Single Objective Real-Parameter Optimization: Algorithm jSO

Janez Brest, Mirjam Sepesy Maučec, Borko Bošković
Faculty of Electrical Engineering and Computer Science
University of Maribor
Smetanova ul. 17, 2000 Maribor, Slovenia
Email: janez.brest@um.si, mirjam.sepesy@um.si, borko.boskovic@um.si

***Abstract*—Solving single objective real-parameter optimization problems, also known as a bound-constrained optimization, is still a challenging task. We can find such problems in engineering optimization, scientific applications, and in other real-world problems. Usually, these problems are very complex and computationally expensive. A new algorithm, called jSO, is presented in this paper. The algorithm is an improved variant of the iL-SHADE algorithm, mainly with a new weighted version of mutation strategy. The experiments were performed on CEC 2017 benchmark functions, which are different from previous competition benchmark functions. A comparison of the proposed jSO algorithm and the L-SHADE algorithm is presented first. From the obtained results we can conclude that jSO performs better in comparison with the L-SHADE algorithm. Next, a comparison of jSO and iL-SHADE is also performed, and jSO obtained better or competitive results. Using the CEC 2017 evaluation method, jSO obtained the best final score among these three algorithms.**

I. Introduction

Single-objective real parameter optimization plays important role in various researches, where increasing efforts are focused on solving complex optimization problems. Several algorithms have been investigated for solving real-parameter problems, and among them, the population based algorithms seem very useful since they are quite simple to implement in any programming language. Recently, we can find two groups of population based algorithms: evolutionary algorithms (EAs), and the group of algorithms inspired by nature, such as bees, ants, or immune systems. In this paper, we are dealing with evolutionary algorithms.

One of the biggest drawbacks of evolutionary algorithms is the loss of diversity in the population. As consequence, an algorithm might show a premature convergence into local optima. A good algorithm needs to apply a population diversity at sufficient high level in early stages of the evolutionary process, while in later stages the algorithm needs to reduce the population diversity in order to increase convergence rate.

We can define the single objective global optimization, called also a bound-constrained optimization, as follows. For an objective function $f(\vec{x})$, an algorithm needs to find variables of vector $\vec{x}$, which minimizes/maximizes $f(\vec{x})$. A number of variables, D, in vector $\vec{x} = \{x_1, x_2, ..., x_D\}$ denotes the dimensionality of the problem. A search space is determined by domains of the variables, i.e. it is defined by their lower and upper bounds, $x_{j,low}$ and $x_{j,upp}$ for $j = 1, 2, ..., D$. In black-box optimization, the task is to solve a global optimization problem without explicit knowledge of the form or structure of the objective function, i.e., f is a black box [1]. Such problems arise in engineering optimization, scientific applications, and in other real-world optimization problems.

Single objective optimization can be included in many other types of optimization, like constrained optimization, multi-modal optimization, multi-objective optimization, etc. However, each type of optimization has some challenges. In a single objective optimization, function f might have several optima and an algorithm is required to find the global one. If an algorithm traps its population into a local optimum, the final result of optimization may be poor. Next challenge is regarding the dimensionality of an optimization problem, since the search space is becoming very huge when the dimension of problem is increased.

The Differential Evolution (DE) [2] algorithm is a stochastic population based EA used for numerical optimization. The DE algorithm is simple for implementation. In recent decades, it has shown robustness, efficiency, and it was very competitive when it was applied to solve real-world optimization problems [3], [4]. DE is suitable especially for optimization over continuous spaces, and, moreover, it might be used also in discrete domains.

The original DE was proposed in 1995, and till now many researches used the original algorithm or improved variants in very different applications. A huge number of improvements were proposed recently [5].

In this paper, we present a new version of DE algorithm (jSO). It is similar to our previously published algorithm, iL-SHADE [7], which was on the third/fourth place on CEC 2016 competition on single objective real-parameter optimization. The iL-SHADE algorithm is an improved version of the L-SHADE algorithm proposed by Tanabe and Fukunaga [6].

A quick look to a history of real-parameter single objective optimization competitions and/or special sessions, tell us that they were organized at CEC in 2005, 2013, 2014, 2015 and 2016. Closely related sessions/competitions such as large-scale global optimization, etc., have been also organized at CEC, and other conferences. Note, that benchmark functions of CEC 2016 and CEC 2017 are different, while the number of benchmark functions is equal in both competitions. Therefore,

978-1-5090-4601-0/17/$31.00

we may not directly compare obtained results for this and previous CEC competitions.

The main contributions of this paper are: (1) A new version of algorithm (jSO) with a weighted variant of mutation strategy is presented, (2) Experiments for CEC 2017 competition are conducted, (3) A detailed comparison of jSO versus L-SHADE is given, and (4) A summary comparison of jSO, iL-SHADE, and L-SHADE is presented.

The structure of the paper is as follows. Section II gives background for this work, where an overview of DE, and a description of the L-SHADE and iL-SHADE algorithms are given. Section III presents our new variant of the algorithm, called jSO, which is used in these experiments. In Section IV experimental results of the jSO algorithm on benchmark functions are presented, and comparison with L-SHADE and iL-SHADE is given. Section V concludes the paper with some final remarks.

II. Background

In this section, we give some backgrounds of DE and related algorithms. Differential evolution (DE) is population based algorithm that belongs to the group of evolutionary algorithms. We can find out that DE shows excellent performance and it is applied in many applications and researches [8], [9], [10], [11], [12], [13]. DE has three control parameters that are required to be set by a user before the evolutionary process starts: F is scaling factor, CR is crossover control parameter, and NP is population size. They are fixed in the original DE algorithm. Although, a user needs to set three parameters, it might be a time consuming task to find *good* values for them. To overcome this issue, several adaptive and self-adaptive DE variants have been proposed. Control parameters F and CR are self-adaptive in SADE [14], jDE [15], JADE [16], etc., and, therefore, the user does not need to set and tune these two control parameters, while the user needs to set NP. Usually, NP remains fixed during the optimization process. Recently, there have been several attempts to adjust this control parameter during the evolutionary process [6], [17].

SHADE [1] is DE-based algorithm which uses success-history based parameter adaptation for CR and F. The Linear Population Size Reduction (LPSR), which continually decreases the population size according to a linear function, was applied into SHADE and the obtained algorithm is called L-SHADE [6]. The SHADE and L-SHADE algorithms have inspired many researches. Improved versions of L-SHADE have been placed on top ranks at CEC 2015 competition: SPS-L-SHADE-EIG [18], DEsPA [19], LSHADE-ND [20], also at CEC 2016: LSHADE_EpSin [21] (an ensemble sinusoidal parameter adaptation is incorporated with L-SHADE), LSHADE44 [22] (L-SHADE with competing strategies), and iL-SHADE [7].

A. Differential Evolution

The DE algorithm [2] is a population based algorithm. Its population P is combined of NP vectors:

$$\mathbf{P}_g = (\vec{x}_{1,g}, \ldots, \vec{x}_{i,g}, \ldots, \vec{x}_{NP,g}),\ i = 1, 2, ..., NP,$$

where g denotes a generation index, $g = 1, 2, \ldots, G_{MAX}$. Each vector consists of D variables:

$$\vec{x}_{i,g} = (x_{i,1,g}, x_{i,2,g}, ..., x_{i,D,g}).$$

A randomly initialized population of NP vectors is evolved throughout G_{MAX} generations guiding the vectors in a search space toward to a global optimum. At the end of the evolutionary process, i.e. after G_{MAX} generations, DE stops and returns the best fitted vector as the final solution.

During each generation DE employs three operations for each individual, namely mutation, crossover and selection.

Mutation: A mutant vector $\vec{v}_{i,g}$ is created using one of the mutation strategies. The 'DE/rand/1' mutation strategy has been introduced in the original DE algorithm [2] and it is one of the most used mutation strategy in DE. This strategy randomly selects two vectors and their difference multiplied by scale factor f is added to third randomly selected vector. We can express this strategy as follows:

$$\vec{v}_{i,g} = \vec{x}_{r_1,g} + F \cdot (\vec{x}_{r_2,g} - \vec{x}_{r_3,g}),$$

where r_1, r_2, and r_3 are indexes within a set of $\{1, NP\}$. The indexes are randomly chosen in such a way that they are pairwise different and also different from index i:

$$r_1 \neq r_2 \neq r_3 \neq i.$$

The other DE mutation strategies are:

- "DE/best/1": $\vec{v}_{i,g} = \vec{x}_{best,g} + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g})$,
- "DE/current to best/1": $\vec{v}_{i,g} = \vec{x}_{i,g} + F(\vec{x}_{best,g} - \vec{x}_{i,g}) + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g})$,
- "DE/best/2": $\vec{v}_{i,g} = \vec{x}_{best,g} + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) + F(\vec{x}_{r_3,g} - \vec{x}_{r_4,g})$,
- "DE/rand/2": $\vec{v}_{i,g} = \vec{x}_{r_1,g} + F(\vec{x}_{r_2,g} - \vec{x}_{r_3,g}) + F(\vec{x}_{r_4,g} - \vec{x}_{r_5,g})$,

where the indexes r_1–r_5 represent the random and mutually different integers generated within the range $\{1, NP\}$ and also different from index i. $\vec{x}_{best}$ is the best vector in a current generation. Each strategy has a different ability to maintain the population diversity which might increase/decrease convergence rate during evolutionary process. A reader is referred to the DE surveys [3], [4], [5].

Crossover: A mutant vector $\vec{v}_{i,g}$ generated by one of the mutation strategies is used in next operation, called crossover. Binomial crossover is widely used in DE, the other crossover is exponential [2], [3]. The former creates a trial vector $\vec{u}_{i,g}$ as follows:

$$u_{i,j,g} = \begin{cases} v_{i,j,g}, & \text{if } rand(0,1) \leq CR \text{ or } j = j_{rand}, \\ x_{i,j,g}, & \text{otherwise,} \end{cases}$$

for $i = 1, 2, ..., NP$ and $j = 1, 2, ..., D$. $CR \in [0, 1]$ is crossover parameter and presents the probability of creating components for a trial vector from a mutant vector. If a component was not selected from the mutant vector, then it is taken from the parent vector $\vec{x}_{i,g}$. Randomly chosen index

$g \leftarrow 1$; Archive $\mathbf{A} \leftarrow \emptyset$
Initialize population $\mathbf{P}_g = (\vec{x}_{i,g}, \ldots, \vec{x}_{NP,g})$ randomly
Set all values in M_F to 0.5; $\Longleftarrow$
Set all values in M_{CR} to 0.8
$k \leftarrow 1$ *// index counter*
while the termination criatera are not meet **do**
  $S_{CR} \leftarrow \emptyset, S_F \leftarrow \emptyset$
  for $i = 1$ **to** NP **do**
    $r_i \leftarrow$ select from $[1, H]$ randomly
    if $r_i = H$ **then**
      $M_{F,r_i} \leftarrow 0.9$
      $M_{CR,r_i} \leftarrow 0.9$
    end if
    if $M_{CR,r_i} < 0$ **then**
      $CR_{i,g} \leftarrow 0$
    else
      $CR_{i,g} \leftarrow \mathcal{N}_i(M_{CR,r_i}, 0.1)$
    end if
    if $g < 0.25 G_{MAX}$ **then**
      $CR_{i,g} \leftarrow \max(CR_{i,g}, 0.7)$ $\Longleftarrow$
    else if $g < 0.5 G_{MAX}$ **then**
      $CR_{i,g} \leftarrow \max(CR_{i,g}, 0.6$ $\Longleftarrow$
    end if
    $F_{i,g} \leftarrow \mathcal{C}_i(M_{F,r_i}, 0.1)$
    if $g < 0.6 G_{MAX}$ **and** $F_{i,g} > 0.7$ **then** $\Longleftarrow$
      $F_{i,g} \leftarrow 0.7$ $\Longleftarrow$
    end if
    $\vec{u}_{i,g} \leftarrow$ current-to-pBest-w/1/bin using Eq. (3) $\Longleftarrow$
  end for
  for $i = 1$ **to** NP **do**
    if $f(\vec{u}_{i,g}) \leq f(\vec{x}_{i,g})$ **then**
      $\vec{x}_{i,g+1} \leftarrow \vec{u}_{i,g}$
    else
      $\vec{x}_{i,g+1} \leftarrow \vec{x}_{i,g}$
    end if
    if $f(\vec{u}_{i,g}) < f(\vec{x}_{i,g})$ **then**
      $\vec{x}_{i,g} \rightarrow \mathbf{A}, CR_{i,g} \rightarrow S_{CR}, F_{i,g} \rightarrow S_F$
    end if
    Shrink $\mathbf{A}$, if necessary
    Update M_{CR} and M_F
    Apply LPSR strategy *// linear population size reduction*
    Update p using Eq. (1)
  end for
  $g \leftarrow g + 1$
end while

Algorithm 1: jSO algorithm

$j_{rand} \in \{1, 2, ..., NP\}$ is responsible for the trial vector to contain at least one component from the mutant vector.

If some variables from the trial vector are out of bounds, a repeat mechanism is applied.

Selection: After crossover operation, the trial vector is evaluated – an objective function $f(\vec{u}_{i,g})$ is calculated. Then selection operation compares two vectors, population vector $\vec{x}_{i,g}$ and its corresponding trial vector $\vec{u}_{i,g}$, according to their objective function values. The better vector will become a member of the next generation. The selection operation for a minimization optimization problem is defined as follow:

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

This selection operation is greedy and it is known for DE, but rarely applied in other EAs.

TABLE I
THE RESULTS OF THE jSO ALGORITHM FOR $D = 10$.

	Best	Worst	Median	Mean	Std.
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_5	0.0000E+00	2.9849E+00	1.9899E+00	1.7558E+00	7.6004E–01
f_6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_7	1.0746E+01	1.3537E+01	1.1750E+01	1.1792E+01	6.0675E–01
f_8	0.0000E+00	2.9849E+00	1.9899E+00	1.9509E+00	7.4352E–01
f_9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_{10}	1.8736E–01	2.4416E+02	1.0307E+01	3.5897E+01	5.5477E+01
f_{11}	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_{12}	0.0000E+00	1.2015E+02	4.1629E–01	2.6621E+00	1.6782E+01
f_{13}	0.0000E+00	5.9511E+00	4.8371E+00	2.9644E+00	2.3534E+00
f_{14}	0.0000E+00	9.9496E–01	0.0000E+00	5.8527E–02	2.3644E–01
f_{15}	4.8541E–05	5.0000E–01	1.7917E–01	2.2084E–01	2.0044E–01
f_{16}	2.4800E–02	1.1402E+00	5.1923E–01	5.6884E–01	2.6440E–01
f_{17}	1.9759E–02	1.4526E+00	4.0314E–01	5.0227E–01	3.4807E–01
f_{18}	2.6385E–06	5.0000E–01	3.7898E–01	3.0800E–01	1.9514E–01
f_{19}	0.0000E+00	3.9161E–02	0.0000E+00	1.0703E–02	1.2543E–02
f_{20}	0.0000E+00	6.2435E–01	3.1217E–01	3.4278E–01	1.2879E–01
f_{21}	1.0000E+02	2.0437E+02	1.0000E+02	1.3238E+02	4.8365E+01
f_{22}	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	0.0000E+00
f_{23}	3.0000E+02	3.0587E+02	3.0000E+02	3.0121E+02	1.5897E+00
f_{24}	1.0000E+02	3.3133E+02	3.2859E+02	2.9660E+02	7.9323E+01
f_{25}	3.9774E+02	4.4338E+02	3.9801E+02	4.0596E+02	1.7478E+01
f_{26}	3.0000E+02	3.0000E+02	3.0000E+02	3.0000E+02	0.0000E+00
f_{27}	3.8901E+02	3.8952E+02	3.8952E+02	3.8939E+02	2.2556E–01
f_{28}	3.0000E+02	6.1182E+02	3.0000E+02	3.3908E+02	9.6547E+01
f_{29}	2.2903E+02	2.4162E+02	2.3318E+02	2.3420E+02	2.9559E+00
f_{30}	3.9450E+02	3.9469E+02	3.9450E+02	3.9452E+02	4.4991E–02

B. L-SHADE and iL-SHADE Algorithms

In this section L-SHADE [6] and iL-SHADE are presented briefly.

The L-SHADE [6] algorithm extends the Success-History based Adaptive DE (SHADE) [1] algorithm with the Linear Population Size Reduction mechanism (LPSR), which after each generation decreases the population size according to a linear function. L-SHADE has been the best ranked DE-based algorithm on CEC 2014 Competition on Real-Parameter Single Objective Optimization. If we look to a history on how an algorithm was inspired by its predecessor, we can see a close relation of the following algorithms: JADE [16], SHADE [1], L-SHADE [6]. Later, several other JADE/(L)SHADE-based algorithms have been proposed.

L-SHADE applies current-to-pBest/1/bin strategy to generate a trial vector, weighted Lehmer mean, $mean_{WL}$, in order to make F, CR control parameters self-adaptive. A historical

TABLE II
THE RESULTS OF THE jSO ALGORITHM FOR $D = 30$.

	Best	Worst	Median	Mean	Std.
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_4	5.8562E+01	6.4117E+01	5.8562E+01	5.8670E+01	7.7797E–01
f_5	3.9798E+00	1.3249E+01	8.0168E+00	8.5568E+00	2.0980E+00
f_6	0.0000E+00	1.3687E–07	0.0000E+00	6.0385E–09	2.7122E–08
f_7	3.6115E+01	4.3093E+01	3.9064E+01	3.8927E+01	1.4594E+00
f_8	4.9748E+00	1.2970E+01	8.9557E+00	9.0918E+00	1.8399E+00
f_9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_{10}	1.0391E+03	2.0415E+03	1.4931E+03	1.5277E+03	2.7716E+02
f_{11}	0.0000E+00	9.1925E+00	1.9899E+00	3.0375E+00	2.6464E+00
f_{12}	3.8638E+00	4.5480E+02	1.3889E+02	1.7038E+02	1.0194E+02
f_{13}	9.9304E–01	2.2459E+01	1.5868E+01	1.4840E+01	4.8312E+00
f_{14}	2.0061E+01	2.4621E+01	2.1245E+01	2.1834E+01	1.2458E+00
f_{15}	3.2488E–01	2.7370E+00	7.7939E–01	1.0879E+00	6.9133E–01
f_{16}	4.0180E+00	2.8350E+02	2.5930E+01	7.8923E+01	8.4769E+01
f_{17}	1.1626E+01	4.7555E+01	3.4588E+01	3.2925E+01	8.0767E+00
f_{18}	4.8587E–01	2.1754E+01	2.0650E+01	2.0411E+01	2.8726E+00
f_{19}	2.0677E+00	1.0861E+01	4.0853E+00	4.5031E+00	1.7323E+00
f_{20}	1.3673E+01	4.1526E+01	2.9302E+01	2.9368E+01	5.8548E+00
f_{21}	2.0507E+02	2.1519E+02	2.0945E+02	2.0929E+02	1.9554E+00
f_{22}	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	0.0000E+00
f_{23}	3.4228E+02	3.6137E+02	3.5050E+02	3.5075E+02	3.2992E+00
f_{24}	4.2196E+02	4.3200E+02	4.2668E+02	4.2646E+02	2.4662E+00
f_{25}	3.8669E+02	3.8672E+02	3.8670E+02	3.8670E+02	7.6811E–03
f_{26}	8.3924E+02	1.0341E+03	9.3040E+02	9.2021E+02	4.2954E+01
f_{27}	4.8203E+02	5.1195E+02	4.9468E+02	4.9739E+02	7.0017E+00
f_{28}	3.0000E+02	4.1398E+02	3.0000E+02	3.0873E+02	3.0250E+01
f_{29}	3.5669E+02	4.5264E+02	4.3313E+02	4.3367E+02	1.3641E+01
f_{30}	1.9550E+03	2.0751E+03	1.9705E+03	1.9712E+03	1.8961E+01

TABLE III
THE RESULTS OF THE jSO ALGORITHM FOR $D = 50$.

	Best	Worst	Median	Mean	Std.
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_4	1.3178E–04	1.4231E+02	2.8513E+01	5.6213E+01	4.8763E+01
f_5	8.9606E+00	2.3886E+01	1.6197E+01	1.6405E+01	3.4620E+00
f_6	0.0000E+00	1.7090E–05	3.1068E–07	1.0933E–06	2.6259E–06
f_7	5.7519E+01	7.4153E+01	6.6640E+01	6.6497E+01	3.4728E+00
f_8	9.9506E+00	2.4053E+01	1.6967E+01	1.6962E+01	3.1354E+00
f_9	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_{10}	2.4048E+03	3.7919E+03	3.2324E+03	3.1398E+03	3.6716E+02
f_{11}	2.1250E+01	3.2211E+01	2.8521E+01	2.7939E+01	3.3284E+00
f_{12}	7.5193E+02	3.0634E+03	1.6533E+03	1.6806E+03	5.2293E+02
f_{13}	4.5847E+00	1.1090E+02	3.7258E+01	3.0599E+01	2.1226E+01
f_{14}	2.0302E+01	2.9662E+01	2.4351E+01	2.4964E+01	1.8734E+00
f_{15}	1.8879E+01	2.8866E+01	2.3420E+01	2.3864E+01	2.4882E+00
f_{16}	1.3473E+02	6.8420E+02	4.7725E+02	4.5052E+02	1.3775E+02
f_{17}	8.5431E+01	4.7697E+02	2.5993E+02	2.8287E+02	8.6142E+01
f_{18}	2.0531E+01	2.9161E+01	2.3874E+01	2.4283E+01	2.0174E+00
f_{19}	1.0385E+01	1.9934E+01	1.3858E+01	1.4139E+01	2.2622E+00
f_{20}	5.1191E+01	3.1793E+02	1.1326E+02	1.4010E+02	7.7375E+01
f_{21}	2.1174E+02	2.2919E+02	2.1918E+02	2.1920E+02	3.7656E+00
f_{22}	1.0000E+02	4.1136E+03	1.0000E+02	1.4872E+03	1.7531E+03
f_{23}	4.1011E+02	4.4029E+02	4.2975E+02	4.3008E+02	6.2364E+00
f_{24}	4.9957E+02	5.1700E+02	5.0728E+02	5.0745E+02	4.1273E+00
f_{25}	4.7735E+02	4.9186E+02	4.8024E+02	4.8088E+02	2.7999E+00
f_{26}	9.6631E+02	1.2556E+03	1.1329E+03	1.1288E+03	5.6167E+01
f_{27}	4.7698E+02	5.3866E+02	5.1250E+02	5.1127E+02	1.1077E+01
f_{28}	4.5885E+02	5.0769E+02	4.5885E+02	4.5981E+02	6.8398E+00
f_{29}	3.2871E+02	3.8576E+02	3.6363E+02	3.6294E+02	1.3157E+01
f_{30}	5.7941E+05	7.2365E+05	5.9048E+05	6.0105E+05	2.9859E+04

memory has H entries ($|M_F| = |M_{CR}| = H$). All three control parameters remain fixed during the evolutionary process in the original DE algorithm [2], while L-SHADE self-adapts scale factor and crossover parameters (F and CR), and shrinks population size (NP) during the evolutionary process.

The iL-SHADE [7] algorithm is an extended version of the L-SHADE algorithm.

Let us summarize the main features that were proposed in the iL-SHADE [7] algorithm:

1) Higher values for CR control parameter are being *propagated* during the optimization process in the following ways:
   - all historical memory values in M_{CR} are initialized to 0.8 (in L-SHADE it is set to 0.5),
   - one historical memory entry (the last one in our case, see Lines 10–13 in Algorithm 1) is set as follows: $M_{CR,H} = 0.9$ and $M_{F,H} = 0.9$, and these values remain unchanged during the evolutionary process. In such a way not only a higher value for CR_i is used more often, but also a higher F_i in a pair. A trial vector has a higher probability of creating components from a mutant vector, if CR value is higher.
2) Memory update mechanism stores historical memory values M_{CR} and M_F of current generation, say g, and uses them weighted equally with the weighted Lehmer means to calculate the historical memory values for the next generation, $g + 1$.
3) Very high values of F and low values of CR are not allowed in an early stage of evolutionary process.
4) After each generation g, p value for current-to-pBest/1 mutation in the next generation, $g + 1$, is computed as follows:

$$p = (\frac{p_{max} - p_{min}}{max\_nfes}) \cdot nfes + p_{min}, \qquad (1)$$

where $nfes$ is the current number of objective function evaluations, and max_nfes is the maximum number of objective function evaluations.

TABLE IV
THE RESULTS OF THE jSO ALGORITHM FOR $D = 100$.

	Best	Worst	Median	Mean	Std.
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	1.2181E+02	4.6953E–07	8.9403E+00	2.4202E+01
f_3	6.4494E–08	1.5008E–05	1.4482E–06	2.3912E–06	2.7250E–06
f_4	8.4524E+01	2.2075E+02	1.9538E+02	1.8963E+02	2.8923E+01
f_5	3.0099E+01	6.0038E+01	4.4049E+01	4.3908E+01	5.6066E+00
f_6	7.6730E–06	3.5617E–03	3.5717E–05	2.0244E–04	6.1988E–04
f_7	1.2880E+02	1.5961E+02	1.4425E+02	1.4490E+02	6.7030E+00
f_8	2.7260E+01	5.4642E+01	4.2250E+01	4.2152E+01	5.5223E+00
f_9	0.0000E+00	5.4385E–01	0.0000E+00	4.5904E–02	1.1493E–01
f_{10}	7.5410E+03	1.1012E+04	9.7507E+03	9.7044E+03	6.8161E+02
f_{11}	4.2724E+01	3.0408E+02	1.0424E+02	1.1317E+02	4.3204E+01
f_{12}	6.1741E+03	3.9130E+04	1.7035E+04	1.8430E+04	8.3505E+03
f_{13}	5.3265E+01	2.5094E+02	1.3958E+02	1.4489E+02	3.8005E+01
f_{14}	4.4073E+01	9.4700E+01	6.3850E+01	6.4341E+01	1.0895E+01
f_{15}	8.6693E+01	2.6150E+02	1.6468E+02	1.6212E+02	3.8054E+01
f_{16}	1.0308E+03	2.4115E+03	1.8817E+03	1.8586E+03	3.4855E+02
f_{17}	6.7483E+02	1.8579E+03	1.2732E+03	1.2780E+03	2.3843E+02
f_{18}	1.0143E+02	2.7887E+02	1.5693E+02	1.6728E+02	3.6499E+01
f_{19}	6.2057E+01	1.5110E+02	1.0635E+02	1.0487E+02	2.0079E+01
f_{20}	6.3020E+02	1.7713E+03	1.3810E+03	1.3760E+03	2.4281E+02
f_{21}	2.4580E+02	2.7964E+02	2.6356E+02	2.6380E+02	6.4286E+00
f_{22}	1.0000E+02	1.1922E+04	1.0662E+04	1.0210E+04	2.1824E+03
f_{23}	5.4323E+02	5.9377E+02	5.7119E+02	5.7115E+02	1.0702E+01
f_{24}	8.8256E+02	9.2249E+02	9.0261E+02	9.0218E+02	7.8932E+00
f_{25}	6.3863E+02	7.7372E+02	7.6163E+02	7.3609E+02	3.5302E+01
f_{26}	3.1188E+03	3.4201E+03	3.2811E+03	3.2673E+03	8.0192E+01
f_{27}	5.3852E+02	6.3018E+02	5.8646E+02	5.8547E+02	2.1672E+01
f_{28}	4.7819E+02	5.7686E+02	5.2367E+02	5.2682E+02	2.7305E+01
f_{29}	8.8939E+02	1.6464E+03	1.2379E+03	1.2566E+03	1.9133E+02
f_{30}	2.1147E+03	2.6477E+03	2.3171E+03	2.3255E+03	1.1878E+02

TABLE V
COMPARISON OF RESULTS OBTAINED BY jSO AND L-SHADE FOR $D = 10$. WILCOXON RANK-SUM TEST ($\alpha = 0.05$).

	jSO		L-SHADE
f_1	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_2	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_3	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_4	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_5	1.7558E+00 ± 7.5255E–01	$+$	2.5771E+00 ± 9.4834E–01
f_6	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_7	1.1792E+01 ± 6.0077E–01	$+$	1.2112E+01 ± 6.7912E–01
f_8	1.9509E+00 ± 7.3619E–01	$+$	2.3820E+00 ± 8.6048E–01
f_9	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_{10}	3.5897E+01 ± 5.4930E+01	$\approx$	3.8612E+01 ± 5.0024E+01
f_{11}	0.0000E+00 ± 0.0000E+00	$\approx$	2.0437E–02 ± 1.0237E–01
f_{12}	2.6621E+00 ± 1.6617E+01	$\approx$	3.3316E+01 ± 5.3716E+01
f_{13}	2.9644E+00 ± 2.3302E+00	$+$	3.8807E+00 ± 2.3273E+00
f_{14}	5.8527E–02 ± 2.3411E–01	$+$	7.7252E–01 ± 9.1909E–01
f_{15}	2.2084E–01 ± 1.9847E–01	$-$	1.4988E–01 ± 1.8764E–01
f_{16}	5.6884E–01 ± 2.6179E–01	$-$	3.8157E–01 ± 1.7650E–01
f_{17}	5.0227E–01 ± 3.4464E–01	$-$	9.7528E–02 ± 1.2790E–01
f_{18}	3.0800E–01 ± 1.9322E–01	$-$	2.1494E–01 ± 1.9689E–01
f_{19}	1.0703E–02 ± 1.2420E–02	$\approx$	7.2615E–03 ± 9.6949E–03
f_{20}	3.4278E–01 ± 1.2752E–01	$-$	1.2242E–02 ± 6.0595E–02
f_{21}	1.3238E+02 ± 4.7889E+01	$+$	1.4933E+02 ± 5.1474E+01
f_{22}	1.0000E+02 ± 1.9886E–13	$\approx$	9.8054E+01 ± 1.3867E+01
f_{23}	3.0121E+02 ± 1.5740E+00	$+$	3.0336E+02 ± 1.4164E+00
f_{24}	2.9660E+02 ± 7.8541E+01	$+$	2.9930E+02 ± 7.9484E+01
f_{25}	4.0596E+02 ± 1.7306E+01	$\approx$	4.0954E+02 ± 1.9770E+01
f_{26}	3.0000E+02 ± 0.0000E+00	$\approx$	3.0000E+02 ± 2.0137E–13
f_{27}	3.8939E+02 ± 2.2333E–01	$\approx$	3.8945E+02 ± 1.7635E–01
f_{28}	3.3908E+02 ± 9.5596E+01	$\approx$	3.2336E+02 ± 8.0154E+01
f_{29}	2.3420E+02 ± 2.9268E+00	$\approx$	2.3452E+02 ± 2.4230E+00
f_{30}	3.9452E+02 ± 4.4548E–02	$\approx$	4.0556E+02 ± 2.0754E+01

In iL-SHADE, the current-to-pBest/1/bin strategy, external archive, linear population size reduction, etc., have been kept unchanged comparing to L-SHADE.

III. JSO – THE NEW VARIANT OF IL-SHADE

In this section, a new variant for solving single-objective real parameter optimization is presented. We name it jSO algorithm. The pseudo-code of the jSO algorithm is given in Algorithm 1. It is extended version of the iL-SHADE [7] algorithm. In Algorithm 1 we marked lines that are new or changed in jSO with respect to the iL-SHADE algorithm with $\Longleftarrow$.

L-SHADE and iL-SHADE apply DE/current-to-pBest/1 mutation strategy to generate a trial vector:

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F(\vec{x}_{pBest,g} - \vec{x}_{i,g}) + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}), \quad (2)$$

while jSO uses a new weighted version of mutation strategy, called DE/current-to-pBest-w/1, as follows:

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F_w(\vec{x}_{pBest,g} - \vec{x}_{i,g}) + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}), \quad (3)$$

where F_w is calculated:

$$F_w = \begin{cases} 0.7 * F, & nfes < 0.2 max\_nfes, \\ 0.8 * F, & nfes < 0.4 max\_nfes, \\ 1.2 * F, & \text{otherwise.} \end{cases} \quad (4)$$

The aim of the presented weighted mutation strategy is to apply a smaller factor F_w to multiply difference of vectors in which $\vec{x}_{pBest,g}$ appears at early stages of the evolutionary process, while in later stages higher factor F_w is used. With factor F_w the vector $\vec{x}_{pBest,g}$ might have a lower and/or higher influence. The values in Eq. (4) and values in Lines 19–27 in Algorithm 1 are set based on some additional experiments, but without fine tuning of these values (parameters).

One can notice that changes and extensions from the iL-SHADE to jSO are minors — the main features of iL-SHADE remain unchanged. And also, the differences between iL-SHADE and L-SHADE are not so big [7]. Therefore, we will mostly focus ourself to compare the performance of jSO

TABLE VI
COMPARISON OF RESULTS OBTAINED BY jSO AND L-SHADE FOR $D = 30$. WILCOXON RANK-SUM TEST ($\alpha = 0.05$).

	jSO		L-SHADE
f_1	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_2	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_3	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_4	5.8670E+01 ± 7.7031E–01	≈	5.8562E+01 ± 4.6102E–14
f_5	8.5568E+00 ± 2.0773E+00	−	6.4115E+00 ± 1.4348E+00
f_6	6.0385E–09 ± 2.6855E–08	≈	2.6769E–08 ± 1.5290E–07
f_7	3.8927E+01 ± 1.4450E+00	−	3.7258E+01 ± 1.2796E+00
f_8	9.0918E+00 ± 1.8218E+00	−	7.1456E+00 ± 1.5779E+00
f_9	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_{10}	1.5277E+03 ± 2.7443E+02	≈	1.5002E+03 ± 1.7337E+02
f_{11}	3.0375E+00 ± 2.6203E+00	+	1.9481E+01 ± 2.4305E+01
f_{12}	1.7038E+02 ± 1.0093E+02	+	1.0099E+03 ± 3.8554E+02
f_{13}	1.4840E+01 ± 4.7836E+00	+	1.6063E+01 ± 4.9664E+00
f_{14}	2.1834E+01 ± 1.2336E+00	≈	2.0091E+01 ± 5.8358E+00
f_{15}	1.0879E+00 ± 6.8452E–01	+	3.2152E+00 ± 1.3212E+00
f_{16}	7.8923E+01 ± 8.3934E+01	≈	4.1378E+01 ± 3.4814E+01
f_{17}	3.2925E+01 ± 7.9971E+00	≈	3.2355E+01 ± 5.9758E+00
f_{18}	2.0411E+01 ± 2.8443E+00	+	2.1232E+01 ± 4.2028E+00
f_{19}	4.5031E+00 ± 1.7152E+00	+	5.2171E+00 ± 1.7886E+00
f_{20}	2.9368E+01 ± 5.7971E+00	≈	3.0451E+01 ± 5.5810E+00
f_{21}	2.0929E+02 ± 1.9362E+00	−	2.0743E+02 ± 1.8419E+00
f_{22}	1.0000E+02 ± 2.8422E–14	≈	1.0000E+02 ± 9.1103E–14
f_{23}	3.5075E+02 ± 3.2667E+00	≈	3.5154E+02 ± 2.2484E+00
f_{24}	4.2646E+02 ± 2.4419E+00	+	4.2746E+02 ± 1.7486E+00
f_{25}	3.8670E+02 ± 7.6056E–03	+	3.8674E+02 ± 2.9143E–02
f_{26}	9.2021E+02 ± 4.2530E+01	+	9.5137E+02 ± 3.7879E+01
f_{27}	4.9739E+02 ± 6.9327E+00	+	5.0482E+02 ± 4.9047E+00
f_{28}	3.0873E+02 ± 2.9952E+01	+	3.3119E+02 ± 5.1065E+01
f_{29}	4.3367E+02 ± 1.3507E+01	≈	4.3424E+02 ± 6.8313E+00
f_{30}	1.9712E+03 ± 1.8774E+01	≈	1.9962E+03 ± 5.2424E+01

TABLE VII
COMPARISON OF RESULTS OBTAINED BY jSO AND L-SHADE FOR $D = 50$. WILCOXON RANK-SUM TEST ($\alpha = 0.05$).

	jSO		L-SHADE
f_1	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_2	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_3	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_4	5.6213E+01 ± 4.8283E+01	≈	6.5558E+01 ± 5.3265E+01
f_5	1.6405E+01 ± 3.4279E+00	−	1.1388E+01 ± 2.1956E+00
f_6	1.0933E–06 ± 2.6000E–06	≈	5.7767E–07 ± 1.2877E–06
f_7	6.6497E+01 ± 3.4386E+00	−	6.3121E+01 ± 1.7759E+00
f_8	1.6962E+01 ± 3.1045E+00	−	1.2136E+01 ± 2.3341E+00
f_9	0.0000E+00 ± 0.0000E+00	≈	3.5109E–03 ± 1.7378E–02
f_{10}	3.1398E+03 ± 3.6355E+02	≈	3.0618E+03 ± 3.4632E+02
f_{11}	2.7939E+01 ± 3.2957E+00	+	4.7502E+01 ± 8.5254E+00
f_{12}	1.6806E+03 ± 5.1777E+02	+	2.2440E+03 ± 5.2256E+02
f_{13}	3.0599E+01 ± 2.1017E+01	+	6.4106E+01 ± 2.6464E+01
f_{14}	2.4964E+01 ± 1.8549E+00	+	2.9408E+01 ± 3.0627E+00
f_{15}	2.3864E+01 ± 2.4637E+00	+	4.0013E+01 ± 1.0250E+01
f_{16}	4.5052E+02 ± 1.3640E+02	−	3.7092E+02 ± 1.2054E+02
f_{17}	2.8287E+02 ± 8.5293E+01	≈	2.5170E+02 ± 6.2340E+01
f_{18}	2.4283E+01 ± 1.9975E+00	+	3.9158E+01 ± 1.1973E+01
f_{19}	1.4139E+01 ± 2.2400E+00	+	2.2502E+01 ± 4.0896E+00
f_{20}	1.4010E+02 ± 7.6613E+01	+	1.4484E+02 ± 5.1325E+01
f_{21}	2.1920E+02 ± 3.7285E+00	−	2.1269E+02 ± 2.7543E+00
f_{22}	1.4872E+03 ± 1.7358E+03	+	2.0199E+03 ± 1.8063E+03
f_{23}	4.3008E+02 ± 6.1750E+00	≈	4.3123E+02 ± 3.8170E+00
f_{24}	5.0745E+02 ± 4.0866E+00	+	5.0999E+02 ± 2.1847E+00
f_{25}	4.8088E+02 ± 2.7723E+00	+	4.8270E+02 ± 4.7639E+00
f_{26}	1.1288E+03 ± 5.5614E+01	+	1.1717E+03 ± 4.5168E+01
f_{27}	5.1127E+02 ± 1.0967E+01	+	5.3658E+02 ± 1.6632E+01
f_{28}	4.5981E+02 ± 6.7724E+00	+	4.7768E+02 ± 2.3496E+01
f_{29}	3.6294E+02 ± 1.3027E+01	−	3.4878E+02 ± 9.7912E+00
f_{30}	6.0105E+05 ± 2.9565E+04	+	6.5897E+05 ± 9.7172E+04

against L-SHADE in the next section, where we will also give summary results for all three algorithms.

IV. EXPERIMENTS

A. Evaluation Method

The CEC 2017 evaluation method combines two scores, defined in Eqs. (6) and (7), to find the final score as follows:

$$score = score_1 + score_2, \tag{5}$$

where

$$score_1 = (1 - \frac{SE - SE_{min}}{SE}) \times 50, \tag{6}$$

Here, SE_{min} is the minimal sum of errors from all the algorithms, and SE is the sum of error values for all dimensions and it is defined as follows:

$$SE = 0.1 \times \sum_{i=1}^{30} ef_{10D} + 0.2 \times \sum_{i=1}^{30} ef_{30D} +$$

$$+0.3 \times \sum_{i=1}^{30} ef_{50D} + 0.4 \times \sum_{i=1}^{30} ef_{100D},$$

where ef is the error values for all the functions.

Then

$$score_2 = (1 - \frac{SR - SR_{min}}{SR}) \times 50, \tag{7}$$

where SR_{min} is the minimal sum of ranks from all the algorithms, and SR is the sum of ranks defined as follows:

$$SR = 0.1 \times \sum_{i=1}^{30} rank_{10D} + 0.2 \times \sum_{i=1}^{30} rank_{30D} +$$

$$+0.3 \times \sum_{i=1}^{30} rank_{50D} + 0.4 \times \sum_{i=1}^{30} rank_{100D}.$$

In the evaluation method, the higher weights are given for higher dimensions.

TABLE VIII
COMPARISON OF RESULTS OBTAINED BY jSO AND L-SHADE FOR $D = 100$. WILCOXON RANK-SUM TEST ($\alpha = 0.05$).

	jSO		L-SHADE
f_1	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_2	8.9403E+00 ± 2.3963E+01	+	3.7908E+05 ± 2.6144E+06
f_3	2.3912E–06 ± 2.6981E–06	+	6.9016E–06 ± 6.7687E–06
f_4	1.8963E+02 ± 2.8638E+01	≈	1.9658E+02 ± 4.6028E+00
f_5	4.3908E+01 ± 5.5513E+00	−	3.6235E+01 ± 5.7543E+00
f_6	2.0244E–04 ± 6.1377E–04	+	5.5323E–03 ± 3.1990E–03
f_7	1.4490E+02 ± 6.6369E+00	−	1.4131E+02 ± 4.5655E+00
f_8	4.2152E+01 ± 5.4679E+00	−	3.4373E+01 ± 5.1524E+00
f_9	4.5904E–02 ± 1.1380E–01	+	4.8584E–01 ± 4.8318E–01
f_{10}	9.7044E+03 ± 6.7490E+02	+	1.0507E+04 ± 4.6023E+02
f_{11}	1.1317E+02 ± 4.2778E+01	+	4.5278E+02 ± 8.9427E+01
f_{12}	1.8430E+04 ± 8.2682E+03	+	2.4887E+04 ± 9.8414E+03
f_{13}	1.4489E+02 ± 3.7631E+01	+	5.7034E+02 ± 4.1447E+02
f_{14}	6.4341E+01 ± 1.0787E+01	+	2.5141E+02 ± 2.9416E+01
f_{15}	1.6212E+02 ± 3.7679E+01	+	2.5659E+02 ± 4.0180E+01
f_{16}	1.8586E+03 ± 3.4512E+02	−	1.6579E+03 ± 2.7759E+02
f_{17}	1.2780E+03 ± 2.3608E+02	−	1.1091E+03 ± 1.9942E+02
f_{18}	1.6728E+02 ± 3.6139E+01	+	2.3964E+02 ± 6.4499E+01
f_{19}	1.0487E+02 ± 1.9881E+01	+	1.7836E+02 ± 2.4235E+01
f_{20}	1.3760E+03 ± 2.4042E+02	+	1.5003E+03 ± 1.9749E+02
f_{21}	2.6380E+02 ± 6.3652E+00	−	2.5935E+02 ± 4.5604E+00
f_{22}	1.0210E+04 ± 2.1609E+03	+	1.1286E+04 ± 5.6467E+02
f_{23}	5.7115E+02 ± 1.0597E+01	−	5.6589E+02 ± 9.0198E+00
f_{24}	9.0218E+02 ± 7.8155E+00	+	9.2037E+02 ± 6.7820E+00
f_{25}	7.3609E+02 ± 3.4954E+01	+	7.5322E+02 ± 2.5773E+01
f_{26}	3.2673E+03 ± 7.9402E+01	+	3.4296E+03 ± 8.3405E+01
f_{27}	5.8547E+02 ± 2.1458E+01	+	6.4291E+02 ± 1.7035E+01
f_{28}	5.2682E+02 ± 2.7036E+01	≈	5.2745E+02 ± 2.1404E+01
f_{29}	1.2566E+03 ± 1.8945E+02	≈	1.2651E+03 ± 1.7645E+02
f_{30}	2.3255E+03 ± 1.1761E+02	+	2.4067E+03 ± 1.5177E+02

B. Experimental Results

We tested the jSO algorithm on scalable benchmark functions, provided for CEC 2017 special session [23]. The dimensions of benchmark functions in this special session are $D = 10, 30, 50$ and 100. The values of the optimal solutions are known in advance for all benchmark functions. An algorithm needs to perform 51 runs for each function, the maximum number of objective function evaluations, max_nfes is $D\times$10,000. There exists the source code of CEC 2017 benchmark functions, but an algorithm is required to use these functions as a black-box, i.e., without explicit knowledge of the structure of benchmark functions.

PC configuration:
System: GNU Linux, CPU: Intel(R) Core(TM) i7-4770 CPU 3.4 GHz, Main memory: 16 GB, Programming language: C++, Algorithm: jSO, Compiler: g++ (GNU Compiler).

The parameters in jSO were kept unchanged according to the parameters setting in the L-SHADE and iL-SHADE algorithms, except the following parameters:

- current-to-pBest-w/1 mutation strategy (current-to-pBest/1 in L-SHADE and iL-SHADE),
- p value for mutation strategy linearly decreases from p_{max} to p_{min} during the evolutionary process, where $p_{max} = 0.25$ in jSO ($p_{max} = 0.2$ in iL-SHADE) and $p_{min} = p_{max}/2$ (in L-SHADE p is fixed to 0.11)
- initial population size $N^{init} = 25\log(D)\sqrt{(D)}$ (in L-SHADE it is set to $18D$, while in iL-SHADE it is set to $12D$),
- historical memory size $H = 5$ ($H = 6$ in L-SHADE and iL-SHADE),
- M_F values are initialized to 0.3 (0.5 in L-SHADE and iL-SHADE).

The obtained results are presented in Tables I, II, III, and IV. In these tables, error values $f(\vec{x}) - f(\vec{x}^*)$, as required in [23], are given. The error values between the best fitness values found in each run out of 51 runs and true optimal value are calculated and then best, worst, median, mean, and standard deviation of the error values are presented in each column in the tables.

Next, we compare the performance of the jSO and L-SHADE algorithms. The results are given in Tables V, VI, VII, and VIII for each dimension. In these tables the mean and standard deviation values are shown for the jSO and L-SHADE algorithms, and the statistical testing is also shown in a separate column. The symbols $+, -, \approx$ indicate that the proposed jSO algorithm performed significantly better ($+$), significantly worse ($-$), or the performance difference is not statistically significant ($\approx$) compared to the L-SHADE algorithm based on the Wilcoxon rank-sum test at the 0.05 significance level.

TABLE IX
SUMMARIZED STATISTICAL TESTINGS INDICATE THAT THE jSO ALGORITHM PERFORMED SIGNIFICANTLY BETTER ($+$), WORSE ($-$), OR THE PERFORMANCE DIFFERENCE IS NOT STATISTICALLY SIGNIFICANT ($\approx$) COMPARED TO L-SHADE AND iL-SHADE, RESPECTIVELY (WILCOXON RANK-SUM TEST AT THE 0.05 SIGNIFICANCE LEVEL).

jSO vs. L-SHADE	+	≈	−
$D = 10$	8	17	5
$D = 30$	11	15	4
$D = 50$	15	9	6
$D = 100$	19	4	7
jSO vs. iL-SHADE	**+**	**≈**	**−**
$D = 10$	7	19	4
$D = 30$	7	13	10
$D = 50$	13	7	10
$D = 100$	13	8	9

Summarized results of the statistical testing are presented in Table IX. If we compare a number of wins ($+$) and loses ($-$), we can see that jSO indicates the better performance than L-SHADE and iL-SHADE on all dimensions, with only one exception (iL-SHADE performed better than jSO on $D = 30$). Superior performance of jSO is seen especially on $D = 100$.

TABLE X
SCORES OF EVALUATION METHOD FOR L-SHADE, IL-SHADE, AND JSO.

L-SHADE	iL-SHADE	jSO
63.64	93.30	**100.0**

TABLE XI
ALGORITHM RUN-TIME COMPLEXITY OF THE JSO ALGORITHM

	T_0 $[s]$	T_1 $[s]$	$\hat{T}_2$ $[s]$	$(\hat{T}_2 - T_1)/T_0$
$D = 10$	0.0730	0.1661	0.2816	1.5822
$D = 30$		0.7579	0.9807	3.0520
$D = 50$		1.8472	2.1541	4.2041

Table X shows the obtained scores according to the evaluation eethod, presented in Section IV-A. In this method, a higher score is better. We can see that jSO obtained the highest score, followed by iL-SHADE and L-SHADE.

Table XI presents the complexity of the jSO algorithm. Notice, this complexity should be considered with some care, since the measured values of run-time are very small.

V. CONCLUSIONS

Differential evolution based algorithms have shown a very good performance on solving numerical optimization problems. In this paper, we present results of our algorithm (jSO) on the set of benchmark functions provided for CEC 2017 special session on single-objective real-parameter optimization.

In this work, we compared our jSO algorithm with the L-SHADE and iL-SHADE algorithms. The jSO algorithm indicates better overall results when comparing to the L-SHADE and iL-SHADE algorithms on all dimensions, with one exception only – iL-SHADE performed better than jSO on $D = 30$. The jSO, iL-SHADE, and L-SHADE algorithms obtained 100.0, 93.30, and 63.64 scores, respectively, according to the CEC 2017 evaluation method, where higher score is better.

ACKNOWLEDGMENT

This work was supported by the Slovenian Research Agency under programs P2-0041 – Computer Systems, Methodologies, and Intelligent Services, and P2-0069 – Advanced Methods of Interaction in Telecommunications.

REFERENCES

[1] R. Tanabe and A. Fukunaga, "Evaluating the performance of SHADE on CEC 2013 benchmark problems," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, June 2013, pp. 1952–1959.

[2] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[3] S. Das and P. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 27–54, 2011.

[4] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 61–106, 2010.

[5] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution–an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[6] R. Tanabe and A. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC2014)*. IEEE, 2014, pp. 1658–1665.

[7] J. Brest, M. S. Maučec, and B. Bošković, "iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization," in *IEEE Congress on Evolutionary Computation (CEC) 2016*. IEEE, 2016, pp. 1188–1195.

[8] U. Mlakar, B. Potočnik, and J. Brest, "A hybrid differential evolution for optimal multilevel image thresholding," *Expert Systems with Applications*, vol. 65, pp. 221–232, 2016.

[9] B. Bošković and J. Brest, "Differential evolution for protein folding optimization based on a three-dimensional AB off-lattice model," *Journal of Molecular Modeling*, vol. 22, pp. 1–15, 2016.

[10] A. Zamuda and J. Brest, "Self-adaptive control parameters randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.

[11] R. P. Parouha and K. N. Das, "A memory based differential evolution algorithm for unconstrained optimization," *Applied Soft Computing*, vol. 38, pp. 501–517, 2016.

[12] I. Fister, P. N. Suganthan, I. F. J. and S. M. Kamal, F. M. Al-Marzouki, M. Perc, and D. Strnad, "Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution," *Nonlinear Dynamics*, vol. 84, pp. 895–914, 2016.

[13] I. Poikolainen, F. Neri, and F. Caraffini, "Cluster-based population initialization for differential evolution frameworks," *Information Sciences*, vol. 297, pp. 216–235, 2015.

[14] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[15] 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 Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[16] J. Zhang and A. Sanderson, "JADE: Adaptive Differential Evolution with Optional External Archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[17] J. Brest and M. S. Maučec, "Population Size Reduction for the Differential Evolution Algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228–247, 2008.

[18] S.-M. Guo, J.-H. Tsai, C.-C. Yang, and P.-H. Hsu, "A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 1003–1010.

[19] N. Awad, M. Ali, and R. Reynolds, "A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 1098–1105.

[20] K. Sallam, R. Sarker, D. Essam, and S. Elsayed, "Neurodynamic differential evolution algorithm and solving cec2015 competition problems," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 1033–1040.

[21] N. H. Awad, M. Z. Ali, P. N. Suganthan, and R. G. Reynolds, "An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems," in *IEEE Congress on Evolutionary Computation (CEC) 2016*. IEEE, 2016, pp. 2958–2965.

[22] R. Poláková, J. Tvrdík, and P. Bujok, "Evaluating the performance of L-SHADE with competing strategies on CEC2014 single parameter-operator test suite," in *IEEE Congress on Evolutionary Computation (CEC) 2016*. IEEE, 2016, pp. 1181–1187.

[23] N. H. Awad, M. Z. Ali, B. Y. Q. J. J. Liang, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization," Nanyang Technological University, Singapore, Tech. Rep., November 2016. [Online]. Available: http://www.ntu.edu.sg/home/epnsugan/