Improving DE-based cooperative coevolution for constrained large-scale global optimization problems using an increasing grouping strategy

To cite this article: A Vakhnin and E Sopov 2020 IOP Conf. Ser.: Mater. Sci. Eng. 734 012099
Improving DE-based cooperative coevolution for constrained large-scale global optimization problems using an increasing grouping strategy

A Vakhnin and E Sopov
Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochny Av., Krasnoyarsk, 660037, Russian Federation
E-mail: alexeyvah@gmail.com, evgenysopov@gmail.com

Abstract. Nowadays, high-dimensional constrained «Black-Box» (BB) optimization problems has become more urgent. At the same time, the constrained large-scale global optimization (cLSGO) problems are not well studied and many modern optimization approaches demonstrate low performance when dealing with cLSGO problems. Evolution algorithms (EAs) has proved their efficiency in solving low-dimensional constrained optimization problems and high-dimensional single-objective optimization problems. In this study, we have proposed a new approach based on the cooperative coevolution (CC) framework and an algorithm for increasing size of variables grouping on the decomposition stage (iCC) when solving cLSGO problems. We have proposed a novel EA that combines SHADE, iCC and ε-constrained method (ε-iCC-SHADE). The proposed optimization algorithm has been investigated using a new cLSGO benchmark, which is based on scalable problems from IEEE CEC 2017 Competition on Constrained Real-Parameter Optimization. The numerical experiments have shown that ε-iCC-SHADE outperforms the early proposed ε-CC-SHADE algorithm which operates with the fixed number of subcomponents.

1. Introduction
The technical progress does not stay still. The complexity of modern problems in different spheres of human activity grows rapidly [1]. As a result, we can observe increasing the number of variables in actual optimization problems. Global BB optimization problems with the number of variables greater than a thousand are called large-scale global optimization problems (LSGO). In [2], the overview of current state of LSGO is presented. Dealing with the high-dimensionality is a challenging task for many optimization techniques. Usually LSGO deals with single objective and no constraints, at the same time, many real-world optimization problems are the constrained optimization problems [3, 4]. The constrained optimization problems (COPs) are known as optimization problems which contain additional requirements and limitations that divide the search space into feasible and infeasible subsets. Many complex COPs deal with both BB objective and BB constraints. Searching for simultaneously the global-best and feasible solution is known as a hard optimization problem. The constrained large-scale global BB optimization combines features of LSGO and COPs and still is not studied. The constrained large-scale global optimization (cLSGO) problem can be formulated as follows:

\[ f(\bar{x}) \rightarrow \min_{\bar{x}} f(x) \]
\[ x_i^L \leq x_i \leq x_i^U, \quad i = 1, n \quad (2) \]
\[ g_j(\vec{x}) \leq 0, \quad j = 1, p \quad (3) \]
\[ h_j(\vec{x}) = 0, \quad j = 1, k \quad (4) \]

where \(\vec{x}\) is vector of objective variables, \(f(\vec{x}): \mathbb{R}^n \rightarrow \mathbb{R}\), \(g(\vec{x})\) and \(h(\vec{x})\) are the objective (fitness) function with \(n\) objective variables, inequality constraints and equality constraints respectively, \(p\) and \(k\) are the numbers of inequality and equality constraints, \(x_i^L\) and \(x_i^U\) are lower and upper bounds for \(i\)-th variable.

According to LSGO survey [2], there exist two popular and effective ways for solving LSGO problems: non-decomposition methods and cooperative coevolution (CC) methods. Non-decomposition methods use special operators and techniques for EAs for dealing with the large number of variables at once [5, 6]. The main idea of CC techniques [7, 8] is to divide the objective vector into smaller parts and to optimize them independently using EAs. For many problems approaches based on CC show better results in solving LSGO problems. CC framework provides a decomposition of a LSGO problem and helps to overcome the «curse of dimension» (CoD) [9].

In this paper, we have proposed and investigated a novel CC approach for solving cLSGO problems, which is able to increase the number of subcomponents during an EA run. The approach is title «iCC». We have chosen SHADE (Success-history based parameter adaptation for Differential Evolution) [10] as the main optimizer in CC. The SHADE approach can effectively self-adapt its parameters, namely \(F\) (scale factor) and \(CR\) (crossover rate). There exist a number of EA modifications for solving constrained optimization problems [11, 12, 13, 14]. A good survey on constrained-handling approaches is proposed in [15]. In our study, we will use the well-known and well-studied \(\epsilon\)-DE method [16]. The whole approach that combines SHADE, iCC and \(\epsilon\)-DE is title as \(\epsilon\)-iCC-SHADE. In this study, we have compared the performance of the proposed algorithm with the previously proposed \(\epsilon\)-CC-SHADE with the fixed number of subcomponents. The rest of the paper is organized as follows. Section 2 describes related work. In Section 3, the proposed approach is described. In Section 4, the experimental setups and results of numerical experiments are discussed. In the Conclusion the results and further research are discussed.

2. Related work

2.1. SHADE algorithm
Differential evolution (DE) is a powerful approach for solving optimization problems of various complexity levels. This approach, first introduced by Storn and Price [17, 18]. DE algorithm and its variants have proved effectiveness many times [19, 20]. Classic DE has three main control parameters: \(pop\_size\) (population size), \(F\) — scale factor, \(CR\) — crossover rate. SHADE is one of DE variants with self-adaptive mechanisms of \(F\) and \(CR\) parameters was created by Tanabe and Fukunaga [10]. SHADE records successful parameter values and use this information to generate new values of \(F\), \(CR\). Besides, SHADE transfers replaced individuals into external archive to maintain previous experience and use they in mutation stage.

2.2. Cooperative Coevolution framework
For today, Cooperative Coevolution is one of the high-performance frameworks for solving LSGO problems. CC was proposed by Potter and De Jong [21, 22]. The main idea of CC is optimization solution vector sequentially in parts with core of some EA (in this paper, we have used SHADE algorithm as a core of CC). As a result, dimension and complexity of LSGO problem decreases. CC has important controlling parameter, the CC performance strongly depends on the number of its subcomponents. As we noted before, LSGO problems are BB problems, the structure of problem is unknown. Thus, in this study, CC performs with equal size of subcomponents. In our study, we have
used the following rule: \( s \cdot m = N \), where \( s \) is number of variables into one subcomponent, \( m \) is the number of subcomponents and \( N \) is total number of variables in the optimization task.

2.3. \( \varepsilon \)DE constrained handlings
In fact, that, there are a lot of constrained-handlings techniques for today [15]. In this study, we have used \( \varepsilon \)DE [16] for constrained-handling. \( \varepsilon \)DE transforms of «Selection» operator in DE using formula (5).

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

(5)

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

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

(6)

\[
G_i(X) = \begin{cases} g_i(X), & \text{if } g_i(X) > 0 \\ 0, & \text{otherwise} \end{cases}
\]

(7)

\[
H_i(X) = \begin{cases} |h_j(X)|, & \text{if } |h_j(X)| - \epsilon > 0 \\ 0, & \text{otherwise} \end{cases}
\]

(8)

In (8) \( \epsilon \) is the tolerance threshold, for all equality constraints it is equal to 0.0001. Analyzing the formula (5), we can conclude that violation values have higher priority than fitness function values in comparing two solutions between each other. To control \( \varepsilon \) parameter in (5), we have used modified (9) formula. We were inspired by [23].

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

(9)

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

Where \( \text{FEV}, \text{MaxFEV} \) are current number of fitness evaluations and maximum budget of fitness evaluation for current run, respectively. \( v(X_{[\theta \cdot \text{pop}\_size]}) \) is a violation value for solution \( X \) with index \( [\theta \cdot \text{pop}\_size] \) after sorting (from best to worse solution). \( c_p \) is a parameter to control the speed of constraints, \( c_p \) is equal to 3. \( \text{pop}\_size \) is a number means population size, \( \theta \) is equal to 0.8. We have tried use different values for \( c_p \) and \( \theta \) parameters, this set of parameters has shown better performance. In (9), we set \( E \) is equal to 0 after evaluated of 80% of total FEV budget, because in the last generations EA has to concentrate search focus in local region.

3. Proposed increasing grouping strategy for Cooperative Coevolution framework
In this section, we will describe our proposed and improved CC approach which is named as «iCC». It is a known fact that EA shows quite good performance, for solving optimization problems, if it is guided the following behavior. In the first iterations, EA should use principles of global search, but near the end of FEV budget, EA should use methods of searching near best found solutions. To control decreasing the number of subcomponents we have used simple rules, see formula (10). Where \( m \) is the number of subcomponents in CC.
\[ m = \begin{cases} 
10, & \text{if } \text{FEV}[0, t_1] \\
8, & \text{if } \text{FEV}[t_1 + 1, t_2] \\
4, & \text{if } \text{FEV}[t_2 + 1, t_3] \\
2, & \text{if } \text{FEV}[t_3 + 1, t_4] \\
1, & \text{if } \text{FEV}[t_4 + 1, \max \text{FEV}] 
\end{cases} \quad (10) \\
\forall i: t_{i+1} - t_i = 0.2 \cdot \max \text{FEV} \]

When we provide decomposition into small groups of variables, an EA is able to converge faster to a perspective region of the search space, especially, if the EA has found an appropriate grouping of interacting variables. At the same time, such a coordinate-wise-like search can be too “greedy”, and larger grouping is more preferable. We will increase sizes of subcomponents during the EA run, and finally, we will finish the search using the whole solution vector \((m = 1)\). We will set the equal fitness evaluation budget for each CC stage.

4. Experimental setup and discussion numerical results

4.1. Benchmark set for Constrained Large-scale Global Optimization Problems

Today no benchmark problems for cLSGO is proposed. In this paper, we have proposed a new benchmark based on scalable problems from IEEE CEC 2017 Competition on Constrained Single Objective Real-Parameter Optimization (CSORPO) [24]. All problems from IEEE CEC 2017 benchmark which do not use low-dimensional transformation matrixes, are scaled up to 1000 dimensions and are included in the new set of cLSGO problems. Detailed information on the cLSGO benchmark is shown in table 1.

| cLSGO problem | CEC’2017 CSORPO | Type of objective | The number and type of constraints |
|---------------|-----------------|------------------|-----------------------------------|
| cLSGO01       | C01             | Non Separable    | Equality: 0, Inequality: 1, Separable |
| cLSGO02       | C03             | Non Separable    | Equality: 1, Inequality: 1, Separable |
| cLSGO03       | C04             | Separable        | Equality: 0, Inequality: 2, Separable |
| cLSGO04       | C06             | Separable        | Equality: 6, Inequality: 0, Separable |
| cLSGO05       | C07             | Separable        | Equality: 2, Inequality: 0 |
| cLSGO06       | C08             | Separable        | Equality: 2, Inequality: 0 |
| cLSGO07       | C09             | Separable        | Equality: 2, Inequality: 0 |
| cLSGO08       | C10             | Separable        | Equality: 2, Inequality: 0 |
| cLSGO09       | C11             | Separable        | Equality: 1, Inequality: 0 |
| cLSGO10       | C12             | Separable        | Equality: 0, Inequality: 2, Separable |
| cLSGO11       | C13             | Non Separable    | Equality: 0, Inequality: 3, Separable |
| cLSGO12       | C14             | Non Separable    | Equality: 1, Inequality: 1, Separable |
| cLSGO13       | C15             | Separable        | Equality: 1, Inequality: 1 |
| cLSGO14       | C16             | Separable        | Equality: 1, Inequality: 1, Separable |
| cLSGO15       | C17             | Non Separable    | Equality: 1, Inequality: 1, Separable |
| cLSGO16       | C18             | Separable        | Equality: 1, Inequality: 2, Non Separable |
| cLSGO17       | C19             | Separable        | Equality: 0, Inequality: 2, Non Separable |
| cLSGO18       | C20             | Non Separable    | Equality: 0, Inequality: 2 |

4.2. Software implementation and setups

All numerical experiments were executed using the following system:

- OS: Ubuntu Linux 18.04 LTS
- CPU: Ryzen 7 1700x (8C/16T) and Ryzen 2700 (8C/16T)
- RAM: 16 GB
- IDE: Code::Blocks 17.12
- Programming Language: C++
- Compiler: g++ (gcc) with -O3 optimization flag

We have used the following main settings for all numerical experiments: the number of variables for all problems is \( D=1000 \), the number of independent runs is 25 runs per a benchmark problem, the maximum number of fitness evaluations is \( \text{Max } F_{E/V} = 3 \times 10^6 \) in each independent run.

In this study, we use the following notation, \( \epsilon \text{-CC-SHADE} (m) \) means that \( \epsilon \text{-CC-SHADE} \) algorithm uses \( m \) subcomponents for CC. The population size is equal to 25, 50, 75 and 100, and \( m \) is equal to 1, 2, 4, 8, 10. For decreasing computation time, we have applied parallel computing with 32 threads using Ryzen 7 1700X and 2700 with 8 CPU cores and 16 threads.

Table 2, table 3, table 4 and table 5 show results of numerical experiments for population size equal to 25, 50, 75 and 100, respectively. The first column is cLSGO problem number. The next columns contain the \( \epsilon \text{-CC-SHADE} (m) \) and \( \epsilon \text{-iCC-SHADE} \) performance values. Each cell contains two values: the median and the constrains violation, evaluated over 25 independent runs. The Figures 1-4 demonstrate ranking results for the EAs performance with population size equal to 25, 50, 75 and 100, respectively. The best EA has the smallest rank, the ranking is based on the median best-found values averaged over all cLSGO benchmark problems.

**Table 2.** The experimental results for \( \epsilon \text{-CC-SHADE} \) and \( \epsilon \text{-iCC-SHADE} \) on the cLSGO benchmark problems, population size is 25

| №  | \( \epsilon \text{-CC-SHADE} (1) \) | \( \epsilon \text{-CC-SHADE} (2) \) | \( \epsilon \text{-CC-SHADE} (4) \) | \( \epsilon \text{-CC-SHADE} (8) \) | \( \epsilon \text{-CC-SHADE} (10) \) | \( \epsilon \text{-iCC-SHADE} \) |
|----|--------------------------------|--------------------------------|-------------------------------|------------------------|---------------------------|-----------------|
| 1  | 1.04E+06                      | 7.55E+05                       | 4.48E+05                     | 2.10E+05               | 1.72E+05                  | 2.50E+05        |
|    | 0.00E+00                      | 0.00E+00                       | 0.00E+00                     | 0.00E+00               | 0.00E+00                  | 0.00E+00        |
| 2  | 1.47E+08                      | 1.74E+08                       | 1.59E+08                     | 1.93E+08               | 2.05E+08                  | 1.51E+08        |
|    | 0.00E+00                      | 0.00E+00                       | 0.00E+00                     | 0.00E+00               | 0.00E+00                  | 0.00E+00        |
| 3  | 3.34E+04                      | 2.89E+04                       | 1.91E+04                     | 1.02E+04               | 8.19E+03                  | 8.25E+03        |
|    | 0.00E+00                      | 0.00E+00                       | 0.00E+00                     | 0.00E+00               | 0.00E+00                  | 0.00E+00        |
| 4  | 1.84E+05                      | 1.91E+05                       | 1.52E+05                     | 1.81E+05               | 1.80E+05                  | 1.80E+05        |
|    | 3.23E-03                      | 1.88E-02                       | 2.36E-02                     | 3.84E-02               | 5.72E-02                  | 7.40E-03        |
| 5  | -1.07E+04                     | -2.27E+03                      | -4.50E+02                    | 1.64E+02               | 1.24E+02                  | -3.50E+03       |
|    | 0.00E+00                      | 0.00E+00                       | 0.00E+00                     | 0.00E+00               | 0.00E+00                  | 0.00E+00        |
| 6  | 1.78E+02                      | 1.53E+02                       | 1.59E+02                     | 1.59E+02               | 1.45E+02                  | 1.55E+02        |
|    | 7.38E+05                      | 5.23E+05                       | 3.48E+05                     | 2.65E+05               | 3.57E+05                  | 2.81E+05        |
| 7  | 1.87E+01                      | 1.95E+01                       | 1.74E+01                     | 1.73E+01               | 1.77E+01                  | 1.80E+01        |
|    | 5.72E+02                      | 2.84E+02                       | 9.48E+00                     | 0.00E+00               | 0.00E+00                  | 0.00E+00        |
| 8  | 1.35E+02                      | 1.41E+02                       | 1.11E+02                     | 9.38E+01               | 1.40E+02                  | 9.32E+01        |
|    | 1.81E+06                      | 1.34E+06                       | 5.96E+05                     | 2.79E+05               | 3.24E+05                  | 3.93E+05        |
| 9  | -2.70E+03                     | -5.80E+03                      | -1.09E+04                    | -5.02E+03              | -4.76E+03                 | -8.48E+03       |
|    | 4.05E+04                      | 2.77E+04                       | 1.39E+04                     | 6.92E+03               | 1.43E+04                  | 9.11E+03        |
| 10 | 2.72E+04                      | 1.57E+04                       | 8.45E+03                     | 6.77E+02               | 1.42E+02                  | 3.75E+01        |
|    | 1.30E+04                      | 7.52E+03                       | 4.13E+03                     | 2.67E+02               | 5.45E+01                  | 1.63E+01        |
| 11 | 2.51E+09                      | 2.73E+09                       | 1.22E+09                     | 4.10E+08               | 2.66E+08                  | 3.38E+08        |
|    | 1.15E+04                      | 1.28E+04                       | 8.87E+03                     | 6.26E+03               | 5.38E+03                  | 5.44E+03        |
| 12 | 2.03E+01                      | 2.01E+01                       | 2.01E+01                     | 2.00E+01               | 2.00E+01                  | 2.00E+01        |
|    | 2.10E+04                      | 2.05E+04                       | 7.86E+03                     | 6.49E+02               | 8.25E+01                  | 2.16E+02        |
| 13 | 1.06E+02                      | 1.06E+02                       | 9.03E+01                     | 9.03E+01               | 7.46E+01                  | 1.12E+02        |
|    | 0.00E+00                      | 0.00E+00                       | 0.00E+00                     | 0.00E+00               | 0.00E+00                  | 0.00E+00        |
| 14 | 3.30E+03                      | 4.19E+03                       | 4.71E+03                     | 5.93E+03               | 6.31E+03                  | 6.13E+03        |
|    | 0.00E+00                      | 0.00E+00                       | 0.00E+00                     | 0.00E+00               | 0.00E+00                  | 0.00E+00        |
| 15 | 3.92E+00                      | 2.00E+00                       | 2.00E+00                     | 2.00E+00               | 2.00E+00                  | 2.00E+00        |
|    | 4.33E+03                      | 5.01E+02                       | 5.01E+02                     | 5.01E+02               | 5.01E+02                  | 5.01E+02        |
| 16 | 4.66E+04                      | 2.50E+04                       | 9.30E+03                     | 3.32E+03               | 3.11E+03                  | 7.37E+03        |
|    | 1.14E+09                      | 3.55E+08                       | 6.44E+07                     | 1.35E+06               | 1.35E+05                  | 8.59E+03        |
Table 3. The experimental results for ε-CC-SHADE and ε-iCC-SHADE on the CLSGO benchmark problems, population size is 50.

| №  | ε-CC-SHADE (1) | ε-CC-SHADE (2) | ε-CC-SHADE (3) | ε-CC-SHADE (4) | ε-CC-SHADE (8) | ε-CC-SHADE (10) | ε-iCC-SHADE |
|----|----------------|----------------|----------------|----------------|----------------|----------------|-------------|
| 1  | 9.71E+05       | 5.87E+05       | 2.52E+05       | 8.97E+04       | 6.78E+04       | 1.35E+05       |             |
|    | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       |             |
| 2  | 8.70E+07       | 1.04E+08       | 1.51E+08       | 1.11E+08       | 1.90E+08       | 9.89E+07       |             |
|    | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       |             |
| 3  | 2.75E+04       | 1.97E+04       | 9.95E+03       | 5.39E+03       | 4.66E+03       | 4.72E+03       |             |
|    | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       |             |
| 4  | 1.61E+05       | 1.74E+05       | 1.64E+05       | 1.79E+05       | 1.65E+05       | 1.34E+05       |             |
|    | 1.96E-03       | 1.39E-02       | 4.28E-02       | 8.48E-02       | 9.98E-02       | 5.23E-03       |             |
| 5  | -1.08E+04      | -4.00E+02      | -3.42E+02      | -4.14E+02      | 8.49E+02       | -1.43E+03      |             |
|    | 0.00E+00       | 4.94E-03       | 1.07E+04       | 1.90E+04       | 2.19E+04       | 0.00E+00       |             |
| 6  | 1.51E+02       | 1.44E+02       | 1.40E+02       | 1.27E+02       | 1.11E+02       | 1.08E+02       |             |
|    | 5.64E+05       | 3.25E+05       | 1.99E+05       | 1.71E+05       | 2.45E+05       | 1.91E+05       |             |
| 7  | 1.74E+01       | 1.76E+01       | 1.59E+01       | 1.51E+01       | 1.56E+01       | 1.48E+01       |             |
|    | 3.47E+01       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       |             |
| 8  | 1.00E+02       | 8.09E+01       | 5.12E+01       | 4.28E+01       | 6.14E+01       | 2.75E+01       |             |
|    | 1.62E+06       | 6.70E+05       | 1.80E+05       | 7.27E+04       | 7.89E+04       | 8.30E+04       |             |
| 9  | -9.67E+03      | 5.43E+02       | -5.95E+03      | -4.74E+03      | -8.26E+03      | -8.23E+03      |             |
|    | 1.61E+04       | 4.67E+03       | 1.33E+03       | 9.04E+02       | 4.25E+03       | 1.57E+03       |             |
| 10 | 6.33E+03       | 5.87E+02       | 4.44E+00       | 3.33E+00       | 3.22E+00       | 2.86E+00       |             |
|    | 2.82E+03       | 2.60E+02       | 0.00E+00       | 0.00E+00       | 4.54E+00       | 0.00E+00       |             |
| 11 | 5.54E+08       | 4.42E+08       | 2.51E+08       | 4.47E+07       | 3.80E+07       | 2.13E+07       |             |
|    | 1.03E+04       | 8.69E+03       | 5.34E+03       | 2.29E+03       | 1.69E+03       | 1.75E+03       |             |
| 12 | 2.01E+01       | 2.00E+01       | 2.00E+01       | 2.00E+01       | 2.00E+01       | 2.00E+01       |             |
|    | 4.61E+03       | 2.59E+02       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       |             |
| 13 | 9.35E+01       | 8.09E+01       | 6.20E+01       | 4.95E+01       | 4.32E+01       | 4.63E+01       |             |
|    | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       |             |
| 14 | 4.68E+03       | 5.99E+03       | 6.75E+03       | 7.10E+03       | 7.13E+03       | 7.14E+03       |             |
|    | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       | 0.00E+00       |             |
| 15 | 5.42E+00       | 2.00E+00       | 2.00E+00       | 2.00E+00       | 2.00E+00       | 2.00E+00       |             |
|    | 7.33E+03       | 5.01E+02       | 5.01E+02       | 5.01E+02       | 5.01E+02       | 5.01E+02       |             |
The experimental results -

\[\begin{array}{cccccc}
\text{No.} & \varepsilon\text{-CC-SHADE (1)} & \varepsilon\text{-CC-SHADE (2)} & \varepsilon\text{-CC-SHADE (4)} & \varepsilon\text{-CC-SHADE (8)} & \varepsilon\text{-CC-SHADE (10)} \\
1 & 1.16E+04 & 4.40E+03 & 1.97E+03 & 1.06E+03 & 1.01E+03 & 8.39E+02 \\
2 & 5.68E+07 & 8.43E+04 & 4.02E+02 & 2.33E+02 & 2.12E+02 & 1.91E+02 \\
3 & 1.06E+03 & 8.73E+02 & 6.00E+02 & 5.82E+02 & 5.69E+02 & 5.20E+02 \\
4 & 7.37E+05 & 7.37E+05 & 7.37E+05 & 7.37E+05 & 7.37E+05 & 7.37E+05 \\
5 & 1.69E+02 & 2.01E+02 & 2.27E+02 & 2.30E+02 & 2.42E+02 & 0.00E+00 \\
\end{array}\]

Figure 2. The ranking of \(\varepsilon\text{-CC-SHADE (m)}\) and \(\varepsilon\text{-iCC-SHADE}\) on the cLSGO 2017, pop_size is 50.

Table 4. The experimental results for \(\varepsilon\text{-CC-SHADE}\) and \(\varepsilon\text{-iCC-SHADE}\) on the cLSGO benchmark problems, population size is 75.
Figure 3. The ranking of $\varepsilon$-CC-SHADE (m) and $\varepsilon$-iCC-SHADE on the cLSGO, pop_size is 75

Table 5. The experimental results for $\varepsilon$-CC-SHADE and $\varepsilon$-iCC-SHADE on the cLSGO benchmark problems, population size is 100

| Nb | $\varepsilon$-CC-SHADE (1) | $\varepsilon$-CC-SHADE (2) | $\varepsilon$-CC-SHADE (4) | $\varepsilon$-CC-SHADE (8) | $\varepsilon$-CC-SHADE (10) | $\varepsilon$-iCC-SHADE |
|----|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|------------------------|
| 1  | 7.79E+05                   | 4.61E+05                   | 1.66E+05                   | 5.57E+04                   | 4.06E+04                   | 8.53E+04               |
|    | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00               |
| 2  | 9.64E+07                   | 8.13E+07                   | 1.51E+08                   | 1.37E+08                   | 1.83E+08                   | 1.04E+08               |
|    | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00               |
| 3  | 2.12E+04                   | 1.16E+04                   | 5.97E+03                   | 3.92E+03                   | 3.35E+03                   | 3.31E+03               |
|    | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00               |
| 4  | 1.32E+05                   | 1.51E+05                   | 1.74E+05                   | 1.78E+05                   | 1.66E+05                   | 1.34E+05               |
|    | 0.00E+00                   | 1.75E-02                   | 8.07E-02                   | 1.81E-01                   | 2.03E-01                   | 1.19E-02               |
| 5  | -1.12E+04                  | -1.05E+03                  | 7.31E+02                   | 1.73E+02                   | -3.32E+02                  | 2.47E+02               |
|    | 0.00E+00                   | 1.69E+04                   | 3.04E+04                   | 3.69E+04                   | 3.80E+04                   | 9.31E+03               |
| 6  | 1.29E+02                   | 1.26E+02                   | 8.37E+01                   | 1.02E+02                   | 1.63E+02                   | 9.82E+01               |
|    | 4.97E+05                   | 2.80E+05                   | 1.21E+05                   | 1.46E+05                   | 1.46E+06                   | 1.40E+05               |
| 7  | 1.65E+01                   | 1.46E+01                   | 1.38E+01                   | 1.25E+01                   | 1.94E+01                   | 1.23E+01               |
|    | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 4.38E+04                   | 0.00E+00               |
| 8  | 9.06E+01                   | 6.43E+01                   | 3.35E+01                   | 1.28E+01                   | 6.08E+01                   | 2.18E+01               |
|    | 1.26E+06                   | 4.31E+05                   | 6.05E+04                   | 6.72E+03                   | 7.52E+04                   | 3.99E+04               |
| 9  | -7.41E+03                  | -2.49E+03                  | -4.81E+03                  | -4.89E+03                  | -3.95E+03                  | -4.29E+03              |
|    | 5.86E+03                   | 8.90E+02                   | 2.58E+02                   | 2.70E+02                   | 1.81E+04                   | 5.19E+02               |
| 10 | 3.53E+02                   | 5.10E+00                   | 2.99E+00                   | 3.22E+00                   | 3.23E+00                   | 3.18E+00               |
|    | 1.05E+02                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00               |
| 11 | 4.79E+08                   | 3.68E+08                   | 2.96E+07                   | 3.25E+06                   | 6.23E+06                   | 2.17E+06               |
|    | 1.02E+04                   | 5.98E+03                   | 2.16E+03                   | 8.47E+02                   | 6.12E+02                   | 7.42E+02               |
| 12 | 2.00E+01                   | 2.00E+01                   | 2.00E+01                   | 2.00E+01                   | 2.00E+01                   | 2.00E+01               |
|    | 1.71E+02                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00               |
| 13 | 7.46E+01                   | 6.20E+01                   | 4.32E+01                   | 3.38E+01                   | 3.38E+01                   | 3.06E+01               |
|    | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00               |
| 14 | 6.33E+03                   | 7.09E+03                   | 7.16E+03                   | 7.03E+03                   | 6.90E+03                   | 6.88E+03               |
|    | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00                   | 0.00E+00               |
| 15 | 2.00E+00                   | 2.00E+00                   | 2.00E+00                   | 2.00E+00                   | 2.00E+00                   | 2.00E+00               |
|    | 5.01E+02                   | 5.01E+02                   | 5.01E+02                   | 5.01E+02                   | 5.01E+02                   | 5.01E+02               |
As we can see in Figures 1-4, the proposed $\varepsilon$-iCC-SHADE performs better than $\varepsilon$-CC-SHADE (m) with different number of subcomponents. Tables 6-9 prove difference in the results of estimation the performance of $\varepsilon$-iCC-SHADE and $\varepsilon$-CC-SHADE (m) using Mann–Whitney U test with normal approximation and tie correction with $p$ value is equal to 0.01. The first column contains algorithms, the second, third and fourth columns contain the numbers of benchmark problems when the first EA has better, worse and equal performance. We have compared proposed $\varepsilon$-iCC-SHADE versus $\varepsilon$-CC-SHADE (m) algorithms. Tables 6-9 show that $\varepsilon$-iCC-SHADE is statistically better than $\varepsilon$-CC-SHADE (m).

Table 6. Mann–Whitney U test, population size is 25

| $\varepsilon$-iCC-SHADE vs. | + (better) | - (worse) | $\approx$ (no. sig) |
|-----------------------------|------------|-----------|---------------------|
| $\varepsilon$-CC-SHADE (1)  | 11         | 3         | 4                   |
| $\varepsilon$-CC-SHADE (2)  | 14         | 2         | 2                   |
| $\varepsilon$-CC-SHADE (4)  | 13         | 2         | 3                   |
| $\varepsilon$-CC-SHADE (8)  | 7          | 3         | 8                   |
| $\varepsilon$-CC-SHADE (10) | 5          | 2         | 11                  |
| Total sum                  | 50         | 12        | 28                  |

Table 7. Mann–Whitney U test, population size is 50

| $\varepsilon$-iCC-SHADE vs. | + (better) | - (worse) | $\approx$ (no. sig) |
|-----------------------------|------------|-----------|---------------------|
| $\varepsilon$-CC-SHADE (1)  | 13         | 4         | 1                   |
| $\varepsilon$-CC-SHADE (2)  | 14         | 2         | 2                   |
| $\varepsilon$-CC-SHADE (4)  | 13         | 2         | 3                   |
| $\varepsilon$-CC-SHADE (8)  | 5          | 2         | 11                  |
| $\varepsilon$-CC-SHADE (10) | 6          | 2         | 10                  |
| Total sum                  | 51         | 12        | 27                  |
5. Conclusions
In this study, we have proposed a novel ε-iCC-SHADE algorithm for constrained large-scale global BB optimization. We have investigated the performance of ε-iCC-SHADE with different population sizes. The experimental results have shown that the proposed iCC approach demonstrates better results than CC with fixed number of subcomponents. At the same time, iCC still has the great potential for improvement, thus in our further works we will try to modify it for adaptive dynamic sizing of groups.

Acknowledgments
This research is supported by the Ministry of Education and Science of Russian Federation within State Assignment № 2.1676.2017/ПЧ.

References
[1] Xiao M, Zhang J, Cai K, Cao X and Tang K 2011 Cooperative co-evolution with weighted random grouping for large-scale crossing waypoints locating in air route network 23rd IEEE International Conference on Tools with Artificial Intelligence pp 215–222
[2] Mahdavi S, Shiri M E and Rahnamayan S 2014 Metaheuristics in large-scale global continues optimization: a survey Information Sciences 295 407–28
[3] Worasuchep C 2008 Solving constrained engineering optimization problems by the constrained PSO-DD 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology pp 5–8
[4] Eskandar H, Sadollah A, Bahreininejad A and Hamdi M 2012 Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems Computers & Structures 151–66
[5] LaTorre A, Muelas S and Peña J-M 2010 A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test Soft Computing 15(11) 2187–99
[6] Wang H, Wu Z, Rahnamayan S, Liu Y and Ventresca M 2011 Enhancing particle swarm optimization using generalized opposition-based learning Information Sciences 181(20) 4699–4714
[7] Omidvar M N, Li X, Mei Y and Yao X 2014 Cooperative co-evolution with differential grouping for large scale optimization IEEE Transactions on Evolutionary Computation 18(3) 378–393
[8] Yang Z, Tang K and Yao X 2008 Large scale evolutionary optimization using cooperative
coevolution Information Sciences 178(15) 2985–99

[9] Marimont R B and Shapiro M B 1979 Nearest neighbour searches and the curse of dimensionality IMA Journal of Applied Mathematics 24(1) 59–70

[10] Tanabe R and Fukunaga A 2013 Success-history based parameter adaptation for differential evolution 2013 IEEE Congress on Evolutionary Computation pp 71–78

[11] Yeniay O 2005 Penalty function methods for constrained optimization with genetic algorithms Math. Comput. Appl. 10 45–56

[12] Homaifar A, Qi C X and Lai S H 1994 Constrained optimization via genetic algorithms Simulation 62(4) 242–53

[13] Richardson J T, Palmer M R, Liepins G and Hilliard M 1989 Some guidelines for genetic algorithms with penalty functions Proceedings of the Third International Conference on Genetic Algorithms pp 191–197

[14] Adeli H and Cheng N 1994 Augmented lagrangian genetic algorithm for structural optimization Journal of Aerospace Engineering 7(1) 104–18

[15] Coello C A 2002 Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art Computer Methods in Applied Mechanics and Engineering 191 1245–87

[16] Takahama T, Sakai S and Iwane N 2006 Solving nonlinear constrained optimization problems by the ε constrained differential evolution 2006 IEEE International Conference on Systems, Man and Cybernetics 3 2322–7

[17] Storn R 1996 On the usage of differential evolution for function optimization Proceedings of North American Fuzzy Information Processing 519–23

[18] Storn R and Price K 1997 Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces Journal of Global Optimization 11 341–59

[19] Vesterstrom J and Thomsen R 2004 A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems Proceedings of the 2004 Congress on Evolutionary Computation 2 1980–7

[20] Alam M N, Das B and Pant V 2015 A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination Electric Power Systems Research 128 39–52

[21] Potter M A and Jong K A 1994 A cooperative coevolutionary approach to function optimization Proceedings of the Third Conference on Parallel Problem Solving From Nature pp 249–57

[22] Potter M A and Jong K A 2000 Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents Evolutionary Computation 8(1) 1–29

[23] Brest J 2009 Constrained real-parameter optimization with ε-self-adaptive differential evolution Studies in Computational Intelligence pp 73–93

[24] Wu G, Mallipeddi R and Suganthan P 2016 Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization