Distance Based Parameter Adaptation for Success-History based Differential Evolution

Adam Viktorin[a], Roman Senkerik[a], Michal Pluhacek[a], Tomas Kadavy[a], Aleš Zamuda[b,*]

[a]*Faculty of Applied Informatics, Tomas Bata University in Zlin, T. G. Masaryka 5555, 760 01 Zlin, Czech Republic*
[b]*Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, 2000 Maribor, Slovenia*

Abstract

This paper proposes a simple, yet effective, modification to scaling factor and crossover rate adaptation in Success-History based Adaptive Differential Evolution (SHADE), which can be used as a framework to all SHADE-based algorithms. The performance impact of the proposed method is shown on the real-parameter single objective optimization (CEC2015 and CEC2017) benchmark sets in 10, 30, 50, and 100 dimensions for all SHADE, L-SHADE (SHADE with linear decrease of population size), and jSO algorithms. The proposed distance based parameter adaptation is designed to address the premature convergence of SHADE–based algorithms in higher dimensional search spaces to maintain a longer exploration phase. This design effectiveness is supported by presenting a population clustering analysis, along with a population diversity measure. Also, the new distance based algorithm versions (Db_SHADE, DbL_SHADE, and DISH) have obtained significantly better optimization results than their canonical counterparts (SHADE, L_SHADE, and jSO) in 30, 50, and 100 dimensional functions.

Keywords: Differential Evolution, Distance Based, Parameter Adaptation, Success-History, Scaling Factor, Crossover Rate

1. Introduction

The Differential Evolution (DE) algorithm was developed in 1995 by Storn and Price [1] and formed a basis for a family of successful algorithms for continuous optimization. The broad research field around DE was summarized most recently in [2].

The continuing research in the DE area provides a plethora of improvements to the original algorithm, and their quality can be tracked easily by following the results of a continuous optimization competition jointly with Congress on Evolutionary Computation (CEC), where DE based algorithms are placed steadily on the top [3, 4, 5, 6, 7, 8, 9, 10].

A common denominator for the best DE variants [11] since 2013 is an algorithm developed by Tanabe and Fukunaga – Success-History based Adaptive Differential Evolution (SHADE) [12].

*Corresponding author
Email addresses: aviktorin@utb.cz (Adam Viktorin), senkerik@utb.cz (Roman Senkerik), pluhacek@utb.cz (Michal Pluhacek), kadavy@utb.cz (Tomas Kadavy), ales.zamuda@um.si (Aleš Zamuda)
URL: ailab.fai.utb.cz (Adam Viktorin)

This algorithm adapts its control parameters, crossover rate *CR* and scaling factor *F*, during the optimization process, in order to suit the given problem, and it also incorporates "current-to-*p*best/1" mutation strategy with external archive of inferior solutions from JADE [13]. The SHADE algorithm was placed 3^{rd} in the CEC2013 competition, and the authors presented an improved version the following year. This version added linear decrease in population size, and the algorithm was titled L-SHADE [14]. L-SHADE won the competition in CEC2014, and the winners of the following years, SPS-L-SHADE-EIG – winner CEC2015 [15], LSHADE_EpSin – joint winner CEC2016 [16], jSO – announced winner at CEC2017 [17], were all based on this algorithm. The versions of SHADE, especially those referred to as L-SHADE, are probably among the most effective evolutionary algorithms [18] and, with exception of the jSO, some of its other variants have already benefited from general enhancements [19]. While new enhancements of these DE algorithms add new mechanism or tune parameters, similar as it has been seen in other optimization algorithms [20], and there are more aspects addressed by different DE application domains, as covered in respective substantial surveys on these domains [21, 22, 23, 24, 25, 26, 27, 18]. Furthermore, theoretical analyses in support of DE also exist, like [28, 29, 30, 31].

All previously mentioned algorithms share the same idea of balancing between exploration and exploitation abilities, and they try to achieve it in various ways. As parameter control is one of the important aspects in evolutionary algorithms [32], and being highlighted experimentally in DE as well [26], the focus of this paper will be on improvement and analysis of an aspect of DE parameter control, specifically the SHADE variants that are currently leading in the DE progress [11].

The motivation behind this research is following two ideas. Firstly, the main aim is to address the premature convergence issue of the SHADE based family of algorithms through the novel distance based parameter adaptation, securing a longer exploration phase mainly in the higher dimensional search space. Secondly, such an adaptation/enhancement should be simple and should not increase the complexity, and not make the understandability and applicability of the proposed algorithm more difficult, as is discussed in the paper [33]. In this paper[1], therefore, a simple update is proposed to the SHADE's original scaling factor and crossover rate adaptation. This modification rewards exploration capabilities more, and it is analyzed experimentally in this paper and shown to be more suitable, especially for problems in higher dimensions using the CEC2015 and CEC2017 benchmarks. The difference between the conference paper [34] and this presented research is as follows: More dimensional settings and two benchmarks suites are used, jSO – as the announced winner of the CEC2017 competition [17] is included in all experiments, and, last but not least, the original population clustering analysis, along with the population diversity measure, are present, supporting the ideas and obtained data concerning the prolonged exploration phase of all used and modified algorithms.

The following Section 2 describes the path from DE algorithm to the SHADE, L-SHADE and jSO, then, distance based parameter adaptation is presented in Section 3. Sections 4 and 5 describe experimental settings and results, respectively. Section 6 provides results discussion and the paper is concluded in Section 7.

[1]Part of this paper was presented at SSCI 2017 conference [34].

2. DE and SHADE

In order to describe the Db_SHADE algorithm, it is important to start from the canonical Differential Evolution (DE) by Storn and Price [1].

The canonical 1995 DE is based on the idea of evolution from a randomly generated set of solutions of the optimization task called population $\boldsymbol{P}$, which has a preset size of NP. Each individual (solution) in the population consists of a vector $\boldsymbol{x}$ of length D (each vector component corresponds to one attribute of the optimized task) and objective function value $f(\boldsymbol{x})$, which mirrors the quality of the solution. The number of optimized attributes D is often referred to as the dimensionality of the problem, and such generated population $\boldsymbol{P}$, represents the first generation of solutions.

The individuals in the population are combined in an evolutionary manner in order to create improved offspring for the next generation. This process is repeated until the stopping criterion is met (either the maximum number of generations, or the maximum number of objective function evaluations, or the population diversity lower limit, or overall computational time), creating a chain of subsequent generations, where each following generation consists of better solutions than those in previous generations – a phenomenon called elitism.

The combination of individuals in the population consists of three main steps: Mutation, crossover, and selection.

In mutation, the attribute vectors of selected individuals are combined in simple vector operations to produce a mutated vector $\boldsymbol{v}$. This operation uses a control parameter — scaling factor F. In the crossover step, a trial vector $\boldsymbol{u}$ is created by selection of attributes either from mutated vector $\boldsymbol{v}$ or the original vector $\boldsymbol{x}$, based on the crossover probability given by the control parameter — crossover rate CR, and finally, in the selection, the fitness $f(\boldsymbol{u})$ of a trial vector is evaluated by an objective function and compared to the fitness $f(\boldsymbol{x})$ of the original vector and the better one is placed into the next generation.

From the basic description of the DE algorithm, it can be seen that there are three control parameters, which have to be set by the user – population size NP, scaling factor F, and crossover rate CR. It was shown in [2, 3], that the setting of these parameters is crucial for the performance of DE. Fine-tuning of the control parameter values is a time-consuming task and, therefore, many state-of-the-art DE variants use self-adaptation in order to avoid this cumbersome task. This is also the case of the SHADE algorithm proposed by Tanabe and Fukunaga in 2013 [12] and since it is used in this paper, the algorithm is described in more detail in the next subsection, along with the novel distance based parameter adaptation.

2.1. SHADE

As already mentioned, the SHADE algorithm was proposed with a self-adaptive mechanism of some of its control parameters in order to avoid their fine-tuning. The control parameters in question are scaling factor F and crossover rate CR. It is fair to mention that the SHADE algorithm is based on Zhang and Sanderson's JADE [16] and shares a lot of its mechanisms [11]. The main difference is in the historical memories $\mathbf{M}_\mathrm{F}$ and $\mathbf{M}_\mathrm{CR}$ for successful scaling factor and crossover rate values with their update mechanism.

The following subsections describe individual steps of the SHADE algorithm: Initialization, mutation, crossover, selection, and historical memory updates, followed by the algorithm's important extension, the linear decrease in population size.

2.1.1. Initialization

The initial population $\boldsymbol{P}$ is generated randomly and for that matter, a Pseudo-Random Number Generator (PRNG) is used with uniform distribution is used. Solution vectors $\boldsymbol{x}$ are generated according to the limits of the solution space – *lower* and *upper* bounds:

$$\boldsymbol{x}_{j,i} = \mathcal{U}\left[lower_j,\ upper_j\right]; \forall j = 1,\ \ldots,\ D; \forall i = 1,\ \ldots,\ NP, \tag{1}$$

where i is the individual index, j is the attribute index, and $\mathcal{U}$ uniform random distribution. The dimensionality of the problem is represented by D, and NP stands for the population size.

Historical memories are preset to contain only 0.5 values for both, scaling factor and crossover rate parameters:

$$M_{\mathrm{CR},i} = M_{\mathrm{F},i} = 0.5; \forall i = 1, \ldots, H, \tag{2}$$

where H is a user-defined size of historical memories.

Also, the external archive of inferior solutions $\boldsymbol{A}$ has to be initialized. Because of no previous inferior solutions, it is initialized empty, $\boldsymbol{A} = \emptyset$, and index k for historical memory updates is initialized to 1.

The following steps are repeated over the generations until the stopping criterion is met.

2.1.2. Mutation

In the original DE mutation step, three mutually different individuals $\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$, $\boldsymbol{x}_{r3}$ are selected randomly from the population and combined. This mutation strategy is titled “rand/1”:

$$\boldsymbol{v}_i = \boldsymbol{x}_{r1} + F\left(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}\right), \tag{3}$$

where $r1 \neq r2 \neq r3 \neq i$, F is the scaling factor value and $\boldsymbol{v}_i$ is the resulting mutated vector.

On the other hand, SHADE’s mutation strategy “current-to-*p*best/1”, introduced in [16], combines four index-wise mutually different vectors in computation of the mutated vector $\boldsymbol{v}$, with the index of $\boldsymbol{x}_{p\mathrm{best}}$ being different from $r1$, $r2$, and i, as:

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F_i\left(\boldsymbol{x}_{p\mathrm{best}} - \boldsymbol{x}_i\right) + F_i\left(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}\right), \tag{4}$$

where $\boldsymbol{x}_{p\mathrm{best}}$ is a randomly selected individual from the best $NP \times p$ individuals in the current population. The p value is generated randomly for each mutation by PRNG, with uniform distribution from the range $[p_{\mathrm{min}}, 0.2]$ and $p_{\mathrm{min}} = 2/NP$. Vector $\boldsymbol{x}_{r1}$ is selected randomly from the current population $\boldsymbol{P}$. Vector $\boldsymbol{x}_{r2}$ is randomly randomly from the union of the current population $\boldsymbol{P}$ and external archive $\boldsymbol{A}$. The scaling factor value F_i is given by:

$$F_i = C\left[M_{\mathrm{F},r}, 0.1\right], \tag{5}$$

where $M_{\mathrm{F},r}$ is a randomly selected value (index r is generated by PRNG from the range 1 to H) from $\mathbf{M}_{\mathrm{F}}$ memory, and C stands for Cauchy distribution. Therefore the F_i value is generated from the Cauchy distribution with location parameter value $\mathbf{M}_{\mathrm{r}}$ and scale parameter value of 0.1. If the generated value F_i is higher than 1, it is truncated to 1, and if F_i is less or equal to 0, it is generated again by (5).

2.1.3. Crossover

In the crossover step, trial vector $\boldsymbol{u}$ is created from the mutated $\boldsymbol{v}$ and original $\boldsymbol{x}$ vectors. For each vector component, a PRNG with uniform distribution is used to generate a random value. If this random value is less or equal to given crossover rate value CR_i, the current vector component will be taken from a trial vector, otherwise, it will be taken from the original vector (6). There is also a safety measure, which ensures that at least one vector component will be taken from the trial vector. This is given by a randomly generated component index j_{rand}:

$$u_{j,i} = \begin{cases} v_{j,i} & \text{if } \mathcal{U}[0,1] \le CR_i \text{ or } j = j_{\text{rand}} \\ x_{j,i} & \text{otherwise} \end{cases} . \tag{6}$$

The crossover rate value CR_i is generated from a Gaussian distribution with a mean parameter value $M_{\text{CR},r}$ selected from the crossover rate historical memory $\mathbf{M}_{\text{CR}}$ by the same index r as in the scaling factor case and Standard Deviation value of 0.1:

$$CR_i = \mathcal{N}[M_{CR,r}, 0.1]. \tag{7}$$

When the generated CR_i value is less than 0, it is replaced by 0, and when it is greater than 1, it is replaced by 1.

2.1.4. Selection

The selection step ensures, that the optimization will progress towards better solutions, because it allows only individuals of better or at least equal objective function value to proceed into the next generation *G+1*:

$$\boldsymbol{x}_{i,G+1} = \begin{cases} \boldsymbol{u}_{i,G} & \text{if } f(\boldsymbol{u}_{i,G}) \le f(\boldsymbol{x}_{i,G}) \\ \boldsymbol{x}_{i,G} & \text{otherwise} \end{cases} , \tag{8}$$

where G is the index of the current generation.

2.1.5. Historical Memory Updates

Historical memories $\mathbf{M}_{\text{F}}$ and $\mathbf{M}_{\text{CR}}$ are initialized according to (2), but their components change during the evolution. These memories serve to hold successful values of F and CR used in the mutation and crossover steps, successful in terms of producing a trial individual better than the original individual. During every single generation, these successful values are stored in their corresponding arrays $\mathbf{S}_F$ and $\mathbf{S}_{CR}$. After each generation, one cell of $\mathbf{M}_{\text{F}}$ and $\mathbf{M}_{\text{CR}}$ memories is updated. This cell is given by the index k, which starts at 1 and increases by 1 after each generation. When it overflows the memory size H, it is reset to 1. The new values of the k-th cell for $\mathbf{M}_{\text{F}}$ and $\mathbf{M}_{\text{CR}}$ are calculated, respectively:

$$M_{F,k} = \begin{cases} \text{mean}_{WL}(\mathbf{S}_F) & \text{if } \mathbf{S}_F \neq \emptyset \\ M_{F,k} & \text{otherwise} \end{cases} , \tag{9}$$

$$M_{CR,k} = \begin{cases} \text{mean}_{WL}(\mathbf{S}_{CR}) & \text{if } \mathbf{S}_{CR} \neq \emptyset \\ M_{CR,k} & \text{otherwise} \end{cases} , \tag{10}$$

where $\text{mean}_{WL}()$ stands for weighted Lehmer mean:

$$\text{mean}_{WL}(\mathbf{S}) = \frac{\sum_{k=1}^{|\mathbf{S}|} w_k \bullet S_k^2}{\sum_{k=1}^{|\mathbf{S}|} w_k \bullet S_k} \tag{11}$$

and the weight vector $\boldsymbol{w}$ is based on the improvement in objective function value between trial and original individuals in current generation G, as follows:

$$w_k = \frac{\text{abs}\left(f\left(\boldsymbol{u}_{k,G}\right) - f\left(\boldsymbol{x}_{k,G}\right)\right)}{\sum_{m=1}^{|\mathbf{S}_{CR}|} \text{abs}\left(f\left(\boldsymbol{u}_{m,G}\right) - f\left(\boldsymbol{x}_{m,G}\right)\right)}. \tag{12}$$

Because both arrays $\mathbf{S}_F$ and $\mathbf{S}_{CR}$ have the same size, it is arbitrary which size will be used for the upper boundary for m in Eq. (12).

The pseudo-codes for both algorithms are shown as Algorithm 1 and Algorithm 2 for DE and SHADE algorithms, respectively.

Algorithm 1 DE

Set *NP, CR, F*, and stopping criterion;
$G = 0$, $\boldsymbol{x}_{\text{best}} = \{\}$;
Randomly initialize (1) population $\boldsymbol{P} = (\boldsymbol{x}_{1,G}, \ldots, \boldsymbol{x}_{NP,G})$;
$\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$;
while stopping criterion not met **do**
  for $i = 1$ to NP **do**
    $\boldsymbol{x}_{i,G} = \boldsymbol{P}[i]$;
    $\boldsymbol{v}_{i,G}$ by mutation (3);
    $\boldsymbol{u}_{i,G}$ by crossover (6);
    if $f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{u}_{i,G}$;
    else
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{x}_{i,G}$;
    end if
    $\boldsymbol{x}_{i,G+1} \rightarrow \boldsymbol{P}_{\text{new}}$;
  end for
  $\boldsymbol{P} = \boldsymbol{P}_{\text{new}}$, $\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$, G++;
end while
return $\boldsymbol{x}_{\text{best}}$ as the best found solution;

2.2. Linear Decrease in Population Size

Linear decrease in population size was introduced to SHADE in [14] to improve its performance. The basic idea is to reduce the population size to promote exploitation in later phases of the evolution. Therefore, a new formula to estimate the population size was formed (13) and a new population size is calculated after each generation. Whenever the new population size NP_{new} is smaller than the current population size NP, the population is sorted according to the objective function value, and the worst $NP - NP_{\text{new}}$ individuals are discarded. The size of the external archive is reduced as well, using the formula:

$$NP_{\text{new}} = \text{round}\left(NP_{\text{init}} - \frac{FES}{MAXFES} * \left(NP_{\text{init}} - NP_{\text{f}}\right)\right), \tag{13}$$

where NP_{init} is the initial population size and NP_f is the end population size. *FES* and *MAXFES* are objective function number evaluations and a maximum number of objective function evaluations, respectively.

Algorithm 2 SHADE

Set NP, H, and stopping criterion;
$G = 0$, $\boldsymbol{x}_{\text{best}} = \{\}$, $k = 1$, $p_{\min} = 2/NP$, $\boldsymbol{A} = \emptyset$;
Randomly initialize (1) population $\boldsymbol{P} = (\boldsymbol{x}_{1,G},\ldots,\boldsymbol{x}_{NP,G})$;
Set $\mathbf{M}_{\text{F}}$ and $\mathbf{M}_{\text{CR}}$ according to (2);
$\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$;
while stopping criterion not met **do**
    $\mathbf{S}_F = \emptyset$, $\mathbf{S}_{CR} = \emptyset$;
    for $i = 1$ to NP **do**
        $r = \mathcal{U}[1, H]$;
        Set F_i by (5) and CR_i by (7);
        $\boldsymbol{x}_{i,G} = \boldsymbol{P}[i]$, $p_i = \mathcal{U}[p_{\min}, 0.2]$;
        $\boldsymbol{v}_{i,G}$ by mutation (4);
        $\boldsymbol{u}_{i,G}$ by crossover (6);
        if $f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G})$ **then**
            $\boldsymbol{x}_{i,G+1} = \boldsymbol{u}_{i,G}$;
            $\boldsymbol{x}_{i,G} \rightarrow \boldsymbol{A}$;
            $F_i \rightarrow \mathbf{S}_F$, $CR_i \rightarrow \mathbf{S}_{CR}$;
        else
            $\boldsymbol{x}_{i,G+1} = \boldsymbol{x}_{i,G}$;
        end if
        if $|\boldsymbol{A}| > NP$ **then**
            Randomly delete an individual from $\boldsymbol{A}$;
        end if
        $\boldsymbol{x}_{i,G+1} \rightarrow \boldsymbol{P}_{\text{new}}$;
    end for
    if $\mathbf{S}_F \neq \emptyset$ **and** $\mathbf{S}_{CR} \neq \emptyset$ **then**
        Update $M_{\text{F},k}$ (9) and $M_{\text{CR},k}$ (10) with Lehmer mean computed by (11) with objective function value improvement based weights from (12), k++;
        if $k > H$ **then**
            $k = 1$;
        end if
    end if
    $\boldsymbol{P} = \boldsymbol{P}_{\text{new}}$, $\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$, G++;
end while
return $\boldsymbol{x}_{\text{best}}$ as the best found solution;

2.3. Weighted Mutation Strategy with Parameterization Enhancements: jSO

The jSO algorithm [17] was announced winner at the CEC 2017 Competition on single objective real-parameter optimization [35], and introduces to L-SHADE mainly a new weighted version of mutation strategy after using, iL-SHADE [36] updates of L-SHADE. The iL-SHADE extends L-SHADE by initializing all values in $\mathbf{M}_{\mathrm{F}}$ and $\mathbf{M}_{\mathrm{CR}}$ at 0.8, additional historical memory entry $M_{\mathrm{F},H} = M_{\mathrm{CR},H} = 0.9$ with weighted Lehmer means to calculate historical memory values, limiting F and CR values in early stages, and computing p for pBest mutation strategy as:

$$p = p_{\min} + \frac{FES}{MAXFES}(p_{\max} - p_{\min}). \tag{14}$$

The jSO algorithm sets p value limits at $p_{\max} = 0.25$ and $p_{\min} = p_{\max}/2$, initial population size at $NP_{\mathrm{init}} = 25\sqrt{D}\log D$, historical memory size $H = 5$, initializes values of M_F at 0.3, and proposes the weighted version mutation strategy current-to-pBest-w/1 for the i-th vector:

$$\mathbf{v}_i = \mathbf{x}_i + F_w(x_{p\mathrm{Best}} - \mathbf{x}_i) + F(\mathbf{x}_{\mathrm{r1}} - \mathbf{x}_{\mathrm{r2}}), \tag{15}$$

where F_w is calculated as:

$$F_w = \begin{cases} 0.7F, & FES < 0.2MAXFES, \\ 0.8F, & FES < 0.4MAXFES, \\ 1.2F, & \text{otherwise.} \end{cases} \tag{16}$$

3. Distance Based Parameter Adaptation

The original adaptation mechanism for scaling factor and crossover rate values uses weighted forms of means (11), where weights are based on the improvement in the objective function value (12). This approach promotes exploitation over exploration and, therefore, might lead to premature convergence, which could be a problem, especially in higher dimensions.

The idea behind the proposed approach is quite simple. Either to improve the exploration ability of the algorithm or to maintain it for a more extended period, it is necessary to force the individuals to explore the search space more intensively and to keep the population diversity higher. By doing that through any adaptation, it would be beneficial to find such a parameter setting that forces the individuals to the desired behavior.

The distance approach is based on the Euclidean distance between the trial and the original individual (17). This increases the complexity of the weight computation slightly by exchanging simple difference for Euclidean distance, nevertheless, the overall experimental complexity of the algorithm is similar (even lower), as given in Tables 21, 22, and 23. The explanation behind the similar complexity of the proposed method is that, in the original weight calculation (12), the differences between objective function values can reach high numerical values, especially in a high dimensional search space. Therefore, the division operation in (12) may take longer time to evaluate, whereas the presented approach (17) with Euclidean distances only considers the individuals' positions within the range of search space (in the case of CEC benchmarks $[-100, 100]$), thus, the numerical range of division operation in weight calculation (17) is substantially lower.

In this modification, scaling factor and crossover rate values connected with the individual that moved the furthest will have the highest weight:

$$w_k = \frac{\sqrt{\sum_{j=1}^{D}\left(u_{k,j,G} - x_{k,j,G}\right)^2}}{\sum_{m=1}^{|S_{CR}|}\sqrt{\sum_{j=1}^{D}\left(u_{m,j,G} - x_{m,j,G}\right)^2}}. \quad (17)$$

Therefore, the exploration ability is rewarded, and this should lead to avoidance of the premature convergence in higher dimensional objective spaces.

The pseudo-code of the Db_SHADE algorithm (extending the SHADE algorithm) is seen as Algorithm 3; for a clear overview, the pseudo-code of the DbL_SHADE algorithm (extending the L-SHADE algorithm) is depicted as Algorithm 4, and the algorithm DISH (extending algorithm jSO) is depicted as Algorithm 5. The modifications in the proposed algorithms from their respective originals are underlined: line 27 in DB_SHADE, line 35 in DbL_SHADE, and line 54 in DISH, which implement the proposed distance based parameter adaptation mechanism for these three new algorithms. The acronym DISH denotes the proposed best performing "**DI**stance Based Parameter Adaptation for **S**ucces-**H**istory based Differential Evolution", as demonstrated experimentally in the next section.

4. Experimental Setting

In order to test the proposed modification experimentally, it was implemented into three algorithms – SHADE (as Db_SHADE), L-SHADE (as DbL-SHADE), and jSO (as DISH). These algorithms were run according to the Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on a Learning-based Real-Parameter Single Objective Optimization (CEC2015) benchmark set in 10, 30, 50 and 100 dimensions [37]. According to the benchmark rules, the stopping criterion was set to $10{,}000 \times D$ objective function evaluations, and 51 independent runs were performed. SHADE's historical memory size H was set to 10, external archive size A was set to NP, and the population size NP was set to 100 [12]. L-SHADE's historical memory size H was set to 6, external archive A size was set to NP, and starting population size NP_{init} was set to $18D$, and the final population size NP_f was set to 4, according to [14]. The jSO's historical memory size H was set to 5, external archive A size was set to NP, and starting population size NP_{init} was set to $25\log D, \sqrt{D}$ and the final population size NP_f was set to 4 [17].

Based on the obtained results for the CEC2015 benchmark set, the most up-to-date SHADE version – jSO was also tested on the CEC2017 benchmark set [35]. Namely, the CEC2015 benchmark set uses a different mathematical formulation of an algorithm for each function (parameters are different, i.e. it is a one function algorithm), whereas CEC2017 uses one algorithm for more functions (whole domain benchmark) [38].

The stated prolonged exploration phase of the distance based parameter adaptation was a subject to study. The population clustering was recorded for each generation, each function in the benchmark, each algorithm and each dimensionality. Also a population diversity was recorded in a similar manner. Both analyses are described in more detail in the following subsections.

4.1. Cluster Analysis

The clustering algorithm selected for this experiment was Density Based Spatial Clustering of Applications with Noise (DBSCAN) [39], which conveniently works on the basis of cluster density, rather than its centre, and, therefore, is able to discover clusters of arbitrary shapes.

The DBSCAN algorithm requires setting of two control parameters and a distance measure. These were set as follows:

Algorithm 3 Db_SHADE

Set NP, H, and stopping criterion;
$G = 0$, $\boldsymbol{x}_{\text{best}} = \{\}$, $k = 1$, $p_{\min} = 2/NP$, $\boldsymbol{A} = \emptyset$;
Randomly initialize (1) population $\boldsymbol{P} = (\boldsymbol{x}_{1,G}, \ldots, \boldsymbol{x}_{NP,G})$;
Set $\mathbf{M}_{\text{F}}$ and $\mathbf{M}_{\text{CR}}$ according to (2);
$\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$;
while stopping criterion not met **do**
    $\mathbf{S}_F = \emptyset$, $\mathbf{S}_{CR} = \emptyset$;
    for $i = 1$ to NP **do**
        $r = \mathcal{U}[1, H]$;
        Set F_i by (5) and CR_i by (7);
        $\boldsymbol{x}_{i,G} = \boldsymbol{P}[i]$, $p_i = \mathcal{U}[p_{\min}, 0.2]$;
        $\boldsymbol{v}_{i,G}$ by mutation (4);
        $\boldsymbol{u}_{i,G}$ by crossover (6);
        if $f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G})$ **then**
            $\boldsymbol{x}_{i,G+1} = \boldsymbol{u}_{i,G}$;
            $\boldsymbol{x}_{i,G} \rightarrow \boldsymbol{A}$;
            $F_i \rightarrow \mathbf{S}_F$, $CR_i \rightarrow \mathbf{S}_{CR}$;
        else
            $\boldsymbol{x}_{i,G+1} = \boldsymbol{x}_{i,G}$;
        end if
        if $|\boldsymbol{A}| > NP$ **then**
            Randomly delete $|\boldsymbol{A}| - NP$ individuals from $\boldsymbol{A}$;
        end if
        $\boldsymbol{x}_{i,G+1} \rightarrow \boldsymbol{P}_{new}$;
    end for
    if $\mathbf{S}_F \neq \emptyset$ **and** $\mathbf{S}_{CR} \neq \emptyset$ **then**
        Update $M_{\text{F},k}$ (9) and $M_{\text{CR},k}$ (10) with Lehmer mean computed by (11) with distance based weights from (17), k++;
        if $k > H$ **then**
            $k = 1$;
        end if
    end if
    $\boldsymbol{P} = \boldsymbol{P}_{\text{new}}$, $\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$, G++;
end while
return $\boldsymbol{x}_{\text{best}}$ as the best found solution;

Algorithm 4 DbL_SHADE

Set NP_{init}, NP_f, H, and stopping criterion;
$NP = NP_{\text{init}}$, $G = 0$, $\boldsymbol{x}_{\text{best}} = \{\}$, $k = 1$, $p_{\min} = 2/NP$, $\boldsymbol{A} = \emptyset$;
Randomly initialize (1) population $\boldsymbol{P} = (\boldsymbol{x}_{1,G}, \ldots, \boldsymbol{x}_{NP,G})$;
Set $\mathbf{M}_{\text{F}}$ and $\mathbf{M}_{\text{CR}}$ according to (2);
$\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$;
while stopping criterion not met **do**
  $\mathbf{S}_F = \emptyset$, $\mathbf{S}_{CR} = \emptyset$;
  for $i = 1$ to NP **do**
    $r = \mathcal{U}[1, H]$;
    Set F_i by (5) and CR_i by (7);
    $\boldsymbol{x}_{i,G} = \boldsymbol{P}[i]$, $p_i = \mathcal{U}[p_{\min}, 0.2]$;
    $\boldsymbol{v}_{i,G}$ by mutation (4);
    $\boldsymbol{u}_{i,G}$ by crossover (6);
    if $f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{u}_{i,G}$;
      $\boldsymbol{x}_{i,G} \to \boldsymbol{A}$;
      $F_i \to \mathbf{S}_F$, $CR_i \to \mathbf{S}_{CR}$;
    else
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{x}_{i,G}$;
    end if
    if $|\boldsymbol{A}| > NP$ **then**
      Randomly delete $|\boldsymbol{A}| - NP$ individuals from $\boldsymbol{A}$;
    end if
    $\boldsymbol{x}_{i,G+1} \to \boldsymbol{P}_{\text{new}}$;
  end for
  Calculate NP_{new} according to (13);
  if $NP_{\text{new}} < NP$ **then**
    Sort individuals in $\boldsymbol{P}$ according to their objective function values and remove $NP - NP_{\text{new}}$ worst ones;
    $NP = NP_{\text{new}}$;
  end if
  if $|\boldsymbol{A}| > NP$ **then**
    Randomly delete $|\boldsymbol{A}| - NP$ individuals from $\boldsymbol{A}$;
  end if
  if $\mathbf{S}_F \neq \emptyset$ **and** $\mathbf{S}_{CR} \neq \emptyset$ **then**
    Update Update $M_{\text{F},k}$ (9) and $M_{\text{CR},k}$ (10) with Lehmer mean computed by (11) with distance based weights from (17), k++;
    if $k > H$ **then**
      $k = 1$;
    end if
  end if
  $\boldsymbol{P} = \boldsymbol{P}_{\text{new}}$, $\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$, G++;
end while
return $\boldsymbol{x}_{\text{best}}$ as the best found solution;

Algorithm 5 DISH

Set NP_{init}, NP_f, H, and stopping criterion;
$NP = NP_{\text{init}}$, $G = 0$, $\boldsymbol{x}_{\text{best}} = \{\}$, $k = 1$, $p_{\min} = 2/NP$, $\boldsymbol{A} = \emptyset$;
Randomly initialize (1) population $\boldsymbol{P} = (\boldsymbol{x}_{1,G}, \ldots, \boldsymbol{x}_{NP,G})$;
Set all values in $\mathbf{M}_{\text{F}}$ to 0.5 and $\mathbf{M}_{\text{CR}}$ to 0.8;
$\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$;
while stopping criterion not met **do**
  $\mathbf{S}_F = \emptyset$, $\mathbf{S}_{CR} = \emptyset$;
  for $i = 1$ to NP **do**
    $r = \mathcal{U}[1, H]$;
    if $r = H$ **then**
      $M_{\text{F},r} = 0.9$;
      $M_{\text{CR},r} = 0.9$;
    end if
    if $M_{\text{CR},r} < 0$ **then**
      $CR_{i,G} = 0$;
    else
      $CR_{i,G} = \mathcal{N}(M_{\text{CR},r}, 0.1)$;
    end if
    Set F_i by (5);
    if $G < 0.6G_{\text{MAX}}$ **and** $F_{i,G} > 0.7$ **then**
      $F_{i,G} = 0.7$;
    end if
    if $G < 0.25G_{\text{MAX}}$ **then**
      $CR_{i,G} = \max(CR_{i,G}, 0.7)$;
    else if $G < 0.5G_{\text{MAX}}$ **then**
      $CR_{i,G} = \max(CR_{i,G}, 0.6)$;
    end if
    $\boldsymbol{x}_{i,G} = \boldsymbol{P}[i]$, $p_i = U[p_{\min}, 0.2]$;
    $\boldsymbol{v}_{i,G}$ by mutation (15);
    $\boldsymbol{u}_{i,G}$ by crossover (6);
    if $f(\boldsymbol{u}_{i,G}) \leq f(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{u}_{i,G}$;
    else
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{x}_{i,G}$;
    end if
    if $f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G} \to \boldsymbol{A}$;
      $F_i \to \mathbf{S}_F$, $CR_i \to \mathbf{S}_{CR}$;
    end if
    if $|\boldsymbol{A}| > NP$ **then**
      Randomly delete $|\boldsymbol{A}| - NP$ individuals from $\boldsymbol{A}$;
    end if
    $\boldsymbol{x}_{i,G+1} \to \boldsymbol{P}_{\text{new}}$;
  end for
  Calculate NP_{new} according to (13);
  if $NP_{\text{new}} < NP$ **then**
    Sort individuals in $\boldsymbol{P}$ according to their objective function values and remove $NP - NP_{\text{new}}$ worst ones;
    $NP = NP_{\text{new}}$;
  end if
  if $|\boldsymbol{A}| > NP$ **then**
    Randomly delete $|\boldsymbol{A}| - NP$ individuals from $\boldsymbol{A}$;
  end if
  if $\mathbf{S}_F \neq \emptyset$ **and** $\mathbf{S}_{CR} \neq \emptyset$ **then**
    Update Update $M_{\text{F},k}$ (9) and $M_{\text{CR},k}$ (10) with Lehmer mean computed by (11) with distance based weights from (17), k++;
    if $k > H$ **then**
      $k = 1$;
    end if
  end if
  $\boldsymbol{P} = \boldsymbol{P}_{\text{new}}$, $\boldsymbol{P}_{\text{new}} = \{\}$, $\boldsymbol{x}_{\text{best}}$ = best from population $\boldsymbol{P}$, G++;
end while
return $\boldsymbol{x}_{\text{best}}$ as the best found solution;

1. Core point distance *Eps* = 1% of the decision space – for the CEC2015 benchmark set *Eps* = 2,
2. Minimal number of points to form a cluster *MinPts* = 4 (minimal number of individuals for mutation),
3. Distance measure equal to Chebyshev distance [40] – if the distance between any corresponding attributes of two individuals is higher than 1% of the decision space, they are not considered directly density-reachable.

4.2. Population Diversity

The used Population Diversity (PD) measure was taken from [41], and is based on the square root of the sum of deviations, Eq. (19), of individual's components from their corresponding means, Eq. (18):

$$\overline{x_j} = \frac{1}{NP} \sum_{i=1}^{NP} x_{j,i}, \tag{18}$$

$$PD = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} \sum_{j=1}^{D} \left(x_{j,i} - \overline{x_j}\right)^2}, \tag{19}$$

where i is the population member iterator and j is the component (dimension) iterator.

5. Results

In this section, several Tables with results are provided, where there are comparisons of reached objective function values between the original algorithm (SHADE, L-SHADE, or jSO) and its modified version with distance based adaptation (Db_SHADE, DbL_SHADE, or DISH, respectively). In the last column of each Table, there is a result of a Wilcoxon rank-sum test with significance level α set to 0.05. When the original version performs significantly better, the "-" sign is used, when the modified version performs significantly better, the "+" sign is used, and when the performance is equal, the "=" sign is used. All Tables provide median and mean values of the 51 runs.

Convergence graphs are given in Figures 1–9. Figures 1–3 depict the difference in convergence between the SHADE and Db_SHADE algorithms in 30D, 50D, and 100D correspondingly, Figures 4–6 depict the difference in convergence between the L-SHADE and DbL_SHADE algorithms in 30D, 50D, and 100D correspondingly and Figures 7–9. It is apparent, that the blue line of distance based version of the algorithms is often slower in convergence, but able to reach better objective function values.

Results in Tables 17–19 present the number of runs where clustering of the population occurred (#runs), mean generation of the first cluster occurrence during those runs (Mean CO), and the mean population diversity in those generations (Mean PD).

Using median values from the results comparison Tables, the Friedman ranking [42] of the algorithms using aggregated functions sets is shown in Tables 24–28, and their post-hoc procedures analysis in Tables 29–33.

Table 1: SHADE vs. Db_SHADE on CEC2015 in 10D.

	SHADE		**Db_SHADE**		
f	Median	Mean	Median	Mean	Result
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.00E+01	1.89E+01	2.00E+01	1.92E+01	=
4	3.07E+00	2.97E+00	3.06E+00	2.98E+00	=
5	2.21E+01	3.42E+01	2.98E+01	4.52E+01	=
6	2.20E-01	2.97E+00	4.16E-01	8.08E-01	=
7	1.67E-01	1.88E-01	1.73E-01	1.91E-01	=
8	8.15E-02	2.69E-01	4.28E-02	2.06E-01	=
9	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=
10	2.17E+02	2.17E+02	2.17E+02	2.17E+02	=
11	3.00E+02	1.66E+02	3.00E+02	2.01E+02	=
12	1.01E+02	1.01E+02	1.01E+02	1.01E+02	=
13	2.78E+01	2.78E+01	2.79E+01	2.76E+01	=
14	2.94E+03	4.28E+03	2.98E+03	4.66E+03	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 2: SHADE vs. Db_SHADE on CEC2015 in 30D.

	SHADE		**Db_SHADE**		
f	Median	Mean	Median	Mean	Result
1	3.73E+01	2.62E+02	2.12E+01	2.42E+02	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.01E+01	2.01E+01	2.01E+01	2.01E+01	=
4	1.41E+01	1.41E+01	1.32E+01	1.31E+01	=
5	1.55E+03	1.50E+03	1.54E+03	1.52E+03	=
6	5.36E+02	5.73E+02	3.37E+02	3.48E+02	+
7	7.17E+00	7.26E+00	6.81E+00	6.74E+00	+
8	1.26E+02	1.21E+02	5.27E+01	7.38E+01	+
9	1.03E+02	1.03E+02	1.03E+02	1.03E+02	=
10	6.27E+02	6.22E+02	5.29E+02	5.32E+02	+
11	4.53E+02	4.50E+02	4.10E+02	4.16E+02	+
12	1.05E+02	1.05E+02	1.05E+02	1.05E+02	=
13	9.52E+01	9.50E+01	9.47E+01	9.50E+01	=
14	3.21E+04	3.24E+04	3.22E+04	3.24E+04	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 3: SHADE vs. Db_SHADE on CEC2015 in 50D.

	SHADE		**Db_SHADE**		
f	Median	Mean	Median	Mean	Result
1	1.81E+04	2.14E+04	3.00E+04	3.27E+04	-
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.01E+01	2.01E+01	2.02E+01	2.02E+01	-
4	3.84E+01	3.92E+01	3.15E+01	3.27E+01	+
5	3.10E+03	3.09E+03	3.06E+03	3.01E+03	=
6	2.87E+03	3.56E+03	2.87E+03	3.91E+03	=
7	4.22E+01	4.25E+01	4.08E+01	4.12E+01	+
8	1.13E+03	1.12E+03	6.62E+02	6.68E+02	+
9	1.06E+02	1.06E+02	1.05E+02	1.05E+02	+
10	1.57E+03	1.59E+03	1.23E+03	1.24E+03	+
11	6.76E+02	6.81E+02	5.83E+02	5.85E+02	+
12	1.08E+02	1.08E+02	1.08E+02	1.08E+02	=
13	1.80E+02	1.80E+02	1.81E+02	1.80E+02	=
14	7.29E+04	6.66E+04	6.96E+04	6.51E+04	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 4: SHADE vs. Db_SHADE on CEC2015 in 100D.

	SHADE		**Db_SHADE**		
f	Median	Mean	Median	Mean	Result
1	2.00E+05	2.20E+05	2.00E+05	2.10E+05	=
2	7.80E-07	7.70E-03	7.00E-10	1.60E-08	+
3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	-
4	1.60E+02	1.60E+02	1.30E+02	1.30E+02	+
5	9.60E+03	9.60E+03	9.40E+03	9.40E+03	=
6	3.50E+04	4.00E+04	3.50E+04	3.80E+04	=
7	1.20E+02	1.30E+02	1.40E+02	1.20E+02	=
8	1.30E+04	1.40E+04	1.10E+04	1.10E+04	=
9	1.10E+02	1.10E+02	1.10E+02	1.10E+02	+
10	4.20E+03	4.20E+03	4.00E+03	4.00E+03	=
11	1.90E+03	1.90E+03	1.70E+03	1.70E+03	+
12	1.20E+02	1.20E+02	1.20E+02	1.20E+02	=
13	3.90E+02	3.90E+02	3.90E+02	3.90E+02	=
14	1.10E+05	1.10E+05	1.10E+05	1.10E+05	=
15	1.10E+02	1.10E+02	1.00E+02	1.00E+02	+

Table 5: L-SHADE vs. DbL_SHADE on CEC2015 in 10D.

	L-SHADE		**DbL_SHADE**		
f	Median	Mean	Median	Mean	Result
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.00E+01	1.87E+01	2.00E+01	1.89E+01	=
4	2.98E+00	2.58E+00	2.99E+00	2.95E+00	=
5	2.87E+01	6.05E+01	1.54E+01	4.23E+01	=
6	4.16E-01	2.84E+00	6.24E-01	7.74E-01	-
7	7.01E-02	1.31E-01	9.49E-02	1.89E-01	=
8	4.21E-01	4.13E-01	3.29E-01	3.44E-01	=
9	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=
10	2.17E+02	2.17E+02	2.17E+02	2.17E+02	=
11	3.00E+02	1.83E+02	3.00E+02	1.95E+02	=
12	1.01E+02	1.01E+02	1.01E+02	1.01E+02	+
13	2.71E+01	2.66E+01	2.69E+01	2.69E+01	=
14	2.94E+03	4.19E+03	2.94E+03	4.77E+03	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 6: L-SHADE vs. DbL_SHADE on CEC2015 in 30D.

	L-SHADE		**DbL_SHADE**		
f	Median	Mean	Median	Mean	Result
1	1.60E+00	6.18E+00	3.86E+00	2.00E+01	-
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	=
4	1.29E+01	1.39E+01	1.29E+01	1.29E+01	=
5	1.44E+03	1.39E+03	1.40E+03	1.41E+03	=
6	7.61E+02	7.71E+02	4.64E+02	4.74E+02	+
7	6.70E+00	6.48E+00	5.91E+00	5.62E+00	+
8	1.51E+02	1.47E+02	1.21E+02	1.14E+02	+
9	1.03E+02	1.03E+02	1.03E+02	1.03E+02	=
10	7.21E+02	7.75E+02	5.99E+02	5.85E+02	+
11	4.77E+02	4.68E+02	4.21E+02	4.33E+02	+
12	1.05E+02	1.05E+02	1.05E+02	1.05E+02	=
13	9.29E+01	9.24E+01	9.32E+01	9.25E+01	=
14	3.33E+04	3.29E+04	3.31E+04	3.25E+04	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 7: L-SHADE vs. DbL_SHADE on CEC2015 in 50D.

	L-SHADE		**DbL_SHADE**		
f	Median	Mean	Median	Mean	Result
1	4.37E+03	6.31E+03	1.17E+04	1.50E+04	-
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	=
4	3.68E+01	3.62E+01	3.09E+01	3.12E+01	+
5	3.07E+03	3.06E+03	2.93E+03	2.90E+03	+
6	2.74E+03	2.75E+03	2.48E+03	2.85E+03	=
7	4.29E+01	4.33E+01	4.20E+01	4.21E+01	+
8	1.15E+03	1.11E+03	8.25E+02	8.28E+02	+
9	1.06E+02	1.06E+02	1.05E+02	1.05E+02	+
10	1.60E+03	1.65E+03	1.41E+03	1.46E+03	+
11	6.93E+02	6.89E+02	5.98E+02	5.97E+02	+
12	1.08E+02	1.08E+02	1.08E+02	1.08E+02	+
13	1.78E+02	1.78E+02	1.79E+02	1.78E+02	=
14	7.30E+04	6.70E+04	5.92E+04	6.37E+04	+
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 8: L-SHADE vs. DbL_SHADE on CEC2015 in 100D.

	L-SHADE		**DbL_SHADE**		
f	Median	Mean	Median	Mean	Result
1	9.10E+04	1.10E+05	1.40E+05	1.60E+05	-
2	1.20E-09	3.80E-09	2.90E-09	7.90E-09	=
3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	=
4	1.50E+02	1.50E+02	1.30E+02	1.30E+02	+
5	9.50E+03	9.50E+03	9.20E+03	9.30E+03	+
6	2.80E+04	3.10E+04	3.30E+04	3.50E+04	-
7	1.10E+02	1.10E+02	1.10E+02	1.10E+02	+
8	8.90E+03	9.80E+03	1.00E+04	1.10E+04	=
9	1.10E+02	1.10E+02	1.10E+02	1.10E+02	+
10	4.50E+03	4.50E+03	4.30E+03	4.30E+03	=
11	1.90E+03	1.90E+03	1.70E+03	1.70E+03	+
12	1.20E+02	1.20E+02	1.20E+02	1.20E+02	=
13	3.90E+02	3.80E+02	3.90E+02	3.90E+02	=
14	1.10E+05	1.10E+05	1.10E+05	1.10E+05	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	+

Table 9: jSO vs. DISH on CEC2015 in 10D.

	jSO		DISH		
f	Median	Mean	Median	Mean	Result
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	=
4	2.00E+00	2.20E+00	2.00E+00	2.10E+00	=
5	1.00E+01	3.20E+01	1.50E+01	4.00E+01	=
6	2.00E+00	3.20E+00	1.40E+00	2.50E+00	=
7	2.90E-02	7.30E-02	2.90E-02	7.50E-02	=
8	4.00E-01	4.20E-01	5.00E-01	5.10E-01	=
9	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=
10	2.20E+02	2.20E+02	2.20E+02	2.20E+02	=
11	3.00E+02	1.70E+02	3.00E+02	2.10E+02	=
12	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=
13	2.80E+01	2.80E+01	2.70E+01	2.70E+01	=
14	2.90E+03	4.70E+03	2.90E+03	4.20E+03	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 10: jSO vs. DISH on CEC2015 in 30D.

	jSO		DISH		
f	Median	Mean	Median	Mean	Result
1	8.80E-02	4.30E-01	7.20E-02	6.70E-01	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.10E+01	2.10E+01	2.10E+01	2.10E+01	=
4	1.40E+01	1.40E+01	1.40E+01	1.40E+01	=
5	1.50E+03	1.50E+03	1.60E+03	1.50E+03	=
6	2.30E+02	2.90E+02	1.80E+02	2.10E+02	+
7	2.80E+00	2.90E+00	2.60E+00	2.80E+00	=
8	5.10E+01	6.20E+01	3.60E+01	6.30E+01	=
9	1.00E+02	1.00E+02	1.00E+02	1.00E+02	+
10	5.00E+02	5.00E+02	4.70E+02	4.70E+02	=
11	4.40E+02	4.40E+02	4.00E+02	4.20E+02	+
12	1.10E+02	1.10E+02	1.10E+02	1.00E+02	=
13	9.40E+01	9.40E+01	9.50E+01	9.50E+01	=
14	3.10E+04	3.20E+04	3.10E+04	3.20E+04	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 11: jSO vs. DISH on CEC2015 in 50D.

	jSO		**DISH**		
f	Median	Mean	Median	Mean	Result
1	7.70E+03	8.80E+03	8.70E+03	1.00E+04	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	2.10E+01	2.10E+01	2.00E+01	2.10E+01	=
4	4.10E+01	4.00E+01	3.40E+01	3.40E+01	+
5	3.30E+03	3.20E+03	3.30E+03	3.20E+03	=
6	2.00E+03	2.10E+03	1.80E+03	1.80E+03	+
7	4.10E+01	4.10E+01	4.10E+01	4.10E+01	=
8	6.20E+02	6.20E+02	5.00E+02	5.10E+02	+
9	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=
10	1.20E+03	1.10E+03	1.10E+03	1.10E+03	+
11	5.10E+02	5.10E+02	4.70E+02	4.80E+02	+
12	1.10E+02	1.10E+02	1.10E+02	1.10E+02	+
13	1.80E+02	1.80E+02	1.80E+02	1.80E+02	+
14	5.90E+04	6.00E+04	5.90E+04	6.20E+04	=
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=

Table 12: jSO vs. DISH on CEC2015 in 100D.

	jSO		**DISH**		
f	Median	Mean	Median	Mean	Result
1	9.10E+04	1.10E+05	1.30E+05	1.40E+05	-
2	5.00E-10	3.30E-09	7.00E-10	3.80E-09	=
3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	=
4	1.80E+02	1.80E+02	1.50E+02	1.50E+02	+
5	10.00E+03	10.00E+03	1.00E+04	10.00E+03	=
6	2.50E+04	2.80E+04	3.20E+04	3.30E+04	-
7	1.10E+02	1.10E+02	1.00E+02	1.10E+02	=
8	7.50E+03	8.10E+03	7.20E+03	7.70E+03	=
9	1.10E+02	1.10E+02	1.10E+02	1.10E+02	+
10	3.90E+03	3.90E+03	3.70E+03	3.80E+03	=
11	1.40E+03	1.40E+03	1.20E+03	1.20E+03	+
12	1.20E+02	1.20E+02	1.20E+02	1.20E+02	+
13	4.00E+02	4.00E+02	4.00E+02	4.00E+02	=
14	1.10E+05	1.10E+05	1.10E+05	1.10E+05	+
15	1.00E+02	1.00E+02	1.00E+02	1.00E+02	+

Table 13: jSO vs. DISH on CEC2017 in $10D$.

	jSO		**DISH**		
f	Median	Mean	Median	Mean	Result
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
5	2.00E+00	1.80E+00	2.00E+00	1.80E+00	=
6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
7	1.20E+01	1.20E+01	1.20E+01	1.20E+01	=
8	2.00E+00	2.00E+00	2.00E+00	2.00E+00	=
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
10	1.00E+01	3.60E+01	7.00E+00	4.40E+01	=
11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
12	4.20E-01	2.70E+00	4.20E-01	3.30E-01	=
13	4.80E+00	3.00E+00	10.00E-01	2.10E+00	=
14	0.00E+00	5.90E-02	0.00E+00	1.20E-01	=
15	1.80E-01	2.20E-01	4.50E-01	3.10E-01	-
16	5.20E-01	5.70E-01	6.30E-01	5.60E-01	=
17	4.00E-01	5.00E-01	3.90E-01	4.40E-01	=
18	3.80E-01	3.10E-01	2.70E-01	2.70E-01	=
19	0.00E+00	1.10E-02	0.00E+00	9.20E-03	=
20	3.10E-01	3.40E-01	3.10E-01	3.40E-01	=
21	1.00E+02	1.30E+02	1.00E+02	1.40E+02	=
22	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=
23	3.00E+02	3.00E+02	3.00E+02	3.00E+02	=
24	3.30E+02	3.00E+02	3.30E+02	2.90E+02	=
25	4.00E+02	4.10E+02	4.00E+02	4.10E+02	=
26	3.00E+02	3.00E+02	3.00E+02	3.00E+02	=
27	3.90E+02	3.90E+02	3.90E+02	3.90E+02	=
28	3.00E+02	3.40E+02	3.00E+02	3.70E+02	=
29	2.30E+02	2.30E+02	2.40E+02	2.40E+02	=
30	4.00E+02	4.00E+02	4.00E+02	4.00E+02	=

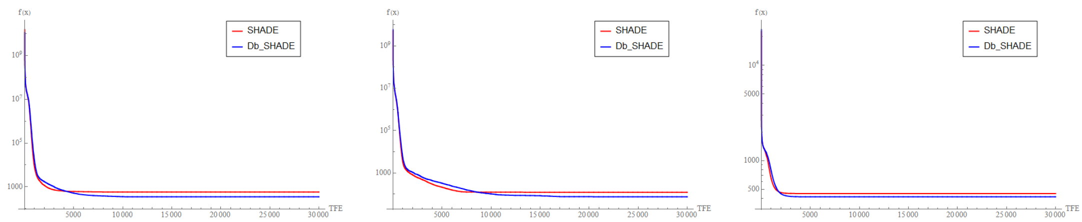

Figure 1: Comparison of selected average convergence between the SHADE and Db_SHADE algorithms on CEC2015 in $30D$. From left $f6, f8$ and $f11$.

Table 14: jSO vs. DISH on CEC2017 in $30D$.

	jSO		**DISH**		
f	Median	Mean	Median	Mean	Result
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
4	5.90E+01	5.90E+01	5.90E+01	5.90E+01	=
5	8.00E+00	8.60E+00	8.00E+00	8.20E+00	=
6	0.00E+00	6.00E-09	0.00E+00	1.30E-08	=
7	3.90E+01	3.90E+01	3.80E+01	3.80E+01	=
8	9.00E+00	9.10E+00	8.00E+00	8.40E+00	=
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
10	1.50E+03	1.50E+03	1.50E+03	1.50E+03	=
11	2.00E+00	3.00E+00	2.00E+00	3.80E+00	=
12	1.40E+02	1.70E+02	1.20E+02	9.40E+01	+
13	1.60E+01	1.50E+01	1.70E+01	1.50E+01	=
14	2.10E+01	2.20E+01	2.20E+01	2.20E+01	=
15	7.80E-01	1.10E+00	9.10E-01	1.10E+00	=
16	2.60E+01	7.90E+01	2.50E+01	8.00E+01	=
17	3.50E+01	3.30E+01	3.50E+01	3.40E+01	=
18	2.10E+01	2.00E+01	2.10E+01	2.00E+01	=
19	4.10E+00	4.50E+00	3.50E+00	4.20E+00	=
20	2.90E+01	2.90E+01	2.80E+01	2.80E+01	=
21	2.10E+02	2.10E+02	2.10E+02	2.10E+02	=
22	1.00E+02	1.00E+02	1.00E+02	1.00E+02	=
23	3.50E+02	3.50E+02	3.50E+02	3.50E+02	=
24	4.30E+02	4.30E+02	4.30E+02	4.30E+02	=
25	3.90E+02	3.90E+02	3.90E+02	3.90E+02	+
26	9.30E+02	9.20E+02	9.30E+02	9.40E+02	-
27	5.00E+02	5.00E+02	4.90E+02	4.90E+02	+
28	3.00E+02	3.10E+02	3.00E+02	3.00E+02	=
29	4.30E+02	4.30E+02	4.40E+02	4.30E+02	=
30	2.00E+03	2.00E+03	2.00E+03	2.00E+03	=

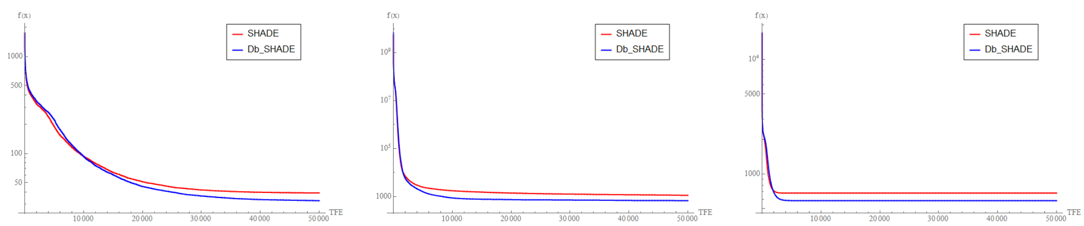

Figure 2: Comparison of selected average convergence between the SHADE and Db_SHADE algorithms on CEC2015 in $50D$. From left $f4$, $f8$ and $f11$.

Table 15: jSO vs. DISH on CEC2017 in $50D$.

	jSO		**DISH**		
f	Median	Mean	Median	Mean	Result
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
4	2.90E+01	5.60E+01	2.90E+01	6.10E+01	=
5	1.60E+01	1.60E+01	1.40E+01	1.40E+01	+
6	3.10E-07	1.10E-06	4.80E-08	9.70E-08	+
7	6.70E+01	6.70E+01	6.40E+01	6.40E+01	+
8	1.70E+01	1.70E+01	1.30E+01	1.40E+01	+
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	=
10	3.20E+03	3.10E+03	3.30E+03	3.20E+03	=
11	2.90E+01	2.80E+01	2.30E+01	2.40E+01	+
12	1.70E+03	1.70E+03	1.20E+03	1.20E+03	+
13	3.70E+01	3.10E+01	1.60E+01	2.70E+01	=
14	2.40E+01	2.50E+01	2.40E+01	2.40E+01	=
15	2.30E+01	2.40E+01	2.00E+01	2.10E+01	+
16	4.80E+02	4.50E+02	4.70E+02	4.50E+02	=
17	2.60E+02	2.80E+02	2.90E+02	3.00E+02	=
18	2.40E+01	2.40E+01	2.20E+01	2.30E+01	+
19	1.40E+01	1.40E+01	1.10E+01	1.10E+01	+
20	1.10E+02	1.40E+02	1.10E+02	1.60E+02	=
21	2.20E+02	2.20E+02	2.20E+02	2.20E+02	+
22	1.00E+02	1.50E+03	1.00E+02	1.80E+03	=
23	4.30E+02	4.30E+02	4.30E+02	4.30E+02	+
24	5.10E+02	5.10E+02	5.10E+02	5.10E+02	=
25	4.80E+02	4.80E+02	4.80E+02	4.80E+02	+
26	1.10E+03	1.10E+03	1.10E+03	1.10E+03	=
27	5.10E+02	5.10E+02	5.10E+02	5.10E+02	+
28	4.60E+02	4.60E+02	4.60E+02	4.60E+02	=
29	3.60E+02	3.60E+02	3.60E+02	3.60E+02	+
30	5.90E+05	6.00E+05	5.90E+05	6.00E+05	=

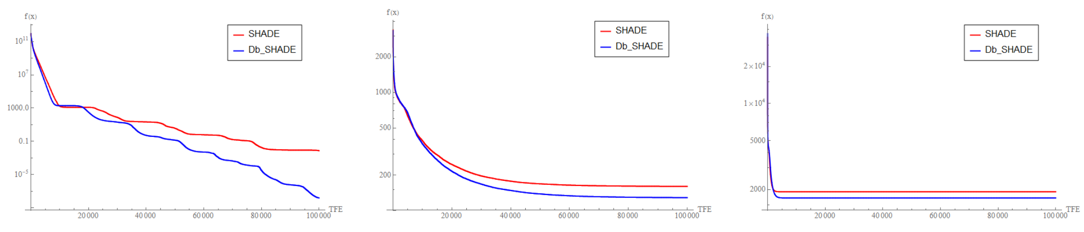

Figure 3: Comparison of selected average convergence between the SHADE and Db_SHADE algorithms on CEC2015 in $100D$. From $f2$, $f4$ and $f11$.

Table 16: jSO vs. DISH on CEC2017 in $100D$.

	jSO		DISH		
f	Median	Mean	Median	Mean	Result
1	0.00E+00	0.00E+00	0.00E+00	8.20E-10	=
2	4.70E-07	8.90E+00	2.90E-07	1.00E+04	=
3	1.50E-06	2.40E-06	1.10E-05	1.60E-05	-
4	2.00E+02	1.90E+02	2.00E+02	2.00E+02	=
5	4.40E+01	4.40E+01	2.80E+01	2.80E+01	+
6	3.60E-05	2.00E-04	4.30E-06	5.70E-06	+
7	1.40E+02	1.50E+02	1.30E+02	1.30E+02	+
8	4.20E+01	4.20E+01	2.90E+01	2.90E+01	+
9	0.00E+00	4.60E-02	0.00E+00	3.50E-03	+
10	9.80E+03	9.70E+03	9.80E+03	9.80E+03	=
11	1.00E+02	1.10E+02	5.20E+01	5.80E+01	+
12	1.70E+04	1.80E+04	1.10E+04	1.20E+04	+
13	1.40E+02	1.50E+02	1.10E+02	1.20E+02	+
14	6.40E+01	6.40E+01	4.00E+01	4.00E+01	+
15	1.70E+02	1.60E+02	7.80E+01	8.90E+01	+
16	1.90E+03	1.90E+03	1.90E+03	1.80E+03	=
17	1.30E+03	1.30E+03	1.30E+03	1.30E+03	=
18	1.60E+02	1.70E+02	9.50E+01	9.90E+01	+
19	1.10E+02	1.10E+02	5.20E+01	5.30E+01	+
20	1.40E+03	1.40E+03	1.50E+03	1.40E+03	=
21	2.60E+02	2.60E+02	2.50E+02	2.50E+02	+
22	1.10E+04	1.00E+04	1.10E+04	1.10E+04	=
23	5.70E+02	5.70E+02	5.70E+02	5.70E+02	+
24	9.00E+02	9.00E+02	8.90E+02	8.90E+02	+
25	7.60E+02	7.40E+02	7.10E+02	7.20E+02	+
26	3.30E+03	3.30E+03	3.10E+03	3.10E+03	+
27	5.90E+02	5.90E+02	5.70E+02	5.70E+02	+
28	5.20E+02	5.30E+02	5.20E+02	5.20E+02	=
29	1.20E+03	1.30E+03	1.30E+03	1.30E+03	=
30	2.30E+03	2.30E+03	2.30E+03	2.30E+03	+

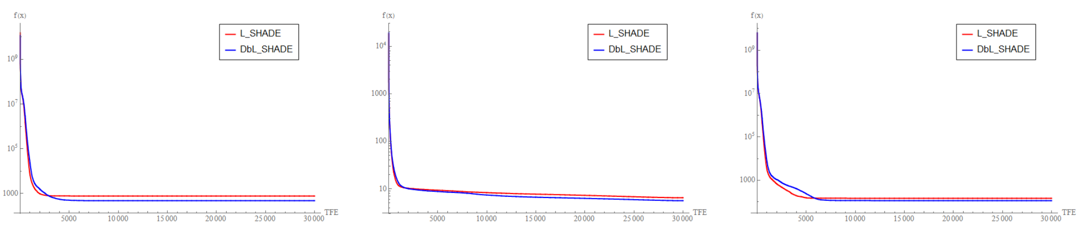

Figure 4: Comparison of selected average convergence between the L-SHADE and DbL_SHADE algorithms on CEC2015 in $30D$. From left $f6$, $f7$ and $f8$.

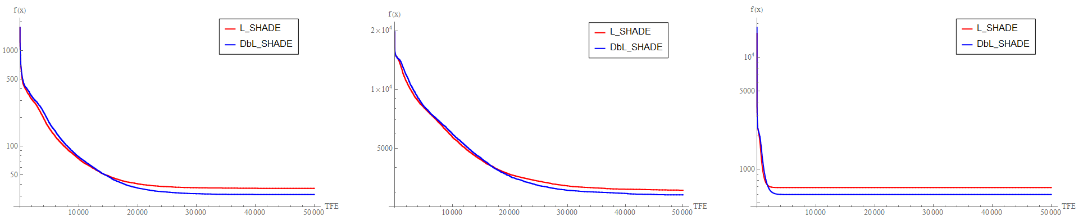

Figure 5: Comparison of selected average convergence between the L-SHADE and DbL_SHADE algorithms on CEC2015 in 50D. From left $f4$, $f5$ and $f11$.

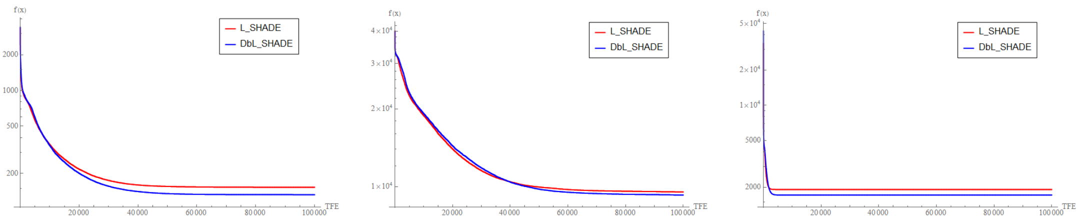

Figure 6: Comparison of selected average convergence between the L-SHADE and DbL_SHADE algorithms on CEC2015 in 100D. From left $f4$, $f5$ and $f11$.

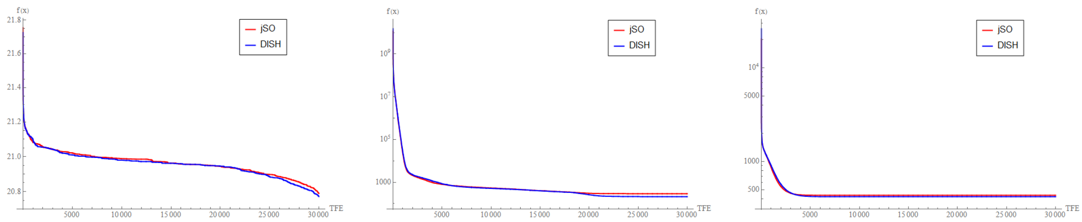

Figure 7: Comparison of selected average convergence between the jSO and DISH algorithms on CEC2015 in 30D. From left $f3$, $f6$ and $f11$.

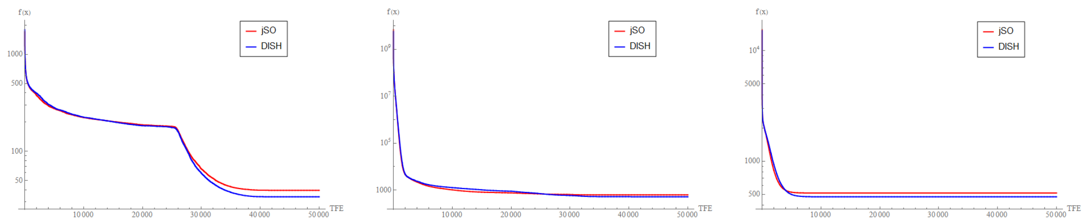

Figure 8: Comparison of selected average convergence between the jSO and DISH algorithms on CEC2015 in 50D. From left $f4$, $f8$ and $f11$.

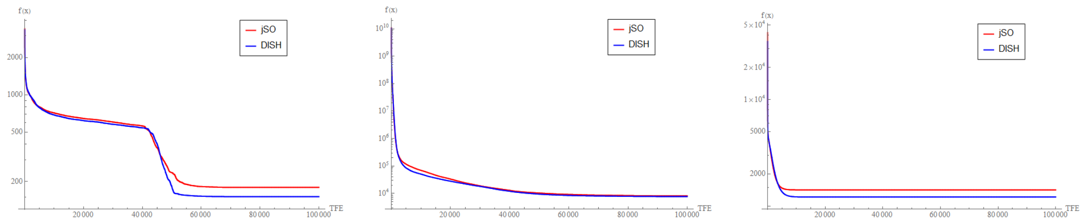

Figure 9: Comparison of selected average convergence between the jSO and DISH algorithms on CEC2015 in 100D. From left $f4$, $f8$ and $f11$.

Table 17: Clustering and population diversity on the CEC2015 in 10D.

	SHADE			Db_SHADE			L-SHADE			DbL_SHADE			jSO			DISH		
f	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD
1	51	1.01E+02	1.44E+01	51	1.16E+02	1.50E+01	51	1.02E+02	1.34E+01	51	1.23E+02	1.46E+01	51	1.22E+02	1.47E+01	51	1.27E+02	1.42E+01
2	51	6.25E+01	3.44E+01	51	7.03E+01	4.07E+01	51	6.01E+01	2.75E+01	51	7.01E+01	3.53E+01	51	6.55E+01	4.13E+01	51	6.72E+01	4.37E+01
3	0	-	-	3	6.47E+02	1.64E+02	0	-	-	5	8.84E+02	1.64E+02	2	1.68E+03	4.12E+00	1	2.11E+03	4.11E+00
4	0	-	-	1	5.48E+02	4.05E+01	2	3.45E+02	5.01E+01	4	3.23E+02	5.25E+01	51	1.54E+03	3.48E+01	51	1.48E+03	3.59E+01
5	0	-	-	0	-	-	0	-	-	0	-	-	50	1.91E+03	8.57E+01	50	2.01E+03	8.25E+01
6	47	6.42E+02	3.08E+01	49	6.56E+02	3.13E+01	48	5.15E+02	2.83E+01	49	5.70E+02	3.07E+01	51	1.10E+03	2.78E+01	51	1.07E+03	2.84E+01
7	1	7.01E+02	3.16E+01	0	-	-	0	-	-	0	-	-	51	1.46E+03	8.94E+00	51	1.55E+03	6.89E+00
8	51	5.07E+02	1.73E+01	51	5.97E+02	1.59E+01	51	4.52E+02	1.46E+01	51	5.26E+02	1.32E+01	51	7.20E+02	2.24E+01	51	7.04E+02	2.32E+01
9	0	-	-	0	-	-	0	-	-	2	8.97E+02	1.46E+01	0	-	-	0	-	-
10	51	1.63E+02	1.02E+01	51	1.96E+02	9.78E+00	51	1.71E+02	9.93E+00	51	2.15E+02	9.34E+00	51	2.06E+02	1.48E+01	51	2.17E+02	1.47E+01
11	33	1.82E+02	1.22E+01	42	2.03E+02	1.13E+01	35	1.60E+02	1.09E+01	39	1.75E+02	1.12E+01	51	2.91E+02	1.10E+01	51	2.68E+02	1.07E+01
12	0	-	-	0	-	-	11	1.47E+03	9.71E+00	12	1.54E+03	8.98E+00	0	-	-	0	-	-
13	0	-	-	0	-	-	0	-	-	0	-	-	47	2.01E+03	1.75E+01	49	2.12E+03	1.66E+01
14	51	8.47E+01	1.16E+01	51	7.77E+01	1.11E+01	51	7.06E+01	7.51E+00	51	7.48E+01	7.89E+00	51	8.35E+01	1.22E+02	51	7.83E+01	9.78E+01
15	51	5.42E+01	5.39E+00	51	6.15E+01	5.46E+00	51	5.76E+01	5.43E+00	51	6.65E+01	5.43E+00	51	5.88E+01	5.23E+00	51	6.04E+01	5.28E+00

Table 18: Clustering and population diversity on the CEC2015 in 30D.

	SHADE			Db_SHADE			L-SHADE			DbL_SHADE			jSO			DISH		
f	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD
1	51	1.74E+02	9.53E+00	51	2.51E+02	1.00E+01	51	1.51E+02	9.18E+00	51	2.22E+02	9.57E+00	51	2.61E+02	8.79E+00	51	2.93E+02	9.38E+00
2	51	7.63E+01	6.80E+00	51	9.50E+01	7.62E+00	51	7.06E+01	6.67E+00	51	9.05E+01	7.18E+00	51	1.13E+02	9.49E+00	51	1.31E+02	1.17E+01
3	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
4	0	-	-	0	-	-	0	-	-	0	-	-	51	4.02E+03	5.71E+01	51	4.06E+03	5.79E+01
5	0	-	-	0	-	-	0	-	-	0	-	-	50	5.43E+03	1.80E+02	51	5.54E+03	1.77E+02
6	51	2.62E+02	9.11E+00	51	4.90E+02	9.38E+00	51	2.00E+02	8.80E+00	51	3.65E+02	8.63E+00	51	1.93E+03	4.64E+01	51	2.40E+03	4.64E+01
7	0	-	-	2	1.40E+03	1.12E+01	7	4.21E+02	1.62E+01	12	6.66E+02	2.11E+01	51	4.21E+03	2.40E+01	51	4.39E+03	2.83E+01
8	51	5.45E+02	1.05E+01	51	8.91E+02	1.23E+01	51	4.26E+02	8.17E+00	51	6.43E+02	1.33E+01	51	3.26E+03	3.52E+01	51	3.30E+03	3.13E+01
9	0	-	-	0	-	-	2	5.82E+02	7.08E+00	0	-	-	23	4.82E+03	6.10E+00	23	4.86E+03	6.14E+00
10	51	3.65E+02	8.50E+00	51	5.01E+02	8.45E+00	51	3.18E+02	8.34E+00	51	4.75E+02	8.25E+00	51	1.22E+03	5.50E+01	51	1.25E+03	5.10E+01
11	51	1.21E+02	8.02E+00	51	1.57E+02	6.82E+00	51	1.13E+02	7.25E+00	51	1.39E+02	6.82E+00	51	1.99E+02	8.25E+00	51	2.24E+02	6.57E+00
12	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
13	0	-	-	0	-	-	0	-	-	0	-	-	43	6.83E+03	7.12E+01	40	7.10E+03	6.28E+01
14	51	1.15E+02	6.96E+00	51	1.44E+02	6.82E+00	51	1.08E+02	6.82E+00	51	1.40E+02	7.03E+00	51	1.94E+02	8.51E+00	51	2.21E+02	1.65E+01
15	51	9.94E+01	6.07E+00	51	1.20E+02	6.18E+00	51	9.36E+01	6.13E+00	51	1.12E+02	6.12E+00	51	1.36E+02	5.86E+00	51	1.53E+02	5.80E+00

Table 19: Clustering and population diversity on the CEC2015 in 50D.

	SHADE			Db_SHADE			L-SHADE			DbL_SHADE			jSO			DISH		
f	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD
1	51	2.17E+02	9.70E+00	51	3.05E+02	9.57E+00	51	1.90E+02	9.86E+00	51	3.04E+02	9.70E+00	51	3.34E+02	9.11E+00	51	4.08E+02	9.21E+00
2	51	9.07E+01	7.01E+00	51	1.09E+02	6.94E+00	51	8.48E+01	7.16E+00	51	1.06E+02	7.03E+00	51	1.43E+02	6.93E+00	51	1.67E+02	7.15E+00
3	0	-	-	0	-	-	0	-	-	1	8.86E+02	3.44E+02	5	1.44E+04	1.15E+02	4	1.27E+04	2.49E+02
4	0	-	-	0	-	-	1	4.86E+02	1.12E+01	0	-	-	51	5.81E+03	6.37E+01	51	5.76E+03	6.28E+01
5	0	-	-	0	-	-	0	-	-	0	-	-	51	8.16E+03	2.40E+02	51	8.54E+03	2.43E+02
6	51	4.72E+02	8.28E+00	51	7.72E+02	8.19E+00	51	3.02E+02	8.08E+00	51	3.69E+02	7.82E+00	51	4.09E+02	1.15E+01	51	5.27E+02	1.41E+01
7	30	4.45E+02	8.95E+00	13	7.71E+02	9.58E+00	41	2.76E+02	8.12E+00	35	5.87E+02	9.05E+00	51	3.58E+03	1.51E+01	51	4.99E+03	1.99E+01
8	50	1.22E+03	1.15E+01	49	1.37E+03	1.14E+01	51	6.35E+02	8.93E+00	51	8.43E+02	9.96E+00	51	1.26E+03	2.31E+01	51	2.23E+03	3.66E+01
9	3	8.21E+02	7.93E+00	0	-	-	11	5.72E+02	8.04E+00	1	9.29E+02	9.67E+00	43	5.74E+03	7.16E+00	34	6.71E+03	6.86E+00
10	51	4.71E+02	7.89E+00	51	5.73E+02	7.89E+00	51	3.78E+02	7.84E+00	51	5.60E+02	7.91E+00	51	8.36E+02	9.10E+00	51	1.42E+03	1.67E+01
11	51	1.27E+02	7.62E+00	51	1.63E+02	7.62E+00	51	1.17E+02	7.48E+00	51	1.56E+02	7.46E+00	51	2.43E+02	7.35E+00	51	2.91E+02	7.35E+00
12	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
13	0	-	-	0	-	-	0	-	-	0	-	-	47	1.12E+04	9.15E+01	46	1.08E+04	1.00E+02
14	51	1.58E+02	7.30E+00	51	1.98E+02	7.21E+00	51	1.41E+02	7.14E+00	51	1.78E+02	7.10E+00	51	2.26E+02	6.87E+00	51	2.62E+02	8.73E+00
15	51	1.59E+02	7.25E+00	51	1.72E+02	7.24E+00	51	1.38E+02	7.08E+00	51	1.56E+02	7.06E+00	51	2.03E+02	6.74E+00	51	2.26E+02	6.69E+00

Table 20: Clustering and population diversity on the CEC2015 in 100*D*.

	SHADE			Db_SHADE			L-SHADE			DbL_SHADE			jSO			DISH		
f	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD	#runs	Mean CO	Mean PD
1	51	2.40E+02	9.25E+00	51	3.07E+02	9.58E+00	51	1.90E+02	9.47E+00	51	2.79E+02	1.01E+01	51	3.81E+02	9.84E+00	51	4.65E+02	9.85E+00
2	51	1.92E+02	8.59E+00	51	1.96E+02	8.59E+00	51	1.56E+02	8.50E+00	51	1.75E+02	8.57E+00	51	2.37E+02	8.12E+00	51	2.62E+02	8.13E+00
3	0	-	-	0	-	-	1	1.48E+03	5.03E+02	1	1.34E+03	5.06E+02	3	2.65E+04	3.25E+02	2	2.65E+04	2.86E+02
4	0	-	-	2	7.25E+02	8.47E+00	0	-	-	1	4.44E+02	8.32E+00	51	6.19E+03	2.99E+01	51	6.10E+03	2.59E+01
5	0	-	-	0	-	-	0	-	-	0	-	-	51	1.45E+04	3.49E+02	51	1.46E+04	3.48E+02
6	51	2.64E+03	7.50E+00	51	4.02E+03	7.32E+00	51	2.02E+03	8.10E+00	51	3.50E+03	7.28E+00	51	2.95E+03	7.53E+00	51	3.82E+03	7.09E+00
7	37	3.87E+02	8.80E+00	26	7.45E+02	8.96E+00	47	2.62E+02	8.28E+00	45	5.35E+02	8.59E+00	51	3.63E+03	9.48E+00	51	6.95E+03	1.26E+01
8	33	6.99E+03	1.08E+01	7	7.64E+03	9.26E+00	51	4.95E+03	8.18E+00	51	7.19E+03	8.56E+00	51	5.54E+03	8.07E+00	51	5.94E+03	8.17E+00
9	10	5.70E+02	9.85E+00	0	-	-	7	7.65E+02	9.48E+00	5	7.57E+02	9.81E+00	50	7.55E+03	8.58E+00	48	1.02E+04	8.35E+00
10	51	1.05E+03	8.14E+00	51	1.04E+03	8.44E+00	51	6.13E+02	7.71E+00	51	8.20E+02	8.10E+00	51	9.96E+02	7.83E+00	51	1.11E+03	7.73E+00
11	51	1.50E+02	8.92E+00	51	1.96E+02	8.78E+00	51	1.34E+02	8.89E+00	51	1.85E+02	8.72E+00	51	3.08E+02	8.51E+00	51	3.86E+02	8.46E+00
12	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-	0	-	-
13	0	-	-	0	-	-	0	-	-	0	-	-	50	1.89E+04	1.73E+02	50	1.92E+04	1.71E+02
14	51	5.66E+02	7.74E+00	51	5.62E+02	7.80E+00	51	3.88E+02	7.53E+00	51	4.34E+02	7.48E+00	51	5.01E+02	7.23E+00	51	5.25E+02	7.21E+00
15	51	4.39E+02	8.85E+00	51	4.30E+02	8.71E+00	51	3.20E+02	8.60E+00	51	3.39E+02	8.78E+00	51	4.12E+02	8.39E+00	51	4.37E+02	8.36E+00

Table 21: Time complexity according to the CEC2015 benchmark - SHADE vs. Db_SHADE.

			SHADE		Db_SHADE	
D	*T0*	*T1*	*T2*	*(T2 - T1)/T0*	*T2*	*(T2 - T1)/T0*
10	2.53E+02	4.02E+02	1.22E+04	4.68E+01	1.22E+04	4.68E+01
30	2.53E+02	1.36E+03	1.52E+04	5.47E+01	1.52E+04	5.49E+01
50	2.53E+02	2.52E+03	1.87E+04	6.41E+01	1.82E+04	6.19E+01
100	2.53E+02	6.49E+03	2.88E+04	8.82E+01	2.79E+04	8.46E+01

Table 22: Time complexity according to the CEC2015 benchmark - L-SHADE vs. DbL_SHADE.

			L-SHADE		DbL_SHADE	
D	*T0*	*T1*	*T2*	*(T2 - T1)/T0*	*T2*	*(T2 - T1)/T0*
10	2.53E+02	4.02E+02	1.20E+04	4.60E+01	1.20E+04	4.58E+01
30	2.53E+02	1.36E+03	2.53E+04	9.45E+01	2.49E+04	9.32E+01
50	2.53E+02	2.52E+03	4.39E+04	1.64E+02	4.31E+04	1.61E+02
100	2.53E+02	6.49E+03	1.31E+05	4.92E+02	1.29E+05	4.85E+02

Table 23: Time complexity according to the CEC2015 benchmark - jSO vs. DISH.

			jSO		DISH	
D	*T0*	*T1*	*T2*	*(T2 - T1)/T0*	*T2*	*(T2 - T1)/T0*
10	2.53E+02	4.02E+02	1.24E+04	4.72E+01	1.23E+04	4.72E+01
30	2.53E+02	1.36E+03	2.50E+04	9.34E+01	2.41E+04	8.98E+01
50	2.53E+02	2.52E+03	4.02E+04	1.49E+02	3.88E+04	1.43E+02
100	2.53E+02	6.49E+03	9.06E+04	3.32E+02	8.91E+04	3.27E+02

Table 24: Friedman ranks for algorithms over CEC2015 (aggregated 10D functions).

Rank	Name	F-rank
0	DISH	3.2
1	jSO	3.3
2	DbL-SHADE	3.4
3	L-SHADE	3.6
4	SHADE	3.6
5	Db_SHADE	4.0

Table 25: Friedman ranks for algorithms over CEC2015 (aggregated 30D functions).

Rank	Name	F-rank
0	DISH	2.8
1	jSO	3.0
2	DbL-SHADE	3.1
3	Db_SHADE	3.6
4	L-SHADE	3.8
5	SHADE	4.6

Table 26: Friedman ranks for algorithms over CEC2015 (aggregated 50D functions).

Rank	Name	F-rank
0	DISH	2.6
1	DbL-SHADE	3.0
2	jSO	3.3
3	Db_SHADE	3.6
4	L-SHADE	4.0
5	SHADE	4.5

Table 27: Friedman ranks for algorithms over CEC2015 (aggregated 100D functions).

Rank	Name	F-rank
0	DISH	2.9
1	jSO	3.1
2	DbL-SHADE	3.4
3	L-SHADE	3.4
4	Db_SHADE	3.6
5	SHADE	4.6

Table 28: Friedman ranks for algorithms over CEC2015 (aggregated all 10D, 30D, 50D, and 100D functions).

Rank	Name	F-rank
0	DISH	2.9
1	DbL-SHADE	3.2
2	jSO	3.2
3	Db_SHADE	3.7
4	L-SHADE	3.7
5	SHADE	4.3

Table 29: Friedman test post-hoc procedures for $\alpha = 0.05$, comparing the CEC2015 aggregate median results ($D = 10$), the reference algorithm is the top performing DISH algorithm. Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value ≤0.01 and for Holm's p ≤0.01, Hommel's p ≤0.01, Holland's p ≤0.0102, Finner's p ≤0.0102, and Li's p ≤0.00814, respectively.

Rank (k)	Algorithm	$z = \frac{R_0 - R_k}{SE}$	p	Holm/Hochberg/Hommel	Holland	Rom	Finner	Li
5	Db_SHADE	1.17	0.242	**0.01**	**0.0102**	0.0105	**0.0102**	**0.00814**
4	SHADE	0.634	0.526	0.0125	0.0127	0.0131	0.0203	**0.00814**
3	L-SHADE	0.586	0.558	0.0167	0.017	0.0167	0.0303	**0.00814**
2	DbL-SHADE	0.342	0.733	0.025	0.0253	0.025	0.0402	**0.00814**
1	jSO	0.195	0.845	0.05	0.05	0.05	0.05	0.05

Table 30: Friedman test post-hoc procedures for $\alpha = 0.05$, comparing the CEC2015 aggregate median results ($D = 30$), the reference algorithm is the top performing DISH algorithm. Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value ≤0.01 and for Holm's p ≤0.0125, Hochberg's p ≤0.01, Hommel's p ≤0.0125, Holland's p ≤0.0127, Rom's p ≤0.0105, Finner's p ≤0.0203, and Li's p ≤0.0121, respectively.

Rank (k)	Algorithm	$z = \frac{R_0 - R_k}{SE}$	p	Holm/Hochberg/Hommel	Holland	Rom	Finner	Li
5	SHADE	2.63	**0.00842**	**0.01**	**0.0102**	**0.0105**	**0.0102**	**0.0121**
4	L-SHADE	1.42	0.157	**0.0125**	**0.0127**	0.0131	0.0203	**0.0121**
3	Db_SHADE	1.07	0.283	0.0167	0.017	0.0167	0.0303	**0.0121**
2	DbL-SHADE	0.439	0.661	0.025	0.0253	0.025	0.0402	**0.0121**
1	jSO	0.293	0.77	0.05	0.05	0.05	0.05	0.05

Table 31: Friedman test post-hoc procedures for $\alpha = 0.05$, comparing the CEC2015 aggregate median results ($D = 50$), the reference algorithm is the top performing DISH algorithm. Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value ≤0.01 and for Holm's p ≤0.0125, Hochberg's p ≤0.01, Hommel's p ≤0.0125, Holland's p ≤0.0127, Rom's p ≤0.0105, Finner's p ≤0.0203, and Li's p ≤0.0215, respectively.

Rank (k)	Algorithm	$z = \frac{R_0 - R_k}{SE}$	p	Holm/Hochberg/Hommel	Holland	Rom	Finner	Li
5	SHADE	2.73	**0.00629**	**0.01**	**0.0102**	**0.0105**	**0.0102**	0.0215
4	L-SHADE	2	0.0454	**0.0125**	**0.0127**	0.0131	0.0203	0.0215
3	Db_SHADE	1.37	0.172	0.0167	0.017	0.0167	0.0303	0.0215
2	jSO	0.976	0.329	0.025	0.0253	0.025	0.0402	0.0215
1	DbL-SHADE	0.537	0.591	0.05	0.05	0.05	0.05	0.05

Table 32: Friedman test post-hoc procedures for $\alpha = 0.05$, comparing the CEC2015 aggregate median results ($D = 100$), the reference algorithm is the top performing DISH algorithm. Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value ≤ 0.01, and for Holm's $p \leq 0.01$, Hommel's $p \leq 0.01$, Holland's $p \leq 0.0102$, Finner's $p \leq 0.0102$, and Li's $p \leq 0.0101$, respectively.

Rank (k)	Algorithm	$z = \frac{R_0 - R_k}{SE}$	p	Holm/Hochberg/Hommel	Holland	Rom	Finner	Li
5	SHADE	2.44	0.0147	**0.01**	**0.0102**	0.0105	**0.0102**	**0.0101**
4	Db_SHADE	0.976	0.329	0.0125	0.0127	0.0131	0.0203	**0.0101**
3	L-SHADE	0.683	0.495	0.0167	0.017	0.0167	0.0303	**0.0101**
2	DbL-SHADE	0.634	0.526	0.025	0.0253	0.025	0.0402	**0.0101**
1	jSO	0.244	0.807	0.05	0.05	0.05	0.05	0.05

Table 33: Friedman test post-hoc procedures for $\alpha = 0.05$, comparing the CEC2015 aggregate median results ($D = 10$, $D = 30$, $D = 50$, and $D = 100$ together), the reference algorithm is the top performing DISH algorithm. Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value ≤ 0.01 and for Holm's $p \leq 0.0125$, Hochberg's $p \leq 0.01$, Hommel's $p \leq 0.0166$, Holland's $p \leq 0.0127$, Rom's $p \leq 0.0105$, Finner's $p \leq 0.0402$, and Li's $p \leq 0.0319$, respectively.

Rank (k)	Algorithm	$z = \frac{R_0 - R_k}{SE}$	p	Holm/Hochberg/Hommel	Holland	Rom	Finner	Li
5	SHADE	4.22	**2.43e-05**	**0.01**	**0.0102**	**0.0105**	**0.0102**	0.0319
4	L-SHADE	2.34	0.0192	**0.0125**	**0.0127**	0.0131	0.0203	0.0319
3	Db_SHADE	2.29	0.0218	0.0167	0.017	0.0167	0.0303	0.0319
2	DbL-SHADE	0.976	0.329	0.025	0.0253	0.025	0.0402	0.0319
1	jSO	0.854	0.393	0.05	0.05	0.05	0.05	0.05

6. Results Discussion

Firstly, the results for the CEC2015 benchmark set is discussed here. As can be seen in Tables 1, 5 and 9, there is not much of an improvement in the performance of the algorithm in lower (10D) dimensional objective space (in all cases combined – 1 improvement and 1 worsening), but the situation is much more interesting in 30 (Tables 2, 6 and 10), 50 (Tables 3, 7 and 11) and 100 (Tables 4, 8 and 12) dimensional objective space. There, the score is 5 improvements against 1 worsening (30D), 6 improvements and 2 worsening (50D), 5 improvements and 1 worsening (100D) in the case of SHADE, 5 improvements against 1 worsening (30D), 9 improvements against 1 worsening (50D), 6 improvements and 2 worsening (100D) in the case of L-SHADE, and 3 improvements (30D), 7 improvements (50D), 6 improvements against 2 worsening (100D) in the case of jSO. Such results lend weight to the assumption that the modification might be useful for preserving exploration ability in higher dimensional objective spaces.

The Friedman ranking in Tables 24–28 and their corresponding post-hoc analyses in Tables 29–33, respectively, further clarify how the DISH (and also DbL_SHADE) improves in ranks considering the compared algorithms and varying through dimensionality. As it might have been not so easy to see from counting Wilcoxon rank-sum outcomes, based on the relative Friedman ranks' improvement in Tables 24–28, the DISH algorithm performance mostly improves gradually with higher dimensions, taking values 3.2, 2.8, 2.6, and 2.9 for the $D = 10$, $D = 30$, $D = 50$, and $D = 100$, respectively. Several overall improvements are also significant, as reported based on the post-hoc procedures in Tables 29–33.

As for the CEC2017 benchmark set, experiments were performed only for the jSO algorithm (winner of the 2017 competition) to confirm the efficiency and robustness of the proposed approach. Results are presented for 10, 30, 50 and 100 dimensional settings in Tables 13, 14, 15, and 16, and they are discussed in detail below.

The findings can be summarized as follows:

- The presented convergence plots (Figures 1–9) show the initially similar convergence of the distance based variants, which, in later stages of the optimization process, nonetheless reach better objective function values.

- The numerical analyses in Tables 17, 18, 19, and 20 also support the introduction of distance based variants. These analyses show for the higher dimensional settings (30D, 50D and 100D), that, for almost all cases (except $f9$ for 50D), the clusters are detected in later stages of a distance based variants' run (see parameter Mean CO). For the original SHADE/L-SHADE/jSO versions, the clustering inside the population is detected earlier. As already mentioned, the lowest dimensional settings of 10D has revealed mixed results. The clusters are occurring even sooner for Db variants (Table 17, $f14$ for SHADE strategies pair, $f11$ for the L-SHADE strategies pair). Furthermore, the occurrence of clusters is not so symmetric between the pairs as for higher dimensions.

- It is possible to link the significantly improving ranking results from Tables 24–28 and clustering analysis in Tables 17–20. Improvements of the results in Tables 2, 3, 4, 6, 7, 8, 10, 11, and 12 denoted by the "+" symbol are always connected with occurrence of later clustering, none at all, or a lesser amount of cluster occurrence from 51 total instances (for the last option, see, for example, column #runs in Table 19, $f7$, both SHADE and L-SHADE pairs). Thus, the higher population diversity is maintained for a higher amount of iterations.

- The clustering analysis reveals the pattern, that the mean population diversity (Tables 17–20) is similar for the compared pairs of SHADE vs. Db_SHADE, L-SHADE vs. DbL_SHADE, and jSO vs. DISH. But, it is necessary to keep in mind, that it was calculated at the time of clusters' first occurrence, which in the case of distance based parameter adaptation, is later in the optimization process. Therefore, the algorithms reach the same stage of optimization, but at different times, which is the evidence of the more extended exploration phase of the distance based variants, and shows that the proposed distance based design of parameter control is effective.

- For the unimodal test functions, the distance based approach proved not to be beneficial. For those cases, the prolonged exploration is not suitable, since the earlier population clustering supports the convergence towards the global extreme. This drawback is further supported by significantly worse optimization results of Db_SHADE (Table 3), DbL_SHADE (Table 7), and jSO (Table 11) for $f1$ in 50D.

- Results for the CEC2017 benchmark set with jSO and DISH counterparts lend weight to the argument that the proposed approach is robust and improves the performance of SHADE family based algorithms significantly in higher dimensions. From Tables 13 – 16, it is possible to see the similar pattern of increasing performance, together with the gradual increase of dimensionality. There is almost no significant difference in the performance for 10 and 30D. For higher dimensional settings of 50 and 100D, there is present a clearly visible boost to the performance of DISH, resulting in 14 wins and zero losses (50D, Table 15), and 19 wins and 1 loss (100D, Table 16). Overall, the results are very convincing, as the winning algorithm of the CEC 2017 competition has been improved significantly.

7. Conclusion

In this paper, a relatively simple and straightforward modification is proposed and tested to the control parameter adaptation in SHADE. The basic idea is that an adaptation mechanism based on the change in position rather than a change in objective function value, may help to avoid the premature convergence of the algorithm in higher dimensional spaces. This idea is confirmed experimentally in the case of the optimization of the CEC2015 benchmark set in 10, 30, 50, and 100 dimensions. Also, the most up-to-date DE variant - jSO enhanced by distance based parameter adaptation is tested on the CEC2017 benchmark set in 10, 30, 50, and 100 dimensions.

The most important feature of this modification is that it can be used easily as a framework, and implemented into any of the SHADE-based state-of-the-art variants. On the other hand, the drawback of this method is a slightly higher computational complexity in determining the weights for historical memory updates of control parameter arrays (distance computation against the simple difference between objective function values). However, time complexity measurements have shown that this drawback plays no role in the overall computation time.

The future work will focus on further experimentation with the proposed modification and applying it to more algorithms and challenges in optimization. For example, the proposed approach might also be useful for constrained problems, where constrained areas would be addressed by increased changes of an individual's components.

Acknowledgments

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project No. LO1303 (MSMT-7778/2014), supported by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089, and supported by the Internal Grant Agency of Tomas Bata University under the Project No. IGA/CebiaTech/2018/003. This work was also funded in part by the Slovenian Research Agency, Project No.: P2-0041. This work is also based upon support by COST (European Cooperation in Science & Technology) under Action CA15140, Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO), and Action IC406, High-Performance Modelling and Simulation for Big Data Applications (cHiPSet), and further supported by resources of the A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin (ailab.fai.utb.cz).

References

[1] K. Price, R. Storn, Differential Evolution-a simple and efficient adaptive scheme for global optimization over continuous space, Technical Report, International Computer Science Institute.

[2] S. Das, S. S. Mullick, P. N. Suganthan, Recent advances in differential evolution – an updated survey, Swarm and Evolutionary Computation 27 (2016) 1–30.

[3] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Bošković, Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 646–657.

[4] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 398–417.

[5] S. Das, A. Abraham, U. K. Chakraborty, A. Konar, Differential Evolution Using a Neighborhood-Based Mutation Operator, IEEE Transactions on Evolutionary Computation 13 (3) (2009) 526–553.

[6] E. Mininno, F. Neri, F. Cupertino, D. Naso, Compact Differential Evolution, IEEE Transactions on Evolutionary Computation 15 (1) (2011) 32–54.

[7] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, M. F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Applied Soft Computing 11 (2) (2011) 1679–1696.

[8] B.-Y. Qu, P. N. Suganthan, J.-J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Transactions on Evolutionary Computation 16 (5) (2012) 601–614.

[9] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P. N. Suganthan, Ensemble of differential evolution variants, Information Sciences 423 (2018) 172–186.

[10] J. Brest, P. Korošec, J. Šilc, A. Zamuda, B. Bošković, M. S. Maučec, Differential evolution and differential ant-stigmergy on dynamic optimisation problems, International Journal of Systems Science 44 (4) (2013) 663–679.

[11] R. D. Al-Dabbagh, F. Neri, N. Idris, M. S. Baba, Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy, Swarm and Evolutionary Computation. DOI: https://doi.org/10.1016/j.swevo.2018.03.008.

[12] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: 2013 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2013, pp. 71–78.

[13] J. Zhang, A. C. Sanderson, JADE: Adaptive Differential Evolution With Optional External Archive, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 945–958.

[14] R. Tanabe, A. S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 1658–1665.

[15] 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: 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2015, pp. 1003–1010.

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

[17] J. Brest, M. S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jSO, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 1311–1318.

[18] A. P. Piotrowski, J. J. Napiorkowski, Step-by-step improvement of JADE and SHADE-based algorithms: Success or failure?, Swarm and Evolutionary Computation. DOI: https://doi.org/10.1016/j.swevo.2018.03.007.

[19] A. P. Piotrowski, aL-SHADE optimization algorithms with population-wide inertia, Information Sciences.

[20] P. N. Suganthan, Particle swarm optimiser with neighbourhood operator, in: 1999 IEEE Congress on Evolutionary Computation (CEC), Vol. 3, IEEE, 1999, pp. 1958–1962.

[21] F. Neri, V. Tirronen, Recent Advances in Differential Evolution: A Survey and Experimental Analysis, Artificial Intelligence Review 33 (1–2) (2010) 61–106.

[22] S. Das, S. Maity, B.-Y. Qu, P. N. Suganthan, Real-parameter evolutionary multimodal optimization – A survey of the state-of-the-art, Swarm and Evolutionary Computation 1 (2) (2011) 71–88.

[23] E. Mezura-Montes, C. A. C. Coello, Constraint-handling in nature-inspired numerical optimization: Past, present and future, Swarm and Evolutionary Computation 1 (4) (2011) 173–194.

[24] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm and Evolutionary Computation 1 (1) (2011) 32–49.

[25] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: A literature review, Swarm and Evolutionary Computation.

[26] A. Zamuda, J. Brest, Self-adaptive control parameters' randomization frequency and propagations in differential evolution, Swarm and Evolutionary Computation 25 (2015) 72–99.

[27] A. P. Piotrowski, Review of Differential Evolution population size, Swarm and Evolutionary Computation 32 (2017) 1–24.

[28] D. Zaharie, Influence of crossover on the behavior of Differential Evolution Algorithms, Applied Soft Computing 9 (3) (2009) 1126–1138.

[29] M. Weber, F. Neri, V. Tirronen, A study on scale factor/crossover interaction in distributed differential evolution, Artificial Intelligence Review 39 (3) (2013) 195–224.

[30] I. Poikolainen, F. Neri, F. Caraffini, Cluster-Based Population Initialization for differential evolution frameworks, Information Sciences 297 (2015) 216–235.

[31] K. R. Opara, J. Arabas, Differential Evolution: A survey of theoretical analyses, Swarm and Evolutionary Computation. DOI: https://doi.org/10.1016/j.swevo.2018.06.010.

[32] G. Karafotias, M. Hoogendoorn, A. Eiben, Parameter Control in Evolutionary Algorithms: Trends and Challenges, Evolutionary Computation, IEEE Transactions on 19 (2) (2015) 167–187.

[33] A. P. Piotrowski, J. J. Napiorkowski, Some metaheuristics should be simplified, Information Sciences 427 (2018) 32–62.

[34] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, A. Zamuda, Distance based parameter adaptation for differential

evolution, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–7.
[35] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, in: Technical Report, Nanyang Technological University Singapore, 2016.
[36] J. Brest, M. S. Maučec, B. Bošković, iL-SHADE: Improved L-SHADE Algorithm for Single Objective Real-Parameter Optimization, in: 2016 IEEE World Congress on Computational Intelligence (IEEE WCCI 2016), Vancouver, Canada, 2016, pp. 1188–1195.
[37] J. Liang, B. Qu, P. Suganthan, Q. Chen, Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-based Real-Parameter Single Objective Optimization, Technical report 201411a, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Nanyang Technological University, Singapore (2014).
[38] A. Zamuda, M. Nicolau, C. Zarges, A black-box discrete optimization benchmarking (bb-dob) pipeline survey: taxonomy, evaluation, and ranking, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO 2018), Vol. ??, 2018, pp. 1777–1782.
[39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, Vol. 96, 1996, pp. 226–231.
[40] M. M. Deza, E. Deza, Encyclopedia of Distances, in: Encyclopedia of Distances, Springer, 2009, pp. 1–583.
[41] R. Poláková, J. Tvrdík, P. Bujok, R. Matoušek, Population-size adaptation through diversity-control mechanism for differential evolution, in: MENDEL, 22th International Conference on Soft Computing, 2016, pp. 49–56.
[42] J. Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, The Journal of Machine Learning Research 7 (2006) 1–30.