iL-SHADE: Improved L-SHADE Algorithm for Single Objective Real-Parameter Optimization

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

***Abstract*—In this paper we present a differential evolution algorithm (iL-SHADE) for solving single objective real-parameter optimization problems. It is an improved version of the well-known L-SHADE algorithm. The experimental results of our algorithm are presented on CEC 2014 benchmark functions. The experiments were performed on 30 benchmark functions and on four different dimensions. The obtained results show that our algorithm performs highly competitive in comparison with the original L-SHADE algorithm.**

I. Introduction

L-SHADE [1] is a state-of-the-art evolutionary algorithm which has been the best ranked (see http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014/CEC2014.htm) Differential Evolution (DE) algorithm on CEC2014 Competition on Real-Parameter Single Objective Optimization [2]. The L-SHADE algorithm extends the Success-History based Adaptive DE (SHADE) [3] algorithm with the linear population size reduction mechanism. SHADE was introduced in 2013, and it is actually an improved version of JADE [4]. SHADE-cc [5] is one of the improved version of the SHADE algorithm.

The global optimization problem can be defined as follows. We need to find variables of vector $\vec{x} = \{x_1, x_2, ..., x_D\}$ which minimizes an objective function $f(\vec{x})$, where D denotes the dimensionality of the problem. Domains of the variables are defined by their lower and upper bounds, $x_{j,low}$ and $x_{j,upp}$ for $j = 1, 2, ..., D$. Therefore, a single objective optimization is called also a bound-constrained black-box optimization. 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 [3]. We can find such problems in engineering optimization and in other real-world problems.

When dealing with the single objective optimization problems, more challenges can arise regarding properties of the problems. Two main challenges are regarding the dimensionality and multi-modality. An algorithm may get into difficulties, either when a dimension of a problem is large, since the search space is very huge, or when solving problems with more optima. Therefore, high dimensional and multi-modal problems are usually more challenging. In case when an optimization function is, for example, multi-modal, an algorithm may be trapped in a local optimum. Therefore, an algorithm needs to try avoiding getting stacked in local optima during the evolutionary process.

Recently, increasing research efforts on handling complex single-objective optimization have been reported, and a variety of algorithms have been presented. Recently, special sessions and competitions on real-parameter single objective optimization were organized at CEC2005, CEC2013, CEC2014 and CEC2015. Closely related were sessions/competitions on large-scale global optimization, which have been also organized at CEC.

Differential Evolution (DE) is a stochastic floating-point encoding evolutionary algorithm that was originally designed for numerical optimization [6]. This simple algorithm has shown its efficiency, robustness and competitiveness when solving real-world optimization problems, especially over continuous spaces [7], [8].

DE has three control parameters: F is amplification factor of the difference vector, CR is crossover control parameter, and NP is population size. They are fixed in the original DE algorithm. One step forward of user's viewpoint presents adaptive and/or self-adaptive mechanisms. Then the user does not need to set/adjust/tune the control parameters. jDE [9] and SADE [10], for example, have self-adaptive control parameters (F and CR), while NP remains a user defined parameter and is not being changed during the optimization process. Later we can find history based mechanisms in a chain of algorithms: JADE – SHADE – L-SHADE, in order to make F and CR self-adaptive, and we can see that the population size also undertakes *evolutionary process*, i.e. NP is fixed in SHADE, while L-SHADE changes it during the optimization process.

In CEC2015 competition [11], the participants were allowed to optimize the parameters of their proposed (hybrid) optimization algorithm for each problem – Learning-Based Problems. On previous CEC competitions an optimization algorithm needs to use the same parameters for solving all benchmark problems. In CEC2015 improved version of L-SHADE have been placed on top ranks: SPS-L-SHADE-EIG [12] rank #1, DEsPA [13] rank #2, and LSHADE-ND [14] rank #3–5. SPS-L-SHADE-EIG extends L-SHADE with an eigenvector-based (EIG) crossover and a successful-parent-selecting (SPS) framework, DEsPA is an enhancement of JADE algorithm which uses a success-based parameter adaptation with resizing population space, and LSHADE-ND is a modified L-SHADE and it uses also neuro-dynamic (ND) mechanism.

978-1-5090-0623-6/16/$31.00

$g \leftarrow 1$, 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_{CR}, M_F to 0.5;
$k \leftarrow 1$ // *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 // $H = 6$
    if $M_{CR,r_i} = \perp$ ***then***
      $CR_{i,g} \leftarrow 0$
    else
      $CR_{i,g} \leftarrow \mathcal{N}_i(M_{CR,r_i}, 0.1)$ *// Normal distribution*
    end if
    $F_{i,g} \leftarrow \mathcal{C}_i(M_{F,r_i}, 0.1)$ *// Cauchy distribution*
    $\vec{u}_{i,g} \leftarrow$ *current-to-pBest/1/bin*
  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 **A**, if necessary
    Update M_{CR} and M_F (Algorithm 2)
    Apply LPSR strategy // *linear population size reduction*
  end for
  $g \leftarrow g + 1$
end while

Algorithm 1: The L-SHADE[1] algorithm

SHADE [3] is an adaptive DE which incorporates success-history based parameter adaptation and one of the state-of-the-art DE algorithms. Success-history based adaptation uses a historical memory M_{CR}, M_F which stores a set of CR, F values that have performed well in the past, and generate new CR, F pairs by directly sampling the parameter space close to one of these stored pairs. L-SHADE [1] further extends SHADE with Linear Population Size Reduction (LPSR), which continually decreases the population size according to a linear function.

In this paper we present an improved version of the L-SHADE algorithm, called iL-SHADE. The main improvements of iL-SHADE are applied in the memory update mechanism.

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 algorithm are given. Section III presents a description of our new variant of the algorithm, called iL-SHADE, which is used in these experiments. In Section IV experimental results of the iL-SHADE algorithm on benchmark functions are presented. Section V concludes the paper with some final remarks.

II. Background

This section gives backgrounds for this work, an overview of DE, and a description of L-SHADE.

if $S_{CR} \neq \emptyset$ **and** $S_F \neq \emptyset$ **then**
  if $M_{CR,k,g} = \perp$ **or** $max(S_{CR}) = 0$ **then**
    $M_{CR,k,g} \leftarrow \perp$
  else
    $M_{CR,k,g+1} \leftarrow mean_{WL}(S_{CR})$
  end if
  $M_{F,k,g+1} \leftarrow mean_{WL}(S_F)$
  $k \leftarrow k + 1$
  if $k > H$ **then**
    $k \leftarrow 1$
  end if
else
  $M_{CR,k,g+1} \leftarrow M_{CR,k,g}$
  $M_{F,k,g+1} \leftarrow M_{F,k,g}$
end if

Algorithm 2: Memory update in the L-SHADE[1] algorithm

A. Differential Evolution

Differential evolution (DE) belongs to the group of evolutionary algorithms. We can find out that DE is very useful in many practical applications and researches [15], [16], [17], [7].

The population P of the DE algorithm [6] consists of NP individuals which are represented as vectors:

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

where g is a generation index $g \in \{1, 2, \ldots, G_{MAX}\}$. Each vector $\vec{x}_{i,g}$ has D variables:

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

DE evolves a randomly initialized population throughout G_{MAX} generations guided the individuals in searching process toward to a global optimum. DE employs three operations for each individual during one generation: mutation, crossover and selection. At the end of evolutionary process, i.e. after G_{MAX} generations, DE stops and returns the best vector found so far as the solution.

During the mutation operation DE creates a mutant vector $\vec{v}_{i,g}$ using one of the mutation strategies. The 'DE/rand/1' strategy has been introduced in the original DE algorithm and it is still very often used in practice. This strategy is defined as follow:

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

where F denotes mutation scale factor, and 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. The mutant vector with index i is created using three vectors, a difference between two of them is multiplied with the scale factor F and then the third vector is added.

The other useful DE strategies [7], [8] are:

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

$g \leftarrow 1$, 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
Set all values in M_{CR} to 0.8 $\Longleftarrow$
$k \leftarrow 1$ *// 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** $\Longleftarrow$
      $M_{F,r_i} \leftarrow 0.9$ $\Longleftarrow$
      $M_{CR,r_i} \leftarrow 0.9$ $\Longleftarrow$
    end if $\Longleftarrow$
    if $M_{CR,r_i} < 0$ **then** $\Longleftarrow$
      $CR_{i,g} \leftarrow 0$
    else
      $CR_{i,g} \leftarrow \mathcal{N}_i(M_{CR,r_i}, 0.1)$
    end if
    if $g < 0.25G_{MAX}$ **then** $\Longleftarrow$
      $CR_{i,g} \leftarrow \max(CR_{i,g}, 0.5)$ $\Longleftarrow$
    else if $g < 0.5G_{MAX}$ **then** $\Longleftarrow$
      $CR_{i,g} \leftarrow \max(CR_{i,g}, 0.25)$ $\Longleftarrow$
    end if $\Longleftarrow$
    $F_{i,g} \leftarrow \mathcal{C}_i(M_{F,r_i}, 0.1)$
    if $g < 0.25G_{MAX}$ **then** $\Longleftarrow$
      $F_{i,g} \leftarrow \min(F_{i,g}, 0.7)$ $\Longleftarrow$
    else if $g < 0.5G_{MAX}$ **then** $\Longleftarrow$
      $F_{i,g} \leftarrow \min(F_{i,g}, 0.8)$ $\Longleftarrow$
    else if $g < 0.75G_{MAX}$ **then** $\Longleftarrow$
      $F_{i,g} \leftarrow \min(F_{i,g}, 0.9)$ $\Longleftarrow$
    end if $\Longleftarrow$
    $\vec{u}_{i,g} \leftarrow$ current-to-pBest/1/bin
  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 (Algorithm 4)
    Apply LPSR strategy *// linear population size reduction*
    Update p using Eq. (1) $\Longleftarrow$
  end for
  $g \leftarrow g + 1$
end while

Algorithm 3: iL-SHADE algorithm

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

if $S_{CR} \neq \emptyset$ **and** $S_F \neq \emptyset$ **then**
  if $M_{CR,k,g} = \perp$ **or** $max(S_{CR}) = 0$ **then**
    $M_{CR,k,g} \leftarrow \perp$
  else
    $M_{CR,k,g+1} \leftarrow (mean_{WL}(S_{CR}) + M_{CR,k,g})/2$ $\Longleftarrow$
  end if
  $M_{F,k,g+1} \leftarrow (mean_{WL}(S_F) + M_{F,k,g})/2$ $\Longleftarrow$
  $k \leftarrow k + 1$
  if $k > H$ **then**
    $k \leftarrow 1$
  end if
else
  $M_{CR,k,g+1} \leftarrow M_{CR,k,g}$
  $M_{F,k,g+1} \leftarrow M_{F,k,g}$
end if

Algorithm 4: Memory update in the iL-SHADE algorithm

Next operation is crossover. It creates a trial vector $\vec{u}_{i,g}$ using *binomial* crossover:

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

for $i = 1, 2, ..., NP$ and $j = 1, 2, ..., D$. CR is crossover parameter within the range $[0, 1)$ and presents the probability of creating components for a trial vector from a mutant vector. Index $j_{rand} \in \{1, 2, ..., NP\}$ is a randomly chosen integer, and at least the component $v_{i,j_{rand},g}$ is surely from a mutant vector. If a component was not selected from the mutant vector, then it is taken from the parent vector $\vec{x}_{i,g}$.

Now, the trial vector is evaluated – an objective function f 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 survive and become a member of the next generation. The selection operation for a minimization optimization problem is defined as follow:

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

B. L-SHADE Algorithm

In this section L-SHADE [1] is briefly presented. L-SHADE is DE-based algorithm with success-history based parameter adaptation (SHADE [3]) and the later is an improved version of JADE [4]. Source code of L-SHADE is available at https://sites.google.com/site/tanaberyoji/home. A pseudo-code of L-SHADE is shown in Algorithms 1 and 2.

SHADE has been the best ranked DE-based algorithm at CEC-2013 Special Session & Competition on Real-Parameter Single Objective Optimization, while among all 21 algorithms it has been on the four-th place after the top three performing algorithms NBIPOP-aCMA-ES [18], iCMAES-ILS [19], and DRMA-LSCh-CMA [20].

L-SHADE combines SHADE and linear population size reduction mechanism, which after each generation decreases the population size according to a linear function. It uses current-to-pBest/1/bin strategy to generate a trial vector, weighted

TABLE I
THE RESULTS OF iL-SHADE FOR $D = 10$. 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.

	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	3.4780E+01	3.4780E+01	3.1370E+01	1.0445E+01
f_5	0.0000E+00	2.0203E+01	2.0005E+01	1.7684E+01	6.5216E+00
f_6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_7	0.0000E+00	1.9678E–02	0.0000E+00	1.4000E–03	4.3830E–03
f_8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_9	0.0000E+00	3.9798E+00	9.9496E–01	1.3071E+00	8.7850E–01
f_{10}	0.0000E+00	6.2454E–02	0.0000E+00	1.3471E–02	2.5943E–02
f_{11}	1.8736E–01	2.4220E+02	1.8472E+01	3.2631E+01	4.8618E+01
f_{12}	2.9572E–07	1.7007E–01	4.6223E–02	5.0280E–02	3.2668E–02
f_{13}	5.9440E–03	5.9737E–02	2.4615E–02	2.8831E–02	1.3213E–02
f_{14}	2.1393E–02	1.4355E–01	6.4297E–02	6.5469E–02	2.4994E–02
f_{15}	2.0716E–01	5.4380E–01	3.7781E–01	3.7227E–01	8.0801E–02
f_{16}	3.4414E–01	1.5609E+00	9.1084E–01	9.6308E–01	3.6262E–01
f_{17}	2.0814E–01	3.8800E+01	1.6194E+00	3.6939E+00	6.3116E+00
f_{18}	8.2715E–03	5.0000E–01	1.5998E–01	1.6950E–01	1.3808E–01
f_{19}	1.2857E–04	1.0159E+00	3.6544E–02	9.1691E–02	2.3246E–01
f_{20}	7.7652E–03	6.2108E–01	3.6850E–01	3.2392E–01	2.1155E–01
f_{21}	8.5228E–06	8.0930E–01	2.7548E–02	1.9975E–01	2.4363E–01
f_{22}	9.7961E–07	2.0004E+01	2.2781E–02	4.3778E–01	2.7958E+00
f_{23}	3.2946E+02	3.2946E+02	3.2946E+02	3.2946E+02	0.0000E+00
f_{24}	1.0000E+02	1.1050E+02	1.0701E+02	1.0583E+02	2.7824E+00
f_{25}	1.0000E+02	2.0137E+02	1.1605E+02	1.2881E+02	3.3072E+01
f_{26}	1.0001E+02	1.0010E+02	1.0003E+02	1.0004E+02	1.9585E–02
f_{27}	6.2384E–01	4.0013E+02	1.3784E+00	1.1842E+02	1.6281E+02
f_{28}	3.5683E+02	4.8084E+02	3.7211E+02	4.0790E+02	4.9609E+01
f_{29}	2.2154E+02	2.2371E+02	2.2176E+02	2.2215E+02	6.0318E–01
f_{30}	4.5429E+02	5.4960E+02	4.6276E+02	4.7555E+02	2.4252E+01

TABLE II
THE RESULTS OF iL-SHADE FOR $D = 30$. 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.

	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	2.0000E+01	2.0415E+01	2.0013E+01	2.0069E+01	1.0049E–01
f_6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_9	1.9899E+00	1.1940E+01	6.9647E+00	6.9062E+00	2.0039E+00
f_{10}	0.0000E+00	4.1638E–02	0.0000E+00	1.1022E–02	1.2734E–02
f_{11}	4.6422E+02	1.6868E+03	1.1672E+03	1.1699E+03	2.8048E+02
f_{12}	6.1444E–02	3.4395E–01	1.4641E–01	1.4807E–01	5.1389E–02
f_{13}	5.0159E–02	1.5002E–01	9.2748E–02	9.4959E–02	2.1182E–02
f_{14}	1.2891E–01	2.7996E–01	1.9818E–01	1.9840E–01	3.4574E–02
f_{15}	1.3193E+00	2.3358E+00	1.8465E+00	1.8449E+00	2.7225E–01
f_{16}	5.2813E+00	9.8492E+00	7.9778E+00	8.0154E+00	9.8132E–01
f_{17}	2.7172E+01	2.4582E+02	1.0424E+02	1.3054E+02	6.5206E+01
f_{18}	1.2605E+00	6.4688E+00	4.1785E+00	3.7780E+00	1.4522E+00
f_{19}	4.6832E–01	4.2177E+00	2.2482E+00	2.2855E+00	7.5547E–01
f_{20}	9.0278E–01	5.3181E+00	2.2294E+00	2.4139E+00	1.0882E+00
f_{21}	1.0586E+00	1.7324E+02	1.4780E+01	5.0691E+01	6.1239E+01
f_{22}	2.0939E+01	1.4790E+02	2.2297E+01	2.9005E+01	2.4204E+01
f_{23}	3.1524E+02	3.1524E+02	3.1524E+02	3.1524E+02	0.0000E+00
f_{24}	2.0000E+02	2.2398E+02	2.2172E+02	2.2046E+02	6.0871E+00
f_{25}	2.0252E+02	2.0269E+02	2.0259E+02	2.0259E+02	4.1734E–02
f_{26}	1.0006E+02	1.0014E+02	1.0009E+02	1.0010E+02	2.0571E–02
f_{27}	3.0000E+02	3.3724E+02	3.0000E+02	3.0073E+02	5.2140E+00
f_{28}	8.0157E+02	8.6817E+02	8.4788E+02	8.4432E+02	1.4806E+01
f_{29}	7.1360E+02	7.2854E+02	7.1518E+02	7.1631E+02	3.3100E+00
f_{30}	4.1724E+02	2.4669E+03	1.0545E+03	1.1984E+03	5.5469E+02

Lehmer mean, $mean_{WL}$ (See Eqs.(7)–(9) in [1]), in order to influence F, CR control parameters adaptation. A historical memory has H entries ($|M_F| = |M_{CR}| = H$).

L-SHADE wins the competition at CEC-2014 Special Session & Competition on Real-Parameter Single Objective Optimization [2]. All three control parameter remain fixed during an optimization process in the original DE algorithm [6], while L-SHADE adapts F and CR, and shrinks NP during an optimization process.

III. THE PROPOSED ALGORITHM

In this section our iL-SHADE algorithm for solving single-objective, global optimization is presented. The iL-SHADE algorithm is an extended version of the L-SHADE [1] algorithm.

The proposed iL-SHADE algorithm differs from the L-SHADE algorithm in using the following mechanisms:

- memory update mechanism stores historical memory values of previous generation and uses them to calculate the historical memory values for the next generation.
- a historical memory has H entries, and one of them contains values that are fixed. This entry is not updated, but its values are used for generating CR_i and F_i control parameters.
- all historical memory values in M_{CR} are initialized to 0.8.
- if M_{CR} is assigned the terminal value $\perp$ then M_{CR} is reset to 0.0.

The pseudo-code of the iL-SHADE algorithm is presented in Algorithms 3 and 4. In both algorithms, we marked lines

TABLE III
THE RESULTS OF iL-SHADE FOR $D = 50$. 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.

	Best	Worst	Median	Mean	Std.
f_1	2.3460E+01	8.2134E+03	1.0890E+03	1.9128E+03	2.0566E+03
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	2.7711E–03	9.8397E+01	9.8103E+01	6.1695E+01	4.7768E+01
f_5	2.0000E+01	2.1096E+01	2.0000E+01	2.0100E+01	2.0180E–01
f_6	0.0000E+00	1.5000E+00	4.5893E–04	6.2683E–02	2.5711E–01
f_7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_9	5.9698E+00	1.6914E+01	1.1940E+01	1.1374E+01	2.6044E+00
f_{10}	2.4993E–02	1.7509E–01	8.0111E–02	8.0426E–02	3.1247E–02
f_{11}	1.8234E+03	3.6626E+03	2.9847E+03	2.9588E+03	4.0428E+02
f_{12}	8.7978E–02	3.4365E–01	1.7206E–01	1.7651E–01	5.1823E–02
f_{13}	7.6665E–02	2.0497E–01	1.4392E–01	1.4190E–01	2.7772E–02
f_{14}	1.8851E–01	3.4104E–01	2.5799E–01	2.6051E–01	3.3346E–02
f_{15}	2.9777E+00	5.1987E+00	4.0281E+00	4.1167E+00	4.8177E–01
f_{16}	1.4671E+01	1.8942E+01	1.6294E+01	1.6360E+01	9.0217E–01
f_{17}	3.3564E+02	1.3619E+03	7.0213E+02	7.1636E+02	2.2673E+02
f_{18}	9.4324E+00	6.2971E+01	2.7077E+01	2.9366E+01	1.0327E+01
f_{19}	5.5698E+00	1.0701E+01	8.0301E+00	8.1581E+00	1.3151E+00
f_{20}	3.1435E+00	1.8496E+01	9.2150E+00	9.4487E+00	2.6791E+00
f_{21}	2.4363E+02	6.4210E+02	3.6992E+02	3.7280E+02	1.1306E+02
f_{22}	2.2988E+01	3.6731E+02	1.4428E+02	1.2020E+02	9.0236E+01
f_{23}	3.4400E+02	3.4400E+02	3.4400E+02	3.4400E+02	0.0000E+00
f_{24}	2.6871E+02	2.7627E+02	2.7433E+02	2.7405E+02	1.5744E+00
f_{25}	2.0484E+02	2.0615E+02	2.0514E+02	2.0519E+02	3.0501E–01
f_{26}	1.0009E+02	2.0000E+02	1.0014E+02	1.0406E+02	1.9575E+01
f_{27}	3.0000E+02	3.6799E+02	3.3116E+02	3.2330E+02	2.2645E+01
f_{28}	1.0529E+03	1.2523E+03	1.1297E+03	1.1330E+03	3.6613E+01
f_{29}	7.3330E+02	9.0117E+02	7.9019E+02	8.1256E+02	4.9211E+01
f_{30}	7.8382E+03	9.4726E+03	8.5653E+03	8.5891E+03	3.7571E+02

TABLE IV
THE RESULTS OF iL-SHADE FOR $D = 100$. 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.

	Best	Worst	Median	Mean	Std.
f_1	7.9224E+04	5.8851E+05	1.8516E+05	1.9394E+05	8.0812E+04
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	1.2600E+02	2.1780E+02	1.5696E+02	1.7346E+02	3.0399E+01
f_5	2.0000E+01	2.0717E+01	2.0031E+01	2.0107E+01	1.7287E–01
f_6	8.1519E–01	1.1459E+01	5.8359E+00	5.8349E+00	2.3198E+00
f_7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_8	2.2404E–05	2.9129E–03	2.3004E–04	3.9692E–04	5.2925E–04
f_9	1.4924E+01	3.4832E+01	2.5121E+01	2.4968E+01	4.6660E+00
f_{10}	5.3794E+00	1.7381E+01	1.0211E+01	1.0482E+01	3.2228E+00
f_{11}	8.0221E+03	1.0824E+04	9.5351E+03	9.4379E+03	6.9454E+02
f_{12}	2.0045E–01	8.9359E–01	3.8552E–01	3.9561E–01	1.2114E–01
f_{13}	1.4860E–01	2.9573E–01	2.2923E–01	2.2813E–01	2.8423E–02
f_{14}	1.7909E–01	2.5782E–01	2.0291E–01	2.0382E–01	1.5239E–02
f_{15}	9.5777E+00	1.6382E+01	1.2455E+01	1.2928E+01	1.5267E+00
f_{16}	3.5228E+01	3.9649E+01	3.8044E+01	3.7996E+01	7.4736E–01
f_{17}	2.7981E+03	5.6538E+03	4.2180E+03	4.1785E+03	6.2911E+02
f_{18}	1.9425E+02	2.8489E+02	2.2742E+02	2.3086E+02	1.9319E+01
f_{19}	8.9930E+01	9.5206E+01	9.2053E+01	9.2132E+01	1.2563E+00
f_{20}	4.6399E+01	1.3397E+02	8.7128E+01	8.8120E+01	1.9660E+01
f_{21}	6.3687E+02	2.5057E+03	1.6320E+03	1.6523E+03	4.1225E+02
f_{22}	6.3723E+02	1.3582E+03	1.0424E+03	1.0360E+03	1.7832E+02
f_{23}	3.4823E+02	3.4823E+02	3.4823E+02	3.4823E+02	0.0000E+00
f_{24}	3.8532E+02	3.9725E+02	3.8977E+02	3.8995E+02	2.7466E+00
f_{25}	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
f_{26}	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
f_{27}	3.1628E+02	4.3786E+02	3.5640E+02	3.6139E+02	2.8965E+01
f_{28}	1.5214E+03	2.3818E+03	2.2084E+03	2.2046E+03	1.2714E+02
f_{29}	7.0733E+02	1.3533E+03	7.6085E+02	8.0594E+02	1.4325E+02
f_{30}	5.5461E+03	1.0175E+04	8.0824E+03	7.9206E+03	9.6872E+02

with $\Longleftarrow$ that are new in iL-SHADE or changed with respect to the L-SHADE algorithm.

Let us summarize and describe the main features that are proposed in the iL-SHADE algorithm:

1) A 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 (i.e. the last one in our case, see Lines 10–13 in Algorithm 3) contains fixed values, i.e. M_{CR,r_i} and M_{F,r_i} are set to 0.9. In such a way not only a higher value for CR_i is used more often, but also a higher F_i as 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, $g+1$. See Lines 5 and 7 in Algorithm 4.
3) Very high values of F and low values of CR are not allowed in an early stage of evolutionary process (Lines 19–23 and 24–31 in Algorithm 3).
4) After each generation g, p value for current-to-pBest/1 mutation in the next generation, $g + 1$, is computed as

TABLE V
COMPARISON OF RESULTS OBTAINED BY iL-SHADE AND L-SHADE FOR $D = 10$.

	iL-SHADE		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	3.1370E+01 ± 1.0343E+01	≈	3.0007E+01 ± 1.1968E+01
f_5	1.7684E+01 ± 6.4574E+00	≈	1.6381E+01 ± 7.3540E+00
f_6	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_7	1.4000E–03 ± 4.3398E–03	+	5.0207E–03 ± 1.2120E–02
f_8	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_9	1.3071E+00 ± 8.6985E–01	+	2.3040E+00 ± 8.2368E–01
f_{10}	1.3471E–02 ± 2.5687E–02	≈	1.2246E–02 ± 2.7709E–02
f_{11}	3.2631E+01 ± 4.8139E+01	≈	3.6160E+01 ± 4.3451E+01
f_{12}	5.0280E–02 ± 3.2346E–02	+	6.7638E–02 ± 1.7475E–02
f_{13}	2.8831E–02 ± 1.3083E–02	+	5.1009E–02 ± 1.6907E–02
f_{14}	6.5469E–02 ± 2.4748E–02	+	8.0008E–02 ± 2.7314E–02
f_{15}	3.7227E–01 ± 8.0005E–02	≈	3.7892E–01 ± 6.5358E–02
f_{16}	9.6308E–01 ± 3.5905E–01	+	1.2212E+00 ± 3.7540E–01
f_{17}	3.6939E+00 ± 6.2494E+00	−	9.9070E–01 ± 8.7412E–01
f_{18}	1.6950E–01 ± 1.3672E–01	≈	2.0031E–01 ± 2.6532E–01
f_{19}	9.1691E–02 ± 2.3017E–01	+	1.2429E–01 ± 2.0154E–01
f_{20}	3.2392E–01 ± 2.0946E–01	−	1.7440E–01 ± 1.7106E–01
f_{21}	1.9975E–01 ± 2.4123E–01	+	4.1243E–01 ± 2.6888E–01
f_{22}	4.3778E–01 ± 2.7683E+00	−	5.0267E–02 ± 4.6226E–02
f_{23}	3.2946E+02 ± 2.8422E–13	≈	3.2946E+02 ± 2.8422E–13
f_{24}	1.0583E+02 ± 2.7550E+00	+	1.0781E+02 ± 1.8812E+00
f_{25}	1.2881E+02 ± 3.2746E+01	≈	1.2933E+02 ± 3.7089E+01
f_{26}	1.0004E+02 ± 1.9392E–02	+	1.0005E+02 ± 1.7274E–02
f_{27}	1.1842E+02 ± 1.6121E+02	≈	5.0305E+01 ± 1.2379E+02
f_{28}	4.0790E+02 ± 4.9120E+01	−	3.8589E+02 ± 3.9653E+01
f_{29}	2.2215E+02 ± 5.9724E–01	≈	2.2204E+02 ± 5.0400E–01
f_{30}	4.7555E+02 ± 2.4013E+01	−	4.6454E+02 ± 9.2636E+00

TABLE VI
COMPARISON OF RESULTS OBTAINED BY iL-SHADE AND L-SHADE FOR $D = 30$.

	iL-SHADE		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	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_5	2.0069E+01 ± 9.9499E–02	+	2.0116E+01 ± 3.3621E–02
f_6	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_7	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_8	0.0000E+00 ± 0.0000E+00	≈	0.0000E+00 ± 0.0000E+00
f_9	6.9062E+00 ± 1.9842E+00	≈	6.8510E+00 ± 1.5612E+00
f_{10}	1.1022E–02 ± 1.2609E–02	+	2.0003E–02 ± 1.7473E–02
f_{11}	1.1699E+03 ± 2.7772E+02	≈	1.2213E+03 ± 1.7937E+02
f_{12}	1.4807E–01 ± 5.0883E–02	+	1.6299E–01 ± 2.4932E–02
f_{13}	9.4959E–02 ± 2.0973E–02	+	1.1545E–01 ± 1.7484E–02
f_{14}	1.9840E–01 ± 3.4233E–02	+	2.3454E–01 ± 2.7345E–02
f_{15}	1.8449E+00 ± 2.6957E–01	+	2.1927E+00 ± 2.2811E–01
f_{16}	8.0154E+00 ± 9.7165E–01	+	8.5660E+00 ± 3.9343E–01
f_{17}	1.3054E+02 ± 6.4563E+01	+	2.1358E+02 ± 1.0563E+02
f_{18}	3.7780E+00 ± 1.4379E+00	+	6.0481E+00 ± 3.0859E+00
f_{19}	2.2855E+00 ± 7.4803E–01	+	3.5946E+00 ± 6.2441E–01
f_{20}	2.4139E+00 ± 1.0774E+00	+	2.9415E+00 ± 1.0209E+00
f_{21}	5.0691E+01 ± 6.0636E+01	≈	8.7216E+01 ± 8.4088E+01
f_{22}	2.9005E+01 ± 2.3965E+01	−	2.7857E+01 ± 1.8702E+01
f_{23}	3.1524E+02 ± 3.5561E–13	≈	3.1524E+02 ± 3.9790E–13
f_{24}	2.2046E+02 ± 6.0271E+00	+	2.2422E+02 ± 1.0477E+00
f_{25}	2.0259E+02 ± 4.1323E–02	≈	2.0261E+02 ± 6.2762E–02
f_{26}	1.0010E+02 ± 2.0369E–02	+	1.0011E+02 ± 1.2946E–02
f_{27}	3.0073E+02 ± 5.1626E+00	≈	3.0000E+02 ± 1.8342E–13
f_{28}	8.4432E+02 ± 1.4660E+01	−	8.3885E+02 ± 1.3115E+01
f_{29}	7.1631E+02 ± 3.2774E+00	≈	7.1631E+02 ± 2.9988E+00
f_{30}	1.1984E+03 ± 5.4922E+02	≈	1.1898E+03 ± 5.1402E+02

follows:

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

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

One can notice, that changes/extensions from L-SHADE to iL-SHADE are not so big — the main features of L-SHADE remains unchanged. The DE strategy current-to-pBest/1/bin, external archive, linear population size reduction, etc. have been kept unchanged. Therefore, we will compare a performance of iL-SHADE against L-SHADE in next section.

IV. EXPERIMENTAL RESULTS

The iL-SHADE algorithm was tested on a set of 30 benchmark functions [2]. The dimensions of benchmark functions are $D = 10, 30, 50$ and 100, and 51 runs of an algorithm were needed for each function. The maximum number of objective function evaluations is $D\times$10,000. The optimal values are known for all benchmark functions.

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

In the experiments, parameters for our algorithm were kept unchanged according to the parameters setting in the L-SHADE algorithm, except the following parameters:

- p value for current-to-pBest/1 mutation linearly decreases from $p_{max} = 0.2$ to $p_{min} = 0.1$ during the evolutionary process (in L-SHADE p is fixed to 0.11),
- $r^{N^{init}}$ is 12 (in L-SHADE it is 18).

The initial population size N^{init} is set to the dimensionality D of the functions multiplied by $r^{N^{init}}$. The calculation of the initial population size is the same in both algorithms.

The obtained results (error values $f(\vec{x}) - f(\vec{x}^*)$) as required

TABLE VII
COMPARISON OF RESULTS OBTAINED BY iL-SHADE AND L-SHADE FOR $D = 50$.

	iL-SHADE		L-SHADE
f_1	1.9128E+03 ± 2.0363E+03	$-$	7.2213E+02 ± 7.6696E+02
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	6.1695E+01 ± 4.7297E+01	$\approx$	6.4626E+01 ± 4.4283E+01
f_5	2.0100E+01 ± 1.9981E–01	$+$	2.0252E+01 ± 4.6847E–02
f_6	6.2683E–02 ± 2.5457E–01	$+$	4.9725E–01 ± 6.4708E–01
f_7	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_8	0.0000E+00 ± 0.0000E+00	$+$	4.9281E–09 ± 9.6145E–09
f_9	1.1374E+01 ± 2.5787E+00	$\approx$	1.1422E+01 ± 2.3726E+00
f_{10}	8.0426E–02 ± 3.0939E–02	$+$	1.2477E–01 ± 3.3960E–02
f_{11}	2.9588E+03 ± 4.0030E+02	$+$	3.2831E+03 ± 3.3141E+02
f_{12}	1.7651E–01 ± 5.1313E–02	$+$	2.2044E–01 ± 2.6505E–02
f_{13}	1.4190E–01 ± 2.7498E–02	$+$	1.6368E–01 ± 2.0719E–02
f_{14}	2.6051E–01 ± 3.3017E–02	$+$	3.0329E–01 ± 2.2080E–02
f_{15}	4.1167E+00 ± 4.7702E–01	$+$	5.2221E+00 ± 4.2343E–01
f_{16}	1.6360E+01 ± 8.9328E–01	$+$	1.6792E+01 ± 4.4278E–01
f_{17}	7.1636E+02 ± 2.2450E+02	$+$	1.4396E+03 ± 3.1198E+02
f_{18}	2.9366E+01 ± 1.0226E+01	$+$	1.0002E+02 ± 1.4931E+01
f_{19}	8.1581E+00 ± 1.3021E+00	$\approx$	8.0939E+00 ± 1.7959E+00
f_{20}	9.4487E+00 ± 2.6527E+00	$+$	1.4306E+01 ± 4.4138E+00
f_{21}	3.7280E+02 ± 1.1195E+02	$+$	5.0642E+02 ± 1.5089E+02
f_{22}	1.2020E+02 ± 8.9347E+01	$\approx$	1.1811E+02 ± 6.8397E+01
f_{23}	3.4400E+02 ± 4.5440E–13	$\approx$	3.4400E+02 ± 4.5558E–13
f_{24}	2.7405E+02 ± 1.5589E+00	$+$	2.7515E+02 ± 6.3438E–01
f_{25}	2.0519E+02 ± 3.0200E–01	$+$	2.0533E+02 ± 3.4392E–01
f_{26}	1.0406E+02 ± 1.9383E+01	$-$	1.0017E+02 ± 1.8723E–02
f_{27}	3.2330E+02 ± 2.2422E+01	$+$	3.3711E+02 ± 3.5587E+01
f_{28}	1.1330E+03 ± 3.6252E+01	$-$	1.1147E+03 ± 3.4293E+01
f_{29}	8.1256E+02 ± 4.8726E+01	$\approx$	8.0532E+02 ± 3.8650E+01
f_{30}	8.5891E+03 ± 3.7201E+02	$\approx$	8.6555E+03 ± 3.6608E+02

TABLE VIII
COMPARISON OF RESULTS OBTAINED BY iL-SHADE AND L-SHADE FOR $D = 100$.

	iL-SHADE		L-SHADE
f_1	1.9394E+05 ± 8.0015E+04	$\approx$	1.7168E+05 ± 5.1877E+04
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	1.7346E+02 ± 3.0099E+01	$-$	1.6212E+02 ± 2.6124E+01
f_5	2.0107E+01 ± 1.7116E–01	$+$	2.0555E+01 ± 3.9808E–02
f_6	5.8349E+00 ± 2.2970E+00	$+$	9.0933E+00 ± 2.6690E+00
f_7	0.0000E+00 ± 0.0000E+00	$\approx$	0.0000E+00 ± 0.0000E+00
f_8	3.9692E–04 ± 5.2403E–04	$+$	1.2548E–02 ± 8.6141E–03
f_9	2.4968E+01 ± 4.6200E+00	$+$	3.3652E+01 ± 6.2040E+00
f_{10}	1.0482E+01 ± 3.1911E+00	$+$	2.6067E+01 ± 5.0688E+00
f_{11}	9.4379E+03 ± 6.8769E+02	$+$	1.0851E+04 ± 5.3633E+02
f_{12}	3.9561E–01 ± 1.1995E–01	$+$	4.2341E–01 ± 4.7061E–02
f_{13}	2.2813E–01 ± 2.8143E–02	$+$	2.4111E–01 ± 2.2058E–02
f_{14}	2.0382E–01 ± 1.5089E–02	$+$	2.1901E–01 ± 1.4371E–02
f_{15}	1.2928E+01 ± 1.5117E+00	$+$	1.6145E+01 ± 1.1058E+00
f_{16}	3.7996E+01 ± 7.4000E–01	$+$	3.9220E+01 ± 4.7065E–01
f_{17}	4.1785E+03 ± 6.2291E+02	$+$	4.4581E+03 ± 7.0390E+02
f_{18}	2.3086E+02 ± 1.9129E+01	$\approx$	2.2411E+02 ± 1.5406E+01
f_{19}	9.2132E+01 ± 1.2439E+00	$+$	9.5817E+01 ± 1.8786E+00
f_{20}	8.8120E+01 ± 1.9467E+01	$+$	1.4335E+02 ± 4.7605E+01
f_{21}	1.6523E+03 ± 4.0818E+02	$+$	2.2699E+03 ± 4.5893E+02
f_{22}	1.0360E+03 ± 1.7656E+02	$+$	1.1194E+03 ± 1.7990E+02
f_{23}	3.4823E+02 ± 1.1369E–13	$\approx$	3.4823E+02 ± 3.1639E–13
f_{24}	3.8995E+02 ± 2.7195E+00	$+$	3.9419E+02 ± 2.0310E+00
f_{25}	2.0000E+02 ± 2.7374E–13	$\approx$	2.0000E+02 ± 2.0054E–13
f_{26}	2.0000E+02 ± 5.3531E–13	$\approx$	1.9804E+02 ± 1.3833E+01
f_{27}	3.6139E+02 ± 2.8680E+01	$+$	3.8075E+02 ± 3.0314E+01
f_{28}	2.2046E+03 ± 1.2589E+02	$+$	2.3056E+03 ± 5.3380E+01
f_{29}	8.0594E+02 ± 1.4184E+02	$\approx$	7.8984E+02 ± 7.7310E+01
f_{30}	7.9206E+03 ± 9.5918E+02	$+$	8.3458E+03 ± 7.1453E+02

in [2] are presented in Tables I, II, III, and IV. 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 a performance of the iL-SHADE and L-SHADE algorithms. The results of SHADE were obtained on the same computer, too. We present the results in Tables V, VI, VII, and VIII for each dimensions. In these tables the mean and standard deviation values are shown for the iL-SHADE and L-SHADE algorithms. The statistical testing on 30 benchmark functions is shown in the last column of the tables. The symbols $+, -, \approx$ indicate that the proposed iL-SHADE algorithm performed significantly better ($+$), significantly worse ($-$), or the performance difference is not statistically significant ($\approx$) compared to the L-SHADE algorithm. We used the Wilcoxon rank-sum test at the 0.05 significance level.

Summarized results of the statistical testing are presented in Table IX. If we compare a number of wins ($+$) and loses ($-$), we can see that iL-SHADE indicates the better performance on all dimensions. There were 120 test functions (30 benchmark functions in each dimensions), L-SHADE preformed better than iL-SHADE in 11 cases, while iL-SHADE wins 60 cases, and the performance difference is not statistically significant

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

iL-SHADE vs. L-SHADE	$+$	$\approx$	$-$
$D = 10$	10	15	5
$D = 30$	13	15	2
$D = 50$	17	10	3
$D = 100$	20	9	1

TABLE X
ALGORITHM COMPLEXITY

	T_0 $[s]$	T_1 $[s]$	$\hat{T}_2$ $[s]$	$(\hat{T}_2 - T_1)/T_0$
$D = 10$	0.0401	0.1407	0.2619	3.0224
$D = 30$		0.6888	0.9112	5.5461
$D = 50$		1.7172	2.0374	7.9850
$D = 100$		6.1493	6.6342	12.0923

in 49 cases.

Table X presents the L-SHADE algorithm complexity.

V. CONCLUSIONS

In this paper we presented the iL-SHADE algorithm. The performance of the algorithm was evaluated on the set of benchmark functions provided for CEC 2014 special session on single-objective real-parameter optimization.

The experimental results give evidence that the iL-SHADE algorithm is highly competitive when comparing to the well-known L-SHADE algorithm on 30 benchmark functions with $D = 10, 30, 50$ and 100 dimensions.

Recently, several DE-based works are related to changing the population size during optimization process, but there are still no general guidelines how to set the population size parameter at the beginning of optimization process, when to change it, either to increase or decrease it, etc. For future research we plan to apply cooperative co-evolution methods to the iL-SHADE algorithm.

ACKNOWLEDGMENT

This work was supported in part by the Slovenian Research Agency under programs P2-0041 and P2-0069.

REFERENCES

[1] 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.

[2] J. J. Liang, B.-Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, Tech. Rep. 201311, 2013. [Online]. Available: http://www.ntu.edu.sg/home/epnsugan/

[3] 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.

[4] 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.

[5] P. Bujok and J. Tvrdík, "Adaptive Differential Evolution: SHADE with Competing Crossover Strategies," in *Artificial Intelligence and Soft Computing: 14th International Conference, ICAISC 2015, Zakopane, Poland, Proceedings, Part I.* Springer International Publishing, 2015, pp. 329–339.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] Q. Chen, B. Liu, Q. Zhang, J. J. Liang, P. N. Suganthan, and B.-Y. Qu, "Problem Definition and Evaluation Criteria for CEC 2015 Special Session and Competition on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization," Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, Nov 2014, Tech. Rep., Nov 2014. [Online]. Available: http://www.ntu.edu.sg/home/epnsugan/

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494615006602

[17] I. Poikolainen, F. Neri, and F. Caraffini, "Cluster-based population initialization for differential evolution frameworks," *Information Sciences*, vol. 297, pp. 216 – 235, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025514010962

[18] I. Loshchilov, "Cma-es with restarts for solving cec 2013 benchmark problems," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, June 2013, pp. 369–376.

[19] T. Liao and T. Stutzle, "Benchmark results for a simple hybrid algorithm on the cec 2013 benchmark set for real-parameter optimization," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, June 2013, pp. 1938–1944.

[20] B. Lacroix, D. Molina, and F. Herrera, "Dynamically updated region based memetic algorithm for the 2013 cec special session and competition on real parameter single objective optimization," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, June 2013, pp. 1945–1951.