Siberian Federal University
79, Svobodny Av., Krasnoyarsk, 660041, Russian Federation

E-mail: alexeyvah@gmail.com

Abstract. Differential Evolution (DE) is one of the most popular evolution algorithms (EAs) for numerical optimization problems with various difficulty levels. Mutation scheme makes a strong influence in DE performance. It produces new trial vectors based on the current population to improve the previous best-found solution. In this paper, we have analyzed the performance of a wide range of various mutation schemes for early proposed ε-CC-SHADE algorithm. We have used scaled benchmark set of constrained large-scale global optimization (cLSGO) problems to investigate ε-CC-SHADE performance. The new benchmark set is based on IEEE CEC 2017 Competition on Constrained Real-Parameter Optimization. The experimental results have shown the dependence of EA performance on the selected mutation scheme for solving optimization problems with high dimension. Based on the numerical results, we have made conclusions for choosing mutation scheme to increase DE performance for cLSGO.

1. Introduction
Actual real-world optimization problems can be characterized by the following properties. The first, these problems are formulated as «Black-Box» (BB) model [1] with constraints. The analytic form of BB problems and constraints are not known, due to the complexity of the fundamental analysis. We can measure only input and output values in BB optimization problems. The second, evaluation budget of the fitness function is limited. This is a consequence of the difficulty of computation evaluation of the potential solution. The third, the dimension of modern optimization problems is high. Optimization tasks with many variables (more than 100 or 1000) called as large-scale global optimization (LSGO) [2]. Constrained large-scale global optimization (cLSGO) problems can be stated as to:

\[
\begin{align*}
\text{minimize } & f(\mathbf{x}) \\
\text{such that } & a_i \leq x_i \leq b_i, \ i = 1,N \\
h_j(\mathbf{x}) & = 0, j = 1,k \\
g_j(\mathbf{x}) & \leq 0, j = 1,p
\end{align*}
\]
where $\mathbf{x}$ is objective vector, $f(\mathbf{x}) : R^N \rightarrow R$ is (fitness) function with $N$ variables. $h(\mathbf{x})$ and $g(\mathbf{x})$ are equality and inequality constraints, respectively. $a_i$ and $b_i$ are lower and upper bounds for $i$-th dimension. The constraints divide search space into feasible and infeasible domains.

LSGO and cLSGO problems are exceedingly difficult for classic optimization techniques. For today, cLSGO problems are not studied well. In addition, cLSGO benchmark set is not exists. This paper presents novel benchmark with cLSGO problems. In this paper, we have studied performance of 12 various mutation schemes for $\varepsilon$-CC-SHADE on proposed cLSGO benchmark. Numerical experiments have shown that $DE/rand/1$, $DE/tour/1$ and $DE/cu-rto-pbest/1(tour)$ mutation strategies demonstrate statistically the best performance than other strategies in solving cLSGO problems for $\varepsilon$-CC-SHADE.

The rest of the paper is organized as follows. Section 2 describes related work. In Section 3, the actual study is described. In Section 4, the experimental setup and results of numerical experiments are discussed. In the conclusion, the results and further research are discussed.

2. Related Work

2.1. Classic Differential Evolution
The first reference about DE was made by [3]. The proposed DE is used for optimization problems and does not use information about function gradient. It is well known that classical optimization algorithms require analytic form of gradient to optimize problem [4]. DE has three main control parameters, such as $NP$ (population size), $F$ (scale factor) and $CR$ (crossover rate). More information about DE and its modifications you can find in this survey [5].

2.1.1. Success-History based Adaptive DE. SHADE was proposed by Tanabe and Fukunaga [6], it is improved version of JADE [7]. SHADE is based on self-adaptation of control DE parameters and external archive to store replaced solutions. SHADE records values of $F$ and $CR$ which show good performance during the optimization process. SHADE and its modifications have shown its effectiveness repeatedly in solving optimization problems [8], [9].

2.2. Cooperative Coevolution framework
Cooperative Coevolution (CC) is a powerful framework to increase the performance of EA in solving optimization problems with many variables. The main idea of CC is decomposing objective vector into small pieces and optimizing them independently to each other using some EA. This approach was proposed in 1994 by Potter and Jong [10]. Many metaheuristics use CC in dealing with LSGO problems [2]. Examples of applying CC can be found in [11], [12].

2.3. $\varepsilon$DE for constraint handling
In this paper, we have used $\varepsilon$DE approach for constraint-handling [13]. There is transformation of classic DE selection operator to (5) in $\varepsilon$DE.

\[
(f(X_1), v(X_1)) <_\varepsilon (f(X_2), v(X_2)) \iff \begin{cases} f(X_1) < f(X_2) \text{ if } v(X_1), v(X_2) \leq \varepsilon \\ f(X_1) < f(X_2) \text{ if } v(X_1) = v(X_2) \\ v(X_1) < v(X_2), \text{ otherwise} \end{cases}
\]

(5)

Where $f(X)$ is fitness value of $X$ solution, $v(X)$ is value of violation which defines as (6):

\[
v(X) = \frac{\sum_{i=1}^{p} g_i(X) + \sum_{i=1}^{k} h_i(X)}{p + k}
\]

(6)

To control $\varepsilon$ parameter in (5), we have used modified (7) and (8) formulas. Original formulas were taken from [14].

\[
\varepsilon = \begin{cases} E, & \text{if } FEV \leq 0.8 \cdot \text{MaxFEV} \\ 0, & \text{otherwise} \end{cases}
\]

(7)
\[ E = \left( 1 - \frac{FEV}{MaxFEV} \right)^c_p \cdot v(X_{\theta \cdot NP}) \]  

where \( FEV \), MaxFEV are current number of fitness evaluations and maximum budget of fitness evaluation, respectively. \( v(X_{\theta \cdot pop \cdot size}) \) is an element of sorted (from best to worse) violation array with \( \theta \cdot NP \) index, rounded to the nearest integer value. The set of parameters values for \( c_p \) and \( \theta \) are equal to 3 and 0.8, respectively. After 80% of \( FEV \) budget is exhausted, we set \( E \) is equal to 0. As a consequence, \( \varepsilon \)-CC-SHADE has to concentrate search focus in local region.

2.4. \( \varepsilon \)-CC-SHADE

Based on 2.1.1., 2.2. and 2.3., we have combined SHADE optimization algorithm, cooperative coevolution approach and cDE constraint handling. As a result, we have designed \( \varepsilon \)-CC-SHADE optimization algorithm for solving cLSGO problems. The numerical results with conclusions are shown in 4 Section.

2.5. Constrained Large-scale Global Optimization Benchmark

There is no exist cLSGO benchmark set for investigating performance of EAs for constrained optimization problems with many variables. This paper presents a novel benchmark set which based on the scalable problems from IEEE CEC 2017 Competition on Constrained Single Objective Real-Parameter Optimization (CSORPO) [15]. We have scaled problems without rotation matrix up to 1000D. This set was named as cLSGO 2017. Problems with rotation matrix were not included in cLSGO 2017. Table 1 contains information about the added benchmark problems into cLSGO 2017 benchmark set.

| CEC’2017 CSORPO | cLSGO 2017 |
|-----------------|------------|
| 1               | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 |

3. Actual Study

As we mentioned before, the DE performance strongly depends on its control parameters and type of mutation scheme. This paper shows the performance of \( \varepsilon \)-CC-SHADE algorithm with various mutation operators. We have used the schemes which you can see in table 2.

| Name mutation scheme | Mutation operator formula | Designation |
|----------------------|---------------------------|-------------|
| DE/rand/1            | \( x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) \) | mut_1       |
| DE/rand/2            | \( x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) + F \cdot (x_{r_4} - x_{r_5}) \) | mut_2       |
| DE/best/1            | \( x_{best} + F \cdot (x_{r_2} - x_{r_5}) \) | mut_3       |
| DE/best/2            | \( x_{best} + F \cdot (x_{r_2} - x_{r_3}) + F \cdot (x_{r_4} - x_{r_5}) \) | mut_4       |
| DE/cut-to-best/1     | \( x_i + F \cdot (x_{best} - x_i) + F \cdot (x_{r_2} - x_{r_5}) \) | mut_5       |
| DE/cut-to-pbest/1    | \( x_i + F \cdot (x_{r_pbest} - x_i) + F \cdot (x_{r_2} - x_{r_5}) \) | mut_6       |
| DE/tour/1            | \( x_{t_1} + F \cdot (x_{t_2} - x_{t_3}) \) | mut_7       |
| DE/tour/2            | \( x_{t_1} + F \cdot (x_{t_2} - x_{t_3}) + F \cdot (x_{t_4} - x_{t_5}) \) | mut_8       |
| DE/best/1(tour)      | \( x_{best} + F \cdot (x_{t_2} - x_{t_5}) \) | mut_9       |
| DE/best/2(tour)      | \( x_{best} + F \cdot (x_{t_2} - x_{t_3}) + F \cdot (x_{t_4} - x_{t_5}) \) | mut_10      |
DE/cut-to-best/1(tour)  \( x_i + F \cdot (x_{\text{best}} - x_i) + F \cdot (x_{t_s} - x_{t_3}) \)  mut_11
DE/cut-to-pbest/1(tour)  \( x_i + F \cdot (x_{p\text{best}} - x_i) + F \cdot (x_{t_s} - x_{t_5}) \)  mut_12

The first column is the name of DE mutation scheme. The second column consists formula of mutation operator. The last column contains short notations for every mutation scheme. This part of them (from mut_1 to mut_6) were taken from [16]. Other mutation schemes (mut_7 to mut_12) were modified using tournament selection. In this study, the size of the tournament was equal to 2. Original SHADE uses mut_6 strategy to produce new individuals. In mut_6, pbest index is randomly selected from part of the best \( NP \cdot p \) \( (p \in (0, 1]) \) individuals in the current population. In this study, \( p \) is equal to 0.1. SHADE stores replaced individuals into external archive to save previous experience about solving problem. If the external archive is full, EA randomly deletes one individual to make space for the newly inserted solution. SHADE uses individuals from the external archive to generate new solutions using mutations of solution vectors. In classic DE, all indices \( r_1 \) are generated randomly from \([1; NP]\). In the case of SHADE, indices \( r_3, r_5, t_3 \) and \( t_5 \) are generated from the interval \([1; NP + |A|]\), indices \( r_1, r_2, r_4, t_1, t_2 \) and \( t_4 \) are generated from the interval \([1; NP]\). A is the external archive, \(|A|\) is the number of individuals who stand in the external archive.

4. Experimental setup and discussion numerical experiments

4.1. Experimental setup
In this paper, all numerical experiments were executed using the following PC system. We have used 2 CPUs (Ryzen 7 1700x and 2700) to parallelize numerical experiments to decrease computation time. \( \varepsilon \)-CC-SHADE were realized on C++ program language.

We have used the following parameters for \( \varepsilon \)-CC-SHADE. The population size for each subcomponent is 50. The size of external archive is 100. The number of subcomponents is 8. The number of independent runs is 25 for each benchmark problem. Fitness budget is \( 3 \cdot 10^6 \) for each run.

4.2. Numerical experiments and results
Table 3 shows results of numerical experiment for some benchmark problem. The first column is cLSGO problem number. The next columns contain the \( \varepsilon \)-CC-SHADE performance values with various mutation schemes from table 2. Each cell contains two values: the median and the constraint violation, evaluated over 25 independent runs. Figure 1 demonstrates average ranking results, based on experimental results, for \( \varepsilon \)-CC-SHADE performance with various mutation schemes. EA with the best mutation scheme has the smallest rank. The ranking values are based on the median best-found values averaged over all cLSGO benchmark problems. As we can note, \( DE/\text{rand}/1 \), \( DE/\text{tour}/1 \) and \( DE/\text{cut-to-pbest}/1(\text{tour}) \) have the smallest ranks. Also, mutation schemes with tournament selection indexes have the smallest rank than similar mutation schemes with random selection indexes in all cases except \( DE/\text{rand}/1 \) versus \( DE/\text{tour}/1 \). Figure 2 shows ranking for three best mutation schemes for every benchmark problem. Table 4 proves difference results of estimation the performance of mutation scheme between each other using Mann–Whitney U test with normal approximation and tie correction with \( p \) value is equal to 0.01. The first column and the first row have designations for mutation schemes. Each cell contains the value which has been calculated as follows. If one mutation scheme (from column) has better, worse or equal performance to another (from row) on the benchmark problem, then add +1, -1 or 0 to this value, respectively. The last column contains a value calculated as the sum of points that a particular mutation scheme has scored.
As we can see, DE/rand/1 versus DE/rand-to-pbest/1(tour) have the same performance (as in ranking test), the difference between these two schemes, using Mann–Whitney U test, is equal to 0. DE/rand/1 versus DE/tour/1 and DE/tour/1 versus DE/rand-to-pbest/1(tour) have quite similar performance, the differences are -3 and 1, respectively.
Table 4. Comparison performance between various mutation schemes using Mann–Whitney U test for ε-CC-SHADE on the eLSGO benchmark problems.

| mut_1 | mut_2 | mut_3 | mut_4 | mut_5 | mut_6 | mut_7 | mut_8 | mut_9 | mut_10 | mut_11 | mut_12 | Total sum |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|-----------|
| 7     | 8     | 12    | 8     | 1     | -3    | 6     | 6     | 10     | 5      | 0      | 60      |
| -7    | -6    | 13    | -8    | -10   | -6    | -11   | -3    | -9     | -11    | -99    | 13      |
| -12   | -11   | -13   | -12   | -12   | -11   | -16   | -11   | -14    | -13    | -137   |         |
| -8    | -4    | 8     | 12    | -     | -11   | -11   | -5    | 2      | 8      | -9     | -12    | -30      |
| -1    | 4     | 10    | 12    | 11    | -4    | 3     | 9     | 10     | 10     | -5     | 59      |
| 3     | 5     | 10    | 12    | 11    | 4     | -     | 5     | 8      | 10     | 7      | 1       | 76       |
| -6    | -4    | 6     | 11    | 5     | -3    | -5    | -     | 5      | 5      | 3      | -2      | 21       |
| -6    | -4    | 11    | 16    | -2    | -9    | -8    | -5    | -9     | -8     | -8     | -4      | -15      |
| -10   | -7    | 3     | 11    | -8    | -10   | -5    | -9    | -9     | -11    | -65    |         |
| -5    | -3    | 9     | 14    | 9     | -10   | -7    | -3    | 8      | 9      | -10    | 11      |
| 0     | 4     | 11    | 13    | 12    | 5     | -1    | 2     | 9      | 11     | 10     | -76     |

5. Conclusions
In this paper, we have studied performance of 12 various mutation schemes for ε-CC-SHADE algorithm for constrained large-scale global BB optimization. The experimental results have shown that DE/rand/1, DE/tour/1 and DE/cu-to-pbest/1(tour) strategies demonstrate better results than the others. Also, we have no found significance differences between these mutation schemes. In further works, we will try to increase ε-CC-SHADE performance using self-adaptive mechanism of choosing mutation strategy during the optimization process.

References
[1] Xiao J, Li W, Li W and Xiao X 2015 Optimization on Black Box Function Optimization Problem Math. Prob. in Eng. pp 1-10 doi:10.1155/2015/647234
[2] Mahdavi S, Shirli M E and Rahnamayan S 2015 Metaheuristics in large-scale global continues optimization: A survey Inf. Sci. 295 pp 407-28 doi:10.1016/j.ins.2014.10.042
[3] Storn R and Price K 1995 Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces Journal of Global Opt. 23
[4] Shi Z-J and Guo J 2009 Global convergence of conjugate gradient method Opt. 58 (2), pp 163-179 doi:10.1080/02331939802434633
[5] Das S and Suganthan P N 2011 Differential Evolution: A Survey of the State-of-the-Art IEEE Trans. on Evol. Comp. 15 (1) pp 4-31 doi:10.1109/tevc.2010.2059031
[6] Tanabe R and Fukunaga 2013 A Success-history based parameter adaptation for Differential Evolution 2013 IEEE Cong. on Evol. Comp. pp 71-78 doi:10.1109/cec.2013.6557555
[7] Zhang J and Sanderson A C 2009 JADE: Adaptive Differential Evolution With Optional External Archive IEEE Tran. on Evol. Comp. 13 (5) pp 945-58 doi:10.1109/tevc.2009.2014613
[8] Kovar S, Valouch J, Kadavy T, Pospisilik M and Adamek M 2018 Enclosure shielding effectiveness calculation using SHADE algorithm 19th Int. Sci. Conf. on Elec. Pow. Eng. (EPE) doi:10.1109/epc.2018.8396021
[9] Viktorin A, Senkerik R, Pluhacek M and Kadavy T 2018 Cluster Occurrence in the DbL_SHADE Population 2018 IEEE Cong. on Evol. Comp. doi:10.1109/cec.2018.8477793
[10] Potter M A and Jong K A 1994 A cooperative coevolutionary approach to function optimization Lec. Not. in Com. Sci. pp 249-57 doi:1007/3-540-58484-6_269
[11] Jiao B, Chen Q and Yan S 2011 A cooperative co-evolution PSO for flow shop scheduling problem with uncertainty JCP. 6 pp 1955-61 10.4304/jcp.6.9.1955-1961.
[12] Boonlong K 2014 Vibration-based damage detection in beams by cooperative coevolutionary genetic algorithm Adv. Mech. Eng.
[13] Takahama T, Sakai S and Iwane N 2006 Solving Nonlinear Constrained Optimization Problems by the ε Constrained Differential Evolution 2006 IEEE Int. Conf. on Sys., Man and Cyber. 3
pp 2322-27. doi:10.1109/icsmc.2006.385209

[14] Brest J 2009 Constrained Real-Parameter Optimization with \(\varepsilon\)-Self-Adaptive Differential Evolution Stud. in Comp. Int. 198 pp 73-93 doi:10.1007/978-3-642-00619-7_4

[15] Wu G, Mallipeddi R and Suganthan P 2017 Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization Tech.1 Rep.

[16] Opara K and Arabas J 2018 Comparison of mutation strategies in Differential Evolution – A probabilistic perspective Swarm and Evol. Comp. 39 doi:10.1016/j.swevo.2017.12.007