An Ensemble Sinusoidal Parameter Adaptation incorporated with L-SHADE for Solving CEC2014 Benchmark Problems

Noor H. Awad[1], Mostafa Z. Ali[2,3], Ponnuthurai N. Suganthan[1], Robert G. Reynolds[4]
[1]Nanyang Technological University, Singapore 639798
School of Electrical & Electronic Engineering
[2]Jordan University of Science & Technology, Jordan 22110
School of Computer Information Systems
[3]Princess Sumayya University for Technology, Jordan
King Hussein Faculty of Computing Sciences
[4]Wayne State University
College of Engineering
Emails: noor0029@e.ntu.edu.sg, mzali.pn@gmail.com, epnsugan@ntu.edu.sg, robert.reynolds@wayne.edu

***Abstract*— An effective and efficient self-adaptation framework is proposed to improve the performance of the L-SHADE algorithm by providing successful alternative adaptation for the selection of control parameters. The proposed algorithm, namely LSHADE-EpSin, uses a new ensemble sinusoidal approach to automatically adapt the values of the scaling factor of the Differential Evolution algorithm. This ensemble approach consists of a mixture of two sinusoidal formulas: A non-Adaptive Sinusoidal Decreasing Adjustment and an adaptive History-based Sinusoidal Increasing Adjustment. The objective of this sinusoidal ensemble approach is to find an effective balance between the exploitation of the already found best solutions, and the exploration of non-visited regions. A local search method based on Gaussian Walks is used at later generations to increase the exploitation ability of LSHADE-EpSin. The proposed algorithm is tested on the IEEE CEC2014 problems used in the Special Session and Competitions on Real-Parameter Single Objective Optimization of the IEEE CEC2016. The results statistically affirm the efficiency and robustness of the proposed approach to obtain better results compared to L-SHADE algorithm and other state-of-the-art algorithms.**

Keywords*— *Differential Evolution; Sinusoidal Fomula, Ensemble approach; Learning-Based Single Objective Optimization.

I. Introduction

The Differential Evolution (DE) algorithm has become one of the most powerful techniques for solving optimization problems [1-3]. DE has produced solid performances when solving a diverse set of problems with varying characteristics such as non-seprability, ill-conditioning, multimodality, in addition to the presence of the curse of dimensionality [4-7]. Based upon experimental and theoretical analysis [8, 9], the performance of DE was shown to be highly dependent on the control settings which are: the scaling factor (*F*), the crossover rate (*CR*), the population size (*NP*), and the chosen mutation/crossover strategies. From this point of view, many different approaches have been proposed in the literature to enhance the performance of DE algorithm. These approaches can be grouped into two main categories. The first category is focused on the development adaptive or self-adaptive techniques that are capable of adjusting the control parameter settings automatically and/or choosing appropriate mutation strategies [10-14]. The second category focusses on hybridizing the DE algorithm with other evolutionary algorithms or local search methods [15, 16]. Some other approaches have attributes from both of these two classes [17, 18].

Many DE-based algorithms were proposed to solve different types of CEC benchmark problems [19]. This is an indication of the usefulness of using DE algorithms for solving varying optimization problems. Among these DE techniques, L-SHADE demonstrated a very good performance in solving the IEEE CEC2014 benchmark set [17]. L-SHADE is an extension version of SHADE algorithm [13] with linear population size reduction. SHADE uses a history-based parameter adaptation scheme based on the JADE algorithm which introduced a novel current/to/pbest mutation strategy [12]. In this paper, the performance of the L-SHADE algorithm is enhanced using a new approach of adapting the scaling factor based on an efficient ensemble sinusoidal scheme. The idea of using the sinusoidal formulas to adapt the *F* and *CR* is not new in DE. The SinDE is an algorithm that uses different sinusoidal formulas to adjust *F* and/or *CR* [20]. In our ensemble approach, a pool of two different sinusoidal adjustments were used: Non-Adaptive Sinusoidal Decreasing Adjustment and Adaptive History-based Sinusoidal Increasing Adjustment. In the context of developing an adaptive sinusoidal approach to adapt control parameter settings in DE, the proposed approach falls in the first class which uses an adaptive history-based scheme to control the settings of *freq* parameter in a sinusoidal formula. The prior successful knowledge is used to adjust the *freq* parameter in each generation. This approach helps in fitting the sinusoidal curve to form more suitable settings of the *scaling* parameter (*F*). This new adaptation scheme provides more suitable settings

978-1-5090-0623-6/16/$31.00 ©2016 IEEE

compared to the L-SHADE algorithm's settings and provides a better balance between exploitation and exploration through successful information sharing. Furthermore, to enhance the exploitation capability of the proposed algorithm, a local search is used at later generations based on Gaussian Walks.

The remainder of the paper is organized as follows. The proposed LSHADE-EpSin is presented in Section 2. The experimental set-up and simulation results are presented in Section 3. Finally, section 4 summarizes the conclusions of this work.

II. L-SHADE WITH AN ENSEMBLE POOL OF SINUSOIDAL PARAMETER ADAPTATION

This section describes the proposed algorithm (LSHADE-EpSin) in which a new adaptation for parameter settings using an ensemble pool of different sinusoidal adjustments is incorporated within L-SHADE.

A. Initialization

The algorithm starts by initializing *NP* *D*-dimensional individuals uniformly distributed within the search space of the problem being solved as follows:

$$x_{i,0}^{j} = x_{\min}^{j} + rand(0,1).(x_{\max}^{j} - x_{\min}^{j}) \quad j = 1, 2, ..., D \tag{1}$$

where *j* is the index of parameter value in the i^{th} individual vector at generation *g*=0, $rand(0,1)$ is a uniformly distributed random generator in the range [0,1] and $X_{\min} = \{x_{\min}^{1}, ..., x_{\min}^{D}\}, X_{\max} = \{x_{\max}^{1}, ..., x_{\max}^{D}\}$ are the lower and upper bounds of each decision variable x_i^j.

B. Current-to-pbest/1 Mutation Strategy with External Archive

The current-to-*p*best/1 mutation strategy was originally proposed by JADE algorithm [12]. It is considered as one of the most powerful mutation strategies for generating promising vectors at each optimization generation [13, 14, 3]. This mutation strategy is an extension of the basic current-to-best/1 where the best individual, $x_{pbest,g}$, is chosen from the top $NP \times p \ (p \in [0,1])$ best individuals of the g^{th} generation as shown in Eq. 2.

$$v_{i,g} = x_{i,g} + F_{i,g}.(x_{pbest,g} - x_{i,g}) + F_{i,g}.(x_{r_1,g} - x_{r_2,g}) \tag{2}$$

where $x_{r_1,g}$ is chosen randomly with an index r_1 selected from [1, *NP*] and $x_{r_2,g}$ is chosen from the union of population P_g and an external archive *A* which stores the inferior parents recently replaced by offspring. Initially the archive is filled with initial population, P_0. At each generation *g*, if the size of *A* exceeds maximum size, *NP*, random individuals are eliminated to free space for newly added individuals.

C. Ensemble of Parameter Adaptation

In the proposed algorithm, an ensemble of parameter settings is used to control the parameter adaptation of current-to-*p*best/1 mutation strategy. The successful setting of the L-SHADE adaptive parameters is incorporated into a new mixture of sinusoidal formulas so as to adapt the scaling factor $F_{i,g}$ at generation *g*. This ensemble consists of two different assignments for control parameters. The first control parameter settings (Sub-section 1) consists of a mixture of two sinusoidal approaches which is used to adapt $F_{i,g}$ and is activated for the first half of generations. The second settings (Sub-section 2) is the L-SHADE adaptation, which are used to adapt $F_{i,g}$ and $CR_{i,g}$ for the second half of generations [17].

1) First Control parameter settings

This control parameter settings is activated at the first half of allowable budget, $g_{s_1} \in [1, \frac{G_{\max}}{2}]$, and consists of a mixture of two different sinusoidal configurations which are:

- Non-Adaptive Sinusoidal Decreasing Adjustment
- Adaptive Sinusoidal Increasing Adjustment

In this sinusoidal pool, one of the two configurations is chosen randomly to adapt $F_{i,g}$ of each individual. The following two sub-sections explain the two sinusoidal adjustments in details.

a) Non-Adaptive Sinusoidal Decreasing Adjustment

In this configuration, we used a decreasing Sine-based formulation to adjust $F_{i,g}$ of each individual in the population at generation g_{s_2} as shown below:

$$F_{i,g_{s_1}} = \frac{1}{2} * \left(\sin(2\pi * freq * g_{s_1} + \pi) * \frac{G_{\max} - g_{s_1}}{G_{\max}} + 1 \right) \tag{3}$$

where *freq* represents the frequency of sinusoidal function, g_{s_2} is the current generation number where $g_{s_1} \in [1, \frac{G_{\max}}{2}]$ and $G_{\max}$ is the maximum allowed generation number. In this configuration, the *freq* parameter is set to a fixed value. This configuration is summarized in Fig 1.

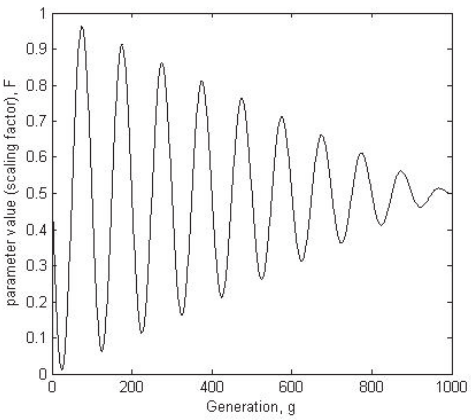

Fig. 1. Sinusoidal decreasing adjustment configuration

b) Adaptive Sinusoidal Increasing Adjustment

In this configuration, we used a different Sine-based formulation as Eq. 11 shows.

$$F_{i,g_{s_1}} = \frac{1}{2} * \left(\sin(2\pi * freq_{i,s_{g_1}} * g_{s_1}) * \frac{g_{s_1}}{G_{\max}} + 1 \right) \quad (4)$$

This formulation represents an increasing adjustment of sinusoidal function as Fig. 2 shows. In this configuration, $freq_{i,s_{g_1}}$ is an adaptive frequency adjusted using an adaptive scheme at each generation g_{s_1} using Cauchy distribution as shown in Eq. 12. This parameter is adapted using a history-based scheme similar as first control parameter settings.

$$freq_{i,s_{g_1}} = randc(\mu freq_{r_i,g_{s_1}}, 0.1) \quad (5)$$

where $\mu freq_{r_i,g_{s_1}}$ is chosen randomly from an external memory M_{freq} which stores the successful mean frequencies for the previous generations in S_{freq}. A random index number, $r_i \in [1, H]$. At the end of generation g_{s_1}, $\mu freq_{r_i,g_{s_1}}$ at index r_i is modified using Lehmer mean as shown in Eq. 7-9.

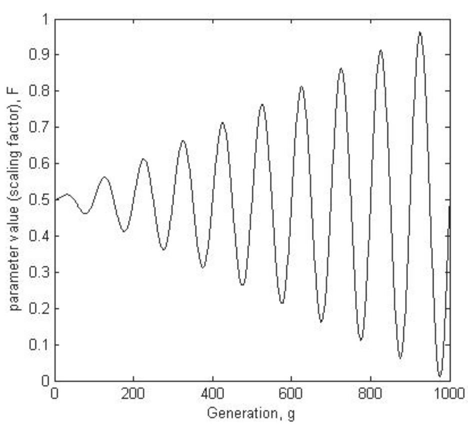

Fig. 2. Sinusoidal increasing adjustment configuration

2) Second Control parameter settings

At each generation $g_{s_2} \in [\frac{G_{\max}}{2}, G_{\max}]$, each individual has its own $F_{i,g}$ and $CR_{i,g}$ parameters that are used to generate new trial vectors. Those two parameters are adapted as shown in Eq. 3 and Eq. 4 using a history-based scheme based on the knowledge from previous generations.

$$F_{i,g} = randc(\mu F_{r_i}, 0.1) \quad (6)$$

$$CR_{i,g} = randn(\mu CR_{r_i}, 0.1) \quad (7)$$

where $randc(\mu F_{r_i}, 0.1)$, $randn(\mu CR_{r_i}, 0.1)$ are two values sampled from Cauchy and Normal distributions, respectively, of mean values equal to μF_{r_i}, μCR_{r_i} and a variance of 0.1.

μF_{r_i}, μCR_{r_i} are chosen randomly from the successful means of previous generations which are stored in a memory *M*. A random index $r_i \in [1, H]$ is selected where *H* is the memory size. Initially the μF, μCR are both set to 0.5 and are updated for the next generations. When $F_i > 1$, it is truncated to 0 and when $F_i \leq 1$, the sampling is repeated until it finds a valid value. In case the value of CR_i outside the range [0,1], it is replaced by 0 or 1 closest to the generated value.

As in the original L-SHADE, the successful values of F, and CR that are able to generate a trial vector better than the parent vector $x_{i,g}$, are recorded in S_F, S_{CR} respectively. At the end of generation g_{s_1}, the memory is updated where the index *k* determines the position in memory to be updated and is incremented when new values enter the memory. The two means μF, μCR at index r_i are modified using the weighted Lehmer mean, $mean_L$, and they are computed as shown in the following equations:

$$\mu F_{k,g_{s_2}+1} = mean_L(S_F) \quad (8)$$

$$\mu CR_{k,g_{s_2}+1} = mean_L(S_{CR}) \quad (9)$$

$$mean_L(S) = \frac{\sum_{k=1}^{|S|} w_k . S_k^2}{\sum_{k=1}^{|S|} w_k . S_k} \quad (10)$$

$$w_k = \frac{\Delta f_k}{\sum_{j=1}^{|S|} \Delta f_j} \quad (11)$$

$$\Delta f_k = |f(u_{k,G}) - f(x_{k,G})| \quad (12)$$

D. Linear population size reduction

It is well known that the selection parameter settings for the scaling factor *F* and crossover rate *CR* impacts the DE performance. The selection of population size *NP* also plays a dominant role in DE performance [3, 6, 14, 21]. In the LSHADE-EpSin algorithm, a linear population size reduction is used to adjust *NP* at each generation *g* as shown in the following equation:

$$NP(g+1) = Round[(\frac{NP_{\min} - NP_{\max}}{FEs_{\max}}).FEs + NP_{\max}] \quad (13)$$

This reduction scheme is called at the end of each generation in order to calculate new *NP* for next generation, $NP(g+1)$. $NP_{\min}$ is set to 4 which is the the minimum number of individuals needed to perform the current-to-*p*best mutation strategy. $NP_{\max}$ is the maximum initial size of the population .

E. Local Search

A local search based on Gaussian Walks is used to increase the exploitation ability of LSHADE-EpSin at later generations when *NP* is decreased to reach 20. When this test condition is satisfied during the run, the local search is activated. Firstly, 10 random individuals are initialized using Eq. 1 and the following Gaussian Walks are performed for 250 generations:

$$y_i = Gaussian(\mu_b, \sigma) + (\varepsilon \times x_{best} - \hat{\varepsilon} \times x_i) \quad (14)$$

where ε and $\hat{\varepsilon}$ are two uniform random numbers in the range [0, 1], x_{best} is the best individual from 10 initial random individuals, x_i is *i*th individual in the 10 individuals group. The mean μ_b is equal to x_{best} while standard deviation σ is computed as shown in Eq. 15 where *G* is the generation number.

$$\sigma = \left| \frac{\log(G)}{G} \times (x_i - x_{best}) \right| \quad (15)$$

After performing the aforementioned Gaussian Walks for G_{LS} generations, the worst 10 individuals from original population are selected as candidates for replacement. In case the new solutions are better than those selected individuals, the replacement occurs. After extensive experiments, G_{LS} is set to 250 generations as it was an enough exploitation window for the entire search process to advance the evolutionary search. Less than this is not enough to utilize the used individuals to find better results, and it was observed that using more than this number will not add to maximum utilization that can be obtained using this organization.

F. Putting it all together

The overall LSHADE-EpSin algorithm is shown in Fig. 3. Lines 1-3 represent the initialization of the population and set-up of the initial values for both settings. Lines 5-10 describe the first control settings as discussed in Sub-Section 1. On the other hand, lines 12-18 describe the sinusoidal settings which are used in the second half of evolutionary search. The current-to-pbest computation is invoked in lines 20-25. Next, the memory is updated and finally the linear population size reduction is computed in lines 27-31. When *NP* is decreased to 20, the local search method is called as described in Section III. E.

Algorithm: LSHADE-EpSin

1. Initialize population at first generation g_0, $P_{g_0} = \langle x_1^{g_0}, \ldots, x_{NP}^{g_0} \rangle$
2. Initialize memory of first control settings *M:* μF and μCR with 0.5
3. Initialize memory of second control settings M_{freq}: $\mu Freq$ with 0.5
4. **While** termination criterion is not met **Do**
5. **if** $g_i \in g_{s_1}$
6. **Call** first control parameter settings
7. Reset successful mean arrays: $S_F, S_{CR} = \phi$
8. Generate a random index $r_i = rand(1, H)$
9. Generate $F_{i,g_{s1}} = randc(\mu F_{r_i}, 0.1)$, $CR_{i,s_{g2}} = randn(\mu CR_{r_i}, 0.1)$
10. **EndIf**
11. **if** $g_i \in g_{s_2}$
12. **Call** second control parameter settings
13. Generate a random number $c = rand(0,1)$
14. **If** $c < 0.5$
15. Generate $F_{i,g_{s2}}$ using Non-Adaptive Decreasing Adjustment Eq. 8
16. **Else**
17. Generate $F_{i,g_{s2}}$ using Adaptive Increasing Adjustment Eq. 9
18. **EndIf**
19. **Generate** CR_i same as first control parameter setting using Eq. 4
20. **End**If
21. **For** *i*=1to *NP*
22. Generate $p_i = rand(0,1) \times N$, $N = NP \times 0.1$
23. Generate new trial vector $u_{i,g}$ using Eq. 2
24. Store successful F_i and CR_i for both control parameter settings
25. **EndFor**
26. Update memory as shown in Eqs. 5, 6 or 10 according to used settings
27. **Call** linear population size reduction
28. **Apply** Eq. 11 to calculate population size for next generation $NP(g+1)$
29. Calculate $NP_{diff} = NP(g) - NP(g+1)$
30. Sort individuals P_g based on functions values
31. Eliminate worst individuals NP_{diff} from P_g
32. If $NP \leq 20$ for the first time
33. Call the local search as described in Section III.E and Eq. 15
34. End
35. **EndWhile**

Fig. 3. Pseudo-code of LSHADE-EpSin algorithm

III. Experimental Results

A. Numerical benchmarks

The performance of the proposed LSHADE-EpSin algorithm is evaluated using a set of problems presented in the CEC2016 competition on real-parameter single objective optimization. In this competition, the CEC 2014 benchmark set was used again [22]. This benchmark contains 30 test functions with a diverse set of characteristics. *D* is the dimensionality of the problem and the functions are tested on 10*D*, 30*D*, 50*D* and 100*D*. In summary, functions 1–3 are unimodal, functions 4–16 are multimodal, functions 17–22 are multimodal, and functions 23–30 are composition functions. More details can be found in [22].

B. Algorithm parameters

The parameter values of LSHADE-EpSin algorithm are set as follows:

- The initial values of all μF , μCR are both set to 0.5. To save resources only one memory structure was used to store both control settings. For the first half of generations, μF and μCR are stored in a two dimensional memory. For the second half, the same two dimensional memory is used in which μF is replaced with $\mu freq$ which is initially set to 0.5. The Memory size *H* is set to 5. For non-adaptive decreasing adjustment, *freq* is set to 0.5.
- The $NP_{\max}$ and $NP_{\min}$ were set to $18*D$ and 4, respectively.
- G_{ls} is set to 250 generations.

C. Algorithm complexity

The algorithm complexity of LSHADE-EpSin is shown in Table 1. The algorithm was coded using Matlab 2013a and was run on a PC with an Intel CPU (3.40GHz) and 8GB RAM. The computational complexity of LSHADE-EpSin algorithm is calculated as described in [22]. In Table 1, T_0 denotes the running time of the following program:

$for\ i = 1:1000000$

$x = 0.55 + (double)i\,; x = x + x\,; x = x\ /\ 2; x = x * x\,;$

$x = sqrt(x); \ \ x = \log(x); x = \exp(x); x = x\ /\ (x + 2);$

end

TABLE I. ALGORITHM COMPLEXITY

	T_0	T_1	$\widehat{T}_2$	$(\widehat{T}_2 - T_1)/T_0$
$D = 10$	0.1598	0.6239	1.0867	2.8963
$D = 30$		1.0129	2.6938	10.5186
$D = 50$		1.2899	3.0572	11.0596
$D = 100$		3.5622	7.2315	22.9616

T_1 is the computing time for f_{18} for 200,000 evaluations while T_2 is the complete running time for the proposed algorithm of f_{18} for 200,000 evaluations. T_2 is evaluated five times, and the mean for T_2 is denoted as $\widehat{T}_2$. Finally, the algorithm complexity is reflected by $\widehat{T}_2, T_1$ and $(\widehat{T}_2 - T_1)/T_0$.

D. Statistical results

The algorithm was run 51 times for each test problem with a number of function evaluations equaling to $10,000 \times D$. When the difference between best solution found and the optimal became less than 10^{-8}, the error was treated as 0. The statistical results of LSHADE-EpSin on *D*=10, 30, 50 and 100 are presented in Tables II-V. Each table gives the best, worst, median, and the mean over the 51 runs of the error value between the best fitness values found in each run and the true optimal value.

TABLE II. STATISTICAL RESULTS OF THE 10-*D* BENCHMARK FUNCTIONS, AVERAGED OVER 51 INDEPENDENT RUNS

	Best	Worst	Median	Mean	Std
F1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F4	0.0000E+00	3.4780E+01	3.4780E+01	3.2052E+01	9.4437E+00
F5	1.6569E-02	2.0016E+01	2.0003E+01	1.4850E+01	8.4197E+00
F6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F7	0.0000E+00	7.3962E-03	0.0000E+00	1.8051E-04	1.0612E-03
F8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F9	9.9497E-01	2.9915E+00	1.9909E+00	1.9330E+00	7.0107E-01
F10	0.0000E+00	6.2454E-02	0.0000E+00	7.3476E-03	2.0322E-02
F11	3.2452E-01	1.2888E+02	1.4069E+01	2.0560E+01	2.4596E+01
F12	4.0197E-02	1.0358E-01	7.4259E-02	7.5656E-02	1.4981E-02
F13	1.5690E-02	7.0062E-02	4.7189E-02	4.6468E-02	1.3782E-02
F14	1.7210E-02	1.7469E-01	7.6356E-02	8.1530E-02	3.5048E-02
F15	2.1497E-01	5.0482E-01	3.7258E-01	3.6378E-01	6.6513E-02
F16	3.0522E-01	1.6150E+00	1.2080E+00	1.1149E+00	2.7315E-01
F17	2.0814E-01	1.3956E+02	1.1381E+01	2.5907E+01	4.0264E+01
F18	1.5491E-03	2.1303E+00	1.8641E-01	2.7592E-01	3.6252E-01
F19	1.7956E-04	1.0295E+00	5.8169E-02	3.0576E-01	4.1906E-01
F20	2.6910E-03	6.2262E-01	1.4565E-01	2.3282E-01	2.0030E-01
F21	1.0293E-06	1.1876E+02	4.8689E-01	3.4475E+00	1.6798E+01
F22	8.1822E-03	2.0125E+01	7.6810E-02	5.2820E-01	2.8032E+00
F23	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F24	1.0001E+02	1.0903E+02	1.0705E+02	1.0632E+02	2.3122E+00
F25	1.0000E+02	2.0000E+02	1.1997E+02	1.3394E+02	3.3295E+01
F26	1.0001E+02	1.0006E+02	1.0004E+02	1.0004E+02	1.5129E-02
F27	5.9437E-01	2.1031E+02	1.2551E+00	4.6958E+01	8.3701E+01
F28	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	3.5660E-12
F29	2.0000E+02	2.2149E+02	2.0000E+02	2.0042E+02	3.0098E+00
F30	2.0000E+02	5.5073E+02	4.6231E+02	3.7242E+02	1.3530E+02

E. Comparing LSHADE-EpSin with other state-of-the-art Differential Evolution algorithms.

In this subsection, the performance of the L-SHADE-EpSin algorithm is compared with the performances of other state-of-the-art Differential Evolution algorithms listed below:

1. Differential Evolution with Linear Population Size Reduction (L-SHADE) [14].
2. Differential Evolution with Success-History Based Parameter Adaptation (SHADE) [13].
3. Adaptive differential evolution with optional external archive (JADE) [12].
4. Differential evolution with an ensemble of parameters and mutation strategies (EPSDE) [23].

TABLE III. STATISTICAL RESULTS OF THE 30-*D* BENCHMARK FUNCTIONS, AVERAGED OVER 51 INDEPENDENT RUNS

	Best	Worst	Median	Mean	Std
F1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F5	2.0064E+01	2.0166E+01	2.0115E+01	2.0117E+01	2.3734E-02
F6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F9	8.3007E+00	1.6855E+01	1.3456E+01	1.3649E+01	1.9097E+00
F10	0.0000E+00	2.0819E-02	0.0000E+00	2.4493E-03	6.7745E-03
F11	5.0766E+02	1.4333E+03	1.1354E+03	1.1017E+03	2.0124E+02
F12	9.0452E-02	1.9381E-01	1.5326E-01	1.5350E-01	2.3631E-02
F13	8.4207E-02	1.7518E-01	1.3599E-01	1.3429E-01	1.9393E-02
F14	1.3105E-01	2.4732E-01	1.9718E-01	1.9505E-01	2.3545E-02
F15	1.8448E+00	2.8021E+00	2.3062E+00	2.3154E+00	2.6176E-01
F16	6.9044E+00	9.1498E+00	8.3866E+00	8.2656E+00	4.8441E-01
F17	3.6028E+01	4.3711E+02	1.6258E+02	1.6568E+02	1.0069E+02
F18	1.2607E+00	1.4727E+01	6.1858E+00	6.2483E+00	2.7992E+00
F19	9.6042E-01	4.4984E+00	2.7535E+00	2.6583E+00	7.5306E-01
F20	7.7056E-01	5.1596E+00	1.8508E+00	1.9705E+00	8.8628E-01
F21	1.4734E+00	2.6054E+02	2.7518E+01	7.3081E+01	7.6133E+01
F22	2.1885E+01	1.4827E+02	2.5286E+01	5.6744E+01	5.1241E+01
F23	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F24	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	5.5501E-10
F25	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F26	1.0009E+02	1.0016E+02	1.0013E+02	1.0013E+02	1.7214E-02
F27	2.0000E+02	3.0000E+02	2.0000E+02	2.0588E+02	2.3763E+01
F28	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F29	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F30	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00

TABLE IV. STATISTICAL RESULTS OF THE 50-*D* BENCHMARK FUNCTIONS, AVERAGED OVER 51 INDEPENDENT RUNS

	Best	Worst	Median	Mean	Std
F1	0.0000E+00	5.3889E-04	0.0000E+00	1.6673E-05	8.4186E-05
F2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F4	1.0231E-01	9.8103E+01	9.8103E+01	5.5906E+01	4.7552E+01
F5	2.0187E+01	2.0329E+01	2.0257E+01	2.0260E+01	3.0625E-02
F6	0.0000E+00	6.5417E-04	1.4309E-04	1.7911E-04	1.6705E-04
F7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F9	1.4389E+01	3.8197E+01	3.1306E+01	3.0212E+01	5.3749E+00
F10	2.3411E-04	1.5468E-01	4.7531E-02	4.9861E-02	2.7260E-02
F11	2.3819E+03	3.5074E+03	3.0914E+03	3.0260E+03	3.1360E+02
F12	1.2900E-01	2.7828E-01	2.0909E-01	2.1034E-01	3.2005E-02
F13	1.2780E-01	2.6077E-01	2.0620E-01	2.0613E-01	2.7022E-02
F14	1.2451E-01	2.2644E-01	1.9760E-01	1.9094E-01	2.2794E-02
F15	4.5970E+00	6.6382E+00	5.6322E+00	5.5699E+00	4.9977E-01
F16	1.4547E+01	1.7484E+01	1.6653E+01	1.6600E+01	5.2168E-01
F17	8.7307E+01	9.5573E+02	3.3013E+02	3.3412E+02	1.5841E+02
F18	6.3016E+00	3.7123E+01	1.9351E+01	1.9277E+01	5.9325E+00
F19	6.3442E+00	1.0757E+01	9.9610E+00	9.6846E+00	1.1238E+00
F20	2.3350E+00	1.0508E+01	5.9906E+00	6.0661E+00	2.0153E+00
F21	1.2305E+02	6.0722E+02	2.9739E+02	3.1607E+02	9.9913E+01
F22	2.7321E+01	1.8019E+02	6.7843E+01	1.0181E+02	5.9160E+01
F23	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F24	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	8.1552E-09
F25	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F26	1.0016E+02	1.0040E+02	1.0020E+02	1.0020E+02	3.4909E-02
F27	2.0000E+02	3.0000E+02	2.0000E+02	2.0196E+02	1.4003E+01
F28	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F29	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F30	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00

5. Differential Evolution with Reduced population size (DynNP-jDE) [21].
6. Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters (CoDE) [16].

The same control parameter values that were suggested in the original papers were used to run the tests. Each algorithm was run for 10,000 x D functions evaluations for each of 51 independent runs.

TABLE V. STATISTICAL RESULTS OF THE 100-*D* BENCHMARK FUNCTIONS, AVERAGED OVER 51 INDEPENDENT RUNS

	Best	Worst	Median	Mean	Std
F1	1.4067E+03	6.0001E+04	1.3793E+04	1.6397E+04	1.1918E+04
F2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F4	1.0739E+02	2.1578E+02	2.0056E+02	1.7927E+02	3.2404E+01
F5	2.0447E+01	2.0624E+01	2.0551E+01	2.0547E+01	3.9744E-02
F6	8.3677E-03	1.5217E+00	2.0110E-02	3.9551E-01	5.2589E-01
F7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F8	4.9488E-04	1.3743E-02	2.3073E-03	2.8971E-03	2.2582E-03
F9	3.7251E+01	1.1503E+02	5.1825E+01	5.4639E+01	1.3789E+01
F10	1.1800E+01	3.0260E+01	1.8646E+01	1.8410E+01	3.5915E+00
F11	8.7917E+03	1.1357E+04	1.0435E+04	1.0404E+04	5.3786E+02
F12	3.2916E-01	5.3862E-01	4.3175E-01	4.2820E-01	4.1276E-02
F13	2.5499E-01	3.6588E-01	3.2033E-01	3.1769E-01	2.5863E-02
F14	1.4808E-01	1.9984E-01	1.7691E-01	1.7652E-01	1.0707E-02
F15	1.4271E+01	1.8575E+01	1.6823E+01	1.6789E+01	8.9663E-01
F16	3.7150E+01	3.9603E+01	3.8473E+01	3.8495E+01	6.1145E-01
F17	1.1334E+03	3.2310E+03	1.9474E+03	2.0153E+03	4.9847E+02
F18	5.2099E+01	1.4562E+02	9.5338E+01	9.5386E+01	2.1281E+01
F19	8.6132E+01	9.2605E+01	8.9228E+01	8.9191E+01	1.1610E+00
F20	1.2669E+01	3.5324E+01	2.1026E+01	2.1154E+01	4.3101E+00
F21	2.1265E+02	1.2260E+03	5.7891E+02	6.1724E+02	2.4514E+02
F22	5.9466E+02	1.6461E+03	1.0774E+03	1.0729E+03	2.0730E+02
F23	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F24	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	1.1744E-09
F25	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F26	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F27	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	1.9773E-04
F28	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F29	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00
F30	2.0000E+02	2.0000E+02	2.0000E+02	2.0000E+02	0.0000E+00

Table VI gives the mean and standard deviation of the best-of-run errors for the 51 independent runs of all used algorithms on 30 benchmark problems for 50*D*. The best error value is marked in bold. This table shows that the LSHADE-EpSin obtained a performance that is at least as good as the other algorithms in 23 out of the 30 functions. LSHADE-EpSin performed equal to or better than the winner of the 2014 competition (L-SHADE), which has the best results in the literature for the 30 CEC2014 functions. This affirms the robustness of LSHADE-EpSin algorithm and the effectiveness of using the proposed sinusoidal ensemble pool of control settings.

Moreover, the Wilcoxon's rank-sum test was used to judge the significance of the results [24]. Table VII summarizes the results of this test at a 0.05 significance level comparing each algorithm versus LSHADE-EpSin on the 30 functions for *D*=30, 50 and 100 . For each of the competitive functions in this table, "–" identifies the number of losses to LSHADE-EpSin, "+" marks the instances where the contestant algorithm exhibits a superior performance to LSHADE-EpSin, and "=" indicates that the performance difference between the competitor and LSHADE-EpSin is not statistically significant. As shown in this table, LSHADE-EpSin clearly has the best overall performance among all the variants and significantly improves upon the performance of L-SHADE algorithm as the number of dimensions increase. It also demonstrates the efficiency of using the adaptive sinusoidal adjustments to adapt the scaling factor by providing a good balance between the exploitation of the already best solutions and exploration of new regions.

TABLE VI. MEAN AND STANDARD DEVIATION OF THE ERROR VALUES FOR FUNCTIONS F1-F30 AVERAGED OVER 51 RUNS @ 50D. BEST ENTRIES ARE MARKED IN BOLDFACE.

	CoDE	dynNP-jDE	SaDE	JADE	SHADE	L-SHADE	LSHADE-EpSin
F1	2.3214E+05	3.2264E+05	9.3147E+05	1.5275E+04	1.8614E+04	5.4839E+02	**1.6673E-05**
	(1.0747E+05)	(1.1734E+05)	(3.1419E+05)	(1.3422E+04)	(1.3638E+04)	(9.3359E+02)	**(8.4186E-05)**
F2	6.5010E+01	3.3251E-07	3.7564E+03	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	(1.9284E+02)	(1.4702E-06)	(3.8115E+03)	**(0.0000E+00)**	**(0.0000E+00)**	**(0.0000E+00)**	**(0.0000E+00)**
F3	2.5387E+01	7.0324E-06	3.4631E+03	4.5134E+03	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	(4.1188E+01)	(4.7809E-05)	(2.2424E+03)	(2.3441E+03)	**(0.0000E+00)**	**(0.0000E+00)**	**(0.0000E+00)**
F4	2.6067E+01	8.1654E+01	8.2104E+01	1.9346E+01	**9.7224E+00**	4.3999E+01	5.5906E+01
	(3.3789E+01)	(2.1947E+01)	(4.4460E+01)	(3.9290E+01)	**(2.9435E+01)**	(4.7625E+01	(4.7552E+01)
F5	**2.0032E+01**	2.0384E+01	2.0733E+01	2.0356E+01	2.0140E+01	2.0262E+01	2.0260E+01
	(6.8626E-02)	(2.9280E-02)	(4.1549E-02)	(3.6845E-02)	(1.9410E-02)	(3.0283E-02)	(3.0625E-02)
F6	8.4983E+00	7.3352E+00	1.8148E+01	1.6589E+01	5.1707E+00	2.8208E-01	**1.7911E-04**
	(3.3570E+00)	(5.6880E+00)	(3.2634E+00)	(6.6285E+00)	(2.4284E+00)	(5.2398E-01)	**(1.6705E-04)**
F7	1.5467E-03	**0.0000E+00**	1.3887E-02	2.0769E-03	3.9093E-03	**0.0000E+00**	**0.0000E+00**
	(3.1935E-03)	**(0.0000E+00)**	(1.5352E-02)	(4.9084E-03)	(7.4077E-03)	**(0.0000E+00)**	**(0.0000E+00)**
F8	4.8773E-01	**0.0000E+00**	1.0925E+00	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	(8.2941E-01)	**(0.0000E+00)**	(1.0945E+00)	**(0.0000E+00)**	**(0.0000E+00)**	**(0.0000E+00)**	**(0.0000E+00)**
F9	8.2582E+01	7.4286E+01	9.1673E+01	5.1867E+01	3.4182E+01	**1.1732E+01**	3.0212E+01
	(1.7230E+01)	(9.3923E+00)	(1.4268E+01)	(8.6673E+00)	(6.1592E+00)	**(2.5019E+00)**	(5.3749E+00)
F10	5.1856E+00	**6.8582E-03**	1.2284E+00	1.2247E-02	1.0287E-02	5.2028E-02	4.9861E-02
	(3.0648E+00)	**(9.7767E-03)**	(1.0759E+00)	(1.3099E-02)	(1.1368E-02)	(2.4097E-02)	(2.7260E-02)
F11	4.3391E+03	4.3297E+03	6.8299E+03	3.8695E+03	3.5552E+03	3.3013E+03	**3.0260E+03**
	(9.4713E+02)	(3.8575E+02)	(1.5764E+03)	(2.7785E+02)	(3.2722E+02)	(3.3270E+02)	**(3.1360E+02)**
F12	**8.7329E-02**	3.6401E-01	1.0986E+00	2.5535E-01	1.6366E-01	2.1362E-01	2.1034E-01
	(4.5691E-02)	(4.8652E-02)	(1.0537E-01)	(3.1321E-02)	(2.1190E-02)	(2.5551E-02)	(3.2005E-02)
F13	3.3636E-01	3.4874E-01	4.3681E-01	3.3223E-01	3.1034E-01	**1.6635E-01**	2.0613E-01
	(4.9736E-02)	(4.1091E-02)	(5.4721E-02)	(4.9590E-02)	(4.5483E-02)	**(1.7505E-02)**	(2.7022E-02)
F14	2.7652E-01	3.0009E-01	3.1170E-01	3.1348E-01	2.9453E-01	2.6990E-01	**1.9094E-01**
	(2.9839E-02)	(2.6630E-02)	(3.2642E-02)	(9.1988E-02)	(5.7219E-02)	(1.8936E-02)	**(2.2794E-02)**
F15	7.4043E+00	1.0077E+01	1.7092E+01	7.0826E+00	5.7376E+00	**5.1011E+00**	**5.5699E+00**
	(1.3955E+00)	(1.1616E+00)	(5.8042E+00)	(9.4236E-01)	(6.1481E-01)	**(4.1540E-01)**	**(4.9977E-01)**
F16	1.8290E+01	1.7551E+01	2.0222E+01	1.7720E+01	1.7444E+01	**1.6816E+01**	**1.6600E+01**
	(8.9113E-01)	(4.6595E-01)	(2.7357E-01)	(4.3341E-01)	(4.2728E-01)	**(4.6135E-01)**	**(5.2168E-01)**
F17	1.5714E+04	1.2583E+04	5.9118E+04	2.3000E+03	2.2347E+03	1.4640E+03	**3.3412E+02**
	(1.3291E+04)	(8.3862E+03)	(3.9220E+04)	(5.9502E+02)	(7.8109E+02)	(3.6469E+02)	**(1.5841E+02)**
F18	3.2916E+02	2.5984E+02	7.5884E+02	1.8203E+02	1.5375E+02	1.0085E+02	**1.9277E+01**
	(3.4839E+02)	(2.4692E+02)	(5.1263E+02)	(4.7150E+01)	(3.6617E+01)	(1.7047E+01)	**(5.9325E+00)**
F19	**6.1691E+00**	1.0832E+01	1.6741E+01	1.3421E+01	9.3810E+00	8.4924E+00	9.6846E+00
	(9.8380E-01)	(8.4404E-01)	(8.7552E+00)	(5.2199E+00)	(3.2466E+00)	(2.0870E+00)	(1.1238E+00)
F20	2.1861E+02	3.9163E+01	9.0611E+02	7.9529E+03	1.9816E+02	5.1411E+01	**6.0661E+00**
	(2.3515E+02)	(1.3337E+01)	(7.3143E+02)	(6.9640E+03)	(5.7311E+01)	(2.0828E+01)	**(2.0153E+00)**
F21	6.4912E+03	2.7281E+03	8.5745E+04	2.4322E+04	1.2297E+03	6.9834E+02	**3.1607E+02**
	(5.5372E+03)	(2.5200E+03)	(4.5410E+04)	(1.6455E+05)	(4.0604E+02)	(2.3873E+02)	**(9.9913E+01)**
F22	6.0223E+02	4.2383E+02	5.1001E+02	5.1982E+02	3.6769E+02	**1.0146E+02**	**1.0181E+02**
	(2.2061E+02)	(1.2826E+02)	(1.7759E+02)	(1.3024E+02)	(1.5946E+02)	**(5.6017E+01)**	**(5.9160E+01)**
F23	3.4400E+02	3.4400E+02	3.4400E+02	3.4400E+02	3.4400E+02	3.4400E+02	**2.0000E+02**
	(1.1482E-13)	(1.1482E-13)	(1.1482E-13)	(3.7646E-13)	(1.1482E-13)	(3.5662E-13)	**(0.0000E+00)**
F24	2.7139E+02	2.6546E+02	2.7596E+02	2.7482E+02	2.7478E+02	2.8547E+02	**2.0000E+02**
	(2.9312E+00)	(1.6705E+00)	(3.0905E+00)	(1.9487E+00)	(1.7334E+00)	(1.9879E+00)	**(8.1552E-09)**
F25	2.0792E+02	2.0691E+02	2.1665E+02	2.1699E+02	2.1217E+02	2.0530E+02	**2.0000E+02**
	(4.7505E+00)	(1.3449E+00)	(9.7720E+00)	(6.7636E+00)	(6.9165E+00)	(3.3379E-01)	**(0.0000E+00)**
F26	1.0618E+02	1.0035E+02	1.9030E+02	1.0237E+02	1.0431E+02	**1.0017E+02**	**1.0020E+02**
	(2.3701E+01)	(4.1976E-02)	(2.9941E+01)	(1.3950E+01)	(1.9529E+01)	**(1.9297E-02)**	**(3.4909E-02)**
F27	5.3680E+02	3.8452E+02	7.8213E+02	4.4058E+02	4.5329E+02	3.0884E+02	**2.0196E+02**
	(6.5540E+01)	(4.8217E+01)	(6.2606E+01)	(5.6802E+01)	(5.6995E+01)	(1.3199E+01)	**(1.4003E+01)**
F28	1.1734E+03	1.1052E+03	1.4354E+03	1.1289E+03	1.1413E+03	9.7358E+02	**2.0000E+02**
	(4.4835E+01)	(3.8147E+01)	(9.8737E+01)	(4.1317E+01)	(5.2616E+01)	(8.3564E+01)	**(0.0000E+00)**
F29	9.2726E+02	9.9798E+02	1.4224E+03	9.0482E+02	8.9071E+02	7.4146E+02	**2.0000E+02**
	(1.3813E+02)	(1.4277E+02)	(3.1715E+02)	(1.2689E+02)	(5.5375E+01)	(4.4718E+01)	**(0.0000E+00)**
F30	9.0213E+03	8.4860E+03	1.1613E+04	9.7197E+03	9.3841E+03	1.1829E+04	**2.0000E+02**
	(5.3860E+02)	(4.2315E+02)	(1.8321E+03)	(8.1792E+02)	(7.8714E+02)	(4.1923E+02)	**(0.0000E+00)**

IV. CONCLUSION

This paper proposed LSHADE-EpSin which is a modified version of the L-SHADE algorithm. The algorithm introduces a self-adaptive framework in order to adapt the control settings during the search. It uses a new ensemble of adaptive sinusoidal approaches in order to adjust the values of the scaling factor automatically. This ensemble approach consists of a mixture of two sinusoidal formulas: the Non-Adaptive Sinusoidal decreasing Adjustment, and the Adaptive History-based Sinusoidal Increasing Adjustment. The proposed algorithm was tested on the benchmarks of the CEC2014 which is used in the Special Session and Competitions on Real-Parameter Single Objective Optimization of the IEEE CEC2016. When compared with other state-of-the-art DE algorithm taken from the literature, it shows a very competitive performance with fast convergence.

TABLE VII. COMPARISON OF LSHADE-ESIN WITH STATE-OF-THE-ART DE ALGORITHMS ON D = 30, 50, 100 USING THE WILCOXON RANK-SUM TEST (SIGNIFICANTLY, $P < 0.05$)

Vs. LSHADE-EpSin		D = 30	D = 50	D = 100
CoDE	+ (win)	1	2	1
	- (lose)	19	23	25
	= (equal)	10	5	4
dynNP-jDE	+ (win)	0	0	3
	- (lose)	20	22	22
	= (equal)	10	8	5
SaDE	+ (win)	0	0	1
	- (lose)	27	29	26
	= (equal)	3	1	3
JADE	+ (win)	0	1	3
	- (lose)	18	21	22
	= (equal)	12	8	5
SHADE	+ (win)	0	1	4
	- (lose)	19	19	20
	= (equal)	11	10	6
L-SHADE	+ (win)	3	2	2
	- (lose)	12	15	18
	= (equal)	15	13	10

Acknowledgment

This research was supported by funding from the Agency for Science, Technology and Research of Singapore (A*STAR) and Nanyang Technological University. Dr. Reynolds also would like to thank NSF Grant for their support under the number: CCF 1451316. This research was also supported by the deanship of research in Jordan University of Science and Technology (JUST). Dr. Mostafa Z. Ali accomplished his major part in this work during his Sabbatical leave from JUST with PSUT, using this fund.

References

[1] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," J. Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[2] M. Lozano, D. Molina, F. Herrera, "Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems," Soft Comput.: Fusion Found., Methodologies Applicat., 15 (2011)11, 2085–2087.

[3] R. Joshi, A.C. Sanderson, "Minimal representation multisensor fusion using differential evolution," IEEE Trans. Syst., Man Cybern. Part A, vol. 29, no. 1, pp. 63–76, Jan. 1999.

[4] S.-M. Guo, J.S.-H. Tsai, C.-C. Yang, 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 Proc. IEEE Congr Evol Comput, pp. 1003 - 1010, 2015

[5] M.Z. Ali, N.H. Awad, P.N. Suganthan, "Multi-population Differential Evolution with Balanced Ensemble of Mutation Strategies for Large-Scale Global Optimization," Applied Soft Computing, V. 33, pp. 304-327, 2015

[6] N.H. Awad, M.Z. Ali, R.G. Reynolds, "A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization," In Proc. IEEE Congr Evol Comput, pp. 1098 - 1105, 2015

[7] M.Z. Ali, N.H. Awad, R. Duwairi, J.Albadarneh, R.G. Reynolds, P.N. Suganthan, "Cluster-Based Differential Evolution with Heterogeneous Influence for numerical optimization," In Proc. IEEE Congr Evol Comput, PP. 393-400, 2015

[8] J. Zhang, A.C. Sanderson, "An approximate Gaussian model of differential evolution with spherical fitness functions," in Proc. IEEE Congr. Evol. Comput., Singapore, Sep. 2007, pp. 2220–2228.

[9] R. Gamperle, S. D. Muller, P. Koumoutsakos, "A parameter study for differential evolution," in Proc. Advances Intell. Syst., Fuzzy Syst.,Evol. Comput., Crete, Greece, 2002, pp. 293–298.

[10] R. Mallipeddi, P. N. Suganthan, "Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies, in Proc. Swarm Evolutionary and Memetic Computing Conference, 2010, pp. 71-78

[11] A. K. Qin, P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in Proc. IEEE Congr. Evol. Comput., vol. 2. Sep. 2005, pp. 1785–1791.

[12] J. Zhang, A.C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," IEEE Trans Evol Comput, vol. 13, no. 5, pp. 945–958, 2009

[13] R. Tanabe, A. Fukunaga, "Success-History Based Parameter Adaptation for Differential Evolution," in IEEE CEC, 2013, pp. 71–78.

[14] R. Tanabe, A. Fukunaga, "Improving the Search Performance of SHADE Using Linear Population Size Reduction," in IEEE CEC, 2014, pp. 1658 - 1665.

[15] S.-Z. Zhao, P. N. Suganthan, S. Das, "Self-adaptive differential evolution with multi-trajectory search for large-scale optimization," Soft Comput.: Fusion Found., Methodologies Applicat., 15(2011) 11, 2175–2185.

[16] Y. Wang, Z. Cai, Q. Zhang, "Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters," IEEE Tran. Evol. Comput., vol. 15, no. 1, pp. 55–66, 2011.

[17] A. LaTorre, S. Muelas, J.-M. Pena, "A mos-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test," Soft Comput.: Fusion Found., Methodologies Applicat., 15(2011) 11, 2187–2199.

[18] M. Ankush, A. K. Das, P. Mukherjee, S. Das and P. N. Suganthan, "Modified Differential Evolution with Local Search Algorithm for Real World Optimization," in Proc. IEEE Congr Evol Comput, 2011, pp. 1565-1572.

[19] http://www.ntu.edu.sg/home/epnsugan/

[20] A. Draa, S. Bouzoubia, I.Boukhalfa, "A sinusoidal differential evolution algorithm for numerical optimisation," Applied Soft Computing, V. 27, I. C, P. 99-126, 2015

[21] J. Brest and M. S. Maucec, "Population size reduction for the differential evolution algorithm," Appl. Intell., vol. 29, no. 3, pp. 228–247, 2008.

[22] J. J. Liang, B. Y. Qu, P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," Zhengzhou University and Nanyang Technological University, Tech. Rep., 2013.

[23] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," Appl. Soft Comput., vol. 11, no. 2, pp. 1679–1696, 2011.

[24] J. Derrac, S. Garcia, D. Molina, F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,"Swarm Evol. Comput., vol. 1, no. 1, pp. 3–18, Mar. 2011