Information Sciences 427 (2018) 32–62

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Some metaheuristics should be simplified

Adam P. Piotrowski*, Jaroslaw J. Napiorkowski

Institute of Geophysics, Polish Academy of Sciences, Ks. Janusza 64, 01-452 Warsaw, Poland

ARTICLE INFO

Article history:
Received 15 March 2017
Revised 1 August 2017
Accepted 15 October 2017
Available online 16 October 2017

Keywords:
Differential evolution
Evolutionary algorithms
Swarm intelligence
Structural bias
CMA-ES
Complexity

ABSTRACT

Users have a large and constantly increasing number of optimization metaheuristics at their disposal. In pursuit of ever better approaches, popular and successful methods are often improved a number of times in a row, or various methods are hybridized. One should, however, pay attention to whether such deliberately improved or hybridized methods do not become unnecessarily complicated, or if some of their elements do not introduce a structural bias. In the first case the algorithm should be simplified by eliminating the unneeded elements. This will make it more clear to the users, and, as shown in this study, may even improve the results. In the second case, operators that are responsible for over-frequent sampling of some part of the search space have to be removed.

In this study we address the problem of over-complexity of metaheuristics, focusing on two joint winners of the CEC2016 competition on single-objective numerical optimization, L-SHADE-EpSin and UMOEA-II algorithms. It is shown that each of them includes a procedure which introduces a structural bias by attracting population towards the origin **O**. As discussed in the text, in the case of seven out of 30 benchmark problems considered in the CEC2016 competition the objective function values in the origin **O** are better than those found by the vast majority of algorithms, hence structural bias affects the results obtained by L-SHADE-EpSin and UMOEA-II. In this work we simplify both algorithms by removing operators that are the main cause of structural bias. Further, we show that in the L-SHADE-EpSin algorithm the Ensemble Sinusoidal adaptation mechanism of control parameter *F*, that is used during the first half of the search, is not needed. Slightly better results on both artificial benchmarks used in the CEC2016 competition and on a wide set of real-world problems may be obtained if, during that period of the search, the value of *F* is simply set to 0.5. Such a simplified algorithm is competitive against a large number of metaheuristics and may be much easier for the users to understand.

1. Introduction

When developing new population-based metaheuristics one may either start from scratch, attempt to improve an already known method, or hybridize properties of various approaches [4,22]. As more and more metaheuristics are proposed, the first approach becomes less likely to be successful, and may easily lead to reduplication of already existing algorithms under a novel name [44]. Indeed, in a number of recent papers researchers expressed concerns whether we need yet another apparently "novel" types of metaheuristics [13,14,44,54], especially since publication of the proof of No Free Lunch Theorems [55]. Hybridization of existing methods or improvement of the selected one by adding new elements are often

* Corresponding author.
E-mail address: adampp@igf.edu.pl (A.P. Piotrowski).

https://doi.org/10.1016/j.ins.2017.10.039
0020-0255/

simpler, have a higher chance of success and better position investigators' work in the context of current research. However, opinions on such approaches are highly diversified; they have been referred to as "adding stuff" in [22], but also praised as "excellent evolution" in [16]. Putting aside such slightly subjective comments, we must admit that adding something new to the already existing algorithms is almost always performed without an in-depth analysis of whether all older operators or procedures are still needed if coupled with the newer ones. Note that discussing the main idea, or philosophy behind the proposed novelty, like for example in [23], is almost never sufficient. As a result, after a few such consecutive additions the novel methods frequently become unnecessarily over-complicated. Both justification and supposed cooperation between their operators may become unclear even to the authors. Some elements of such constructed algorithms may affect the clarity of the method discouraging its potential users, or lead to mistakes during implementation [3]. What is even worse, the more complicated algorithm, the higher the risk that some of its elements introduce structural bias [25] that leads to artificially sampling some part of the search space more frequently than the others.

As an example imagine that, in a specific procedure, the locations of the current individuals along the coordinates in the search space are multiplied by 0.5. In such a trivial example the population of structurally biased algorithm will be attracted towards the origin $\mathbf{O}=[0^1, 0^2, ..., 0^D]$ (where D is the problem dimensionality) without any reason justified by the fitness landscape. This may artificially help finding good solutions if they are close to the origin or negatively affect the search by wasting the function calls if the solutions in and around the origin are poor. Using an empirical approach proposed in [25], in [35] it was shown that a weak structural bias is present in a large number of metaheuristics (although not in all of them), but its causes are often complicated and hard to find. However, some algorithms are very strongly biased. If the sources of their bias can be determined, they should be removed.

In the present study we address both the problem of structural bias and unnecessary complication of metaheuristics. As an example we use two algorithms that have been declared joint winners of the CEC2016 competition on single-objective numerical optimization, namely L-SHADE-EpSin [1] and UMOEA-II [12]. To facilitate the discussion, our key results are organized into three main sections (2-4).

First, in Section 2 we briefly discuss L-SHADE-EpSin and UMOEA-II algorithms and trace the history of their developments.

In Section 3 we address the problem of structural bias. We show that specific operators of both L-SHADE-EpSin and UMOEA-II are structurally biased and under specific circumstances lead the population "artificially" towards the origin **O**. The presence of structurally biased operators in both algorithms affect the results of the CEC2016 competition, as in seven out of 30 benchmark minimization problems used in CEC2016 the values of objective function in the neighbourhood of the origin are better than the values that are found by almost all unbiased algorithms. We removed such structurally biased parts of L-SHADE-EpSin and UMOEA-II, and report the results obtained.

Then, in Section 4 we show that the main novelty of the L-SHADE-EpSin algorithm, a complicated mechanism of Ensemble Sinusoidal adaptation of control parameter F that is applied during the first half of the search, is not needed, and that slightly better results, both on CEC2014 benchmarks [27] (that were used during the CEC2016 competition) and on 22 real-world problems from various fields of science [6] may be obtained by setting F to 0.5 for the first half of the search. Hence, the only successful novelty of L-SHADE-EpSin compared with L-SHADE proposed in 2014 by Tanabe and Fukunaga [47] is that in L-SHADE-EpSin the parameter F is managed differently during the first and the second half of the search.

Finally, comparison between such simplified algorithms and various kinds of metaheuristics is given in Section 5. Conclusions are summarized in Section 6.

2. Step-by-step development of L-SHADE-EpSin and UMOEA-II

In Section 2.1 we discuss the complicated history of the development of L-SHADE-EpSin algorithm, and define its specific features. In Section 2.2 we briefly discuss the UMOEA-II algorithm and point to its operators that introduce structural bias.

2.1. L-SHADE-EpSin

The L-SHADE-EpSin algorithm [1] is a good example of a method that has been developed by step-by-step improvements of the previous variants, starting from the Differential Evolution algorithm invented by Storn and Price in 1990's [45]. To discuss the details of L-SHADE-EpSin and understand its features, we should start from its predecessors.

2.1.1. Differential evolution

Without the loss of generality we consider numerical minimization problems by searching for the global optimum solution $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) = \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \tag{1}$$

where $f(\mathbf{x})$ is the real-valued function, $\mathbf{x}$ is a D-dimensional vector, $\mathbf{x} = \{x^1, ..., x^D\}$with the domain $\Omega\subseteq\mathbf{R}^D$. L-SHADE-EpSin is based on L-SHADE [47], that itself is a much modified version of population-based Differential Evolution (DE) algorithm [45], the starting point for our discussion. In DE, in generation $g=0$, NP individuals (solution vectors) $\mathbf{x}_{i,g} = \{x^1_{i,g}, ..., x^D_{i,g}\}$, $i=1,...,NP$ are randomly initialized from the uniform distribution

$$x^j_{i,0} = L^j + rand^j_i(0,1)\cdot(U^j - L^j);\ j = 1, ..., D;\ i = 1, ..., NP \tag{2}$$

Basic DE pseudocode:

1. set generation counter $g = 0$ and problem-related information: D (problem dimensionality), $MNFC$ (maximum number of function calls);
2. set control parameters: F, CR, NP (user specified, no general rule is available [45]);
3. set L^d and U^d as lower and upper bounds of d-th variable ($d = 1,\ldots,D$);
4. initialize population $\mathbf{P}$ of individuals $\mathbf{x}_{i,g}$ according to eq. (2);
5. find objective function values $f(\mathbf{x}_{i,g})$ for all individuals in $\mathbf{P}$;
6. set the number of function calls used $NFC = NP$;
7. **while** $NFC < MNFC$
8. $g = g + 1$;
9. **for** $i = 1$: NP
10. perform mutation following eq. (3);
11. perform crossover following eq. (4);
12. perform bounds handling and selection (eq. (5));
13. $NFC = NFC + 1$;
14. **end for;**
15. **end while;**

Fig. 1. The pseudocode of basic DE algorithm.

where $rand_i{}^j(0,1)$ is a random value within [0,1] interval generated separately for each jth element of ith individual and L^j and U^j are pre-set bounds that define the subset $\prod_{j=1}^{D}[L^j, U^j]$ of the search space $\mathbf{R}^D$. In each subsequent generation g for each individual, the basic DE performs mutation

$$\mathbf{v}_{i,g} = \mathbf{x}_{r1,g} + F \cdot (\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}) \tag{3}$$

crossover

$$u_{i,g}^{j} = \begin{cases} v_{i,g}^{j} & if \quad rand_i^j(0,1) \le CR \quad or \quad j = j_{rand,i} \\ x_{i,g}^{j} & otherwise \end{cases} \tag{4}$$

followed by some boundary handling [21], and selection operations

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & if \quad f(\mathbf{u}_{i,g}) \le f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & otherwise \end{cases} \tag{5}$$

In Eq. (3) $r1$, $r2$ and $r3$ are randomly selected integers from the range [1, NP], such that $r1 \neq r2 \neq r3 \neq i$. In Eq. (4) $j_{rand,i}$ is a randomly selected integer from [1, D] range; CR (Eq. 4) and F (Eq. 3) are two additional control parameters of the algorithm (next to population size NP) that are to be defined by the user. After all the above procedures, the NP individuals for the $g+1$ generation are created. The algorithm performs generation by generation until stopping conditions are reached, often defined as the maximum number of function calls ($MNFC$). The pseudocode of the basic DE algorithm is given in Fig. 1.

Currently a large number of DE variants exist and many of them have been developed by modifying the previous ones. For details, see review papers [7,30]. Here we are interested in the path that led to the concept of L-SHADE-EpSin.

One of the main research goals in DE is setting the efficient adaptation rules for DE control parameters and proposing more efficient mutation strategies. In the JADE variant introduced in 2009, Zhang and Sanderson [57] proposed both a new DE mutation strategy and a concept of specific parameter adaptation rules that updates F and CR values in each generation in such a way that they would differ for each individual (hence i index is added, F_i and CR_i) and promote those that are successful in generating offspring that win the selection procedure.

Soon after, in [32], the more complicated variant of the JADE adaptation scheme was proposed, which takes into account the rate of improvement of each successful individual and its control parameter values. Later, in 2013, Tanabe and Fukunaga [46] proposed SHADE, which extends the JADE concept from [57] with the modification given in [32], by introducing a memory that stores a number of successful historical values of DE control parameters. All DE variants mentioned above used a fixed user-specified population size, although in the meantime a number of DE versions with variable population size have been proposed (see a review in [36]). In 2014 the variant of SHADE with linear population size reduction (L-SHADE) was introduced [47]. Without any links to SHADE, another control parameter adaptation method based on sinusoidal formulas (SinDE) was proposed in [8]. The L-SHADE-EpSin algorithm is the modification of L-SHADE (that, as discussed above, has itself a long and complicated history of small step-size improvements) in that sinusoidal formulas inspired by SinDE and a local search procedure were added. Hence, before discussing L-SHADE-EpSin, we have to describe L-SHADE [47].

2.1.2. L-SHADE

The L-SHADE algorithm initializes the population according to Eq. (2) like the basic DE. However, the population size in L-SHADE is initially large and decreases with the number of function calls - this is marked with index g (NP_g).

Five memory sets are required in L-SHADE. In order to maintain diversity, an external archive **A** (a set which maximum size varies with generations $|\mathbf{A}| = NA_g$, but is initially empty $\mathbf{A} = \emptyset$; NA_g is linearly related to NP_g by $NA_g = \lceil rar \cdot NP_g \rceil$, where *rar* is another control parameter of the algorithm, hence NA_g also decreases with *g*) is used to preserve parent vectors which are worse than the trial vectors. The next two memories **MF, MCR** of predefined size $|\mathbf{MF}| = |\mathbf{MCR}| = H$ (*H* is another control parameter of the algorithm) store a set of *CR* and *F* values that performed well in the past. All elements in both sets are set initially to 0.5. The initially empty (and emptied at the beginning of each generation) last two sets, **SF** and **SCR**, store the individual-specific Cr_i and F_i values that succeeded in generating a trial vector which is better than the parent in a particular generation. The size of these two sets varies during the run.

At the beginning of each generation, for each individual $\boldsymbol{x}_{i,g}$ in population **P** of current size $|\mathbf{P}| = NP_g$, control parameters' values F_i and CR_i are generated as follows

$$ri = randi(1, H) \tag{6}$$

$$F_i = randc_i(MF_{ri}, 0.1) \tag{7}$$

$$CR_i = \begin{cases} 0 & if \quad MCR_{ri} = -1 \\ randn_i(MCR_{ri}, 0.1) & otherwise \end{cases} \tag{8}$$

where *randi* is a random integer generated within a specified interval [1, *H*], $randc_i$ and $randn_i$ are random numbers generated respectively from Cauchy and Normal distributions; all three are generated independently for each individual $\boldsymbol{x}_{i,g}$. When either F_i or CR_i is outside [0,1] interval, it is set to the closest bound (0 or 1), with the exception of the case $F_i \leq 0$, when F_i is regenerated. The reason why MCR_k may equal -1 will be clarified later. Then, in each generation g, the following DE/current-to-*p*best/1 mutation strategy taken from JADE [57] is used

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{pbest,g} - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \tag{9}$$

where $\mathbf{x}_{pbest,\,g}$ is, for each *i* and *g*, randomly selected from the *p*% of best individuals in the population **P**, and *r2* is randomly selected from the union of the current population **P** and external archive **A**. External archive is initially empty, and during the run is gradually filled with parent individuals $\boldsymbol{x}_{i,g}$ that loose the selection procedure (Eq. (5)) against the offspring, until the maximum archive size equal to NA_g is reached. Then each time the new individual $\boldsymbol{x}_{i,g}$ is to be moved into archive after losing selection, one randomly selected individual from the archive is discarded (as NA_g decreases with *g*, a respective number of randomly selected individuals is also discarded whenever NA_g decrease). After mutation the crossover is performed, but as CR_i values vary for each individual (and generation *g*), Eq. (4) needs to be re-written as

$$u_{i,g}^{j} = \begin{cases} v_{i,g}^{j} & if \quad rand_i^j(0,1) \leq CR_i \quad or \quad j = j_{rand,i} \\ x_{i,g}^{j} & otherwise \end{cases} \tag{10}$$

After crossover, some bounds handling procedure is needed; often the one described in [57] is used. Then selection in L-SHADE is performed as in classical DE following Eq. (5), but, if the parent individual loses, it is moved to the archive **A**. In L-SHADE the control parameters' values F_i and CR_i, which in generation *g* were used by those newly created individuals that have won the selection procedure (Eq. (5)) are stored in memory sets **SCR** and **SF**. At the beginning of each generation *g*, both memories are cleared ($\mathbf{SCR} = \emptyset$, $\mathbf{SF} = \emptyset$). Hence after each generation *g* they include only control parameters' values that turned out successful in this generation *g*. After generation *g*, if at least one improvement was noted (hence **SCR** and **SF** are not empty) the *k*th element of **MF** and **MCR** (*k* is initialized with 1 and after each generation is increased by 1 until it reaches *H*; when $k = H$ in particular generation *g*, in the generation $g+1$ index *k* is set back to 1) is replaced as follows

$$MF_k = \frac{\sum_{l=1}^{|SF|} w_l \cdot SF_l^2}{\sum_{l=1}^{|SF|} w_l \cdot SF_l} \tag{11}$$

$$MCR_k = \frac{\sum_{l=1}^{|SCR|} w_l \cdot SCR_l^2}{\sum_{l=1}^{|SCR|} w_l \cdot SCR_l} \tag{12}$$

$$w_l = \frac{\Delta f_l}{\sum_{a=1}^{|SCR|} \Delta f_a} \tag{13}$$

$$\Delta f_l = \left| f(u_{l,g}) - f(x_{l,g}) \right| \tag{14}$$

Note that always $|SF| \equiv |SCR|$. If improvements were obtained only with CR_i control parameters that were equal to 0, then MCR_k is set to -1 (instead of using Eq. (12)). If there were no improvements in generation *g*, MF_k and MCR_k remain unchanged.

L-SHADE pseudocode:

---

1. set generation counter $g = 0$ and problem-related information: D (problem dimensionality), $MNFC$ (maximum number of function calls);
2. set control parameters: $NP_{g=0} = 18{\cdot}D$, $NP_{min} = 4$, $H = 6$, $p\% = 11\%$, $rar = 2.6$ (tuned settings in [47]);
3. create memory sets: **MF**, **MCR** ($|\mathbf{MF}| = |\mathbf{MCR}| = H$, $MF_i = MCR_i = 0.5$, $i = 1{:}H$), **A**, **SF, SCR** ($|\mathbf{A}| = 0$);
4. set L^d and U^d as lower and upper bounds of d-th variable ($d = 1,\ldots,D$);
5. initialize population **P** of individuals $\mathbf{x}_{i,g}$ according to eq. (2);
6. find objective function values $f(\mathbf{x}_{i,g})$ for all individuals in **P**;
7. set the number of function calls used $NFC = NP_0$;
8. find the best solution found so far $\mathbf{x}_{best,0}$;
9. in the first generation the initial population is used, hence $NP_1 = NP_0$;
10. **while** $NFC < MNFC$
11. $g = g + 1$;
12. $|\mathbf{SF}| = |\mathbf{SCR}| = 0$, $|\mathbf{A}_{\mathbf{temp}}| = 0$;
13. **for** $i = 1{:}\ NP_g$
14. generate r_i according to eq. (6);
15. generate CR_i according to eq. (8);
16. generate F_i according to eq. (7);
17. choose randomly $\mathbf{x}_{pbest,g}$ for i among round($NP_g \cdot p\%$) best individuals in **P;**
18. perform mutation following eq. (9);
19. perform crossover following eq. (10);
20. perform bounds handling and selection (eq. (5));
21. **if** $f(\mathbf{u}_{i,g}) \le f(\mathbf{x}_{i,g})$
22. $\mathbf{x}_{i,g} \rightarrow \mathbf{A}_{\mathbf{temp}}$, $CR_i \rightarrow \mathbf{SCR}$, $F_i \rightarrow$ **SF;**
23. **end if;**
24. $NFC = NFC + 1$;
25. **end for;**
26. reduce population size to NP_{g+1} following eq. (15);
27. $\mathbf{A} \leftarrow \mathbf{A} \cup \mathbf{A}_{\mathbf{temp}}$;
28. reduce the archive size to $NA_{g+1} = \lceil rar \cdot NP_{g+1} \rceil$ by discarding random individuals from **A**;
29. update **MF**, **MCR** following eqs (11), (12);
30. **end while;**

Fig. 2. The pseudocode of L-SHADE algorithm.

At the end of each generation the linear population size reduction is performed by discarding the $NP_{g-1} - NP_g$ worst individuals from the population **P**, with

$$NP_{g+1} = round\left(\frac{NP_{\min} - NP_0}{MNFC} \cdot NFC + NP_0\right) \tag{15}$$

where *round* procedure returns the nearest integer number; NP_0 is the population size at the beginning of the search; NP_{min} is the lowest possible population size; *NFC* is the number of already used function calls; *MNFC* is the maximum number of allowed function calls. The pseudocode of L-SHADE variant is given in Fig. 2.

2.1.3. L-SHADE-EpSin

The L-SHADE-EpSin [1] differs from L-SHADE in two elements.

Firstly, during the first 50% of *MNFC* the values of control parameters F_i are updated using a function randomly selected out of the following two sinusoidal functions, namely non-adaptive sinusoidal decreasing adjustment and adaptive sinusoidal increasing adjustment described by Eqs. (16) and (17)

$$F_i = 0.5 \cdot \left(\sin(2\pi \cdot freq \cdot g + \pi) \cdot \frac{g_{\max} - g}{g_{\max}} + 1\right) \tag{16}$$

$$F_i = 0.5 \cdot \left(\sin(2\pi \cdot fq_i \cdot g) \cdot \frac{g}{g_{\max}} + 1\right) \tag{17}$$

During the second 50% of *MNFC*, the values of parameters F_i are updated as in L-SHADE. In Eqs. (16) and (17) g and g_{max} are the current generation number and the predefined maximum number of generations, *freq*, is a control parameter with

a fixed predefined value, and fq_i is yet another control parameter, whose values are adapted in a similar way as the values of CR_i. As long as $NFC \le 0.5 \cdot MNFC$ at the beginning of each generation g for each individual $\boldsymbol{x}_{i,g}$ in population **P,** the specific value of fq_i is generated as

$$fq_i = randc_i(MFQ_{ri}, 0.1) \tag{18}$$

where $randc_i$ is a random number generated from the Cauchy distribution and ri is the index generated by Eq. (6). The value of fq_i is regenerated if $fq_i \le 0$ or truncated to 1 if $fq_i > 1$. MFQ_{ri} is selected randomly from an additional external memory **MFQ** ($|\mathbf{MFQ}| = H$, each element is set initially to 0.5) that stores the mean values of successful fq_i from the former generations. Values of fq_i that are successful in a particular generation are remembered in the set **SFQ** whose size depends on the number of successful offspring in a particular generation. At the beginning of each generation **SFQ** is cleared ($\mathbf{SFQ} = \emptyset$), and during a given generation each value of fq_i associated with each offspring $\boldsymbol{u}_{i,g}$ that wins selection procedure according to Eq. (5) is remembered within memory set **SFQ**. Note that in L-SHADE-EpSin fq_i is remembered within **SFQ** irrespectively which equation, (16) or (17), was used to generate $\boldsymbol{u}_{i,g}$, even though fq_i had no impact on $\boldsymbol{u}_{i,g}$ if Eq. (16) was used. After generation g, if at least one improvement was noted, the kth element of **MFQ** (k is the same index as used in Eqs. (11) and (12)) is replaced by

$$MFQ_k = \frac{\sum_{l=1}^{|SFQ|} w_l \cdot SFQ_l{}^2}{\sum_{l=1}^{|SFQ|} w_l \cdot SFQ_l} \tag{19}$$

Note that w_l is defined as in Eqs. (13) and (14).

Secondly, when the population size NP_g of L-SHADE-EpSin is reduced to or below 20 for the first time in a particular run, the local search procedure is implemented. This procedure uses 10 new individuals $\boldsymbol{y}_{i,g}$ that are initialized randomly according to

$$y_{i,0}^j = L^j + rand_i^j(0,1) \cdot (U^j - L^j); \quad j = 1, ..., D; \quad i = 1, ..., 10 \tag{20}$$

and then perform a kind of Gaussian walk for 250 generations gg (hence 2500 function calls), according to the procedure defined in [1] as

$$\mathbf{y}_{i,gg} = N(\mathbf{y}_{best,gg-1}, \sigma) + rand(0,1) \times \mathbf{y}_{best,gg-1} - rand(0,1) \times \mathbf{y}_{i,gg-1} \tag{21}$$

with

$$\sigma = \left| \frac{\ln(g)}{g} \times (\mathbf{y}_{i,gg-1} - \mathbf{y}_{best,gg-1}) \right| \tag{22}$$

where $\mathbf{y}_{best,gg-1}$ is the best among 10 individuals generated in generation *gg-1,* and $N(\mathbf{y}_{best,gg-1}, \sigma)$indicates normal distribution with mean $\mathbf{y}_{best,gg-1}$ and standard deviation σ. Note that in Eq. (22) g is the generation counter from the main DE procedure, gg is the generation counter in the local search. However, in the L-SHADE-EpSin MATLAB code available from CEC2016 web page (http://www.ntu.edu.sg/home/EPNSugan) one finds an important difference compared with the procedure described in the paper [1] – instead of Eq. (21) the following procedure is implemented in the code:

$$\mathbf{y}_{i,gg} = N(\mathbf{y}_{best,gg-1}, \sigma) + N(0,1) \times \mathbf{y}_{best,gg-1} - N(0,1) \times \mathbf{y}_{i,gg-1} \tag{23}$$

Note that in both procedures defined by Eqs. (21) and (23) a scalar value is generated, either from $rand(0,1)$ or $N(0,1)$, to be multiplied by the vector $\mathbf{y}_{best,gg-1}$, and another scalar value is generated to be multiplied by a vector $\mathbf{y}_{i,gg-1}$. However, $N(\mathbf{y}_{best}, \sigma)$ generates a D-dimensional vector. Each time the local search procedure finds $\mathbf{y}_{i,gg}$ that is better than the worst individual $\mathbf{x}_{i,g}$ in **P,** $\mathbf{x}_{i,g}$ is substituted by $\mathbf{y}_{i,gg}$. However, note that the local search procedure itself is non-elitist, which means that the individuals from generation gg ($\mathbf{y}_{i,gg} \quad i = 1, ..., 10$) are discarded after generation $gg+1$.

The pseudocode of L-SHADE-EpSin is given in Fig. 3. The following control parameter values are used in L-SHADE-EpSin [1]: $NP_0 = 18 \cdot D$, $NP_{min} = 4$, $H = 5$, $freq = 0.5$, $rar = 1.4$ and $p\% = 11\%$. Some values are specified according to the official code obtained from the CEC2016 web page, as they were not given in the paper [1]. The settings of H and rar used in L-SHADE-EpSin [1] slightly differ from the values proposed for L-SHADE in [47]), what may be found by comparing pseudocodes in Figs 2 and 3.

2.2. UMOEA-II

United multi-operator Evolutionary Algorithms-II (UMOEA-II) [12], a modified version of UMOEA [11], is a multi-algorithm composed of CMA-ES [19] and DE [45]. Here we present only the main features of UMOEA-II that are needed to understand its general behaviour, with special attention paid to the part that causes structural bias. For the detailed description of UMOEA-II the reader is referred to [12].

UMOEA-II is a population-based algorithm. In generation $g = 0$, the population **P** of size NP is randomly initialized and divided into two subpopulations, $NP1$ and $NP2$, that are assigned to DE ($NP1$) or CMA-ES ($NP2$) sub-algorithms. In each generation g the search is performed either by only DE procedures (only $NP1$ is used), only by CMA-ES procedures (only $NP2$ is used), or by both of them. The search is divided into cycles; at the beginning of each cycle the probabilities of

L-SHADE-EpSin pseudocode:

---

1. set generation counter $g = 0$ and problem-related information: D (problem dimensionality), $MNFC$ (maximum number of function calls);
2. set control parameters: $NP_{g=0} = 18 \cdot D$, $NP_{min} = 4$, $H = 5$, $p\% = 11\%$, $rar = 1.4$, $freq = 0.5$;
3. create memory sets: **MF**, **MCR, MFQ** ($|\mathbf{MF}| = |\mathbf{MCR}| = |\mathbf{MFQ}| = H$, $MF_i = MCR_i = MFQ_i = 0.5$, $i = 1{:}H$), **A**, **SF, SCR, SFQ** ($|\mathbf{A}| = 0$);
4. set L^d and U^d as lower and upper bounds of d-th variable ($d = 1,\ldots,D$);
5. initialize population **P** of individuals $\mathbf{x}_{i,g}$ according to eq. (2);
6. find objective function values $f(\mathbf{x}_{i,g})$ for all individuals in **P**;
7. set the number of function calls used $NFC = NP_0$;
8. find the best solution found so far $\mathbf{x}_{best,0}$;
9. in the first generation the initial population is used, hence $NP_1 = NP_0$;
10. **while** $NFC < MNFC$
11. $g = g + 1$;
12. $|\mathbf{SF}| = |\mathbf{SCR}| = |\mathbf{SFQ}| = 0$, $|\mathbf{A}_{\mathbf{temp}}| = 0$;
13. **for** $i = 1{:}\ NP_g$
14. generate r_i according to eq. (6);
15. generate CR_i according to eq. (8);
16. **if** $NFC < 0.5 \cdot MNFC$
17. generate fq_i according to eq. (18);
18. **if** $rand(0,1) < 0.5$
19. generate F_i according to eq. (16);
20. **else**
21. generate F_i according to eq. (17);
22. **end if;**
23. **else**
24. generate F_i according to eq. (7);
25. **end if;**
26. choose randomly $\mathbf{x}_{pbest,g}$ for i among round($NP_g \cdot p\%$) best individuals in **P;**
27. perform mutation following eq. (9);
28. perform crossover following eq. (10);
29. perform bounds handling and selection (eq. (5));
30. **if** $f(\mathbf{u}_{i,g}) \le f(\mathbf{x}_{i,g})$
31. $\mathbf{x}_{i,g} \rightarrow \mathbf{A}_{\mathbf{temp}}$, $CR_i \rightarrow \mathbf{SCR}$, $F_i \rightarrow \mathbf{SF}$, $fq_i \rightarrow \mathbf{SFQ}$;
32. **end if;**
33. $NFC = NFC + 1$;
34. **end for;**
35. reduce population size to NP_{g+1} following eq. (15);
36. $\mathbf{A} \leftarrow \mathbf{A} \cup \mathbf{A}_{\mathbf{temp}}$;
37. reduce the archive size to $NA_{g+1} = \lceil rar \cdot NP_{g+1} \rceil$ by discarding random individuals from **A**;
38. update **MF**, **MCR, MFQ** following eqs (11), (12), (19);
39. **if** $NP_{g+1} \le 20$ **and** $NP_g > 20$
40. generate new 10 individuals following eq. (20);
41. perform local search for 2500 function calls (eq. (22) and either (21) or (23)) modifying population **P** each time the improvement is found;
42. $NFC = NFC + 2500$;
43. **end if;**
44. **end while;**

Fig. 3. The pseudocode of L-SHADE-EpSin algorithm.

Table 1
51-Run mean performance of various algorithms on 50-dimensional CEC2014 problems. Results are collected from the literature.

problem	L-SHADE-EpSin [1]	UMOEA-II [12]	L-SHADE [1]	ETI-SHADE [9]	GA-MPC [35]	DNS-PSO [35]
F1	1.667E-05	1.834E-03	5.484E+02	5.168E+04	4.058E+05	8.482E+05
F2	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.472E+03	7.124E+03
F3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	9.828E+00	1.583E+03
F4	5.591E+01	3.849E+01	4.400E+01	1.834E+01	6.429E+01	8.583E+01
F5	2.026E+01	2.000E+01	2.026E+01	2.030E+01	2.028E+01	2.113E+01
F6	1.791E-04	2.030E-01	2.821E-01	8.501E-01	5.195E+00	3.294E+01
F7	0.000E+00	0.000E+00	0.000E+00	1.450E-04	2.465E-03	1.040E-02
F8	0.000E+00	0.000E+00	0.000E+00	0.000E+00	5.198E+01	1.967E+02
F9	3.021E+01	4.100E+00	1.173E+01	3.810E+01	5.796E+01	2.301E+02
F10	4.986E-02	5.431E-01	5.203E-02	5.736E-01	1.201E+03	4.096E+03
F11	3.026E+03	3.470E+03	3.301E+03	3.848E+03	3.479E+03	6.219E+03
F12	2.103E-01	1.303E-01	2.136E-01	2.469E-01	8.070E-02	2.447E+00
F13	2.061E-01	1.466E-01	1.664E-01	1.437E-01	4.358E-01	5.606E-01
F14	1.909E-01	3.041E-01	2.699E-01	2.926E-01	3.392E-01	3.969E-01
F15	5.570E+00	5.288E+00	5.101E-00	5.651E+00	5.540E+00	2.111E+01
F16	1.660E+01	1.826E+01	1.682E+01	1.696E+01	1.672E+01	2.123E+01
F17	3.341E+02	1.089E+03	1.464E+03	1.931E+03	2.623E+04	1.539E+05
F18	1.928E+01	7.276E+01	1.009E+02	8.095E+01	5.197E+02	1.123E+03
F19	9.685E+00	8.787E+00	8.492E+00	8.466E+00	5.792E+00	1.650E+01
F20	6.066E+00	1.363E+01	5.141E+01	5.897E+01	2.363E+02	4.318E+02
F21	3.161E+02	4.521E+02	6.983E+02	7.733E+02	1.541E+04	1.143E+05
F22	1.018E+02	1.982E+02	1.015E+02	2.723E+02	9.802E+02	1.058E+03
F23	**2.000E+02**	**2.000E+02**	3.440E+02	3.440E+02	3.370E+02	*2.000E+02*
F24	**2.000E+02**	**2.000E+02**	2.585E+02	2.727E+02	2.689E+02	*2.000E+02*
F25	**2.000E+02**	**2.000E+02**	2.053E+02	2.080E+02	2.005E+02	*2.000E+02*
F26	1.002E+02	1.001E+02	1.002E+02	1.022E+02	1.004E+02	1.967E+02
F27	**2.020E+02**	**2.000E+02**	3.088E+02	3.567E+02	1.207E+03	2.308E+02
F28	**2.000E+02**	**2.000E+02**	9.736E+02	1.107E+03	3.693E+02	2.155E+02
F29	**2.000E+02**	8.085E+02	7.415E+02	8.423E+02	2.157E+02	1.851E+06
F30	**2.000E+02**	8.619E+03	1.183E+04	8.920E+03	1.127E+03	1.259E+04

using DE (*prob1*) and CMA-ES (*prob2*) are set to 1 and information sharing between *NP1* and *NP2* is performed by substituting some of the worst individuals within the poorer sub-population by the best individuals from the better one (see [12] for technical details). Later, during each cycle *prob1* and *prob2* are adaptively modified during the run depending on two criteria: the quality and the diversity of solutions found by the particular procedure (again, for details see [12]). Two sub-populations do not communicate with each other. At the beginning of each generation within a cycle two random numbers are generated (*rand1*(0,1) and *rand2*(0,1)). If *rand1*(0,1) $<$ *prob1* the DE procedures are applied to sub-population *NP1* and if *rand2*(0,1) $<$ *prob2* the CMA-ES procedures are applied to sub-population *NP2*. When *rand1*(0,1) $<$ *prob1* and *rand2*(0,1) $<$ *prob2*, DE and CMA-ES procedures are preformed together. In addition, to improve the exploitation at the later stages of the run, when $NFC > 0.75 \cdot MNFC$, the local search procedure based on the interior point [31] may be used for a pre-specified number of function evaluations with some low probability related to the efficiency of the local search.

The length of the cycle (*CL*) in UMOEA-II may have large impact on the results, but is not precisely specified in [12]. In the UMOEA-II code available from CEC2016 web page (http://www.ntu.edu.sg/home/EPNSugan) one finds that *CL* may vary from 50 for 10-dimensional problems to 150 for 50-dimensional ones. Hence we related *CL* to *D* approximately by $CL = 25 + \lceil 2.5 \cdot D \rceil$, where *D* is the problem dimensionality.

Among two algorithms used within UMOEA-II, the CMA-ES algorithm follows standard operators and control parameters given in [19,20], for details see [12]. The DE variant used by UMOEA-II is called MODE [12]. It follows the basic DE mutation-crossover-selection scheme Eqs. (3)–(5) with the L-SHADE-based concept of linear population size reduction and control parameter adaptation Eqs. (6)–(15). There is, however, one fundamental difference between L-SHADE and MODE that is used in UMOEA-II. MODE [12] allows each individual to be evolved by one out of the three following mutation strategies: 1. DE/current-to-pbest defined by Eq. (9), 2. DE/current-to-pbest without archive (where $\mathbf{x}_{r2,\,g}$ in Eq. (9) is chosen from $\mathbf{P}$, not the union of $\mathbf{P} \cup \mathbf{A}$), and 3. a strategy defined as

$$\mathbf{v}_{i,g} = F_i \cdot \mathbf{x}_{r1,g} + (\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}) \tag{24}$$

where $\mathbf{x}_{r2,\,g}$ is for particular i and g randomly selected from the 50% of best individuals in the population $\mathbf{P}$, whereas $\mathbf{x}_{r1,\,g}$ and $\mathbf{x}_{r3,\,g}$ are randomly selected from the whole $\mathbf{P}$. The probabilities of choosing each strategy are initially equal to 1/3, but are modified during the run according to their successfulness (for details see [12]). However, the probability of using each mutation strategy cannot fall below 0.1.

Table 2
51-run based standard deviation of the performance of various algorithms on 50-dimensional CEC2014 problems. Results are collected from the literature.

problem	L-SHADE-EpSin [1]	UMOEA-II [12]	L-SHADE [1]	ETI-SHADE [9]	GA-MPC [35]	DNS-PSO [35]
F1	8.419E-05	1.149E-03	9.336E+02	2.653E+04	1.318E+05	2.770E+05
F2	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.440E+03	8.689E+03
F3	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.302E+01	6.183E+02
F4	4.755E+01	4.840E+01	4.763E+01	3.729E+01	3.013E+01	4.697E+01
F5	3.063E-02	1.981E-04	3.028E-02	9.085E-02	2.131E-01	3.734E-02
F6	1.671E-04	4.503E-01	5.239E-01	8.322E-01	2.011E+00	5.914E+00
F7	0.000E+00	0.000E+00	0.000E+00	1.036E-03	4.341E-03	1.347E-02
F8	0.000E+00	0.000E+00	0.000E+00	0.000E+00	8.935E+00	3.375E+01
F9	5.375E+00	1.946E+00	2.502E+00	9.809E+00	1.582E+01	4.011E+01
F10	2.726E-02	6.347E-01	2.410E-02	3.363E-01	4.847E+02	7.333E+02
F11	3.136E+02	4.825E+02	3.327E+02	5.436E+02	8.014E+02	6.939E+02
F12	3.201E-02	7.147E-02	2.555E-02	7.072E-02	3.333E-02	1.011E+00
F13	2.702E-02	3.845E-02	1.751E-02	3.144E-02	6.651E-02	9.241E-02
F14	2.279E-02	2.284E-02	1.894E-02	4.003E-02	9.011E-02	2.368E-01
F15	4.998E-01	8.169E-01	4.154E-01	1.221E+00	1.210E+00	6.574E+00
F16	5.217E-01	7.392E-01	4.614E-01	7.356E-01	1.185E+00	4.030E-01
F17	1.584E+02	3.447E+02	3.647E+02	5.522E+02	1.071E+04	9.584E+04
F18	5.933E+00	1.740E+01	1.705E+01	4.425E+01	8.871E+02	1.123E+03
F19	1.124E+00	1.979E+00	2.087E+00	4.591E+00	1.442E+00	3.127E+00
F20	2.015E+00	3.955E+00	2.083E+01	2.949E+01	1.968E+02	1.775E+02
F21	9.991E+01	1.328E+02	2.387E+02	2.411E+02	1.450E+04	5.246E+04
F22	5.916E+01	9.632E+01	5.602E+01	1.540E+02	2.333E+02	3.955E+02
F23	**0.000E+00**	**0.000E+00**	3.566E-13	3.028E-13	4.177E-13	*2.272E-08*
F24	**8.155E-09**	**2.548E-13**	1.988E+00	1.982E+00	6.764E+00	*6.492E-05*
F25	**0.000E+00**	**0.000E+00**	3.338E-01	3.369E+00	1.608E-01	*0.000E+00*
F26	3.491E-02	3.561E-02	1.930E-02	1.397E+01	4.635E-02	1.815E+01
F27	**1.400E+01**	**0.000E+00**	1.320E+01	4.294E+01	1.350E+02	7.613E+01
F28	**0.000E+00**	**0.000E+00**	8.356E+01	4.817E+01	4.161E+00	6.760E+01
F29	**0.000E+00**	4.643E+01	4.472E+01	5.909E+01	2.745E+00	1.013E+07
F30	**0.000E+00**	4.648E+02	4.192E+02	5.624E+02	3.297E+02	2.600E+03

3. Structural bias in L-SHADE-EpSin and UMOEA-II

In this section we find and remove the sources of structural bias in both winners of the IEEE CEC2016 competition, L-SHADE-EpSin and UMOEA-II algorithms. First, in 3.1, we refer to the results obtained by these algorithms on the 50-dimensional CEC2014 benchmark problems that were reported in their respective source papers. We point out seven problems for which, as reported in the literature, such results seriously differ from the results achieved by various other algorithms. We find that these seven problems share similar features, namely that the solutions located in the close neighbourhood of the origin **O**, although the optimum is not located there, are of very good quality, better than solutions found by almost all optimizers. We also note that the objective function value in the origin **O** is approximately equal to the objective function values found by both L-SHADE-EpSin and UMOEA-II. This suggests that both algorithms may artificially facilitate sampling points in the vicinity of the origin **O**. In Sections 3.2 and 3.3 the operators that introduce structural bias to L-SHADE-EpSin and UMOEA-II are identified and removed. In the same two sections the results obtained for the 50-dimensional CEC2014 problems by both algorithms with removed structurally biased parts (called L-SHADE-EpSin-NLS and UMOEA-II-M) are reported and compared with those obtained by original, biased versions.

3.1. Results obtained by L-SHADE-EpSin and UMOEA-II algorithms according to the literature

In the source papers of the L-SHADE-EpSin [1] and UMOEA-II [12] algorithms, 51-run averaged results and the respective standard deviations obtained for 30 benchmark problems from CEC2014 set are reported (note that test set CEC2014 was used during the CEC2016 competition). To simplify the discussion, let us concentrate on the results obtained for 50-dimensional cases that are given in Tables 1 and 2 of this paper, but the same conclusions could be drawn for other dimensionalities considered in [1,12]. To allow comparison of the results obtained by L-SHADE-EpSin and UMOEA-II, Tables 1 and 2 also include recently published results (from [1,9,35]) reached by selected other algorithms on 50-dimensional CEC2014 problems:

- L-SHADE [47], that was the best among 6 widely praised algorithms used as competitors for L-SHADE-EpSin in [1],
- ETI-SHADE proposed and tested in [9],
- GA-MPC [10] – the winner of CEC2011 competition [6] on real-world problems that was tested on the CEC2014 set in our recent study [35] (note however that in that study only 30 runs were performed on each problem),
- DNS-PSO [51] in which the essential presence of the structural bias was detected in [35].

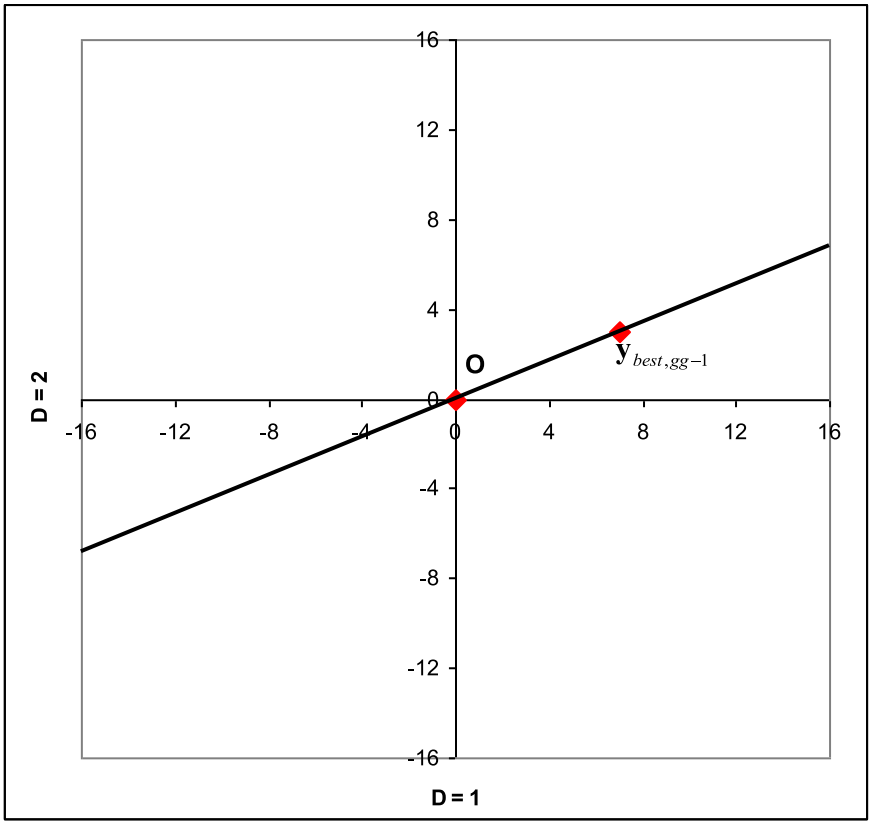

Fig. 4. The locations (denoted as solid line) in 2-dimensional search space that may be reached by the best individual $\mathbf{y}_{best,gg-1}$ of the local search procedure according to Eq. (26).

Comparing the results obtained by L-SHADE-EpSin and UMOEA-II with those from the literature we quickly note that L-SHADE-EpSin seems to perform outstandingly well for problems F23-F25 and F27-F30, and UMOEA-II for problems F23-F25 and F27-F28. For the mentioned problems both algorithms almost always reach the objective function values of $f(\mathbf{x}) \approx 200$. Such good results on these problems are unachievable by other state-of-the-art methods as one may find from [1] and Table 1 of this paper. However, there is an exception – DNS-PSO performed similarly to L-SHADE-EpSin and UMOEA-II on problems F23-F25. As shown in [35], DNS-PSO is contaminated by strong structural bias introduced by a specific components devoted to performing local and global searches, which drags the algorithm towards the origin **O**. In the case of 50-dimensional CEC2014 problems F23-F30, the objective value of $f(\mathbf{O})$ equals 200, which is almost exactly the same value as obtained by DNS-PSO on problems F23-F25, by L-SHADE-EpSin on F23-F25 and F27-F30, and by UMOEA-II on F23-F25 and F27-F28. Note that in the case of all CEC2014 problems $f(\mathbf{x}^*)=0$, where $\mathbf{x}^*$ is the global optimum. Hence solutions of F23-F30 in $f(\mathbf{O})$ are not the global optima, but (with exception of problem F26) are better than the solutions achieved by almost all algorithms. This raises the question whether the apparently good performances of L-SHADE-EpSin and UMOEA-II on such problems are not the effect of some structural bias that attract these algorithms towards the origin, as was the case of DNS-PSO.

3.2. Structural bias in L-SHADE-EpSin algorithm

In this section we discuss the evidence that the structural bias of L-SHADE-EpSin is introduced by the local search procedure (see Eqs. (20)–(23) and lines 39–43 in pseudocode in Fig. 3) that favours sampling points close to the origin **O**. Note that we have to consider two variants of local search procedure:

- V1, the option described in the paper [1] (Eq. (21)), and
- V2, the option implemented in the MATLAB code available from CEC2016 conference web page http://www.ntu.edu.sg/home/EPNSugan (Eq. (23)).

There are two important features of the local search strategy used by L-SHADE-EpSin. **Firstly**, although each time the local search procedure finds $\mathbf{y}_{i,gg}$ that is better than the worst individual $\mathbf{x}_{i,g}$ in **P**, $\mathbf{x}_{i,g}$ is substituted by $\mathbf{y}_{i,gg}$, but the local search procedure itself does not benefit from individuals stored in **P**. The local search procedure is also non-elitist, which

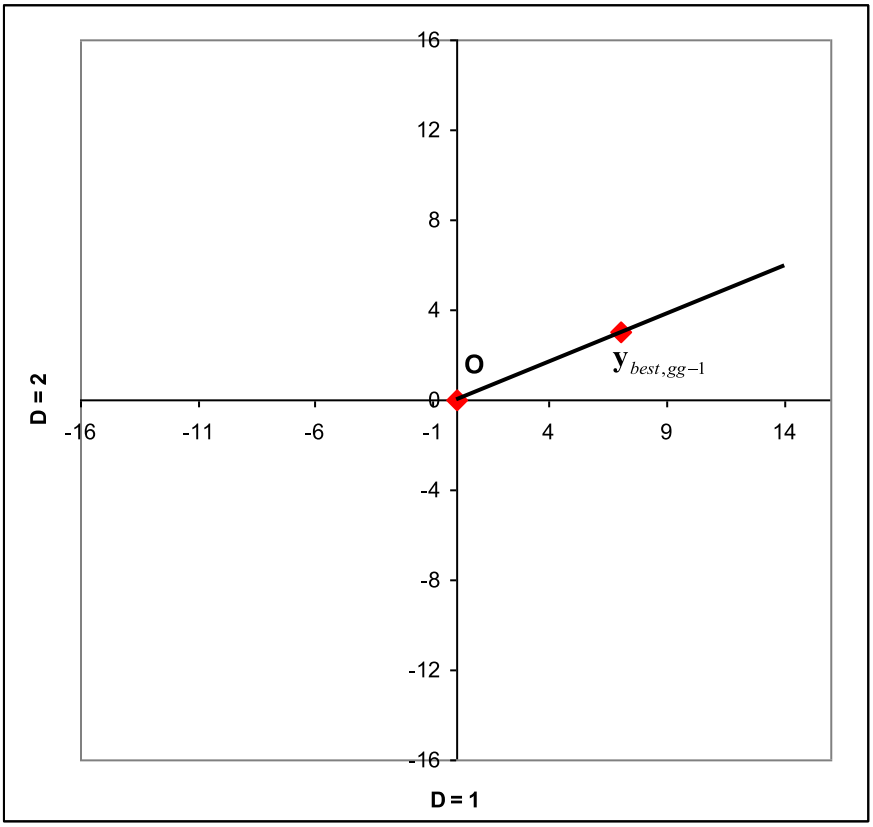

Fig. 5. The locations (denoted as solid line segment) in 2-dimensional search space that may be reached by the best individual $\mathbf{y}_{best,gg-1}$ of the local search procedure according to Eq. (25).

means that the individuals from generation gg ($\mathbf{y}_{i,gg}$ $i = 1, ..., 10$) are discarded after generation $gg+1$. As a result, $\mathbf{y}_{best,gg-1}$ is always the best among 10 solutions sampled in the generation gg-1, but may be poorer than those sampled in preceding generations (gg-2, gg-3 and so on). **Secondly**, the local search is performed in each generation by all 10 individuals (including the best one) using the same equation (either (21) or (23)), which has an important impact on the search.

We have to discuss separately two cases:

- C1, that occurs for the best individual, $\mathbf{y}_{i,gg-1} = \mathbf{y}_{best,gg-1}$, and
- C2, that occurs for the remaining nine individuals, $\mathbf{y}_{i,gg-1} \neq \mathbf{y}_{best,gg-1}$.

After these comments, we may discuss the source of structural bias in L-SHADE-EpSin.

First, we pay attention to the position of the points sampled from normal distribution $N(\mathbf{y}_{best,gg-1}, \sigma)$. For the case C1, when $\mathbf{y}_{i,gg-1} = \mathbf{y}_{best,gg-1}$, we find that $\sigma = 0$ according to Eq. (22). Hence $N(\mathbf{y}_{best,gg-1}, \sigma) = \mathbf{y}_{best,gg-1}$. This is valid for both V1 and V2, so we may rewrite Eqs. (21) and (23) for the variant V1 as

$$\mathbf{y}_{i,gg} = \mathbf{y}_{best,gg-1} + rand_1(0,1) \times \mathbf{y}_{best,gg-1} - rand_2(0,1) \times \mathbf{y}_{best,gg-1} \tag{25}$$

and for the variant V2 as

$$\mathbf{y}_{i,gg} = \mathbf{y}_{best,gg-1} + N_1(0,1) \times \mathbf{y}_{best,gg-1} - N_2(0,1) \times \mathbf{y}_{best,gg-1} \tag{26}$$

Note that in Eq. (25) a scalar value is generated from $rand_1(0,1)$ and another scalar value from $rand_2(0,1)$; both scalars are independently multiplied by the same vector $\mathbf{y}_{best,gg-1}$. Similarly, in Eq. (26), one scalar value is generated from $N_1(0, 1)$, and another scalar value from $N_2(0, 1)$; both are also multiplied by the same vector $\mathbf{y}_{best,gg-1}$. Hence, according to Eq. (26), the best individual may move only along the line that goes through two points: the position of the current best individual $\boldsymbol{y}_{best,gg-1}$ and the origin **O,** see Fig. 4. In case of Eq. (25), available points are even more restricted, to a line segment [**O**, $2 \times \mathbf{y}_{best,gg-1}$], as indicated in Fig. 5. The position of the current best individual $\mathbf{y}_{best,gg-1}$ is not defined a priori and would change during the run when another individual becomes the best, but the position of the second point (**O**) is fixed for good, hence privileged. This makes points located in the neighbourhood of the origin favoured in sampling when C1 takes place.

Now, let us consider C2 (when $\mathbf{y}_{i,gg-1} \neq \mathbf{y}_{best,gg-1}$), the search pattern performed by the remaining nine individuals according to non-simplified Eqs. (21) or (23). The moves of individuals in C2 are not restricted to any direction. However,

Table 3
51-Run mean performance of various variants of L-SHADE-EpSin and its local search procedures applied separately to 50-dimensional CEC2014 problems. V1 – refers to the variant of local search procedure described in the paper [1], V2 – refers to the variant of the local search procedure used in the official L-SHADE-EpSin code available on CEC2016 web page, L-SHADE-EpSin-NLS – refers to the L-SHADE-EpSin variant without local search procedure, f(**O**) – denotes the value of the objective function in the origin of the coordinate system. All algorithms are tested with the maximum number of function calls set to 500,000, but V1 and V2 local search procedures used alone are also tested with 10,000 and 2500 function calls (as denoted in respective brackets in column titles).

problem	f(**O**)	L-SHADE-EpSin-V1	L-SHADE-EpSin-V2	L-SHADE-EpSin-NLS	V1	V1 (10,000)	V1 (2500)	V2	V2 (10,000)	V2 (2500)
F1	1.665E+10	4.203E-04	5.089E-03	8.120E-06	3.275E+09	4.080E+09	4.318E+09	1.370E+09	2.309E+09	2.798E+09
F2	1.996E+11	3.288E-14	3.399E-14	3.288E-14	1.244E+11	1.596E+11	1.625E+11	9.409E+10	1.200E+11	1.333E+11
F3	6.963E+08	5.796E-14	5.684E-14	5.684E-14	1.546E+05	1.664E+05	1.731E+05	1.286E+05	1.638E+05	1.980E+05
F4	7.259E+04	6.029E+01	5.853E+01	6.237E+01	3.145E+04	4.692E+04	4.784E+04	1.869E+04	2.784E+04	3.445E+04
F5	2.169E+01	2.025E+01	2.025E+01	2.023E+01	2.124E+01	2.131E+01	2.131E+01	2.114E+01	2.125E+01	2.131E+01
F6	9.074E+01	2.046E-04	1.822E-04	1.040E-02	6.979E+01	7.454E+01	7.425E+01	6.484E+01	7.011E+01	7.205E+01
F7	1.879E+03	1.070E-13	9.808E-14	9.808E-14	1.156E+03	1.492E+03	1.538E+03	8.831E+02	1.123E+03	1.245E+03
F8	9.088E+02	1.048E-09	1.474E-09	4.429E-10	6.022E+02	6.560E+02	6.708E+02	5.410E+02	6.050E+02	6.271E+02
F9	1.011E+03	2.839E+01	2.995E+01	2.938E+01	6.701E+02	7.437E+02	7.568E+02	6.224E+02	6.902E+02	7.268E+02
F10	1.843E+04	4.900E-02	5.402E-02	2.945E-02	1.316E+04	1.465E+04	1.494E+04	1.183E+04	1.310E+04	1.397E+04
F11	1.833E+04	3.177E+03	3.090E+03	3.105E+03	1.433E+04	1.497E+04	1.521E+04	1.334E+04	1.454E+04	1.499E+04
F12	1.395E+01	2.096E-01	2.132E-01	2.023E-01	3.939E+00	4.842E+00	5.267E+00	3.184E+00	4.482E+00	4.967E+00
F13	9.717E+00	2.068E-01	2.092E-01	2.031E-01	7.379E+00	8.416E+00	8.402E+00	6.304E+00	7.254E+00	7.677E+00
F14	4.796E+02	1.877E-01	1.867E-01	1.965E-01	2.892E+02	3.850E+02	3.808E+02	2.232E+02	2.838E+02	3.098E+02
F15	2.739E+07	5.568E+00	5.422E+00	5.367E+00	1.804E+06	6.331E+06	7.331E+06	7.433E+05	1.722E+06	2.737E+06
F16	2.501E+01	1.659E+01	1.650E+01	1.658E+01	2.311E+01	2.336E+01	2.336E+01	2.252E+01	2.311E+01	2.327E+01
F17	3.878E+09	3.998E+02	3.531E+02	3.633E+02	4.224E+08	5.655E+08	5.464E+08	8.789E+07	2.148E+08	3.524E+08
F18	3.821E+10	1.771E+01	1.698E+01	1.745E+01	1.022E+10	1.802E+10	1.738E+10	3.582E+09	7.361E+09	1.016E+10
F19	8.929E+03	9.921E+00	9.778E+00	9.555E+00	1.378E+03	2.287E+03	2.650E+03	5.431E+02	9.120E+02	1.180E+03
F20	3.218E+09	6.349E+00	6.614E+00	6.181E+00	1.616E+05	2.897E+05	5.846E+05	6.734E+04	2.237E+05	4.713E+05
F21	1.867E+09	3.302E+02	3.111E+02	3.108E+02	6.455E+07	1.112E+08	1.188E+08	1.291E+07	3.190E+07	5.412E+07
F22	6.109E+06	8.779E+01	1.236E+02	1.064E+02	2.896E+04	2.078E+05	2.081E+05	3.198E+03	5.098E+03	1.173E+04
F23	**2.000E+02**	**2.000E+02**	**2.000E+02**	**3.440E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**
F24	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.679E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**
F25	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.049E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**
F26	2.000E+02	1.002E+02	1.002E+02	1.002E+02	2.000E+02	2.000E+02	1.994E+02	2.000E+02	2.000E+02	2.000E+02
F27	**2.000E+02**	**2.981E+02**	**2.020E+02**	**3.116E+02**	**2.000E+02**	**1.168E+03**	**2.195E+03**	**2.000E+02**	**2.000E+02**	**2.480E+02**
F28	**2.000E+02**	**2.000E+02**	**2.000E+02**	**1.141E+03**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**
F29	**2.000E+02**	**2.000E+02**	**2.000E+02**	**8.038E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**
F30	**2.000E+02**	**2.000E+02**	**2.000E+02**	**8.429E+03**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**	**2.000E+02**

each move starts from the point generated from $N(\mathbf{y}_{best,gg-1}, \sigma)$, to which locations of the best individual $\mathbf{y}_{best,gg-1}$ and the current position $\mathbf{y}_{i,gg-1}$ of ith individual, each multiplied by a different random number, are added. The distance between the starting position $N(\mathbf{y}_{best,gg-1}, \sigma)$ and the location of the best individual $\mathbf{y}_{best,gg-1}$ depends only on σ. The values of σ, when related to the search space, are always low for CEC2014 problems, due to the ln(g)/g factor in Eq. (22) and the fact that in L-SHADE-EpSin the local search procedure is used in the very late part of the search. For example, in 50-dimensional cases of CEC2014 problems, when $MNFC = 500{,}000$, $NP_0 = 18{\cdot}D = 900$ and $NP_{min} = 4$, the local search procedure of L-SHADE-EpSin is initialized after generation $g \approx 2100$ (see Eq. (22)), hence $\ln(g)/g \approx 0.0036$. Because $U^j - L^j = 200 \quad j = 1, ..., D$, one obtains $y^j{}_{i,gg-1} - y^j{}_{best,gg-1} \leq 200$ and $\sigma^j < 0.73$. As a result, the base point $N(\mathbf{y}_{best,gg-1}, \sigma)$ is at least in 99.8% of cases within $[\mathbf{y}_{best,gg-1} - 2.2, \mathbf{y}_{best,gg-1} + 2.2]$. For randomly generated individuals the expected values of σ^j are smaller, hence $N(\mathbf{y}_{best,gg-1}, \sigma)$ are expected even closer to $\mathbf{y}_{best,gg-1}$. Hence, as CEC2014 problems are bounded within [−100,100] for each dimension, the generated base point $N(\mathbf{y}_{best,gg-1}, \sigma)$ for the move of each ith individual from C2 is almost always relatively close to $\mathbf{y}_{best,gg-1}$. Moreover, the more clustered individuals are, the closer to $\mathbf{y}_{best,gg-1}$ the base point $N(\mathbf{y}_{best,gg-1}, \sigma)$ will be located.

Above we have discussed that in the local search procedure the base points $N(\mathbf{y}_{best,gg-1}, \sigma)$ from which the moves of individuals within local search procedure starts are located either in $\mathbf{y}_{best,gg-1}$ (in case C1), or very close to $\mathbf{y}_{best,gg-1}$ (in case C2). Let us now consider each out of two possible moves defined by Eqs. (21) and (25) for V1, or Eqs. (23) and (26) for V2.

First, consider V1 and C1 (Eq. (25)). As discussed above (see also Fig. 5), the best point may move only along a line segment between **O** and $2 \times \mathbf{y}_{best,gg-1}$, as $-1 \leq rand_1(0, 1) - rand_2(0, 1) \leq 1$. Depending on whether $rand_1(0, 1) < rand_2(0, 1)$ or $rand_1(0, 1) > rand_2(0, 1)$, the best individual may move, with equal probability, either towards **O** or in the opposite direction. The step size depends on the distance between $\mathbf{y}_{best,gg-1}$ and **O** (the length of the vector $\mathbf{y}_{best,gg-1}$); hence this size is re-scaled each time $\mathbf{y}_{best}$ changes its position. The smaller $|\mathbf{y}_{best,gg} - \mathbf{O}|$ is, the shorter steps. This has an important impact on the search, as it favours sampling points close to **O** due to the following reason. Irrespective where $\mathbf{y}_{best,gg-1}$ is located, if the values $rand_1(0, 1) \approx 0$ and $rand_2(0, 1) \approx 1$ are generated, the origin **O** is reached immediately according to Eq. (25). As only points along a line segment between **O** and $2 \times \mathbf{y}_{best,gg-1}$ are available, once $\mathbf{y}_{best,gg-1}$ is sampled relatively close to **O**, points located more than twice further from **O** are unavailable in a single step according to Eq. (25) (until a different individual becomes the best one). We may illustrate this on the following examples. The new point $\mathbf{y}_{i,gg}$ generated by Eq. (25) may be located over twice closer to **O** then the current best individual in one generation with probability $\Pr(|\mathbf{y}_{i,gg} - \mathrm{O}| < 0.5 \cdot |\mathbf{y}_{best,gg-1} - \mathrm{O}|) \approx 12.5\%$, but, if $\mathbf{y}_{i,gg}$ becomes $\mathbf{y}_{best,gg}$, getting back from point $\mathbf{y}_{best,gg}$ to $\mathbf{y}_{best,gg-1}$ will not be possible in a single generation, as steps larger than

Table 4
Standard deviation of performance of various variants of L-SHADE-EpSin and its local search procedures applied separately to 50-dimensional CEC2014 problems. V1 – refers to the variant of local search procedure described in the paper [1], V2 – refers to the variant of the local search procedure used in the official L-SHADE-EpSin code available on CEC2014 web page, L-SHADE-EpSin-NLS – refers to the L-SHADE-EpSin variant without local search procedure. All algorithms are tested with the maximum number of function calls set to 500,000, but V1 and V2 local search procedures used alone are also tested with 10,000 and 2500 function calls (as denoted in respective brackets in column titles).

problem	L-SHADE-EpSin-V1	L-SHADE-EpSin-V2	L-SHADE-EpSin-NLS	V1	V1 (10,000)	V1 (2500)	V2	V2 (10,000)	V2 (2500)
F1	3,000E-03	3,607E-02	4,519E-05	1,141E+09	1,995E+09	1,769E+09	1,624E+08	3,925E+08	4,695E+08
F2	1,044E-14	1,140E-14	1,044E-14	9,578E+09	1,508E+10	1,683E+10	6,854E+09	1,274E+10	1,227E+10
F3	7,960E-15	1,137E-14	0,000E+00	1,253E+04	2,005E+04	2,672E+04	7,969E+03	1,859E+04	3,229E+04
F4	4,758E+01	4,779E+01	4,686E+01	5,659E+03	9,308E+03	1,025E+04	2,239E+03	4,210E+03	5,261E+03
F5	3,224E-02	3,688E-02	4,803E-02	8,255E-02	6,148E-02	5,932E-02	3,603E-02	4,741E-02	3,732E-02
F6	1,822E-04	2,004E-04	7,266E-02	2,973E+00	3,726E+00	3,655E+00	1,458E+00	1,952E+00	2,082E+00
F7	2,702E-14	3,951E-14	3,951E-14	1,387E+02	1,591E+02	1,600E+02	5,902E+01	8,876E+01	1,360E+02
F8	3,580E-09	4,624E-09	2,959E-09	4,483E+01	3,762E+01	3,905E+01	1,355E+01	1,959E+01	2,638E+01
F9	6,104E+00	5,953E+00	5,620E+00	4,655E+01	4,703E+01	4,924E+01	2,049E+01	3,132E+01	3,184E+01
F10	2,402E-02	2,663E-02	1,715E-02	8,382E+02	9,677E+02	8,754E+02	3,615E+02	5,336E+02	5,160E+02
F11	3,202E+02	3,331E+02	3,175E+02	7,425E+02	7,351E+02	7,966E+02	3,705E+02	4,745E+02	5,127E+02
F12	2,535E-02	2,486E-02	2,679E-02	7,619E-01	8,485E-01	8,121E-01	3,060E-01	4,027E-01	5,111E-01
F13	1,855E-02	2,089E-02	1,712E-02	4,397E-01	5,318E-01	6,218E-01	2,694E-01	2,926E-01	3,837E-01
F14	2,232E-02	2,142E-02	2,251E-02	4,373E+01	4,139E+01	4,324E+01	1,323E+01	2,440E+01	2,798E+01
F15	5,240E-01	5,340E-01	4,666E-01	1,123E+06	3,031E+06	3,864E+06	1,466E+05	5,041E+05	1,150E+06
F16	3,416E-01	4,332E-01	4,519E-01	1,860E-01	2,660E-01	2,914E-01	1,680E-01	1,521E-01	2,190E-01
F17	1,613E+02	1,641E+02	1,524E+02	1,798E+08	2,788E+08	2,606E+08	2,253E+07	5,954E+07	9,742E+07
F18	6,538E+00	5,303E+00	5,939E+00	4,078E+09	7,271E+09	7,116E+09	7,224E+08	1,755E+09	3,092E+09
F19	8,883E-01	1,192E+00	1,222E+00	5,623E+02	1,279E+03	1,323E+03	7,438E+01	1,899E+02	2,680E+02
F20	2,056E+00	1,997E+00	2,404E+00	6,142E+04	2,622E+05	1,416E+06	1,903E+04	1,211E+05	3,160E+05
F21	1,009E+02	1,116E+02	1,243E+02	3,871E+07	1,060E+08	8,237E+07	2,832E+06	1,007E+07	2,131E+07
F22	5,306E+01	6,610E+01	6,366E+01	2,754E+04	3,826E+05	3,006E+05	3,381E+02	1,187E+03	7,672E+03
F23	0,000E+00	0,000E+00	2,930E-13	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00
F24	8,609E-09	5,015E-09	1,284E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	3,275E-09
F25	0,000E+00	0,000E+00	1,319E-01	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00
F26	1,939E-02	2,776E-02	2,075E-02	0,000E+00	0,000E+00	3,945E+00	0,000E+00	0,000E+00	0,000E+00
F27	4,424E+01	1,400E+01	2,005E+01	0,000E+00	1,126E+03	9,388E+02	0,000E+00	0,000E+00	3,430E+02
F28	0,000E+00	0,000E+00	3,527E+01	0,000E+00	0,000E+00	8,239E-05	0,000E+00	0,000E+00	5,569E-13
F29	0,000E+00	0,000E+00	2,959E+01	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00
F30	0,000E+00	0,000E+00	4,067E+02	0,000E+00	0,000E+00	2,009E-13	0,000E+00	0,000E+00	2,009E-13

$2 \times \mathbf{y}_{best,\,gg}$will not be available. The new point $\mathbf{y}_{i,\,gg}$ generated by Eq. (25) may be located more than ten times closer to origin $\mathbf{O}$ than $\mathbf{y}_{best,gg-1}$ with probability 0.5%. But, if $\mathbf{y}_{i,\,gg}$ becomes $\mathbf{y}_{best,\,gg}$, going back is impossible during a single generation, and of extremely low probability even within 10 generations (assuming that any individual located further from $\mathbf{O}$ does not become the best during that time).

The moves by nine individuals according to Eq. (21) (V1 and C2) are not performed along any line. However, as the point generated by $N(\mathbf{y}_{best,gg-1}, \sigma)$ is almost always close to $\mathbf{y}_{best,gg-1}$, and the value of $rand_1(0, 1) \in [0, 1]$, the individual $\mathbf{y}_{i,\,gg}$ will be dragged towards the interval defined by $[\mathbf{y}_{best,gg-1}, 2 \times \mathbf{y}_{best,gg-1}]$. The smaller value of $rand_2(0, 1)$ is generated, the closer to this interval the new point is located. As a result, these nine individuals are dragged towards $[\mathbf{y}_{best,gg-1}, 2 \times \mathbf{y}_{best,gg-1}]$.

Let us summarise the effects of the local search procedure defined by variant V1. The current best individuals $\mathbf{y}_{best,\,gg}$ are dragged towards the origin $\mathbf{O}$, and the other individuals are dragged towards the $[\mathbf{y}_{best,\,gg}, 2 \times \mathbf{y}_{best,\,gg}]$ interval. As a result, the probability of sampling points close to $\mathbf{O}$ is much higher than the probability of sampling points located in other parts of the search space. Moreover, if points located close to $\mathbf{O}$ are good ones (hence, if sampled, such points often become $\mathbf{y}_{best,\,gg}$), after some time all 10 individuals would be attracted towards them and points located far from $\mathbf{O}$ will become unavailable. On the contrary, when points located close to $\mathbf{O}$ are poor, although frequently sampled such points would not become $\mathbf{y}_{best,\,gg}$ and individuals of the local search procedure would not converge towards $\mathbf{O}$.

Now let us discuss the variant V2. Here the situation is slightly different, as values sampled from Normal distributions $N_1(0, 1)$ and $N_2(0, 1)$ are unbounded. However, the probability of sampling points close to, or far from $\mathbf{O}$ is also highly uneven. In case C1, when $\mathbf{y}_{i,gg-1} = \mathbf{y}_{best,gg-1}$ (Eq. (26)) one may note that, for example, sampling of points located ten times closer to the origin $\mathbf{O}$ than the current best individual would occur with probability $\Pr(|\mathbf{y}_{best,gg} - \mathrm{O}| < 0.1 \cdot |\mathbf{y}_{best,gg-1} - \mathrm{O}|) = \Pr((1 + N_1(0, 1) - N_2(0, 1)) \in [-0.1, 0.1]) \approx 4.4\%$. However, if $\mathbf{y}_{i,\,gg}$ become $\mathbf{y}_{best,\,gg}$, the probability of sampling points located 10 times farther from $\mathbf{O}$ than $\mathbf{y}_{best,\,gg}$ in one step would be very low, as $\Pr(|\mathbf{y}_{best,gg} - \mathrm{O}| > 10 \cdot |\mathbf{y}_{best,gg-1} - \mathrm{O}|) = (\Pr((N_1(0, 1) - N_2(0, 1)) < -11) + \Pr((N_1(0, 1) - N_2(0, 1)) > 9)) \approx 0\%$Note also that in the case of D-dimensional problems the probability of randomly sampling from uniform distribution the points within subset of $V^l \in [-10, 10]^D$ when sampling is made within $V^S \in [-100, 100]^D$ is $\frac{1}{10^D}$. Hence, although sampling the points within $V^l \in [-10, 10]^D$ randomly has extremely low probability when D is not very small, the best individual sample points within $V^l \in [-10, 10]^D$ with probability $\geq 4.4\%$, irrespective where it is located.

In case C2 (i.e. when $\mathbf{y}_{i,gg-1} \neq \mathbf{y}_{best,gg-1}$ see Eq. (23)) of variant V2 a similar situation is observed. As $N(\mathbf{y}_{best,gg-1}, \sigma)$ is almost always close to $\mathbf{y}_{best,gg-1}$, if $-1.1 < N_1(0, 1) < -0.9$ and $-0.1 \leq N_2(0, 1) \leq 0.1$ the point $\mathbf{y}_{i,\,gg}$ reached according to

L-SHADE-EpSin-NLS pseudocode:

---

1. set generation counter $g = 0$ and problem-related information: D (problem dimensionality), $MNFC$ (maximum number of function calls);
2. set control parameters: $NP_{g=0} = 18 \cdot D$, $NP_{min} = 4$, $H = 5$, $p\% = 11\%$, $rar = 1.4$, $freq = 0.5$;
3. create memory sets: **MF**, **MCR, MFQ** ($|\mathbf{MF}| = |\mathbf{MCR}| = |\mathbf{MFQ}| = H$, $MF_i = MCR_i = MFQ_i = 0.5$, $i = 1{:}H$), **A**, **SF, SCR, SFQ** ($|\mathbf{A}| = 0$);
4. set L^d and U^d as lower and upper bounds of d-th variable ($d = 1,\ldots,D$);
5. initialize population **P** of individuals $\mathbf{x}_{i,g}$ according to eq. (2);
6. find objective function values $f(\mathbf{x}_{i,g})$ for all individuals in **P**;
7. set the number of function calls used $NFC = NP_0$;
8. find the best solution found so far $\mathbf{x}_{best,0}$;
9. in the first generation the initial population is used, hence $NP_1 = NP_0$;
10. **while** $NFC < MNFC$
11. $g = g + 1$;
12. $|\mathbf{SF}| = |\mathbf{SCR}| = |\mathbf{SFQ}| = 0$, $|\mathbf{A}_{\mathbf{temp}}| = 0$;
13. **for** $i = 1{:}\ NP_g$
14. generate r_i according to eq. (6);
15. generate CR_i according to eq. (8);
16. **if** $NFC < 0.5 \cdot MNFC$
17. generate fq_i according to eq. (18);
18. **if** $rand(0,1) < 0.5$
19. generate F_i according to eq. (16);
20. **else**
21. generate F_i according to eq. (17);
22. **end if;**
23. **else**
24. generate F_i according to eq. (7);
25. **end if;**
26. choose randomly $\mathbf{x}_{pbest,g}$ for i among round($NP_g \cdot p\%$) best individuals in **P;**
27. perform mutation following eq. (9);
28. perform crossover following eq. (10);
29. perform bounds handling and selection (eq. (5));
30. **if** $f(\mathbf{u}_{i,g}) \le f(\mathbf{x}_{i,g})$
31. $\mathbf{x}_{i,g} \rightarrow \mathbf{A}_{\mathbf{temp}}$, $CR_i \rightarrow \mathbf{SCR}$, $F_i \rightarrow \mathbf{SF}$, $fq_i \rightarrow \mathbf{SFQ}$;
32. **end if;**
33. $NFC = NFC + 1$;
34. **end for;**
35. reduce population size to NP_{g+1} following eq. (15);
36. $\mathbf{A} \leftarrow \mathbf{A} \cup \mathbf{A}_{\mathbf{temp}}$;
37. reduce the archive size to $NA_{g+1} = \lceil rar \cdot NP_{g+1} \rceil$ by discarding random individuals from **A**;
38. update **MF**, **MCR, MFQ** following eqs (11), (12), (19);
39. **end while;**

Fig. 6. The pseudocode of L-SHADE-EpSin-NLS algorithm.

Eq. (23) will be on average 10 times closer to the origin **O** than the one among $\mathbf{y}_{best,gg-1}$ and $\mathbf{y}_{i,gg-1}$ that is further from **O**. The probability of both $-1.1 < N_1(0,1) < -0.9$ and $-0.1 \le N_2(0,1) \le 0.1$ is $\approx 0.38\%$. As there are nine individuals in each generation that perform moves according to C2, such situations happen on average once in 20 generations. Also, $\mathbf{y}_{i,gg-1}$ is attracted by $\mathbf{y}_{best,gg-1}$. The probability that $\mathbf{y}_{i,\,gg}$ will be located, on average, 10 times closer to $\mathbf{y}_{best,gg-1}$ than $\mathbf{y}_{i,gg-1}$ (i.e. the probability that $-0.1 < N_1(0,1) < 0.1$ and $-1.1 \le N_2(0,1) \le -0.9$) is also $\approx 0.38\%$. As a result, on average, in every 20 generations one individual comes very close to $\mathbf{y}_{best,gg-1}$. Based on the discussion given in the previous paragraph, because $\mathbf{y}_{best,gg-1}$ is dragged by **O**, the probability of sampling the point in the vicinity of **O** by any individual is much higher than sampling points located somewhere else.

Above we showed that both variants of the local search procedure used by the L-SHADE-EpSin algorithm, the one introduced in the paper [1] (V1) and another (V2) implemented in the code available from CEC2016 conference web page (http://www.ntu.edu.sg/home/EPNSugan) are structurally biased, i.e. the probability that they sample points located close to

Table 5
51-Run mean performance of L-SHADE-EpSin variants V1 and V2 when one of individuals in initial population is, or is not located in the origin **O**. Tests are made on 50-dimensional CEC2014 problems. V1 – refers to the variant of local search procedure described in the paper [1], V2 – refers to the variant of the local search procedure used in the official L-SHADE-EpSin code available on CEC2016 web page, f(**O**) – denotes the value of the objective function in the origin of the coordinate system.

problem	f(**O**)	L-SHADE-EpSin-V1		L-SHADE-EpSin-V2	
		O not included in the initial population	**O** included in the initial population	**O** not included in the initial population	**O** included in the initial population
F1	1.665E + 10	4.203E-04	1,242E-04	5.089E-03	3.045E-04
F2	1.996E + 11	3.288E-14	3,288E-14	3.399E-14	3.567E-14
F3	6.963E + 08	5.796E-14	5,796E-14	5.684E-14	5.684E-14
F4	7.259E + 04	6.029E + 01	5,866E + 01	5.853E + 01	5.848E + 01
F5	2.169E + 01	2.025E + 01	2,025E + 01	2.025E + 01	2.025E + 01
F6	9.074E + 01	2.046E-04	1,987E-04	1.822E-04	1.036E-02
F7	1.879E + 03	1.070E-13	9,808E-14	9.808E-14	9.808E-14
F8	9.088E + 02	1.048E-09	4,957E-09	1.474E-09	8.151E-10
F9	1.011E + 03	2.839E + 01	2,941E + 01	2.995E + 01	3.118E + 01
F10	1.843E + 04	4.900E-02	5,513E-02	5.402E-02	5.619E-02
F11	1.833E + 04	3.177E + 03	3,127E + 03	3.090E + 03	3.022E + 03
F12	1.395E + 01	2.096E-01	2,093E-01	2.132E-01	2.154E-01
F13	9.717E + 00	2.068E-01	2,042E-01	2.092E-01	2.052E-01
F14	4.796E + 02	1.877E-01	1,842E-01	1.867E-01	1.930E-01
F15	2.739E + 07	5.568E + 00	5,581E + 00	5.422E + 00	5.536E + 00
F16	2.501E + 01	1.659E + 01	1,667E + 01	1.650E + 01	1.658E + 01
F17	3.878E + 09	3.998E + 02	3,615E + 02	3.531E + 02	3.523E + 02
F18	3.821E + 10	1.771E + 01	1,808E + 01	1.698E + 01	1.769E + 01
F19	8.929E + 03	9.921E + 00	9,910E + 00	9.778E + 00	9.445E + 00
F20	3.218E + 09	6.349E + 00	6,163E + 00	6.614E + 00	5.470E + 00
F21	1.867E + 09	3.302E + 02	3,209E + 02	3.111E + 02	3.209E + 02
F22	6.109E + 06	8.779E + 01	1,078E + 02	1.236E + 02	1.132E + 02
F23	**2.000E + 02**	**2.000E + 02**	**2,000E + 02**	**2.000E + 02**	**2.000E + 02**
F24	**2.000E + 02**	**2.000E + 02**	**2,000E + 02**	**2.000E + 02**	**2.000E + 02**
F25	**2.000E + 02**	**2.000E + 02**	**2,000E + 02**	**2.000E + 02**	**2.000E + 02**
F26	2.000E + 02	1.002E + 02	1,002E + 02	1.002E + 02	1.002E + 02
F27	**2.000E + 02**	**2.981E + 02**	**2,000E + 02**	**2.020E + 02**	**2.000E + 02**
F28	**2.000E + 02**	**2.000E + 02**	**2,000E + 02**	**2.000E + 02**	**2.000E + 02**
F29	**2.000E + 02**	**2.000E + 02**	**2,000E + 02**	**2.000E + 02**	**2.000E + 02**
F30	**2.000E + 02**	**2.000E + 02**	**2,000E + 02**	**2.000E + 02**	**2.000E + 02**

the origin **O** is much higher than the probability that they sample points located in other parts of the search space. If the solutions located close to the origin **O** are not attractive (poor), this leads only to wasting the function calls. However, if good solutions are located close to the origin **O** (as in case of CEC2014 problems F23-F30), this biases the results. If good solutions are located anywhere else in the search space, they are not found by the local search procedure used by L-SHADE-EpSin.

To confirm this empirically, in Table 3 we list the mean (respective standard deviations are given in Table 4) results obtained within 51 runs on 50-dimensional CEC2014 problems by a few variants of the L-SHADE-EpSin algorithm and its local search procedures used alone, namely:

- L-SHADE-EpSin with local search procedure V2 (L-SHADE-EpSin-V2) given in the code available on the CEC2016 page (http://www.ntu.edu.sg/home/EPNSugan),
- L-SHADE-EpSin with local search procedure V1 (L-SHADE-EpSin-V1) discussed in the paper [1] (obtained by modifying the code of the local search procedure mentioned above),
- L-SHADE-EpSin without any local search (L-SHADE-EpSin-NLS); note that L-SHADE-EpSin-NLS differs from L-SHADE-EpSin only by skipping local search procedure, see pseudocode in Fig. 6,
- the local search procedure V2 used alone (skipping all other procedures of L-SHADE-EpSin),
- the local search procedure V1 used alone (skipping all other procedures of L-SHADE-EpSin).

The maximum number of function calls is set to 500,000, the value suggested for CEC2014 problems in [27], but local search procedures are also tested with *MNFC* set to 2500 (the value they in fact use within L-SHADE-EpSin algorithm, when 10 individuals are run for 250 generations [1]) and 10,000. In Table 3 we also show the objective function value of each problem for the origin **O**. Note that the objective function value for the global optimum of each problem equals 0. The control parameter values of all those procedures are the same as suggested in [1].

From Tables 3 and 4 we see that the results obtained by 3 versions of the L-SHADE-EpSin algorithm clearly differ from the results obtained by local search procedures V1 and V2 for problems F1-F22. In the case of all problems F1-F22 the results obtained by the L-SHADE-EpSin variants are always better than those obtained by V1 and V2 procedures with 500,000 function calls, and almost for each case the difference is of orders of magnitude. However, in the case of problems F23-F25 and F27-F30 the local search procedures lead to the same results as two versions of the L-SHADE-EpSin algorithm that use local search; L-SHADE-EpSin-NLS that does not use any local search procedure performs much worse. The function values found for problems F23-F26 and F28-F30 by the local search procedures are almost identical to the values noted in the

Table 6
51-Run mean performances (accompanied by respective standard deviations given in *italics*) of UMOEA-II and UMOEA-II-M algorithms on 50-dimensional CEC2014 problems. The maximum number of function calls is set to 500,000. Problems on which UMOEA-II takes advantage due to the use of structurally biased mutation strategy are bolded. f(**O**) – denotes the value of the objective function in the origin of the coordinate system.

problem	f(**O**)	UMOEA-II mean	UMOEA-II-M mean	UMOEA-II st. dev.	UMOEA-II-M st. dev.
F1	1.665E + 10	2.019E-03	1.377E-03	*1.265E-03*	*9.888E-04*
F2	1.996E + 11	3.700E-13	9.251E-14	*3.449E-13*	*2.532E-14*
F3	6.963E + 08	1.148E-13	7.356E-14	*3.854E-14*	*2.616E-14*
F4	7.259E + 04	9.618E + 00	9.790E + 00	*2.946E + 01*	*2.942E + 01*
F5	2.169E + 01	2.000E + 01	2.000E + 01	*1.963E-04*	*2.239E-04*
F6	9.074E + 01	3.514E-01	3.567E-01	*6.472E-01*	*6.036E-01*
F7	1.879E + 03	1.137E-13	1.115E-13	*0.000E + 00*	*1.592E-14*
F8	9.088E + 02	1.890E-12	1.331E-12	*1.133E-12*	*1.076E-12*
F9	1.011E + 03	4.383E + 00	3.595E + 00	*1.702E + 00*	*1.599E + 00*
F10	1.843E + 04	7.422E-01	2.026E-01	*7.503E-01*	*2.262E-01*
F11	1.833E + 04	3.544E + 03	3.380E + 03	*5.606E + 02*	*5.995E + 02*
F12	1.395E + 01	1.111E-01	1.069E-01	*5.801E-02*	*5.736E-02*
F13	9.717E + 00	1.502E-01	1.429E-01	*3.617E-02*	*3.829E-02*
F14	4.796E + 02	3.016E-01	2.965E-01	*2.100E-02*	*2.319E-02*
F15	2.739E + 07	5.033E + 00	5.299E + 00	*8.772E-01*	*1.025E + 00*
F16	2.501E + 01	1.811E + 01	1.839E + 01	*6.151E-01*	*9.177E-01*
F17	3.878E + 09	1.211E + 03	1.014E + 03	*3.169E + 02*	*2.932E + 02*
F18	3.821E + 10	7.684E + 01	6.865E + 01	*1.945E + 01*	*2.010E + 01*
F19	8.929E + 03	8.953E + 00	9.167E + 00	*2.225E + 00*	*1.723E + 00*
F20	3.218E + 09	1.469E + 01	1.354E + 01	*4.582E + 00*	*4.583E + 00*
F21	1.867E + 09	4.477E + 02	4.200E + 02	*1.246E + 02*	*1.169E + 02*
F22	6.109E + 06	1.613E + 02	1.835E + 02	*9.931E + 01*	*1.158E + 02*
F23	2.000E + 02	**2.000E + 02**	**3.440E + 02**	*0.000E + 00*	*3.912E-13*
F24	2.000E + 02	**2.000E + 02**	**2.662E + 02**	*1.413E-13*	*8.440E + 00*
F25	2.000E + 02	**2.000E + 02**	**2.045E + 02**	*0.000E + 00*	*1.995E + 00*
F26	2.000E + 02	1.001E + 02	1.001E + 02	*3.477E-02*	*3.118E-02*
F27	2.000E + 02	**2.000E + 02**	**3.443E + 02**	*0.000E + 00*	*3.577E + 01*
F28	2.000E + 02	**2.000E + 02**	**1.117E + 03**	*0.000E + 00*	*2.858E + 01*
F29	2.000E + 02	8.102E + 02	8.051E + 02	*4.269E + 01*	*4.472E + 01*
F30	2.000E + 02	8.738E + 03	8.768E + 03	*3.383E + 02*	*5.009E + 02*

Table 7
51-run mean performances of UMOEA-II, when one of individuals in initial population is, or is not located in the origin **O**. Tests are made on 50-dimensional CEC2014 problems. f(**O**) – denotes the value of the objective function in the origin of the coordinate system.

problem	f(**O**)	UMOEA-II **O** not included in the initial population	UMOEA-II- **O** included in the initial population
F1	1.665E + 10	2.019E-03	2.298E-03
F2	1.996E + 11	3.700E-13	2.970E-13
F3	6.963E + 08	1.148E-13	1.081E-13
F4	7.259E + 04	9.618E + 00	7.694E + 00
F5	2.169E + 01	2.000E + 01	2.000E + 01
F6	9.074E + 01	3.514E-01	3.653E-01
F7	1.879E + 03	1.137E-13	1.450E-04
F8	9.088E + 02	1.890E-12	1.926E-12
F9	1.011E + 03	4.383E + 00	4.179E + 00
F10	1.843E + 04	7.422E-01	5.629E-01
F11	1.833E + 04	3.544E + 03	3.528E + 03
F12	1.395E + 01	1.111E-01	1.238E-01
F13	9.717E + 00	1.502E-01	1.404E-01
F14	4.796E + 02	3.016E-01	2.988E-01
F15	2.739E + 07	5.033E + 00	5.320E + 00
F16	2.501E + 01	1.811E + 01	1.828E + 01
F17	3.878E + 09	1.211E + 03	1.126E + 03
F18	3.821E + 10	7.684E + 01	7.799E + 01
F19	8.929E + 03	8.953E + 00	9.440E + 00
F20	3.218E + 09	1.469E + 01	1.377E + 01
F21	1.867E + 09	4.477E + 02	4.674E + 02
F22	6.109E + 06	1.613E + 02	1.743E + 02
F23	2.000E + 02	**2.000E + 02**	**2.000E + 02**
F24	2.000E + 02	**2.000E + 02**	**2.000E + 02**
F25	2.000E + 02	**2.000E + 02**	**2.000E + 02**
F26	2.000E + 02	1.001E + 02	1.001E + 02
F27	2.000E + 02	**2.000E + 02**	**2.000E + 02**
F28	2.000E + 02	**2.000E + 02**	**2.000E + 02**
F29	2.000E + 02	8.102E + 02	**2.000E + 02**
F30	2.000E + 02	8.738E + 03	**2.000E + 02**

Table 8
51-Run mean performances (accompanied by the respective standard deviations given in *italics*) of L-SHADE-50, L-SHADE-EpSin (V1 and V2), L-SHADE-EpSin-NLS and L-SHADE on 50-dimensional CEC2014 problems.

problem	L-SHADE-50 mean	L-SHADE-EpSin-V1 mean	L-SHADE-EpSin-V2 mean	L-SHADE-EpSin-NLS mean	L-SHADE mean	L-SHADE-50 st. dev.	L-SHADE-EpSin-V1 st. dev.	L-SHADE-EpSin-V2 st. dev.	L-SHADE-EpSin-NLS st. dev.	L-SHADE st. dev.
F1	**1.253E-06**	4.203E-04	5.089E-03	8.120E-06	1.529E+03	*7.253E-06*	*3.000E-03*	*3.607E-02*	*4.519E-05*	*2.036E+03*
F2	3.511E-14	3.288E-14	3.399E-14	**3.288E-14**	4.012E-14	*1.218E-14*	*1.044E-14*	*1.140E-14*	*1.044E-14*	*1.625E-14*
F3	5.684E-14	5.796E-14	5.684E-14	5.684E-14	**5.573E-14**	*0.000E+00*	*7.960E-15*	*1.137E-14*	*0.000E+00*	*7.960E-15*
F4	5.871E+01	6.029E+01	**5.853E+01**	6.237E+01	6.578E+01	*4.758E+01*	*4.758E+01*	*4.779E+01*	*4.686E+01*	*4.422E+01*
F5	2.024E+01	2.025E+01	2.025E+01	**2.023E+01**	2.025E+01	*3.353E-02*	*3.224E-02*	*3.688E-02*	*4.803E-02*	*2.958E-02*
F6	1.918E-04	2.046E-04	**1.822E-04**	1.040E-02	1.436E-01	*2.122E-04*	*1.822E-04*	*2.004E-04*	*7.266E-02*	*4.219E-01*
F7	1.025E-13	1.070E-13	9.808E-14	9.808E-14	**4.235E-14**	*3.414E-14*	*2.702E-14*	*3.951E-14*	*3.951E-14*	*5.551E-14*
F8	**6.983E-11**	1.048E-09	1.474E-09	4.429E-10	9.718E-11	*2.536E-10*	*3.580E-09*	*4.624E-09*	*2.959E-09*	*1.183E-10*
F9	2.794E+01	2.839E+01	2.995E+01	2.938E+01	**1.191E+01**	*6.457E+00*	*6.104E+00*	*5.953E+00*	*5.620E+00*	*1.952E+00*
F10	3.006E-02	4.900E-02	5.402E-02	**2.945E-02**	4.971E-02	*1.896E-02*	*2.402E-02*	*2.663E-02*	*1.715E-02*	*2.265E-02*
F11	**3.039E+03**	3.177E+03	3.090E+03	3.105E+03	3.237E+03	*2.832E+02*	*3.202E+02*	*3.331E+02*	*3.175E+02*	*2.609E+02*
F12	2.105E-01	2.096E-01	2.132E-01	**2.023E-01**	2.119E-01	*2.513E-02*	*2.535E-02*	*2.486E-02*	*2.679E-02*	*2.808E-02*
F13	2.005E-01	2.068E-01	2.092E-01	2.031E-01	**1.555E-01**	*2.267E-02*	*1.855E-02*	*2.089E-02*	*1.712E-02*	*2.156E-02*
F14	1.920E-01	1.877E-01	**1.867E-01**	1.965E-01	3.131E-01	*2.374E-02*	*2.232E-02*	*2.142E-02*	*2.251E-02*	*5.892E-02*
F15	5.370E+00	5.568E+00	5.422E+00	5.367E+00	**5.068E+00**	*5.531E-01*	*5.240E-01*	*5.340E-01*	*4.666E-01*	*3.755E-01*
F16	**1.644E+01**	1.659E+01	1.650E+01	1.658E+01	1.680E+01	*4.838E-01*	*3.416E-01*	*4.332E-01*	*4.519E-01*	*4.827E-01*
F17	3.605E+02	3.998E+02	**3.531E+02**	3.633E+02	1.718E+03	*1.932E+02*	*1.613E+02*	*1.641E+02*	*1.524E+02*	*4.598E+02*
F18	1.878E+01	1.771E+01	**1.698E+01**	1.745E+01	1.066E+02	*5.576E+00*	*6.538E+00*	*5.303E+00*	*5.939E+00*	*1.649E+01*
F19	9.369E+00	9.921E+00	9.778E+00	9.555E+00	**7.962E+00**	*1.257E+00*	*8.883E-01*	*1.192E+00*	*1.222E+00*	*1.739E+00*
F20	**6.150E+00**	6.349E+00	6.614E+00	6.181E+00	1.486E+01	*1.956E+00*	*2.056E+00*	*1.997E+00*	*2.404E+00*	*4.551E+00*
F21	3.317E+02	3.302E+02	3.111E+02	**3.108E+02**	6.318E+02	*1.171E+02*	*1.009E+02*	*1.116E+02*	*1.243E+02*	*1.780E+02*
F22	1.150E+02	**8.779E+01**	1.236E+02	1.064E+02	1.008E+02	*6.060E+01*	*5.306E+01*	*6.610E+01*	*6.366E+01*	*6.656E+01*
F23	3.440E+02	2.000E+02	2.000E+02	3.440E+02	3.440E+02	*3.326E-13*	*0.000E+00*	*0.000E+00*	*2.930E-13*	*3.178E-13*
F24	2.678E+02	2.000E+02	2.000E+02	2.679E+02	2.749E+02	*1.328E+00*	*8.609E-09*	*5.015E-09*	*1.284E+00*	*7.618E-01*
F25	2.049E+02	2.000E+02	2.000E+02	2.049E+02	2.052E+02	*1.629E-01*	*0.000E+00*	*0.000E+00*	*1.319E-01*	*2.798E-01*
F26	**1.002E+02**	**1.002E+02**	**1.002E+02**	**1.002E+02**	**1.002E+02**	*2.412E-02*	*1.939E-02*	*2.776E-02*	*2.075E-02*	*1.925E-02*
F27	3.076E+02	2.981E+02	2.020E+02	3.116E+02	3.441E+02	*1.682E+01*	*4.424E+01*	*1.400E+01*	*2.005E+01*	*2.863E+01*
F28	1.140E+03	2.000E+02	2.000E+02	1.141E+03	1.119E+03	*3.729E+01*	*0.000E+00*	*0.000E+00*	*3.527E+01*	*4.057E+01*
F29	8.095E+02	2.000E+02	2.000E+02	8.038E+02	8.031E+02	*3.849E+01*	*0.000E+00*	*0.000E+00*	*2.959E+01*	*3.203E+01*
F30	8.501E+03	2.000E+02	2.000E+02	8.429E+03	8.666E+03	*3.120E+02*	*0.000E+00*	*0.000E+00*	*4.067E+02*	*3.305E+02*
nr. of wins	6	2	6	6	7					

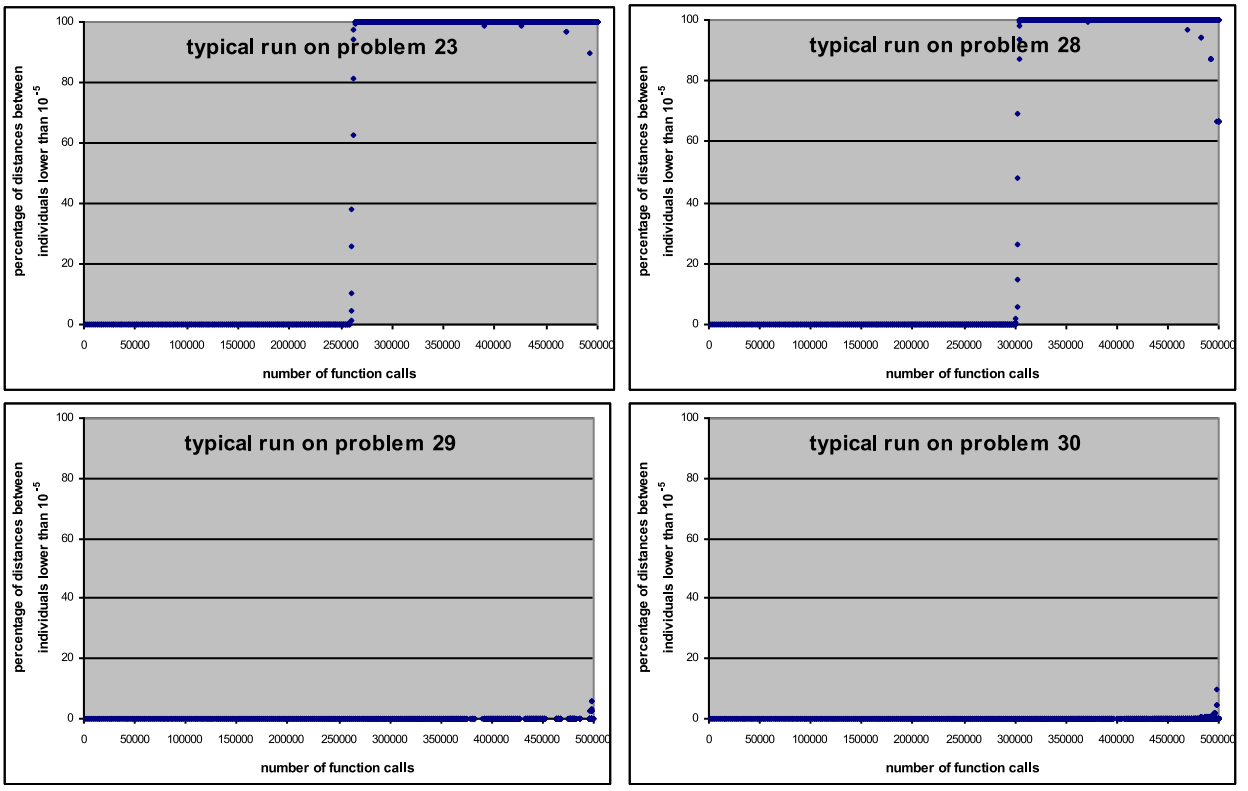

Fig. 7. The percentage of distances between individuals that are lower than 10^{-5} during a typical UMOEA-II run on 50-dimensional problems F23, F28, F29 and F30 from CEC2014 set with respect to the number of function calls used.

Table 9
Number of 50-dimensional CEC2014 problems on which L-SHADE-50 is better/worse than each among 4 other SHADE-L-based variants. In case of L-SHADE-EpSin-V1 and L-SHADE-EpSin-V2 only 23 problems are considered (F1-F22 and F26), in case of L-SHADE-SpSin-NLS and L-SHADE – all 30 problems are considered.

	L-SHADE-EpSin-V1	L-SHADE-EpSin-V2	L-SHADE-EpSin-NLS	L-SHADE
L-SHADE-50 better	16	13	15	19
L-SHADE-50 worse	6	8	11	9

origin **O**. This confirms that local search procedures are useless for problems F1-F22 and F26; on the contrary, all L-SHADE-EpSin procedures apart from the local search are useless for problems F23-F25 and F27-F30. Procedures V1 and V2 lead to similar results for each problem apart from F27. For problem F27, local search procedure V2 reaches values closer to **O** than the procedure V1 (described in the paper [1]), leading to much better results of L-SHADE-EpSin-V2 than L-SHADE-EpSIn-V1. We may also note that no variant of the L-SHADE-EpSin algorithm is able to reach solutions better than those found by the local search procedures for all problems F23-F30 except for the problem F26. But in the case of problem F26, solutions better than those found by the local search procedures are found by L-SHADE-EpSin even when it does not use any local search (L-SHADE-EpSin-NLS). Hence, although in the case of problem F26 both variants of the local search procedure also converge towards **O**, L-SHADE-EpSin finds better solutions elsewhere – like in the case of problems F1-F22.

Empirical tests discussed above confirm that the good results of L-SHADE-EpSin on problems F23-F25 and F27-F30 are the effect of a structurally biased local search procedure, which favours sampling points in the neighbourhood of the origin **O**. Hence, for the further discussion and future applications we recommend using the variant of L-SHADE-EpSin without the local search procedure (L-SHADE-EpSin-NLS). If our arguments about why to skip the local search procedure that drags algorithm towards the origin **O** were not convincing for some readers, we may suggest another test. Let us note that not all current DE variants are initialized randomly within the bounds [37]. Hence, we may suggest that during initialization each algorithm should sample the point in the origin **O**. In other words, the point located in the origin **O** is included into the initial population as one of individuals. Let us then run the L-SHADE-EpSin algorithm (in variants V1 and V2) 51 times on each CEC2014 problem. In Table 5 we give a comparison between results obtained when the point in the origin is, and when it is not included in the initial population. One may easily find that for a specific variant (V1 or V2) the results are almost

L-SHADE-50 pseudocode:

---

1. set generation counter $g = 0$ and problem-related information: D (problem dimensionality), $MNFC$ (maximum number of function calls);
2. set control parameters: $NP_{g=0} = 18 \cdot D$, $NP_{min} = 4$, $H = 5$, $p\% = 11\%$, $rar = 1.4$;
3. create memory sets: **MF**, **MCR** ($|\mathbf{MF}| = |\mathbf{MCR}| = H$, $MF_i = MCR_i = 0.5$, $i = 1{:}H$), **A**, **SF, SCR,** ($|\mathbf{A}| = 0$);
4. set L^d and U^d as lower and upper bounds of d-th variable ($d = 1,\ldots,D$);
5. initialize population **P** of individuals $\mathbf{x}_{i,g}$ according to eq. (2);
6. find objective function values $f(\mathbf{x}_{i,g})$ for all individuals in **P**;
7. set the number of function calls used $NFC = NP_0$;
8. find the best solution found so far $\mathbf{x}_{best,0}$;
9. in the first generation the initial population is used, hence $NP_1 = NP_0$;
10. **while** $NFC < MNFC$
11. $g = g + 1$;
12. $|\mathbf{SF}| = |\mathbf{SCR}| = 0$, $|\mathbf{A}_{\mathbf{temp}}| = 0$;
13. **for** $i = 1{:}\ NP_g$
14. generate r_i according to eq. (6);
15. generate CR_i according to eq. (8);
16. **if** $NFC < 0.5 \cdot MNFC$
17. $F_i = 0.5$;
18. **else**
19. generate F_i according to eq. (7);
20. **end if;**
21. choose randomly $\mathbf{x}_{pbest,g}$ for i among round($NP_g \cdot p\%$) best individuals in **P;**
22. perform mutation following eq. (9);
23. perform crossover following eq. (10);
24. perform bounds handling and selection (eq. (5));
25. **if** $f(\mathbf{u}_{i,g}) \le f(\mathbf{x}_{i,g})$
26. $\mathbf{x}_{i,g} \rightarrow \mathbf{A}_{\mathbf{temp}}$, $CR_i \rightarrow \mathbf{SCR}$, $F_i \rightarrow$ **SF;**
27. **end if;**
28. $NFC = NFC + 1$;
29. **end for;**
30. reduce population size to NP_{g+1} following eq. (15);
31. $\mathbf{A} \leftarrow \mathbf{A} \cup \mathbf{A}_{\mathbf{temp}}$;
32. reduce the archive size to $NA_{g+1} = \lceil rar \cdot NP_{g+1} \rceil$ by discarding random individuals from **A**;
33. update **MF**, **MCR** following eqs (11) and (12);
34. **end while;**

Fig. 8. The pseudocode of L-SHADE-50 algorithm.

identical, when the solution in the origin is or is not included (the only larger difference, noted for problem F6 when the V2 variant is used, occur because once among 51 runs the L-SHADE-EpSin variant V2 with **O** included in the initial population got stuck in a poor local minimum with function value ≈ 0.51; this may always happen for stochastic algorithms). The initial individual located in the origin is quickly removed from the population for problems F1-F22, as it is a poor one. However, in case of problems F23-F25 and F27-F30 it is, and always remains during the whole run, the best solution found (despite not being located in the global optimum). Hence, in the case of problems F23-F25 and F27-F30 for which L-SHADE-EpSin seems to perform so well, all its search is equivalent to sampling the point in the origin (which is a much cheaper solution, requiring just a single function call).

3.3. Structural bias in UMOEA-II algorithm

UMOEA-II is a procedure even more complicated than L-SHADE-EpSin, but it seems that its structurally biased part, which, like in case of L-SHADE-EpSin, drags the population towards the origin **O**, may be identified more simply. UMOEA-II is composed of two sub-algorithms, CMA-ES and MODE; here we pay attention to the second. Among three mutation

Table 10
Summary of CEC 2011 real-world problems [6].

problem	dimensionality	topic
F1	6	parameter estimation for frequency-modulated sound waves (music/engineering)
F2	30	Lennard-Jones potential problem (physics)
F3	1	bifunctional catalyst blend optimal control (chemistry)
F4	1	optimal control of a non-linear stirred tank reactor (chemistry)
F5	30	Tersoff potential function for silicon – variant 1 (physics)
F6	30	Tersoff potential function for silicon – variant 2 (physics)
F7	20	spread spectrum radar polly phase code design (communication)
F8	7	transmission network expansion planning (power engineering)
F9	126	large scale transmission pricing problem (economy/power engineering)
F10	12	circular antenna array design (communication systems/physics)
F11.1	120	dynamic economic dispatch – variant 1 (economy/power engineering)
F11.2	216	dynamic economic dispatch – variant 2 (economy/power engineering)
F11.3	6	static economic load dispatch – variant 1 (economy/ power engineering)
F11.4	13	static economic load dispatch – variant 2 (economy/ power engineering)
F11.5	15	static economic load dispatch – variant 3 (economy/ power engineering)
F11.6	40	static economic load dispatch – variant 4 (economy/ power engineering)
F11.7	140	static economic load dispatch – variant 5 (economy/ power engineering)
F11.8	96	hydrothermal scheduling – variant 1 (economy/ hydraulics/ engineering)
F11.9	96	hydrothermal scheduling – variant 2 (economy/ hydraulics/ engineering)
F11.10	96	hydrothermal scheduling – variant 3 (economy/ hydraulics/ engineering)
F12	26	spacecraft trajectory optimization – Messenger (space missions/ engineering)
F13	22	spacecraft trajectory optimization – Cassini (space missions/ engineering)

strategies used within MODE, the one defined by Eq. (24) much differs from the mutation strategies used by DE algorithms [7], as it multiplies the base point $\mathbf{x}_{r1,g}$ to which the differential vector is added by a scalar value F_i. The adaptation of F_i in MODE [12] follows the SHADE-based scheme (see [46]), hence $F_i \in (0, 1]$. Hence, due to multiplying all the coordinates of the base point $\mathbf{x}_{r1,g}$ used in mutation defined by Eq. (24) by a value from (0,1] interval, the base point is moved towards the origin $\mathbf{O}$. Hence, the offspring $\mathbf{v}_{i,g}$ may be located anywhere in the search space only as long as the two points within the difference vector $(\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g})$ are far from each other. If individuals start clustering towards some local optima, which, after some time, frequently happens in DE (see [34,53]) then for pairs of individuals located close to the same local optimum $|\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}| \to 0$, hence $\mathbf{v}_{i,g}$moves towards $\mathbf{O}$. The more individuals cluster close to each other, the more frequently the offspring will be dragged towards $\mathbf{O}$. How many generations would be needed to reach the origin $\mathbf{O}$ when $|\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}| \to 0$ depends on the distribution of available F_i values. In MODE F_i is generated from the Cauchy distribution with scale parameter $\gamma = 0.1$ [12,46] and location parameter x_0 that is adapted within [0,1]. As a result even if the location parameter $x_0 = 1$ (when the probability of sampling small values of F_i is the lowest), the probability of generating values of F_i lower than |0.01| is $\Pr(-0.01 < F_i < 0.01) \approx 1/1600$. This probability increases to $\approx 1/400$ when $x_0 = 0.5$ and to $\approx 1/16$ when $x_0 = 0$. Moreover, F_i may relatively frequently be 100 times lower (|0.0001|). If for example $x_0 = 0.1$, $\Pr(-0.0001 < F_i < 0.0001) \approx 1/3100$, and if $x_0 = 0$ $\Pr(-0.0001 < F_i < 0.0001) \approx 1/1600$. Even if $x_0 = 0.5$, $\Pr(-0.0001 < F_i < 0.0001) \approx 1/36000$. As a result, when the mutation strategy defined by Eq. (24) is used (note that depending on the generation, this strategy is applied to between 10% and 80% of MODE individuals) and some individuals within the population converge very close to each other (which is frequent in DE after sufficient number of function calls is used, see [34,53]), the points close to the origin $\mathbf{O}$ are sampled with excessive probability. As in the case of 50-dimensional CEC2014 problems there are 500,000 function calls allowed per algorithm's run, the probabilities of sampling a point very close to the origin $\mathbf{O}$ given above means that**,** irrespectively where the base points are located, points located very close to $\mathbf{O}$ will be sampled in almost every run.

Note also that MODE is an elitist approach (like each DE algorithm, but opposed to local search procedures discussed in the previous sub-section for L-SHADE-EpSin), hence new individuals are accepted only if they are better than the parent, and are retained until a better offspring is generated. As a result, since any individual gets close to the $\mathbf{O}$, if solutions in the vicinity of $\mathbf{O}$ are better than those located somewhere else, sampling of points located closer and closer to $\mathbf{O}$ will become frequent. On the other hand, if solutions close to $\mathbf{O}$ are of poor quality, such biased sampling leads only to wasting function calls.

Interestingly, despite the discussed bias, UMOEA-II does not find good solutions located close to $\mathbf{O}$ for problems F29 and F30 (see Table 1), as L-SHADE-EpSin did. The reason is that much fewer MODE individuals cluster very close to the same point when UMOEA-II is run on problems F29 or F30 than when it is run on problems F23-F28. This is illustrated in Fig. 7, which shows the percentages of distances between MODE individuals that are lower than 10^{-5} within all distances between individuals which belong to the MODE sub-population (the number of such distances equals $(NP^2_{MODE} - NP_{MODE})/2$), during a typical UMOEA-II run on the 50-dimensional version of CEC2014 problems F23, F28, F29 and F30. Separate dots represent situations when some individuals converge very close to each other when all others remained further scattered (the case in bottom figures), or when some individuals suddenly left the cluster for short period of time (the cases in upper figures).

The structural bias of UMOEA-II may be easily removed by discarding or modifying the mutation strategy given in Eq. (24). In the simplest case (we call it UMOEA-II-M) the mutation strategy given in Eq. (24) may be modified such that

Table 11
51-Run mean performances (accompanied by the respective standard deviations given in *italics*) of L-SHADE-50, L-SHADE-EpSin (V1 and V2), L-SHADE-EpSin-NLS and L-SHADE on real-world CEC2011 problems. In the last row the number of problems for which particular algorithm performed best is counted.

problem	L-SHADE-50 mean	L-SHADE-EpSin-V1 mean	L-SHADE-EpSin-V2 mean	L-SHADE-EpSin-NLS mean	L-SHADE mean	L-SHADE-50 st. dev.	L-SHADE-EpSin-V1 st. dev.	L-SHADE-EpSin-V2 st. dev.	L-SHADE-EpSin-NLS st. dev.	L-SHADE st. dev.
F1	5.639E-01	9.729E-01	1.688E+00	1.814E+00	**1.403E-04**	*2.286E+00*	*3.438E+00*	*3.961E+00*	*3.965E+00*	*6.445E-04*
F2	**-2.641E+01**	-2.589E+01	-2.596E+01	-2.621E+01	-2.612E+01	*5.586E-01*	*5.808E-01*	*4.893E-01*	*5.589E-01*	*6.266E-01*
F3	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	*1.293E-19*	*1.980E-19*	*1.546E-19*	*1.681E-19*	*2.071E-19*
F4	1.883E+01	**1.447E+01**	1.500E+01	1.811E+01	1.608E+01	*3.118E+00*	*1.075E+00*	*1.933E+00*	*3.318E+00*	*3.022E+00*
F5	**-3.657E+01**	-3.645E+01	-3.635E+01	-3.639E+01	-3.622E+01	*5.004E-01*	*5.049E-01*	*5.388E-01*	*5.298E-01*	*7.687E-01*
F6	-2.912E+01	-2.915E+01	-2.915E+01	**-2.916E+01**	**-2.916E+01**	*2.423E-01*	*7.773E-03*	*9.929E-03*	*4.242E-03*	*4.780E-03*
F7	1.176E+00	1.151E+00	1.166E+00	**1.128E+00**	1.174E+00	*8.807E-02*	*9.184E-02*	*8.263E-02*	*9.031E-02*	*9.378E-02*
F8	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	*0.000E+00*	*0.000E+00*	*0.000E+00*	*0.000E+00*	*0.000E+00*
F9	**1.767E+03**	1.909E+03	1.946E+03	1.824E+03	2.648E+03	*3.557E+02*	*3.411E+02*	*3.344E+02*	*3.613E+02*	*4.364E+02*
F10	**-2.162E+01**	**-2.162E+01**	-2.161E+01	-2.161E+01	**-2.162E+01**	*9.250E-02*	*9.395E-02*	*7.489E-02*	*9.911E-02*	*8.700E-02*
F11	5.208E+04	5.209E+04	5.220E+04	**5.196E+04**	5.199E+04	*5.801E+02*	*6.045E+02*	*5.819E+02*	*5.062E+02*	*5.026E+02*
F12	1.774E+07	1.777E+07	1.777E+07	**1.772E+07**	1.775E+07	*6.544E+04*	*5.856E+04*	*6.322E+04*	*4.425E+04*	*5.684E+04*
F13	1.545E+04	1.545E+04	1.545E+04	1.545E+04	**1.544E+04**	*1.092E+01*	*1.336E+01*	*1.216E+01*	*1.092E+01*	*6.156E-01*
F14	1.811E+04	1.810E+04	1.811E+04	1.810E+04	**1.809E+04**	*4.052E+01*	*4.401E+01*	*4.259E+01*	*3.374E+01*	*3.338E+01*
F15	**3.274E+04**	**3.274E+04**	**3.274E+04**	**3.274E+04**	**3.274E+04**	*8.292E+00*	*8.287E+00*	*1.326E-01*	*1.157E+01*	*3.312E-01*
F16	**1.234E+05**	1.235E+05	**1.234E+05**	1.236E+05	1.240E+05	*5.011E+02*	*5.306E+02*	*5.276E+02*	*4.599E+02*	*5.583E+02*
F17	1.848E+06	1.845E+06	**1.842E+06**	**1.842E+06**	1.866E+06	*1.157E+04*	*1.001E+04*	*1.077E+04*	*1.215E+04*	*9.826E+03*
F18	9.300E+05	**9.299E+05**	9.301E+05	9.300E+05	9.317E+05	*9.915E+02*	*9.665E+02*	*1.184E+03*	*9.818E+02*	*9.626E+02*
F19	9.386E+05	9.387E+05	9.385E+05	**9.384E+05**	9.397E+05	*1.078E+03*	*8.111E+02*	*1.021E+03*	*1.151E+03*	*9.004E+02*
F20	**9.298E+05**	9.300E+05	9.300E+05	9.301E+05	9.315E+05	*8.560E+02*	*1.279E+03*	*1.068E+03*	*1.161E+03*	*1.157E+03*
F21	**1.546E+01**	1.565E+01	1.571E+01	1.551E+01	1.618E+01	*8.110E-01*	*6.467E-01*	*6.138E-01*	*7.746E-01*	*7.609E-01*
F22	1.638E+01	1.669E+01	1.594E+01	1.565E+01	**1.339E+01**	*2.472E+00*	*2.372E+00*	*2.463E+00*	*2.421E+00*	*1.900E+00*
nr. of wins	10	6	5	9	9					

Table 12
Number of real-world CEC2011 problems on which L-SHADE-50 is better/worse than each among 4 other SHADE-L-based variants.

problem	L-SHADE-EpSin-V1	L-SHADE-EpSin-V2	L-SHADE-EpSin-NLS	L-SHADE
L-SHADE-50 better	14	11	11	10
L-SHADE-50 worse	6	9	9	8

$\mathbf{x}_{r1,\,g}$ is not multiplied by F_i

$$\mathbf{v}_{i,g} = \mathbf{x}_{r1,g} + (\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}) \tag{27}$$

The mutation strategy defined in Eq. (27) is a classical DE/rand/1 approach proposed in the first version of DE discussed by Storn and Price in 1997 [45] with scaling factor F set to 1. Note that the only difference between UMOEA-II and UMOEA-II-M is that UMOEA-II-M uses Eq. (27) instead of Eq. (24) as a third mutation strategy option within the MODE procedure. Of course, other choices for the third mutation strategy may be more efficient, but here our main goal is to eliminate the structural bias (in the simplest possible way), not searching for the further modifications of UMOEA-II. The results of both the UMOEA-II and UMOEA-II-M procedures obtained during 51 runs on the 50-dimensional version of CEC2014 problems with maximum number of function calls set to 500,000 (as defined in [27]) are given in Table 6. The results from both variants were obtained during 51 independent runs using the code available from CEC2014 web page. Note, however, that the CMA-ES procedure implemented within official UMOEA-II code did not handle the bounds during first 50% of function calls. As solutions out of the bounds should be considered unfeasible, we removed this restriction from the official UMOEA-II code and bounds are always handled during the entire runs.

From Table 6 we observe that the mean results for the 51-runs for problems F23-F25 and F27-F28 are no longer equal to 200, the objective function value noted in **O**. This suggests that the fact that in mutation defined by Eq. (24) base vector was multiplied by $F_i \in (0, 1]$ was the main source of the structural bias in the UMOEA-II procedure. Obviously, results obtained by UMOEA-II-M on problems F23-F25 and F27-F28 are poorer than those reported for structurally biased UMOEA-II. However, for all other problems the differences between UMOEA-II and UMOEA-II-M are relatively small. We noticed that, surprisingly, for 16 out of the remaining 25 problems UMOEA-II-M achieved a slightly better performance than UMOEA-II, and for only 7 problems – slightly poorer. As the differences are small, this may occur just by chance. However, as frequent sampling of points located close to **O** for most problems does not lead to any improvement, the fact that UMOEA-II-M avoids such waste of function calls may be the reason of its marginally better performance on most problems.

Like in the case of L-SHADE-EpSin, we may also show what happens if one individual of the initial UMOEA-II population is located in the origin **O** (such an individual should be included into *NP1*, the part of the population that is used by structurally-biased MODE) – the results are given in Table 7. The conclusions are similar to those discussed for L-SHADE-EpSin – when the function value of the origin is poor, an individual located in **O** is quickly removed from the population, even though the algorithm is structurally biased towards sampling the region around **O**; however, when **O** is a good solution, UMOEA-II is unable to improve it during any run, and the final performance of UMOEA-II is equal the objective function value sampled in **O** by the initial population, making the whole search procedure useless.

We should also note here that UMOEA-II with one individual located in **O** in the initial generation was stuck in a very poor local optimum in one among 51 runs on problem F7, which highly affected the average performance, even though in all other 50 runs the results were the same as in the case when no individual from the initial population was placed in **O**. We draw the readers attention to this as it indicate the fragility of the comparison of metaheuristics.

4. Simplification of L-SHADE-EpSin

In Section 2.1.3 it is pointed out that L-SHADE [47] and L-SHADE-EpSin [1] algorithms differ mainly in two elements: L-SHADE-EpSin uses a local search procedure and introduces a novel adaptation scheme that is implemented during the first half of the search. However, as shown in Section 3.2, the local search procedure used by L-SHADE-EpSin is structurally biased and favours sampling solutions located close to the origin **O**, hence it needs to be removed. The algorithm with the structurally biased part removed has been called L-SHADE-EpSin-NLS (see pseudocode in Fig. 6). Hence, apart from minor control parameter settings, there is only one difference between L-SHADE-EpSin-NLS and L-SHADE [47] – during the first 50% of function calls for adaptation of the control parameter F_i L-SHADE-EpSin-NLS uses a specific scheme based on two sinusoidal functions; during the second 50% of function calls L-SHADE and L-SHADE-EpSin perform the same operations. As one may conclude from the discussion in Section 2.1.3 and pseudocode given in Fig. 3, the sinusoidal-based adaptation scheme is complicated and introduces additional parameters, one of which (fq_i) needs to be adapted during the run.

The impact of the adaptation mechanisms on the algorithms' performance may be complicated and not necessarily positive [56]. We suggest that the whole procedure of adaptation of control parameter F_i by means of sinusoidal functions is not needed and may be skipped - this will further significantly simplify the algorithm. We, however, do not claim that L-SHADE-EpSin should be reduced to L-SHADE. As we show empirically below, the important novelty introduced by L-SHADE-EpSin is the different approach to control parameter F_i adaptation during the first and the second half of the search. Let us label

Table 13
Algorithms tested. Abbreviations: DE – Differential Evolution; PSO – Particle Swarm Optimization; GA – Genetic Algorithm; D – problem dimensionality. [*] – marks algorithms which MATLAB codes were obtained from authors of the source papers or relevant web pages (see Comments).

short name	description	reference	year	population size	Comments
ALC-PSO	PSO with an aging leader and challengers	[5]	2013	20	The initial particle velocities are set to 0. Velocities are restricted to be not larger than 50% of the bounds span.
AMALGAM [*]	A multialgorithm genetically adaptive method for single objective optimization	[50]	2009	variable, initialized with 10 when $D < 20$; 15 when $D < 40$; 20 when $D \geq 40$	AMALGAM variant that merges CMA-ES, GA and PSO is used, as suggested and described in [50]. AMALGAM makes a number of sub-runs within time budget; for the conditions of commencing sub-run, see [50]. In each consecutive sub-run the population is twice larger, until it become 32 times larger than in the first sub-run. The MATLAB code of AMALGAM has been obtained from prof. Vrugt, first author of [50].
AM-DEGL	Adaptive memetic DEGL	[33]	2013	5D within [10,500]	Adaptive Memetic DE variant, based on DEGL, SADE and NMA. Note that the number of local nearest neighbours is assumed to be not smaller than 6 in this study.
cNrGA [*]	Non-revisiting GA with adaptive mutation using constant memory	[28]	2016	100	A variant of GA that remembers most of its history to avoid re-evaluations of already sampled solutions. The code of cNrGA has been obtained from author's web page www.ee.cityu.edu.hk/~syyuen/Public/Code.html.
CoBiDE	Differential Evolution based on covariance matrix learning and bimodal distribution parameter setting	[52]	2014	60	One of the first two DE variants (the second has been introduced in [18]) that put attention to the problem of coordinate system rotation.
GA-MPC [*]	Genetic algorithm with multi-parent crossover	[10]	2011	90	GA that was the winner of CEC2011 competition. The MATLAB code of GA-MPC has been submitted from CEC2011 web page (www3.ntu.edu.sg/home/epnsugan/index_files/CEC11-RWP/CEC11-RWP.htm). One change to the original code has been made – if objective function is "not a number" (what may happen in a few CEC2011 problems even if solutions within a bounds are sampled) in the original code 0 had been set; in the modified code a very large number (10^{100}) is set in such a case.
GLPSO	Genetic Learning PSO	[17]	2016	50	A recent PSO variant that mixes various elements from PSO and GA.
HCLPSO [*]	Heterogeneous comprehensive learning PSO	[29]	2015	40	A modified variant of very popular CLPSO algorithm [26]. The maximum velocities are set to 20% of the bounds span. The initial velocities are generated randomly form uniform distribution within the allowed margin. The code has been obtained from one of co-authors of [29] paper.
IDE [*]	DE with an Individual-independent mechanism	[48]	2015	50 if $D \leq 10$; 200 if $D > 50$; linear approximation within [50,200] interval if $10 < D \leq 50$	A novel variant of DE that introduces concepts of individual-dependent setting of control parameters and mutation strategies that are based on function values found by particular individuals. The code has been obtained from one of co-authors of [48] paper.
IILPSO	PSO with Interswarm Interactive Learning Strategy	[39]	2016	50	A multi-swarm PSO variant with specific scheme of information sharing between swarms. The maximum velocities are set to 12.5% of the bounds span. The initial velocities are generated randomly form uniform distribution within the allowed margin.
LBBO [*]	Linearized Biogeography-based Optimization	[43]	2014	50	Biogeography-based optimization [42] variant that uses additional local search technique and is able to re-initialize during run. The code of LBBO has been submitted from Prof. Simon's web page (http://academic.csuohio.edu/simond/bbo/linearized/). A change to Systematic Search procedure has been made, to prevent it from using too many function calls during run (in the original code in this procedure it was not checked whether the number of allowed function calls has been exceeded or not).
MDE-pBX	Modified DE with p-best crossover	[24]	2012	100	MDE_pBX introduces another DE crossover and mutation operators.
PMS	Parallel Memetic Structures	[2]	2013	1	PMS is non-population based heuristic, partly based on RA and some DE operators. Similarly to the approach proposed in [22], its construction is based on Ockham's razor principle, that is so frequently violated in modern metaheuristics.
PSO-iw	PSO with inertia weight	[41]	1998	40	The initial particle velocities are randomly generated within 20% of the bounds span. During run velocities are also restricted to be not higher than 20% of the bounds span.
SADE	Self-adaptive DE	[38]	2009	50	Probably the most popular adaptive DE variant. The learning period (LP in [38]) has been set to 40, as a modest choice according to parameter sensitivity presented in [38].

Table 14
51-Run mean performances of 17 algorithms on 50-dimensional CEC2014 problems.

	L-SHADE-50	UMOEA-II-M	ALC-PSO	AMALGAM	AM-DEGL	cNrGA	CoBiDE	GA-MPC	GLPSO
F1	1.25E-06	1.38E-03	2.15E+06	1.45E-13	2.21E+04	8.74E+06	2.80E+05	4.00E+05	5.94E+05
F2	3.51E-14	9.25E-14	6.94E+04	8.69E-14	2.08E-07	3.02E+05	1.30E-01	4.65E+03	7.41E+03
F3	5.68E-14	7.36E-14	1.52E+04	8.92E-14	9.65E-09	3.16E+04	4.62E-03	1.55E+01	2.17E+03
F4	5.87E+01	9.79E+00	1.07E+02	5.40E+00	1.27E+01	1.07E+02	3.37E+01	5.68E+01	7.22E+01
F5	2.02E+01	2.00E+01	2.10E+01	2.01E+01	2.00E+01	2.00E+01	2.03E+01	2.03E+01	2.09E+01
F6	1.92E-04	3.57E-01	3.58E+01	7.30E-02	2.43E+01	2.43E+01	8.16E+00	4.89E+00	2.15E+01
F7	1.03E-13	1.11E-13	6.27E-02	1.40E-13	2.56E-03	1.91E-01	4.46E-15	3.00E-03	1.14E-02
F8	6.98E-11	1.33E-12	1.49E-02	4.89E-02	3.00E+01	7.91E-01	2.01E-14	5.05E+01	1.40E-13
F9	2.79E+01	3.60E+00	2.24E+02	4.46E+00	5.63E+01	8.03E+01	8.86E+01	5.47E+01	1.07E+02
F10	3.01E-02	2.03E-01	3.46E+02	1.66E+01	1.01E+03	5.94E-01	1.31E+01	1.13E+03	1.59E+00
F11	3.04E+03	3.38E+03	5.38E+03	4.16E+02	5.47E+03	4.39E+03	3.77E+03	3.58E+03	4.97E+03
F12	2.10E-01	1.07E-01	1.03E+00	6.59E-02	1.78E-01	1.38E-01	3.07E-01	7.91E-02	1.29E-01
F13	2.00E-01	1.43E-01	5.85E-01	1.57E-01	3.01E-01	3.52E-01	3.63E-01	4.25E-01	4.49E-01
F14	1.92E-01	2.96E-01	6.01E-01	3.01E-01	2.72E-01	5.31E-01	2.78E-01	3.73E-01	3.76E-01
F15	5.37E+00	5.30E+00	3.34E+01	4.80E+00	6.42E+00	2.27E+01	6.17E+00	5.52E+00	1.36E+01
F16	1.64E+01	1.84E+01	2.11E+01	1.79E+01	1.93E+01	1.82E+01	1.78E+01	1.72E+01	1.77E+01
F17	3.60E+02	1.01E+03	5.97E+05	2.27E+03	1.38E+03	3.59E+06	1.09E+04	2.91E+04	9.64E+04
F18	1.88E+01	6.86E+01	2.56E+03	1.51E+02	6.95E+01	1.55E+03	6.85E+01	3.23E+02	1.39E+03
F19	9.37E+00	9.17E+00	2.91E+01	1.13E+01	1.18E+01	2.27E+01	6.85E+00	5.87E+00	1.36E+01
F20	6.15E+00	1.35E+01	5.26E+03	1.83E+02	3.65E+01	7.39E+04	3.12E+01	2.17E+02	1.57E+03
F21	3.32E+02	4.20E+02	1.98E+05	1.57E+03	8.02E+02	3.64E+06	2.91E+03	1.75E+04	6.61E+04
F22	1.15E+02	1.83E+02	1.24E+03	2.23E+02	4.80E+02	9.97E+02	5.16E+02	9.05E+02	8.97E+02
F23	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.37E+02	3.44E+02
F24	2.68E+02	2.66E+02	2.76E+02	2.57E+02	2.73E+02	2.67E+02	2.68E+02	2.69E+02	2.70E+02
F25	2.05E+02	2.05E+02	2.17E+02	2.04E+02	2.05E+02	2.18E+02	2.07E+02	2.00E+02	2.24E+02
F26	1.00E+02	1.00E+02	1.62E+02	1.00E+02	1.00E+02	1.41E+02	1.04E+02	1.00E+02	1.59E+02
F27	3.08E+02	3.44E+02	1.41E+03	3.41E+02	3.82E+02	9.79E+02	4.72E+02	1.22E+03	9.67E+02
F28	1.14E+03	1.12E+03	3.08E+03	1.14E+03	1.21E+03	1.50E+03	1.20E+03	3.68E+02	1.76E+03
F29	8.09E+02	8.05E+02	8.05E+06	7.15E+02	7.22E+02	4.17E+03	9.69E+02	2.15E+02	1.44E+06
F30	8.50E+03	8.77E+03	2.59E+04	9.48E+03	9.56E+03	1.46E+04	8.71E+03	1.23E+03	1.24E+04
	HCLPSO	**IDE**	**IILPSO**	**LBBO**	**MDE_pBX**	**PMS**	**PSO-iw**	**SADE**	
F1	6.27E+05	1.16E+06	6.95E+05	1.21E+05	7.09E+04	4.61E+05	8.30E+06	1.40E+05	
F2	1.41E+02	1.10E+00	3.47E+03	2.22E+03	5.77E+03	7.14E+03	2.23E+04	4.62E+00	
F3	1.50E+03	4.20E+00	1.82E+03	6.10E-06	7.82E-03	3.01E+04	8.18E+02	9.31E+02	
F4	8.81E+01	5.51E+01	1.10E+02	2.47E+01	3.46E+01	7.47E+01	2.17E+02	3.82E+01	
F5	2.02E+01	2.03E+01	2.00E+01	2.00E+01	2.05E+01	2.00E+01	2.09E+01	2.05E+01	
F6	1.47E+01	2.79E-02	3.77E+01	3.74E+01	5.76E+00	3.82E+01	2.69E+01	1.55E+01	
F7	6.99E-03	3.09E-03	2.09E-02	1.93E-03	5.12E-03	1.17E-02	1.19E-02	1.30E-02	
F8	4.61E-13	5.07E-07	9.48E-01	0.00E+00	5.56E+01	1.67E-07	8.92E+01	3.90E-02	
F9	1.02E+02	5.85E+01	2.17E+02	2.95E+02	1.21E+02	3.96E+02	1.44E+02	6.81E+01	
F10	2.87E+00	3.96E+01	6.20E+00	6.84E+01	1.53E+03	7.51E-02	2.23E+03	2.20E-01	
F11	4.06E+03	4.34E+03	4.97E+03	5.92E+03	8.32E+03	6.35E+03	5.90E+03	5.33E+03	
F12	1.51E-01	3.63E-01	1.95E-01	3.75E-01	8.15E-01	1.76E-01	7.30E-01	6.17E-01	
F13	3.45E-01	3.04E-01	4.59E-01	4.46E-01	2.99E-01	5.16E-01	5.49E-01	4.06E-01	
F14	2.78E-01	2.74E-01	3.41E-01	4.07E-01	3.11E-01	4.57E-01	5.61E-01	3.05E-01	
F15	1.11E+01	6.33E+00	4.17E+01	2.69E+01	8.47E+00	6.18E+01	1.50E+01	1.54E+01	
F16	1.79E+01	1.89E+01	1.85E+01	1.93E+01	2.01E+01	1.88E+01	2.02E+01	1.89E+01	
F17	1.38E+05	4.93E+03	2.51E+05	4.20E+04	4.64E+03	1.76E+05	1.02E+06	2.20E+04	
F18	1.75E+02	5.08E+01	6.73E+02	1.60E+02	3.37E+02	2.62E+03	2.86E+04	6.47E+02	
F19	1.50E+01	9.64E+00	2.55E+01	6.09E+01	9.63E+00	2.53E+01	5.61E+01	1.67E+01	
F20	1.57E+03	3.63E+01	2.38E+03	1.99E+02	1.85E+02	3.23E+04	6.32E+02	1.07E+03	
F21	1.21E+05	1.20E+03	1.03E+05	6.04E+04	1.30E+03	2.42E+05	4.42E+05	1.33E+04	
F22	6.56E+02	3.37E+02	1.18E+03	1.50E+03	8.17E+02	1.51E+03	8.79E+02	4.99E+02	
F23	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.47E+02	3.44E+02	
F24	2.60E+02	2.57E+02	2.64E+02	2.73E+02	2.79E+02	2.87E+02	2.75E+02	2.77E+02	
F25	2.12E+02	2.07E+02	2.36E+02	2.18E+02	2.11E+02	2.24E+02	2.20E+02	2.22E+02	
F26	1.02E+02	1.08E+02	1.53E+02	1.14E+02	1.02E+02	1.04E+02	1.73E+02	1.34E+02	
F27	5.60E+02	3.10E+02	1.43E+03	1.32E+03	5.62E+02	1.32E+03	1.18E+03	7.44E+02	
F28	1.65E+03	1.25E+03	4.39E+03	3.30E+03	1.17E+03	5.68E+03	2.63E+03	1.22E+03	
F29	1.22E+03	9.57E+02	7.00E+05	2.17E+03	9.20E+02	6.22E+06	4.33E+07	1.00E+03	
F30	1.05E+04	9.67E+03	1.43E+04	1.03E+04	9.50E+03	1.30E+04	4.92E+04	1.07E+04	

Table 15
Standard deviations of the results obtained on 50-dimensional CEC2014 problems.

	L-SHADE-50	UMOEA-II-M	ALC-PSO	AMALGAM	AM-DEGL	cNrGA	CoBiDE	GA-MPC	GLPSO
F1	7.25E-06	9.89E-04	9.77E+05	4.32E-14	3.02E+04	3.29E+06	1.24E+05	1.55E+05	2.96E+05
F2	1.22E-14	2.53E-14	2.61E+05	1.83E-14	1.11E-06	1.90E+06	3.70E-01	5.29E+03	8.34E+03
F3	0.00E+00	2.62E-14	7.33E+03	5.94E-14	4.42E-08	1.38E+04	6.87E-03	2.69E+01	2.06E+03
F4	4.76E+01	2.94E+01	3.87E+01	2.10E+01	3.23E+01	2.99E+01	3.54E+01	3.39E+01	3.20E+01
F5	3.35E-02	2.24E-04	8.60E-02	1.54E-01	2.64E-03	1.17E-02	2.80E-01	2.25E-01	3.82E-01
F6	2.12E-04	6.04E-01	3.92E+00	2.16E-01	5.81E+00	3.57E+00	3.11E+00	1.77E+00	3.26E+00
F7	3.41E-14	1.59E-14	1.91E-01	4.87E-14	5.53E-03	1.38E-01	2.23E-14	4.91E-03	1.30E-02
F8	2.54E-10	1.08E-12	1.06E-01	2.46E-01	6.54E+00	9.19E-01	4.38E-14	1.20E+01	1.06E-13
F9	6.46E+00	1.60E+00	4.71E+01	3.64E+00	1.53E+01	1.59E+01	1.54E+01	1.25E+01	2.17E+01
F10	1.90E-02	2.26E-01	1.78E+02	2.96E+01	3.23E+02	9.83E-01	3.28E+00	5.23E+02	1.96E+00
F11	2.83E+02	6.00E+02	8.08E+02	5.02E+02	7.91E+02	8.98E+02	6.41E+02	9.74E+02	7.25E+02
F12	2.51E-02	5.74E-02	5.55E-01	7.39E-02	7.45E-02	3.60E-02	2.92E-01	3.00E-02	5.28E-02
F13	2.27E-02	3.83E-02	9.48E-02	4.57E-02	5.53E-02	7.21E-02	6.40E-02	7.63E-02	7.78E-02
F14	2.37E-02	2.32E-02	3.09E-01	4.44E-02	4.35E-02	2.55E-01	2.65E-02	1.63E-01	1.45E-01
F15	5.53E-01	1.02E+00	8.71E+00	7.76E-01	2.54E+00	7.08E+00	1.10E+00	1.28E+00	3.27E+00
F16	4.84E-01	9.18E-01	4.53E-01	1.33E+00	6.36E-01	8.17E-01	8.75E-01	1.09E+00	9.34E-01
F17	1.93E+02	2.93E+02	3.19E+05	5.06E+02	5.65E+02	1.70E+06	7.84E+03	1.64E+04	4.63E+04
F18	5.58E+00	2.01E+01	2.00E+03	5.01E+01	2.91E+01	1.12E+03	4.31E+01	4.86E+02	1.14E+03
F19	1.26E+00	1.72E+00	1.83E+01	3.69E+00	3.38E+00	1.36E+01	1.28E+00	1.35E+00	2.59E+00
F20	1.96E+00	4.58E+00	1.76E+03	1.18E+02	1.63E+01	3.45E+04	9.54E+00	1.99E+02	1.02E+03
F21	1.17E+02	1.17E+02	1.59E+05	4.11E+02	3.45E+02	2.97E+06	3.68E+03	1.54E+04	4.10E+04
F22	6.06E+01	1.16E+02	3.37E+02	1.25E+02	1.98E+02	3.56E+02	1.72E+02	2.73E+02	2.47E+02
F23	3.33E-13	3.91E-13	4.96E-02	6.29E-04	3.03E-13	2.28E-02	3.68E-13	6.29E-13	1.81E-07
F24	1.33E+00	8.44E+00	4.50E+00	1.31E+00	4.35E+00	6.55E+00	2.85E+00	6.24E+00	6.96E+00
F25	1.63E-01	2.00E+00	7.88E+00	2.02E+00	2.53E+00	3.71E+00	1.18E+00	1.01E-01	6.69E+00
F26	2.41E-02	3.12E-02	6.31E+01	8.04E-02	5.75E-02	7.81E+01	1.95E+01	5.67E-02	4.95E+01
F27	1.68E+01	3.58E+01	1.25E+02	2.64E+01	1.11E+02	9.57E+01	7.19E+01	1.83E+02	9.97E+01
F28	3.73E+01	2.86E+01	7.77E+02	7.19E+01	6.48E+01	1.16E+02	6.51E+01	3.90E+00	4.87E+02
F29	3.85E+01	4.47E+01	1.53E+07	1.41E+02	1.04E+02	5.09E+03	2.15E+02	3.06E+00	7.17E+06
F30	3.12E+02	5.01E+02	3.72E+04	6.62E+02	8.90E+02	2.81E+03	4.64E+02	4.13E+02	2.30E+03
	HCLPSO	**IDE**	**IILPSO**	**LBBO**	**MDE_pBX**	**PMS**	**PSO-iw**	**SADE**	
F1	2.16E+05	3.32E+05	3.75E+05	1.52E+05	3.19E+04	2.60E+05	1.13E+07	9.86E+04	
F2	1.96E+02	1.63E+00	4.99E+03	7.07E+03	7.23E+03	9.07E+03	1.04E+05	2.02E+01	
F3	8.12E+02	2.78E+00	1.11E+03	1.14E-05	1.83E-02	1.39E+04	1.06E+03	1.76E+03	
F4	1.48E+01	3.57E+01	3.65E+01	4.11E+01	3.63E+01	5.25E+01	7.52E+01	3.41E+01	
F5	5.75E-02	6.35E-02	8.92E-03	2.91E-06	3.35E-01	7.76E-03	1.69E-01	4.78E-02	
F6	3.42E+00	5.63E-02	3.52E+00	4.20E+00	2.48E+00	3.75E+00	4.36E+00	3.58E+00	
F7	6.89E-03	4.62E-03	4.11E-02	4.25E-03	7.63E-03	1.46E-02	1.47E-02	1.68E-02	
F8	9.49E-14	1.68E-06	9.01E-01	0.00E+00	1.23E+01	2.67E-08	1.64E+01	1.95E-01	
F9	1.73E+01	8.64E+00	3.54E+01	4.69E+01	3.08E+01	9.66E+01	3.24E+01	1.28E+01	
F10	1.65E+01	5.83E+01	2.70E+00	9.57E+01	5.43E+02	2.74E-02	6.33E+02	3.18E-01	
F11	4.33E+02	5.94E+02	6.77E+02	8.13E+02	1.12E+03	9.40E+02	8.73E+02	7.47E+02	
F12	5.17E-02	6.54E-02	4.58E-02	1.64E-01	3.59E-01	5.32E-02	2.62E-01	7.96E-02	
F13	5.79E-02	2.63E-02	8.56E-02	9.50E-02	7.38E-02	9.46E-02	9.96E-02	5.22E-02	
F14	2.91E-02	2.67E-02	1.23E-01	7.23E-02	4.38E-02	1.99E-01	3.16E-01	3.25E-02	
F15	3.08E+00	1.21E+00	1.86E+01	9.59E+00	4.06E+00	2.20E+01	4.02E+00	4.42E+00	
F16	7.54E-01	5.51E-01	7.18E-01	7.11E-01	8.26E-01	7.52E-01	8.06E-01	4.25E-01	
F17	1.23E+05	3.17E+03	3.32E+05	7.06E+04	2.57E+03	1.01E+05	1.23E+06	2.52E+04	
F18	8.45E+01	2.15E+01	7.10E+02	6.30E+01	6.21E+02	1.25E+03	1.92E+05	4.44E+02	
F19	2.85E+00	9.76E-01	1.17E+01	2.27E+01	6.58E+00	1.97E+01	2.39E+01	9.59E+00	
F20	1.10E+03	9.22E+00	1.85E+03	3.64E+02	8.91E+01	1.43E+04	3.11E+02	1.44E+03	
F21	6.47E+04	2.46E+02	5.26E+04	1.33E+05	5.34E+02	2.07E+05	5.11E+05	2.39E+04	
F22	2.54E+02	1.05E+02	3.33E+02	3.19E+02	4.34E+02	3.77E+02	2.88E+02	1.45E+02	
F23	9.77E-13	4.35E-13	7.02E-03	2.98E-04	4.94E-13	3.28E-05	1.34E+00	2.87E-13	
F24	3.71E+00	2.33E+00	7.06E+00	4.69E+00	3.16E+00	4.79E+01	4.47E+00	3.10E+00	
F25	2.46E+00	6.24E-01	8.06E+00	6.81E+00	5.25E+00	1.49E+01	4.00E+00	7.53E+00	
F26	1.40E+01	2.71E+01	5.02E+01	3.47E+01	1.40E+01	1.95E+01	4.48E+01	4.74E+01	
F27	1.58E+02	2.31E+01	2.86E+02	9.75E+01	7.54E+01	2.58E+02	1.92E+02	8.39E+01	
F28	1.62E+02	8.56E+01	1.06E+03	8.32E+02	8.43E+01	1.37E+03	6.73E+02	7.16E+01	
F29	1.91E+02	1.26E+02	4.98E+06	9.47E+02	8.32E+01	1.36E+07	8.06E+07	1.65E+02	
F30	1.01E+03	4.92E+02	3.68E+03	1.53E+03	7.99E+02	2.10E+03	3.32E+04	1.20E+03	

Table 16
51-run mean performances of 17 algorithms on CEC2011 problems.

	L-SHADE-50	UMOEA-II-M	ALC-PSO	AMALGAM	AM-DEGL	cNrGA	CoBiDE	GA-MPC	GLPSO
F1	5.64E-01	4.00E-12	1.11E+01	4.79E+00	1.58E+00	1.39E+01	0.00E+00	1.58E+00	1.01E+01
F2	-2.64E+01	2.70E+01	-1.87E+01	-1.83E+01	-2.66E+01	-2.36E+01	-2.11E+01	-2.74E+01	-2.69E+01
F3	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05
F4	1.88E+01	1.94E+01	1.39E+01	1.47E+01	2.02E+01	1.86E+01	1.93E+01	1.42E+01	1.56E+01
F5	-3.66E+01	3.55E+01	-3.20E+01	-2.99E+01	-3.48E+01	-3.32E+01	-3.49E+01	-3.40E+01	-3.45E+01
F6	-2.91E+01	2.90E+01	-2.46E+01	-2.34E+01	-2.84E+01	-2.56E+01	-2.88E+01	-2.78E+01	-2.66E+01
F7	1.18E+00	7.43E-01	1.14E+00	6.05E-01	9.33E-01	1.34E+00	6.46E-01	1.03E+00	9.53E-01
F8	2.20E+02	2.20E+02	2.20E+02	2.20E+02	2.20E+02	2.20E+02	2.20E+02	2.20E+02	2.20E+02
F9	1.77E+03	3.39E+03	5.91E+05	3.01E+03	4.46E+03	1.72E+03	1.76E+03	3.89E+03	2.43E+03
F10	-2.16E+01	2.16E+01	-1.62E+01	-1.94E+01	-2.17E+01	-1.45E+01	-2.17E+01	-2.17E+01	-2.07E+01
F11.1	5.21E+04	1.97E+05	8.16E+05	5.21E+04	5.40E+04	5.25E+04	5.23E+04	5.26E+04	5.25E+04
F11.2	1.77E+07	1.76E+07	1.88E+07	1.74E+07	1.91E+07	1.76E+07	1.77E+07	1.08E+06	1.79E+07
F11.3	1.54E+04	1.94E+04	1.55E+04	1.55E+04	1.54E+04	1.55E+04	1.54E+04	1.54E+04	1.55E+04
F11.4	1.81E+04	1.84E+04	1.92E+04	1.81E+04	1.81E+04	1.92E+04	1.81E+04	1.83E+04	1.92E+04
F11.5	3.27E+04	6.49E+04	3.30E+04	3.28E+04	3.28E+04	3.30E+04	3.27E+04	3.28E+04	3.30E+04
F11.6	1.23E+05	1.26E+05	1.39E+05	1.36E+05	1.25E+05	1.38E+05	1.28E+05	1.37E+05	1.39E+05
F11.7	1.85E+06	3.79E+06	2.13E+06	2.15E+06	1.93E+06	1.97E+07	1.91E+06	2.15E+06	2.05E+06
F11.8	9.30E+05	9.30E+05	1.18E+06	9.57E+05	9.40E+05	1.01E+06	9.38E+05	9.92E+05	9.54E+05
F11.9	9.39E+05	9.39E+05	1.47E+06	9.59E+05	9.59E+05	1.73E+06	9.51E+05	1.21E+06	1.26E+06
F11.10	9.30E+05	9.30E+05	1.14E+06	9.53E+05	9.41E+05	1.01E+06	9.38E+05	1.05E+06	9.48E+05
F12	1.55E+01	2.14E+01	1.66E+01	1.62E+01	1.60E+01	1.81E+01	1.40E+01	1.25E+01	1.55E+01
F13	1.64E+01	2.96E+01	2.23E+01	1.89E+01	1.56E+01	2.14E+01	1.25E+01	1.05E+01	2.12E+01
	HCLPSO	**IDE**	**IILPSO**	**LBBO**	**MDE_pBX**	**PMS**	**PSO-iw**	**SADE**	
F1	9.58E-01	1.11E+00	1.20E+01	2.96E+00	3.25E+00	1.34E+01	1.15E+01	6.06E-02	
F2	-2.67E+01	-2.65E+01	-2.64E+01	-2.69E+01	-2.48E+01	-2.04E+01	-2.30E+01	-2.37E+01	
F3	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	1.15E-05	
F4	1.41E+01	1.46E+01	1.42E+01	1.39E+01	1.63E+01	1.40E+01	1.41E+01	1.76E+01	
F5	-3.55E+01	-3.53E+01	-3.44E+01	-3.60E+01	-3.30E+01	-3.42E+01	-3.29E+01	-3.51E+01	
F6	-2.82E+01	-2.87E+01	-2.50E+01	-2.73E+01	-2.40E+01	-2.83E+01	-2.27E+01	-2.90E+01	
F7	1.05E+00	8.96E-01	1.26E+00	8.00E-01	1.10E+00	1.23E+00	9.86E-01	1.12E+00	
F8	2.20E+02	2.20E+02	2.23E+02	2.20E+02	2.20E+02	1.57E+03	2.21E+02	2.20E+02	
F9	1.62E+03	6.22E+03	6.28E+03	6.00E+04	2.21E+03	1.32E+03	2.45E+05	1.96E+03	
F10	-2.12E+01	-2.16E+01	-1.80E+01	-1.47E+01	-2.16E+01	-1.25E+01	-1.82E+01	-2.16E+01	
F11.1	5.27E+04	5.85E+04	5.22E+04	5.14E+04	5.25E+04	5.23E+04	1.94E+05	5.31E+04	
F11.2	1.75E+07	1.91E+07	1.76E+07	2.19E+07	1.74E+07	1.76E+07	2.41E+07	1.75E+07	
F11.3	1.54E+04	1.54E+04	1.55E+04	1.55E+04	1.54E+04	1.55E+04	1.55E+04	1.54E+04	
F11.4	1.91E+04	1.84E+04	1.91E+04	1.83E+04	1.81E+04	1.90E+04	1.91E+04	1.83E+04	
F11.5	3.30E+04	3.28E+04	3.31E+04	3.29E+04	3.28E+04	3.32E+04	3.31E+04	3.28E+04	
F11.6	1.37E+05	1.35E+05	1.43E+05	1.53E+05	1.29E+05	1.47E+05	1.41E+05	1.32E+05	
F11.7	2.00E+06	1.96E+06	2.08E+06	4.43E+07	1.93E+06	2.74E+06	1.97E+06	1.92E+06	
F11.8	9.57E+05	9.60E+05	1.23E+06	3.46E+06	9.39E+05	8.35E+06	9.76E+05	9.57E+05	
F11.9	1.39E+06	1.38E+06	1.75E+06	3.11E+06	9.50E+05	6.64E+06	1.27E+06	1.27E+06	
F11.10	9.51E+05	9.62E+05	1.21E+06	2.72E+06	9.39E+05	7.23E+06	9.63E+05	9.51E+05	
F12	1.50E+01	1.42E+01	1.52E+01	1.57E+01	1.58E+01	2.50E+01	1.66E+01	1.62E+01	
F13	1.86E+01	1.61E+01	2.34E+01	2.00E+01	1.85E+01	2.89E+01	2.37E+01	1.54E+01	

L-SHADE-50 an algorithm that works exactly like L-SHADE-EpSin-NLS but keeps the control parameters of F_i set to 0.5 for each individual during the first 50% of function calls. The pseudocode of L-SHADE-50 is given in Fig. 8.

In Table 8 the mean results (accompanied by the respective standard deviations) obtained during 51 runs on 50-dimensional CEC2014 problems by the L-SHADE-50 algorithm are compared against respective results achieved by L-SHADE-EpSin-V1, L-SHADE-EpSin-V2, L-SHADE-EpSin-NLS and L-SHADE. Due to the structural bias present in L-SHADE-EpSin-V1 and L-SHADE-EpSin-V2 that seriously affect the results obtained for problems F23-F25 and F27-F30, these 7 problems are disregarded when comparing all 5 variants together. Also, as all 5 variants are very similar to each other, instead of using average ranking and statistical tests, we count the number of problems for which the particular algorithm wins against the four others (including draws), and then present (see Table 9) one-by-one comparisons of L-SHADE-50 against each other L-SHADE-based variant. In one-by-one comparisons of L-SHADE-50 against L-SHADE and L-SHADE-EpSin-NLS all 30 problems are considered, as such algorithms do not include the structurally biased local search procedure; comparisons against L-SHADE-EpSin-V1 and L-SHADE-EpSin-V2 is based on 23 problems (F1-F22 and F26).

Among 23 problems considered for all 5 L-SHADE variants, L-SHADE performs best for 7 problems, L-SHADE-50, L-SHADE-EpSin-NLS and L-SHADE-EpSin-V2 for 6, and L-SHADE-EpSin-V1 for only 2 problems. Although L-SHADE gained the largest number of wins, from Table 9 we note that the number of 50-dimensional CEC2014 problems for which L-SHADE-50 performs better than each out of 4 competitors is always higher than the number of problems for which it performs worse.

Table 17
Standard deviations of the results obtained on CEC2011 problems.

	L-SHADE-50	UMOEA-II-M	ALC-PSO	AMALGAM	AM-DEGL	cNrGA	CoBiDE	GA-MPC	GLPSO
F1	2.29E+00	5.01E-12	5.83E+00	5.92E+00	4.02E+00	6.93E+00	0.00E+00	4.01E+00	5.75E+00
F2	5.59E-01	7.01E-01	4.65E+00	5.92E+00	1.13E+00	4.16E+00	2.79E+00	7.01E-01	1.58E+00
F3	1.29E-19	2.19E-19	3.40E-19	1.28E-18	2.20E-19	4.77E-13	2.05E-19	1.71E-21	1.47E-19
F4	3.12E+00	2.86E+00	1.49E-01	4.58E-01	2.50E+00	3.17E+00	2.74E+00	1.39E+00	2.65E+00
F5	5.00E-01	8.91E-01	3.21E+00	2.86E+00	1.24E+00	1.78E+00	1.09E+00	1.49E+00	1.86E+00
F6	2.42E-01	5.09E-01	2.90E+00	3.15E+00	1.07E+00	3.25E+00	6.92E-01	1.75E+00	3.13E+00
F7	8.81E-02	1.49E-01	2.02E-01	9.09E-02	1.20E-01	1.85E-01	1.09E-01	2.87E-01	1.52E-01
F8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F9	3.56E+02	6.86E+02	2.92E+05	1.01E+03	6.71E+03	8.62E+02	4.57E+02	6.74E+03	8.69E+02
F10	9.25E-02	9.89E-02	3.93E+00	2.42E+00	1.25E-01	2.08E+00	1.06E-01	1.21E-01	1.69E+00
F11.1	5.80E+02	3.94E+05	5.96E+05	5.97E+02	4.15E+03	5.54E+02	6.17E+02	1.09E+03	6.83E+02
F11.2	6.54E+04	2.85E+05	1.38E+06	2.61E+04	6.17E+05	9.20E+04	6.65E+04	2.29E+04	1.08E+05
F11.3	1.09E+01	1.70E+04	1.69E+01	9.33E+00	9.66E+00	2.00E+01	6.16E-01	1.85E+00	2.21E+01
F11.4	4.05E+01	2.09E+02	1.52E+02	3.52E+01	4.14E+01	1.61E+02	3.60E+01	1.11E+02	1.91E+02
F11.5	8.29E+00	7.97E+04	7.73E+01	5.62E+01	7.94E+00	8.36E+01	2.10E+01	2.40E+01	6.14E+01
F11.6	5.01E+02	1.10E+03	5.04E+03	5.51E+03	6.12E+02	2.54E+03	1.26E+03	2.37E+03	4.01E+03
F11.7	1.16E+04	1.32E+07	2.12E+05	3.02E+05	1.38E+04	4.47E+07	1.38E+04	2.62E+05	1.76E+05
F11.8	9.91E+02	1.92E+03	4.08E+05	3.63E+04	3.13E+03	1.32E+05	2.22E+03	4.52E+04	2.28E+04
F11.9	1.08E+03	1.70E+03	2.73E+05	2.34E+04	2.79E+04	3.28E+05	1.92E+04	1.42E+05	1.40E+05
F11.10	8.56E+02	1.95E+03	3.28E+05	1.06E+04	2.79E+03	1.36E+05	2.02E+03	1.14E+05	6.94E+03
F12	8.11E-01	6.73E+00	3.56E+00	2.11E+00	2.04E+00	2.61E+00	2.13E+00	3.61E+00	2.90E+00
F13	2.47E+00	1.31E+01	3.90E+00	2.21E+00	3.20E+00	3.24E+00	2.83E+00	2.57E+00	2.91E+00
	HCLPSO	**IDE**	**IILPSO**	**LBBO**	**MDE_pBX**	**PMS**	**PSO-iw**	**SADE**	
F1	2.98E+00	3.41E+00	5.47E+00	5.14E+00	5.47E+00	6.17E+00	4.39E+00	1.36E-01	
F2	1.31E+00	9.99E-01	2.49E+00	1.14E+00	2.07E+00	5.09E+00	4.14E+00	1.59E+00	
F3	1.51E-19	1.58E-19	1.62E-19	9.36E-15	2.07E-19	1.30E-18	1.47E-19	1.34E-19	
F4	2.80E-01	1.49E+00	2.39E-01	4.13E-02	3.16E+00	2.52E-01	2.65E-01	3.39E+00	
F5	9.83E-01	1.06E+00	1.54E+00	7.37E-01	2.23E+00	1.34E+00	2.48E+00	8.44E-01	
F6	1.14E+00	8.05E-01	2.98E+00	1.97E+00	2.95E+00	8.75E-01	3.23E+00	1.40E-01	
F7	1.55E-01	1.26E-01	1.95E-01	1.20E-01	3.36E-01	2.26E-01	1.36E-01	1.79E-01	
F8	0.00E+00	0.00E+00	1.83E+01	0.00E+00	0.00E+00	1.53E+03	5.32E+00	0.00E+00	
F9	8.24E+02	1.08E+03	7.52E+03	2.77E+04	6.62E+02	6.19E+02	1.08E+05	5.08E+02	
F10	2.28E-01	1.49E-01	2.82E+00	2.96E+00	1.59E-01	2.08E+00	4.12E+00	1.60E-01	
F11.1	6.72E+02	1.45E+03	3.84E+02	4.81E+02	5.79E+02	5.53E+02	1.54E+05	3.21E+03	
F11.2	5.70E+04	8.21E+04	7.73E+04	2.84E+05	5.97E+04	1.02E+05	1.02E+06	6.60E+04	
F11.3	1.85E+00	9.39E-01	1.37E+01	3.71E+00	6.51E+00	3.74E+01	1.29E+01	2.42E+00	
F11.4	1.36E+02	1.07E+02	1.45E+02	5.92E+01	3.78E+01	3.10E+02	1.74E+02	7.63E+01	
F11.5	3.68E+01	3.27E+01	9.31E+01	4.13E+01	8.96E+01	1.11E+02	9.21E+01	5.45E+01	
F11.6	2.22E+03	1.28E+03	4.91E+03	9.33E+03	2.88E+03	6.81E+03	5.29E+03	1.89E+03	
F11.7	1.23E+05	2.08E+04	2.46E+05	2.76E+07	7.87E+04	7.18E+05	8.93E+04	1.44E+04	
F11.8	3.15E+04	1.13E+04	4.03E+05	3.83E+06	2.84E+03	5.53E+06	1.75E+05	5.34E+04	
F11.9	2.23E+05	6.70E+04	4.84E+05	2.59E+06	1.24E+04	3.61E+06	2.27E+05	2.12E+05	
F11.10	2.15E+04	1.22E+04	4.55E+05	2.31E+06	2.08E+03	5.15E+06	7.85E+04	5.22E+04	
F12	1.67E+00	1.76E+00	2.81E+00	2.34E+00	1.21E+00	1.00E+01	2.83E+00	1.37E+00	
F13	2.70E+00	3.40E+00	2.63E+00	2.83E+00	3.27E+00	6.43E+00	3.07E+00	2.88E+00	

L-SHADE-50 is better than L-SHADE on 19 problems, worse only on 9. Hence, L-SHADE-50, a simplified variant of L-SHADE-EpSin may be considered at least no poorer than other L-SHADE-based variants.

The whole discussion given so far was based on 50-dimensional CEC2014 problems. However, CEC2014 problems represent artificial functions that have no relation to practical problems. To find if the conclusions will still be valid for real-world problems we test the same five L-SHADE-based variants on 22 CEC2011 real-world problems [6] of various dimensionalities and difficulties (for brief description of the problems, see Table 10). We perform the same analysis as for CEC2014 problems, and the results are given in Tables 11 and 12.

As there is little reason to think that, in the case of real-world problems, the good values of objective function will be found close to **O**, we should expect that the effect of structural bias would not favour L-SHADE-EpSin variants on such problems. Indeed, from Table 11 we find that both L-SHADE-SpSin-V1 and L-SHADE-EpSin-V2 variants enjoy fewer wins than each of the three other algorithms, although the differences are small. Similarly to what was observed for CEC2014 problems in one-by-one comparisons, L-SHADE-50 gains slightly more wins than losses against each of the 4 competitors. We are far from claiming that such a result confirms its better performance; as differences are small, most statistical tests would show no difference. What we rather learn from this test is that most modifications proposed within the L-SHADE-EpSin algorithm in [1] are practically unimportant, with exception of setting the values of F_i for all individuals to 0.5 during the first half of function calls. This allows us to highly simplify the L-SHADE-EpSin algorithm.

Table 18
Mean rankings of algorithms averaged over all CEC2014 and CEC2011 problems.

	CEC2014	CEC2011
L-SHADE-50	3.18	5.20
UMOEA-II-M	3.02	10.77
ALC-PSO	14.83	12.66
AMALGAM	4.15	8.61
AM-DEGL	6.42	7.07
cNrGA	12.17	12.20
CoBiDE	6.18	4.64
GA-MPC	6.97	7.07
GLPSO	10.82	9.16
HCLPSO	8.43	7.34
IDE	6.78	7.61
IILPSO	12.63	11.68
LBBO	10.82	9.80
MDE_pBX	9.12	7.20
PMS	13.03	12.50
PSO-iw	14.63	12.36
SADE	9.82	7.11

5. Comparison of simplified algorithms against various other metaheuristics

Finally, one may be interested in how L-SHADE-50 and UMOEA-II-M perform against other kinds of metaheuristics. To avoid repeating comparisons of similar methods, in this section we do not consider algorithms based on JADE [57] or SHADE [46]. The performance of L-SHADE-50 and UMOEA-II-M is compared on CEC2011 and 50-dimensional versions of CEC2014 problems against the performance of 15 other metaheuristics listed in Table 13, including some state-of-the-art and some relatively new algorithms. The list given in Table 13 contains five various DE variants not related to JADE or SHADE, five Particle Swarm Optimization (PSO) variants, two Genetic Algorithms (including GA-MPC, the winner of CEC2011 competition) and three other types of heuristics (multialgorithm AMALGAM, Biogeography-based algorithm LBBO and non-population based PMS). Codes that were obtained from other researchers or their web pages are marked with [*] in Table 13. Based on the comparative study by Helwig et al. [21] for PSO, among these 15 algorithms those whose codes were not obtained from their authors are tested with the reflection (rebounding) method as a bounds handling technique.

Each algorithm is run 51 times on every problem. The maximum number of function calls are set to 150,000 for CEC2011 and to $10{,}000{\cdot}D$ (hence 500,000) for 50-dimensional CEC2014 problems, as suggested in source papers [6,27]. The 51-run mean performances (and respective standard deviations) are given in Tables 14–17. In Table 18 the averaged rankings of algorithms on CEC2011 and CEC2014 problems are given. Such averaged rankings are obtained as follows. First for every problem the best function values found during each run by the specific algorithm are averaged over 51 runs. Then algorithms are ranked from the one with the lowest 51-run averaged objective function value (which is considered the best and receives rank 1st), to the one with the highest 51-run averaged objective function value (considered the worst and ranked 17th). As sometimes results obtained by various algorithms may differ just marginally, which would have no practical meaning, it is assumed in this paper that when the difference between the 51-run averaged objective function values achieved by some algorithms is smaller than 10^{-10} for a particular problem, such algorithms are considered as performing equally well on this problem and receive equal rank (for example, rank 1.5 if such a low difference is noted between the two best ones). Finally, ranks achieved by each algorithm are averaged over all 22 (in case of CEC2011 set) or 30 (in case of CEC2014 set) problems, giving just two final numbers per algorithm shown in Table 18: the averaged rankings on CEC2011 and CEC2014 test sets.

Finally, the statistical significance of pair-wise comparisons among 17 algorithms is verified by means of the Friedman's test with post-hoc Shaffer's static procedure [40] at $\alpha = 0.05$ (see Tables 19–20), as suggested in [15,49] for experiments where a fair comparison of a number of already existing methods is needed. The respective codes of statistical tests were obtained from www.cmpe.boun.edu.tr/~ulas/m2test/details.php [49].

As we can find from Table 18, UMOEA-II-M and L-SHADE-50 are the best two methods according to tests based on 50-dimensional CEC2014 problems, with almost the same performance. The difference in their averaged rankings is marginal and statistically insignificant (see Table 19). However, the differences between averaged rankings of either UMOEA-II-M or L-SHADE-50 and each out of the worst 10 algorithms (among 15 competitors) are statistically significant at $\alpha = 0.05$.

In case of real-world CEC2011 problems the results obtained by these two algorithms seriously differ. UMOEA-II-M is ranked only 12th best out of 17 algorithms, L-SHADE-50 retains the second best position – this time slightly loosing to CoBiDE. The difference between the performance of L-SHADE-50 and UMOEA-II-M is statistically significant (see Table 20). We may note that the difference in averaged rankings obtained by UMOEA-II-M on both sets is by far the largest compared to the differences observed for each of the 16 other algorithms under consideration. Hence, according to the empirical comparison against a large number of versatile metaheuristics on both artificial benchmarks and real-world problems

Table 19
The statistical significance of pair-wise comparisons among all algorithms on 50-dimensional CEC2014 problems by means of Friedman's test with post-hoc Shaffer's procedure at 5% significance level. 1 – difference between two algorithms is statistically significant; empty cell – difference between two algorithms is not statistically significant.

CEC2014	L-SHADE-50	UMOEA-II-M	ALC-PSO	AMALGAM	AM-DEGL	cNrGA	CoBiDE	GA-MPC	GLPSO	HCLPSO	IDE	IILPSO	LBBO	MDE_pBX	PMS	PSO-iw	SADE
L-SHADE-50			1			1			1	1		1	1	1	1	1	1
UMOEA-II-M			1			1			1	1		1	1	1	1	1	1
ALC-PSO	1	1		1	1		1	1		1	1			1			1
AMALGAM			1			1			1			1	1	1	1	1	1
AM-DEGL			1			1						1			1	1	
cNrGA	1	1		1	1		1	1			1						
CoBiDE			1			1			1			1	1		1	1	
GA-MPC			1			1						1			1	1	
GLPSO	1	1		1			1										
HCLPSO	1	1	1												1	1	
IDE			1			1						1			1	1	
IILPSO	1	1		1	1		1	1			1						
LBBO	1	1		1			1										
MDE_pBX	1	1	1	1												1	
PMS	1	1		1	1		1	1		1	1						
PSO-iw	1	1		1	1		1	1		1	1			1			1
SADE	1	1	1	1												1	

Table 20
The statistical significance of pair-wise comparisons among all algorithms on real-world CEC2011 problems by means of Friedman test with post-hoc Shaffer's procedure at 5% significance level. 1 – difference between two algorithms is statistically significant; empty cell – difference between two algorithms is not statistically significant.

CEC2011	L-SHADE-50	UMOEA-II-M	ALC-PSO	AMALGAM	AM-DEGL	cNrGA	CoBiDE	GA-MPC	GLPSO	HCLPSO	IDE	IILPSO	LBBO	MDE_pBX	PMS	PSO-iw	SADE
L-SHADE-50		1	1			1						1			1	1	
UMOEA-II-M	1						1										
ALC-PSO	1				1		1	1						1			1
AMALGAM																	
AM-DEGL			1												1		
cNrGA	1						1										
CoBiDE		1	1			1						1			1	1	
GA-MPC			1												1		
GLPSO																	
HCLPSO																	
IDE																	
IILPSO	1						1										
LBBO																	
MDE_pBX			1														
PMS	1				1		1	1									1
PSO-iw	1						1										
SADE			1												1		

L-SHADE-50 may be considered a highly competitive algorithm. However, this cannot be confirmed for UMOEA-II-M, due to its poor performance on real-world problems.

6. Conclusions

A large number of metaheuristics are introduced each year, and some of them seem to be successful for various kinds of problems. However, in this study it is shown that some metaheuristics have to be simplified because they contain operators that structurally bias their search by favouring sampling from some parts of the decision space. Other metaheuristics should be simplified as they contain unneeded or even harmful operators. This may be the effect of "adding stuff" [22] while developing metaheuristics by making step-by-step modifications to older and simpler variants. In this paper the discussion is based on an example of two joint winners of the CEC2016 competition on Evolutionary Computation, L-SHADE-EpSin [1] and UMOEA-II [12]. Yet, we are deeply convinced that a number of other metaheuristics also contain operators that need to be simplified.

It is shown that both L-SHADE-EpSin and UMOEA-II algorithms contain structurally biased operators that are dragging them towards the origin **O** in the decision space. As in the case of seven out of 30 CEC2014 problems that were used during the CEC2016 competition, points located in the origin **O** have better objective function values than solutions often found by other algorithms (although the global optimum is located somewhere else) - such bias seriously affects the performance of both L-SHADE-EpSin and UMOEA-II algorithms. The behaviour of structurally biased operators is discussed and, we hope, clarified in this study. We argue that the operators that structurally bias the algorithms by excessively sampling points located close to the origin **O** are unfair and have to be removed. It is also shown that the variants of L-SHADE-EpSin and UMOEA-II with biased operators removed may perform even better on problems for which good solutions are not located close to the origin **O**.

The L-SHADE-EpSin algorithm with the structurally biased operator removed may be further significantly simplified. We discuss the fact that L-SHADE-EpSin has been created by step-by-step modifications of simpler DE variants and that the path to its development is quite complicated. L-SHADE-EpSin combines the features of L-SHADE [47] with that of SinDE [8] in a specific way: the adaptation of scaling factor F_i, which is one of the DE control parameters, is performed by means of complicated sinusoidal-based functions during the first half of search (which is inspired by SinDE), but other search procedures, including the adaptation of other control parameters and adaptation of F_i during the second half of the search are based on L-SHADE. We show that the entire sinusoidal-based function adaptation part of L-SHADE-EpSin is not needed, and the feature which leads to some improvement of L-SHADE-EpSin over L-SHADE is rather the different management of control parameter F_i during the first and second half of the search. Instead of using a complicated sinusoidal-based adaptation mechanism, F_i may be simply set to 0.5 during the first half of the search. The results achieved by such a simplified algorithm on both artificial benchmarks from CEC2014 and real-world problems from CEC2011 are slightly better than the results obtained by non-simplified L-SHADE-EpSin or by L-SHADE algorithms.

We also show that the proposed simplified variant of L-SHADE-EpSin is highly competitive on a wide collection of both artificial benchmarks and real-world problems against 15 various metaheuristics, namely: 5 variants of Differential Evolution that are much different from L-SHADE-EpSin, 5 variants of Particle Swarm Optimization, 2 variants of Genetic Algorithms (including the winner of CEC2011 competition), a multialgorithm, a variant of a Biogeography-based algorithm and a variant of a non-population based heuristic. This confirms that simplification of some metaheuristics may not only make them more transparent and easier to use, but also improve their performance.

Acknowledgement

This work was supported within statutory activities No 3841/E-41/S/2016 of the Ministry of Science and Higher Education of Poland.

We would like to thank Prof. Ponnuthurai N. Suganthan for providing the MATLAB codes of MDE_pBX and HCLPSO algorithms, Prof. Jasper A. Vrugt for providing the MATLAB code of AMALGAM and Dr Tang for providing the MATLAB code of IDE. We are also grateful to the authors of [1,10,12,28,43,49] for making the codes of L-SHADE-EpSin, UMOEA-II, GA-MPC, LBBO and cNrGA algorithms, as well as the code of Shaffer's post-hoc statistical procedure widely available on the web.

References

[1] 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, Proc of the IEEE Congress on Evolutionary Computation, 2016, doi:10.1109/CEC.2016.7744163.
[2] F. Caraffini, F. Neri, G. Iacca, A. Mol, Parallel memetic structures, Inf. Sci. 227 (2013) 60–82.
[3] F. Caraffini, B. Passow, F. Neri, G. Iacca, *Re*-sampled inheritance search: high performance despite the simplicity, Soft Comput. 17 (12) (2013) 2235–2256.
[4] F. Caraffini, F. Neri, L. Picinali, An analysis on separability for memetic computing automatic design, Inf. Sci. 265 (2014) 1–22.
[5] W.N. Chen, J. Zhang, Y. Lin, N. Chen, Z.H. Zhan, H.S.H. Chung, Y. Li, Y.H. Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2) (2013) 241–258.
[6] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria For CEC 2011 Competition On Testing Evolutionary Algorithms On Real World Optimization Problems, Jadavpur Univ., Nanyang Technol. Univ., Kolkata, India, December 2010.
[7] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution – an updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[8] A. Draa, S. Bouzoubia, I. Boukhalfa, A sinusoidal differential evolution algorithm for numerical optimization, Appl. Soft Comput. 27 (2015) 99–126.
[9] W. Du, S.Y.S. Leung, Y. Tang, A.V. Vasilakos, Differential Evolution with event-triggered impulsive control, IEEE Trans. Cybern. 47 (1) (2017) 244–257.
[10] S.M. Elsayed, R.A. Sarker, D.L. Essam, GA with a new multi-parent crossover for constrained optimization, IEEE Congress Evol. Comput. (2011) 857–864.
[11] S.M. Elsayed, R. Sarker, D.L. Essam, N.M. Hamza, Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization, in: Proc of the IEEE Congress on Evolutionary Computation, Beijing, China, 2014, pp. 1650–1657.
[12] S. Elsayed, N. Hamza, R. Sarker, Testing united multi-operator evolutionary algorithms-II on single objective optimization problems, Proc of the IEEE Congress on Evolutionary Computation, 2016, doi:10.1109/CEC.2016.7744164.
[13] I. Fister Jr, U. Mlakar, J. Brest, I. Fister, A new population-based nature-inspired algorithm every month: is the current era coming to the end? in: Proceedings of the 3rd Student Computer Science Research Conference, Ljubljana, Slovenia, University of Primorska Press, 2016, pp. 33–37.
[14] S. Fong, X. Wang, Q. Xu, R. Wong, J. Fiaidhi, S. Mohammed, Recent advances in metaheuristic algorithms: does the Makara dragon exist? J. Supercomput. 72 (2016) 3764–3786.
[15] S. Garcia, F. Herrera, An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[16] C. Garcia-Martinez, P.D. Gutierrez, D. Molina, M. Lozano, F. Herrera, Since CEC 2005 competition on Real-parameter optimisation: a decade of research, progress and comparative analysis's weakness, Soft Comput. (2017) (in press), doi:10.1007/s00500-016-2471-9.
[17] Y.J. Gong, J.J. Li, Y. Zhou, Y. Li, H.S.H. Chung, Y.H. Shi, J. Zhang, Genetic learning particle swarm optimization, IEEE Trans. Cybern. 46 (10) (2016) 2277–2290.
[18] S.M. Guo, C.C. Yang, Enhancing Differential Evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (1) (2015) 31–49.
[19] N. Hansen, S. Muller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (1) (2003) 1–18.
[20] N. Hansen, S. Finck, R. Ros, A. Auger, Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions, 2009 INRA Technical Report RR-6829.
[21] S. Helwig, J. Branke, S. Mostaghim, Experimental analysis of bound handling techniques in particle swarm optimization, IEEE Trans. Evol. Comput. 17 (2) (2013) 259–271.
[22] G. Iacca, F. Neri, E. Mininno, Y.S. Ong, M.H. Lim, Ockham's Razor in memetic computing: three stage optimal memetic exploration, Inf. Sci. 188 (2012) 17–43.
[23] G. Iacca, F. Caraffini, F. Neri, Multi-strategy coevolving aging particle optimization, Int. J. Neural Syst. 24 (2014) art. no. 1450008.
[24] S.M. Islam, S. Das, S. Ghosh, S. Roy, P.N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2) (2012) 482–500.
[25] A.V. Kononova, D.W. Corne, P. De Wilde, V. Shneer, F. Caraffini, Structural bias in population-based algorithms, Inf. Sci. 298 (2015) 468–490.

[26] J.J. Liang, A.K. Qin, P.N. Suganthan, S Baskar, Comprehensive Learning Particle Swarm Optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[27] 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, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013 Technical Report 201311.
[28] Y. Lou, S.Y. Yuen, Non-revisiting genetic algorithm with adaptive mutation using constant memory, Memetic Comput. 8 (2016) 189–210.
[29] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.
[30] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (1–2) (2010) 61–106.
[31] Y. Nestevrov, A. Nemirovskii, Y. Ye, Interior-point polynomial algorithms in convex programming, SIAM (1994).
[32] F. Peng, K. Tang, G. Chen, Z. Yao, Multi-start JADE with knowledge transfer for numerical optimization, in: Proc of the IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 1889–1895.
[33] A.P. Piotrowski, Adaptive memetic differential evolution with global and local neighborhood-based mutation operators, Inf. Sci. 241 (2013) 164–194.
[34] A.P. Piotrowski, Differential evolution algorithms applied to neural network training suffer from stagnation, Appl. Soft Comput. 21 (2014) 382–406.
[35] A.P. Piotrowski, J.J. Napiorkowski, Searching for structural bias in particle swarm optimization and differential evolution algorithms, Swarm Intell. 10 (2016) 307–353.
[36] A.P. Piotrowski, Review of differential evolution population size, Swarm Evol. Comput. 32 (2017) 1–24.
[37] I. Poikolainen, F. Neri, F. Caraffini, Cluster-based population initialization for differential evolution frameworks, Inf. Sci. 297 (2015) 216–235.
[38] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential Evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.
[39] Q. Qin, S. Cheng, Q. Zhang, L. Li, Y. Shi, PSO with interswarm interactive learning strategy, IEEE Trans. Cybern. 46 (10) (2016) 2238–2251.
[40] J.P. Shaffer, Modified sequentially rejective multiple test procedures, J. Am. Statist. Assoc. 81 (395) (1986) 826–831.
[41] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in:, in: Proceeding in IEEE Congress on Evolutionary Computation (CEC), 1998, pp. 69–73.
[42] D. Simon, Biogeography-Based Optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[43] D. Simon, M.G.H. Omran, M. Clerc, Linearized biogeography-based optimization with re-initialization and local search, Inf. Sci. 267 (2014) 140–157.
[44] K. Sorensen, Metaheuristics—the metaphor exposed, Int. Trans. Oper. Res. 22 (2015) 3–18.
[45] R. Storn, K.V. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[46] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: Proc. IEEE Congress on Evolutionary Computation, Cancun, Mexico, 2013, pp. 71–78.
[47] R. Tanabe, A. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proc. IEEE Congress on Evolutionary Computation, Bejing, China, 2014, pp. 1658–1665.
[48] L. Tang, Y. Dong, J. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (4) (2015) 560–574.
[49] A. Ulas, O.T. Yildiz, E. Alpaydin, Cost-conscious comparison of supervised learning algorithms over multiple data sets, Pattern Recognit. 45 (2012) 1772–1781.
[50] J.A. Vrugt, B.A. Robinson, J.M. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput. 13 (2) (2009) 243–259.
[51] H. Wang, H. Sun, C. Li, S. Rahnamayan, J.S. Pan, Diversity enhanced particle swarm optimization with neighborhood search, Inf. Sci. 223 (2013) 119–135.
[52] Y. Wang, H.X. Li, T.W. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.
[53] M. Weber, F. Neri, V. Tirronen, Distributed differential evolution with explorative–exploitative population families, Genet. Program. Evolvable Mach. 10 (4) (2009) 343–371.
[54] T. Weise, R. Chiong, K. Tang, Evolutionary optimization: Pitfalls and booby traps, J. Comput. Sci. Tech. 27 (5) (2012) 907–936.
[55] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[56] A. Zamuda, J. Brest, Self-adaptive control parameters' randomization frequency and propagations in differential evolution, Swarm Evol. Comput. 25 (2015) 72–99.
[57] J. Zhang, Z.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.