Applied Soft Computing 61 (2017) 468–485

Contents lists available at ScienceDirect

Applied Soft Computing

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

Deterministic metaheuristic based on fractal decomposition for large-scale optimization

A. Nakib[a,*], S. Ouchraa[a], N. Shvai[a], L. Souquet[b], E.-G. Talbi[c]

[a] Université Paris-Est, Laboratoire LISSI, 122 Rue Paul Armangot, 94400 Vitry sur Seine, France
[b] Data ScienceTech Institute, DSTI Labs, 950 Route des Colles, Les Templiers, 061410 Biot, France
[c] INRIA Lille – Nord Europe Parc Scientifique de la Haute Borne, 40, Avenue Halley, Bat A, Villeneuve d'Ascq, France

ARTICLE INFO

Article history:
Received 20 February 2017
Received in revised form 16 July 2017
Accepted 18 July 2017
Available online 9 August 2017

Keywords:
Large-scale optimization
Metaheuristics
Geometric fractal decomposition
Local search continuous optimization

ABSTRACT

In this work a new method based on geometric fractal decomposition to solve large-scale continuous optimization problems is proposed. It consists of dividing the feasible search space into sub-regions with the same geometrical pattern. At each iteration, the most promising ones are selected and further decomposed. This approach tends to provide a dense set of samples and has interesting theoretical convergence properties. Under some assumptions, this approach covers all the search space only in case of small dimensionality problems. The aim of this work is to propose a new algorithm based on this approach with low complexity and which performs well in case of large-scale problems. To do so, a low complex method that profits from fractals properties is proposed. Then, a deterministic optimization procedure is proposed using a single solution-based metaheuristic which is exposed to illustrate the performance of this strategy. Obtained results on common test functions were compared to those of algorithms from the literature and proved the efficiency of the proposed algorithm.

1. Introduction

In the last decade the complexity of the problems has increased with the increase of the CPUs' power and the decrease of memory costs. Indeed, the appearance of clouds and other supercomputers provide the possibility to solve large-scale optimization problems.

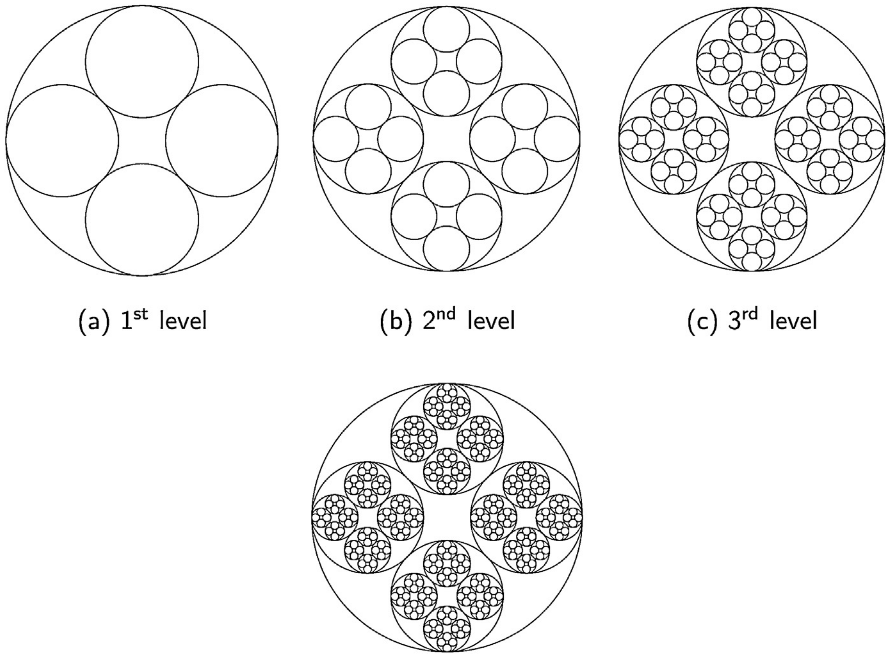
Black-box optimization has attracted many researchers from operational research communities; this class of optimization problem characterizes most real world problems, where there is a lack of information about the objective function. In this case, the most used methods are population-based metaheuristics [1–5]. In the literature many authors proposed several ideas to enhance the performance of metaheuristics, the reader can refer to [6] for more details.

In [7], authors pointed out the fact that the stochastic nature of metaheuristics algorithms is a limiting factor, in some applications, when it comes to safety critical applications where repeatability is important. Typically in these cases, metaheuristics can be used to improve the parameter settings of deterministic algorithms. Moreover, when the metaheuristics are efficient, some of them are difficult to implement. Moreover, the justification of an obtained solution can also be difficult because the method used to deduce is based on a complex stochastic search rather than on a deterministic approach.

In large-scale problems the complexity comes from the fact that local minima (and maxima) are rare compared to saddle points. Indeed, some points around a saddle point have greater fitness than the saddle point, while others have a lower fitness value. This phenomena can be explained by the fact that at a saddle point, the Hessian matrix has both positive and negative eigenvalues. Then, points lying along eigenvectors associated with positive eigenvalues have greater fitness than the saddle point, while points lying along negative eigenvalues have a lower value. Consequently, a saddle point can be considered as a local minimum along one cross-section of the fitness function and a local maximum along another cross-section. For instance, let $f: R^n \to R$ be a function of this type, the expected ratio of the number of saddle points to local optima grows exponentially with n. To understand the perception following this behavior, one can see that the Hessian matrix at a local minimum has only positive eigenvalues, but the Hessian matrix at a saddle point has both positive and negative eigenvalues. Assume that the sign of each eigenvalue is generated by flipping a coin, then, in one dimension, it is easy to obtain a local minimum by flipping a coin and getting heads once. In n-dimension case, it is exponentially unlikely that all n coins tossed will have heads outcome. For all these reasons, the use of sub-gradient or stochastic gradient-based algorithms are more recommended for large-scale problems.

[*] Corresponding author.

http://dx.doi.org/10.1016/j.asoc.2017.07.042
1568-4946/© 2017 Published by Elsevier B.V.


(a) 1st level

(b) 2nd level

(c) 3rd level

(d) 4th level

Fig. 1. Illustration of the fractal decomposition of the search space: the depth of the decomposition equal to 4.

Moreover, the hybridization with metaheuristics can enhance their performance.

In this work, the performance analysis of the proposed algorithm, called the Fractal Decomposition based Algorithm (FDA), is performed on large-scale global optimization test functions with dimensions going from 50 to 1000. The obtained results, and the comparison with other metaheuristics designed to solve the same types of problems, show the efficiency and the competitiveness of the proposed algorithm to solve large scale optimization problems. It should be noted that the proposed algorithm is a single solution-based metaheuristic while other competing algorithms are population-based metaheuristic.

The rest of this paper is organized as follows. Section 2 presents the related works. In Section 3, the principle of the geometric fractal decomposition is exposed, while, the coverage of the search space is discussed in Section 4. The proposed algorithm is detailed in Section 5, obtained results are pointed out and discussed in Section 6. Conclusions and future work are in the last section.

2. Related work

The concept of fractals is a line of research not much exploited in optimization so far. In general, this concept is related to the problem of representation using the geometric fractal decomposition of the feasible space.

In [8], the authors designed a framework for bound-constrained black-box global optimization algorithms which partition the search domain over multiple scales. Among the different state-of-the-art algorithms analyzed, the DIviding RECTangles (DIRECT) algorithm has received great success to solve optimal design problems. The original algorithm [9] is based on partitioning the feasible search space into a growing number of hyperintervals. Then, at each iteration, the most promising ones are selected for further partitioning. DIRECT takes profit from information on the objective function behavior to perform optimization. However, when the problem dimension increases (dimension >10) its performance decreases drastically, in terms of computation time and quality of the final solution.

In 1999, Demirhan introduced a metaheuristic for global optimization based on geometric partitioning of the search space called FRACTOP [10]. The geometrical form used in the proposed method is the hypercube. Indeed, the closure of the feasible region is divided into 2^n subregions with n corresponding to the problem dimension. Then, a number of solutions are collected randomly from each subregion or using a metaheuristic such as simulated annealing [11] or genetic algorithm [4]. After that, a guidance system is set through fuzzy measures to lead the search to the promising subregions simultaneously and discard the other regions from partitioning. In fact, the belief property that a subregion contains the global optimum is estimated using the belief measure of the previous level and the evidence of the sample of solutions collected from each subregion of the current level.

The main advantage of the decomposition procedure used in FRACTOP is that there is no overlap avoiding to visit the same local area more than once. Thus, that makes the proposed approach efficient for low dimensions as presented in the original paper. However, the decomposition procedure generates 2^n subregions. Hence, when n is higher, the complexity of the partitioning method increases exponentially. Besides, the global optimum may be missed by discarding some subregions. Then, this algorithm must visit 2^{50} subregions if it is run on problems of dimension 50.

Another trial using a representation based on the fractal geometry for evolutionary algorithms called Multiple Optima Sierpinski Searcher was also proposed in [12]. As its name suggests, the fractal geometrical form chosen for this method is the Sierpinski triangle generated using the chaos game which consists of moving a point repeatedly from a vertex to another selected randomly. Besides, in order to reduce the computational cost, the located optima are stored and manipulated using strings of characters that specifies them. The author proposed to face this limit to select $n+1$ generators instead of 2^n generator samples. However, the search will not cover all the feasible regions and it will have regions that will

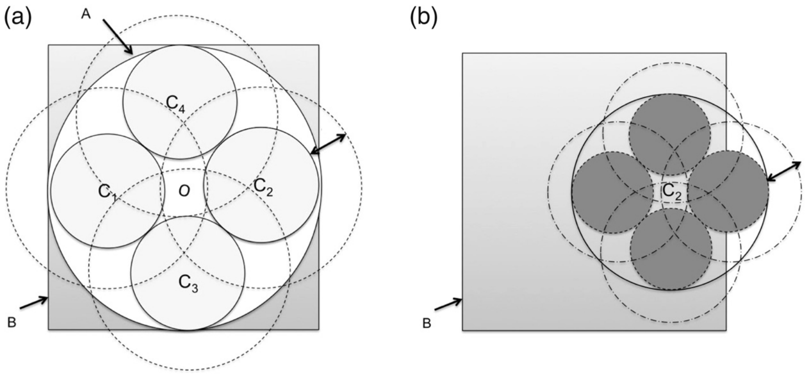


Fig. 2. Illustration of the decomposition procedure in the case of a 2D search space, where A is the biggest hypersphere inside the search space (B), C_1, C_2, C_3, and C_4 are centers of hyperspheres at the first level.

never be visited using fractals. This was only a primary work and was never extended because of the complexity of this approach.

Other metaheuristics (genetic algorithms, differential evolution and particle swarm optimization) based on geometric decomposition of the search space were also proposed in the literature. However, they do not perform well when dealing with large-scale problems.

3. Geometric fractal decomposition

In this work, a geometrical fractal decomposition based on hypersphere as an elementary geometric form is considered. This choice was motivated by its low complexity and flexibility to cover a part of the search space. Indeed, it is easy to go from one center of a hypersphere to another analytically without storing them.

An example of a geometric fractal decomposition is presented in Fig. 1, where an example of a fractal dimension of four is illustrated for four levels of decompositions.

Indeed, there are many schemes to divide an N-dimension search space. Several methods were tested as the Descartes theorem but the complexity at the generalization to the N-dimensional increases exponentially. As the goal is to find a scalable method that allows the whole search space to be decomposed, we propose to use overlapped hyperspheres. Then, a geometric decomposition without central hyperspheres was considered. Indeed, such a recursive division of the search space with a fixed number of hyperspheres at each level is called a fractal decomposition and, the number of hyperspheres inside a hypersphere can be seen as *the fractal dimension*.

The advantages of this decomposition are summarized in the following propositions:

Proposition 1. *When $2 \times D$ equal hyperspheres are inscribed within a bigger hypersphere in an D-dimensional space, then the ratio (δ) between their radii does not depend on D and is equal to $\delta = (1 + \sqrt{2})$.*

The proof of this proposition can be easily done using geometric properties of hyperspheres.

Proposition 2. *The hyperspheres fractal decomposition allows the centers and radius of the hyperspheres to be found analytically.*

Proof. To decompose the search space, the biggest hypersphere within the search domain is used with a center $\vec{C}^{(1)}$ and its radius r obtained using expressions (1) and (2), respectively.

$$\vec{C}_j^{(1)} = L + \frac{(U-L)}{2}, \quad \text{for } j = 1, 2, \ldots, N \tag{1}$$

where $\vec{C}^{(1)}$ is the coordinate of the center of the biggest hypersphere within the search space, and r is its radius.

$$r = \frac{(U-L)}{2} \tag{2}$$

where U is the upper bound, L is the lower bound of the whole search space, and D is the dimension of the space.

$$\vec{C}_j^{(i)} = \vec{C}_j^{(i)} + (-1)^i \times \left((r - r') \times \vec{e}_j\right) \tag{3}$$

where $\vec{e}_j$ is the unit vector at the dimension j, and i is the index of the hypersphere $i = 1, \ldots 2\,D$, and $r' = r/\delta$. □

4. Coverage of the search space via the fractal decomposition

As it can be seen in Fig. 1, the geometric fractal decomposition does not cover all the space. Thus, to overcome this problem, an increase of the radius (inflation) of each hypersphere of the following level must be performed. This increase produces overlaps between hyperspheres and, thus, allows all the space to be covered as shown in Fig. 2. So, the ratio between the inflated and the original radii of the hyperspheres, called relaxation coefficient, is named α.

In the following, the expression giving the value of α the lower bound of this coefficient is presented.

4.1. Relaxation at the first level

In this section the value of the relaxation coefficient (α) at the first level is calculated.

Denote the radius of parent hypersphere r, and the radius of child hyperspheres r'.

Let us consider the centers of small hyperspheres are located in points $O_i^{\pm} = (0, \ldots, 0, \pm d, 0, \ldots, 0)$, where the only non-zero entry $\pm d$ is in ith position, and $d = r - r' = r\left(1 - \frac{1}{1+\sqrt{2}}\right)$. The minimal required relaxation coefficient value α_n s.t. any arbitrary point from inside of parent-hypersphere is covered by one of the inflated (relaxed) child-hyperspheres with radius $r'' = \alpha_n r'$. To do this, we first estimate the inflated radius r_A for an arbitrary fixed point A, and then r'' will be defined as $r'' = \max_A r_A$. Let a point $A(x_1, \ldots, x_D)$ lie inside the parent-hypersphere, i.e. $x_1^2 + x_2^2 + \ldots + x_D^2 \le r^2$. The minimal required value of the radius r_A to cover A is defined by:

$$\begin{aligned} r_A^2 &= \min_{i,\pm} \| O_i^{\pm} A \|^2 \\ &= \min_{i,\pm} \left(\sum_{j \neq i} x_j^2 + (|x_i| \pm d)^2 \right). \end{aligned} \tag{4}$$

As $|x_i| \geq 0$, and $d > 0$, then $(|x_i| - d)^2 < (|x_i| + d)^2$. Hence, $r_A^2 = \min_i \left(\sum_{i \neq j} x_j^2 + (|x_i| - d)^2 \right)$ (we denote $a^2 = \|OA\|^2 = \sum_{i=1}^{n} x_i^2$), then:

$$
\begin{aligned}
r_A^2 &= \min_i \left(\sum_{i \neq j} x_j^2 + (|x_i| - d)^2 \right) \\
&= a^2 - d(2\max_i |x_i| - d).
\end{aligned} \tag{5}
$$

As A was considered as an arbitrary fixed point, to cover all points inside the hypersphere, we need to maximize the relaxation of the radius r_A over A. Thus:

$$
\begin{aligned}
r''^2 &= \max_{\|OA\|^2 \leq r^2} r_A^2 \\
&= \max_{0 \leq a \leq r} \max_{\|OA\| = a} (a^2 - d(2\max_i |x_i| - d))
\end{aligned} \tag{6}
$$

Now, we show that the point A cannot be a *maximum point* unless all of its coordinates are equal up to the sign. Without loss of generality, we assume that $x_i \geq 0$ for all $1 \leq i \leq D$ (indeed, the sign of x_i does not influence the value of the considered function under *max* operator). Moreover, assume that there exists an index s such that $x_s \neq \max_k x_k$. Let $i_1, i_2, \ldots, i_m$ be the indices of the highest coordinates of A, and j be the entry of the second highest coordinates:

$$
\begin{aligned}
&x_{i_1} = x_{i_2} = \ldots = x_{i_m} = \max_k x_k \\
&x_j = \max_{k \neq i_1, i_2, \ldots, i_m} x_k
\end{aligned} \tag{7}
$$

Under this assumption it is possible to *shift* the point A along the sphere $\|OA\| = a$ such as the maximized function is increasing, and thus the point A cannot be a point of maximum.

We Consider another point, A', with coordinates $(x'_1, \ldots, x'_D)$, where:

$$x'_k = x_k, \ \ k \neq i_1, i_2, \ldots, i_m, j, \tag{8}$$

$$x'_i = x_i - \delta^-, \ \ i = i_1, i_2, \ldots, i_m, \tag{9}$$

$$x'_j = x_j + \delta^+, \tag{10}$$

and $0 < \delta^-, \delta^+ < \frac{1}{2}(x_{i_1} - x_j)$ are such that $\|OA'\|^2 = \|OA\|^2 = a^2$. Then:

$$x'_i = x_i - \delta^- > x_j + \delta^+ = x'_j, \ \ i = i_1, i_2, \ldots, \tag{11}$$

$$x'_j = x_j + \delta^+ > x_j \geq x_k = x'_k, \ \ k \neq i_1, i_2, \ldots, i_m, j. \tag{12}$$

Hence, $x'_{i_1} = x'_{i_2} = \ldots x'_{i_m} = \max_k x'_k$. Denote $f(A) = (a^2 - d(2\max_k |x_k| - d))$, where as before $a^2 = \|OA\|^2$. Then:

$$
\begin{aligned}
f(A') &= (a^2 - d(2x'_{i_1} - d)) \\
&= (a^2 - d(2x_{i_1} - 2\delta^- - d)) \\
&= (a^2 - d(2x_{i_1} - d)) + 2\delta^- d \\
&> f(A).
\end{aligned} \tag{13}
$$

In other terms:

$$\operatorname{argmax}_{\|OA\| = a} (a^2 - d(2\max_i |x_i| - d)) \neq A. \tag{14}$$

One can also remark that the maximum exists. Indeed, function $f(A)$ is continuous, and can be considered as a function on a compact of lower dimension, defined by equality $\|OA\| = a$. Hence, by extreme value theorem, it reaches its maximum, and based on the inequality above, we infer that the point of maximum has to have equal coordinates (up to the sign). In particular, the maximum is reached at the point A^* having coordinates $\left(\frac{a}{\sqrt{D}}, \ldots, \frac{a}{\sqrt{D}} \right)$. Back to the expression for the relaxation radius r'', we get:

$$
\begin{aligned}
r''^2 &= \max_{0 \leq a \leq r} \max_{\|OA\| = a} (a^2 - d(2\max_i |x_i| - d)) \\
&= \max_{0 \leq a \leq r} \left(a^2 - \frac{2ad}{\sqrt{D}} + d^2 \right).
\end{aligned} \tag{15}
$$

The quadratic function of a under the max operator reaches its maximum on one of interval ends. Denote $g(a) = a^2 - \frac{2ad}{\sqrt{D}} + d^2$. Then:

$$g(0) = d^2 = \lambda^2 r^2, \tag{16}$$

$$g(r) = r^2 - \frac{2nd}{\sqrt{D}} + d^2 = r^2 - \frac{2\lambda r^2}{\sqrt{D}} + \lambda^2 r^2, \tag{17}$$

where $\lambda = 1 - \frac{1}{1+\sqrt{2}} = \frac{\sqrt{2}}{1+\sqrt{2}} \approx 0.59$. From this, it is obvious to see that for $D \geq 2 \frac{2\lambda}{\sqrt{D}} < 1$, and so $g(r) > g(0)$. Finally:

$$
\begin{aligned}
r''^2 &= \max_{0 \leq a \leq r} \left(a^2 - \frac{2ad}{\sqrt{D}} + d^2 \right) \\
&= g(r) \\
&= r^2 - \frac{2\lambda r^2}{\sqrt{D}} + \lambda^2 r^2 \\
&= r^2 \left(1 - \frac{2\lambda}{\sqrt{D}} + \lambda^2 \right),
\end{aligned} \tag{18}
$$

and, thus:

$$= \sqrt{5 + 2\sqrt{2} - \frac{2\sqrt{2}(1+\sqrt{2})}{\sqrt{D}}}. \tag{19}$$

If one would like to set the unified relaxation coefficient, it would have to be:

$$\alpha^{(1)} = \sup_{D \geq 1} \alpha_D = \sup_{D \geq 1} \sqrt{5 + 2\sqrt{2} - \frac{2\sqrt{2}(1+\sqrt{2})}{\sqrt{D}}} = \sqrt{5 + 2\sqrt{2}} \approx 2.80. \tag{20}$$

4.2. *Lower bound estimation of α*

In the previous paragraph, we pointed out the 1*st*-level. During the proof we established that the most distant point of initial hypersphere to the centers of 1*st*-level hyperspheres is the point $M\left(\frac{r}{\sqrt{n}}, \frac{r}{\sqrt{D}}, \ldots, \frac{r}{\sqrt{D}} \right)$, given here up to the sign of indices (because the symmetry does not influence the distance from the given point to the set of hyperspheres centers). We will now assume that this point is the most distant point for higher levels of decomposition as well. Even if it is not the case, we will obtain the lower bound for the relaxation coefficient, as point the M still needs to be covered.

Let us find, among a k-level decomposition hyperspheres centers', the closest one to the point M. Consider the process of recursive fractal division where at each level, we consider only one of the obtained hyperspheres as a process of approaching the point M. This procedure has a finite number of steps, which are allowed to be made only along one of the axes. The most effective strategy would be then to move along the axis by which the current position and the target point have the biggest difference. Indeed, without loss of generality, we assume that our goal is to approach the point $Q(x_1, x_2, \ldots, x_n)$ from point $O(0, 0, \ldots, 0)$ with a step of length s. Assume that after this step, we moved to the point $O_i^{\pm}$, where $OO_i^{\pm} = \pm s e_i$. Then:

$$
\begin{aligned}
d^2(O_i^{\pm}, Q) - d^2(O, Q) &= (x_i - (\pm s))^2 - x_i^2 \\
&= s^2 - 2(\pm s) x_i.
\end{aligned} \tag{21}
$$

In the previous expression (21) the first element does not depend on i, while, the second one is the smallest and yields the smallest value (i.e. biggest absolute value) when $\left|x_i\right|$ is the largest possible and sign $\pm s$ = sign $\pm x_i$ (i.e. the step was made in the direction of x_i, not the opposite one).

Denote the radius of the parent-hypersphere $r = r_0$ and the radius of child-hyperspheres of l-level decomposition as r_l. The step on level l is then equal to $s_l = r_{l-1} - r_l$. Then,

$$r_l = \beta r_{l-1}, \tag{22}$$

where

$$\beta = \frac{1}{1+\sqrt{2}}. \tag{23}$$

Thus:

$$s_l = r_{l-1} - r_l = \beta^{l-1} r - \beta^l r. \tag{24}$$

and

Consider the case $n \geq l$ according the approaching strategy given above, one of the points close, (O_l) to the point M will have the following coordinates:

$$(s_1, s_2, \ldots, s_l, 0, \ldots, 0) = (r - \beta r, \beta r - \beta^2 r, \ldots, \beta^{l-1} r - \beta^l r, 0, \ldots, 0). \tag{25}$$

Then, the squared distance between the points O_l and M, i.e. the distance we need to cover with the inflated hypersphere, is equal to:

$$\begin{aligned} d^2(O_k, M) = & \left(r - \beta r - \frac{r}{\sqrt{n}}\right)^2 + \left(\beta r - \beta^2 r - \frac{r}{\sqrt{n}}\right)^2 + \ldots \\ & + \left(\beta^{l-1} r - \beta^l r - \frac{r}{\sqrt{n}}\right)^2 + (n-l)\frac{r^2}{n} \end{aligned} \tag{26}$$

Then, the relaxation coefficient $\alpha_n^{(l)}$ is thus equal to:

$$\begin{aligned} \alpha_n^{(l)} & = \frac{d(O_l, M)}{r_l} = \frac{d(O_l, M)}{\beta^l r} \\ & = \sqrt{\sqrt{2}\left((1+\sqrt{2})^{2l} - 1\right) + 1 - \frac{2(1+\sqrt{2})^l\left((1+\sqrt{2})^l - 1\right)}{\sqrt{n}}} \end{aligned} \tag{27}$$

Now consider the case $n < l$. Then, one of the closest points to the point M is the point O_l with coordinates:

$$(s_1, s_2, \ldots, s_n + \ldots s_l) = (r - \beta r, \beta r - \beta^2 r, \ldots, \beta^{n-1} r - \beta^l r). \tag{28}$$

Similarly to the previous case:

$$\begin{aligned} d^2(O_l, M) = & \left(r - \beta r - \frac{r}{\sqrt{n}}\right)^2 \\ & + \left(\beta r - \beta^2 r - \frac{r}{\sqrt{n}}\right)^2 + \ldots + \left(\beta^{n-1} r - \beta^l r - \frac{r}{\sqrt{n}}\right)^2 \end{aligned} \tag{29}$$

Then, the relaxation coefficient $\alpha_n^{(l)}$ is equal to:

$$\begin{aligned} \alpha_n^{(l)} & = \frac{d(O_l, M)}{r_l} \\ & = \sqrt{\sqrt{2}((\delta^{2l} + c^{2l-2n+1}) + 1 - 2\delta^{l-n+1} - \frac{2}{\sqrt{n}}\delta^l(\delta^l - 1)} \end{aligned} \tag{30}$$

where $\delta = (1+\sqrt{2})$ defined in property 1.

Consequently, combining both cases, we obtain the following formula:

$$\alpha_n^{(l)} \geqslant \begin{cases} Eq.(27), & n \geq l, \\ Eq.(30), & \text{otherwise} \end{cases} \tag{31}$$

5. Proposed fractal decomposition based algorithm

In this section, the proposed algorithm called FDA, that profits from the fractal decomposition, is presented. To find the global optimal solution (if it is known), the obvious way is to explore exhaustively all inflated last level child-hyperspheres, however, it is too time consuming. To overcome this problem, two heuristics were proposed: the first one, called *promising hypersphere selection heuristic*: it allows selecting the most promising hypersphere for further decomposition. This heuristic is used during the exploration phase. The second heuristic is performed at the last level, called intensification local search heuristic (ILS): its aim is finding the best solution inside a reduced sub-region. This second heuristic is used for the intensification phase.

Moreover, as of the proposed fractal decomposition, there is no need to save all information about visited hyperspheres by FDA: all decomposition can be reconstructed analytically. Then, only the best positions met are saved. Indeed, the expression (3) is used to compute centers' positions without any past position.

An overview of the proposed algorithm is presented in Algorithm 1, while, the full description is presented in Algorithm 2 and 3. As pointed out before, it uses the hypersphere H as a geometrical form to represent the search space which is repetitively divided into 2 × *dimension* child-hyperspheres CHs_i where $i = 1, \ldots, 2D$. This decomposition choice is explained in Proposition 1. Then, the quality q_i^l of each child-hypersphere CHs_i is evaluated using the procedure described in Section 5.1. Afterwards, the child-hyperspheres are sorted and, that with the best quality is chosen to be the next hypersphere to be visited (decomposed). This procedure allows the algorithm to guide the search to the most promising region and lead the optimization to start at the best position.

The sub-region of the search space limited by a child-hypersphere CHs_i is defined by:

$$CHs_i = \{x \in \mathfrak{R}^D : x - r^{(i)} \leq x \leq x + r^{(i)}\} \tag{32}$$

with $i \in L^m$, where $r^{(i)}$ is the radius of the child-hypersphere i, and L^m is the set of indices that constitute the hypersphere at level m, respectively. At each iteration, the most promising child-hypersphere is selected for further decomposition.

Once the last level, called the fractal depth (k), is reached, the intensive local search procedure (ILS) is applied to the sorted CHs_i.

When all of hyperspheres of the last level are visited, the search is raised up, using a moving-up procedure (Algorithm 4), to another region via the previous depth (level), by replacing the current hypersphere (H) with the following child-hypersphere CHs_i from the sorted list. The process is repeated until one of the stopping criterion is reached or the child-hyperspheres from all the levels were visited.

Algorithm 1. Overview of FDA

```
set $k = 5$,   ▷ $k$ is the maximum deep of the fractal decomposition
$\omega_{\min} = 1 \times e^{-20}$,   ▷ $\omega_{\min}$ is the tolerance threshold
$NBEval = 1$,   ▷ the counter of the evaluations of the objective function
Compute $\vec{C}$   ▷ the barycenter of the search space
$BestSol = f(\vec{C})$,   ▷ $BestSol$ is the best objective function value found so
far $bestPosition = \vec{C}$.   ▷ $bestPosition$ is the corresponding position
Compute the radius $r$ using Eq. (2)
Set the level variable: $l = 1$
while Stopping criteria are not reached do
  Decompose the current hypersphere $H$ using the Fractal procedure
  using expression (3)
  for $2 \times D$ $l$-level hypersphere do
  Apply the promising hypersphere selection procedure described in
  Section 5.1
  end for
  Sort the $2 \times D$ hyperspheres at the current $l$-level
  Replace the current hypersphere $H$ by the first of the sorted
  hyperspheres at the current level
  if $l == k$ then
```

```
for Each $2 \times D$ of $k$th-level hypersphers do
  Apply the ILS heuristics described in Section 5.3
end for
if stopping criterion is not reached then
  Apply the move-up Procedure (Algorithm 4)
end if
  else
Go to next level: $l = l + 1$
  end if
end while
return BestSol and bestPosition
```

Algorithm 2. Full description of FDA (Part 1)

```
set $k = 5$    ▷ $k$ is the maximum Deep of the fractal decompostion,
$\omega_{\min} = 1 \times e^{-20}$    ▷ $\omega_{\min}$ is the tolerance threshold,
$\varphi = 0.5$    ▷ $\varphi$ is the step-size reduction factor,
$\alpha = 1.75$    ▷ $\alpha$ is the inflation coefficient
NBEval = 1    ▷ the counter of the evaluations of the objective function,
Compute $\vec{C}$    ▷ the barycenter of the search space.
$BestSol = f(\vec{C})$    ▷ BestSol is the best objective function value found so
far, $bestPosition = \vec{C}$    ▷ bestPosition the corresponding position
Compute the radius $r$ using Eq. (2)
Set the level variable: $l = 1$
while Stopping criteria are not reached do
  Decompose the current hypersphere $H$ via the expression (3)
  for $2 \times D$ $l$-level hypersphere do
Compute $g_1, g_2$ and $g_c$ using the expression (34) and evaluate the
quality of the HyperSphere $q$, using the expression (33)
NBEval = NBEval + 3
  end for
  Sort the $2 \times D$ hyperspheres at the current $l$-level
  Replace the current hypersphere $H$ by the first of the sorted
hyperspheres at the current level
  if $l == k$ then
for Each $2 \times D$ of $k$th-level hypersphers do
  Set $\vec{x}_C = \vec{C}$ the center of the current hypersphere $H$
  Evaluate the objective function of the solution $\vec{x}_C$
  NBEval = NBEval + 1
  Set the step size $\omega$, to the radius of the current hypersphere $H$
  while $\omega \geq \omega_{\min}$ do
for Each dimension $i = 1, \ldots, D$ do
  $\vec{x}_L = \vec{x}_C - \omega \times \vec{e}_i$ and $\vec{x}_R = \vec{x}_C + \omega \times \vec{e}_i$
  Evaluate the fitness of $\vec{x}_L$ and $\vec{x}_R$
  NBEval = NBEval + 2
```

Algorithm 3. Full description of FDA (Part 2)

```
  $\vec{x}_C$= Argmin $\{f(\vec{x}_C), f(\vec{x}_L), f(\vec{x}_R)\}$
end for
if No improvment of the fitness $\vec{x}_C$ then
  Decrease the step size $\omega$: $\omega = \omega \times \varphi$
end if
if $f(\vec{x}_C) < BestSol$ then
  $BestSol = f(\vec{x}_C)$
  $bestPosition = \vec{x}_C$
end if
  end while
end for
if stopping criterion is not reached then
  Set $l$    ▷ the Current level,
  Set $N$ to the number of explored hypershperes at level $l-1$
  while $N == 2 \times D$ do
$l = l - 1$
Update $N$ to the number of explored Hyperspheres at level $l$.
  end while
  if $l == 1$ then
All hyperspheres have been explored
Stopping criterion is satisfied
  else
Update the position of the current hypersphere by the next
unexplored hypersphere at the current level $l$
  end if
end if
  else
Go to next level: $l = l + 1$
  end if
end while
return the best solution BestSol and its coordinates bestPosition
```

As it can be noticed, the proposed approach can be compared to the depth-first branch and bound technique (B&B) often used in combinatorial optimization. The main difference is that in our case each branch represents a part of the search space rather than a singular part of the whole solution. Compared to the B&B, some areas are split and, areas that do not seem to be hopeless are further investigated while the most promising one is searched more intensively.

5.1. Promising hypersphere selection (exploration strategy)

The aim of this procedure is to select the most promising region that might contain the best solution or the global optimum. To do so, each hypersphere i created by the decomposition procedure is evaluated.

It is important to mention that each time a solution is evaluated during the exploration phase, a track of the best solution (*BestSol*) and its coordinates (*bestPosition*) is saved.

The first step, to evaluate the hypersphere quality, is to generate two points $\vec{s}_1$ and $\vec{s}_2$ following expression (35) and (36). Then, for positions $\vec{s}_1, \vec{s}_2$ and the center of the current hypersphere $\vec{C}^l$, their fitnesses, f_1, f_2 and f_c, respectively, are calculated as well as their, corresponding distances to the best position found so far (*BSF*) via the Euclidean distance. The last step consists of computing the slope at the three positions ($\vec{s}_1, \vec{s}_2$ and $\vec{C}^l$), referred as g_1, g_2 and g_c. This is performed by taking the ratio between the fitness (f_1, f_2 and f_c) and their corresponding distances. Then, the quality for the current hypersphere will be represented by the highest ratio among g_1, g_2 and g_c, denoted by q:

$$q = \max\{g_1, g_2, g_c\} \tag{33}$$

with:

$$g_1 = \frac{f(\vec{s_1})}{\left\|\vec{s_1} - BSF\,\right\|},\quad g_2 = \frac{f(\vec{s_2})}{\left\|\vec{s_2} - BSF\,\right\|} \text{ and } g_c = \frac{f(\vec{C}^l)}{\left\|\vec{C}^l - BSF\right\|} \tag{34}$$

where:

$$\vec{s_1} = \vec{C}^l + \alpha \frac{r_l}{\sqrt{D}} \times \vec{e}_d, \text{ for } d = 1, 2, \ldots, D \tag{35}$$

$$\vec{s_2} = \vec{C}^l - \alpha \frac{r_l}{\sqrt{D}} \times \vec{e}_d, \text{ for } d = 1, 2, \ldots, D \tag{36}$$

5.2. Multilevel search strategy

At each level, hyperspheres that have not yet been decomposed are stored in a list, sorted by their quality score, evaluated during the exploration strategy detailed in Section 5.1.

In the case where all the spheres at a level have been explored without reaching the stopping criterion, the next hypersphere in the upper level's $(l-1)$ list, is then chosen to be decomposed. If all the hyperspheres in the upper level have been explored, then, a move to $l-2$ is performed and so on, until exploring the whole search space or the stopping criterion is satisfied as detailed in Algorithm 4.

Algorithm 4. The move-up procedure

```
Set $l$    ▷ Current level
Set $N$    ▷ number of explored hypershperes at level $l-1$
Set $D$    ▷ dimension of the problem
while $N == 2 \times D$ do
  $l = l - 1$
  Update $N$ to number of explored Hyperspheres at level $l$
end while
if $l == 1$ then
  All hyperspheres have been explored
  Stopping criterion is satisfied
else
  Update the position of the current hypersphere by the next
unexplored hypersphere at the current level $l$
end if
```

Table 1
Functions F_1–F_{11}.

Function	Name	Definition
F1	Shifted sphere function	$\sum_{i=1}^{D} z_i^2 + f\_bias, z = x - o$
F2	Shifted Schwefel Problem 2.21	$\max\{\lvert z_i\rvert, 1 \le i \le D\} + f\_b\ ias, z = x - o$
F3	Shifted Rosenbrock's Function	$\sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1^2)) + f\_bias, z = x - o$
F4	Shifted Rastrigin's Function	$\sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) + f\_bias, z = x - o$
F5	Shifted Griewank's Function	$\sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_bias, z = x - o$
F6	Shifted Ackley's Function	$-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)\right) + 20 + e + f\_bias, z = x - o$
F7	Schwefel's Problem 2.22	$\sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$
F8	Schwefel's Problem 1.2	$\sum_{i=1}^{D} (\sum_{j=1}^{D} x_j)^2$
F9	Extended f10	$\sum_{i=1}^{D-1} f_{10}(x_i, x_{i+1})) + f_{10}(x_D, x_1)$ $f_{10}(x, y) = (x^2 + y^2)^{0.25}(\sin^2(50(x^2 + y^2)^{0.1}) + 1)$
F10	Bohachevsky	$\sum_{i=1}^{D-1} (x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$
F11	Schaffer	$(x_i^2 + x_{i+1}^2)^{0.25}(\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1)$

Table 2
Properties of functions F_1–F_{11}.

Function	Range	Optimum	U/M	Shifted	Separable	Can be optimized dimension by dimension
F1	$[-100, 100]^D$	−450	U	√	√	√
F2	$[-100, 100]^D$	−450	U	√		
F3	$[-100, 100]^D$	390	M	√		√
F4	$[-5, 5]^D$	−330	M	√	√	√
F5	$[-600, 600]^D$	−180	M	√		
F6	$[-32, 32]^D$	−140	M	√	√	√
F7	$[-10, 10]^D$	0	U		√	√
F8	$[-65.536, 65.536]^D$	0	U			
F9	$[-100, 100]^D$	0	U			√
F10	$[-15, 15]^D$	0	U			
F11	$[-100, 100]^D$	0	U			

Table 3
Properties of functions F_{12}–F_{19}.

Function	F_{ns}	F'	m_{ns}	Range	$f(x^*)$
F12	$NS - F_9$	F_1	0.25	$[-100, 100]^D$	0
F13	$NS - F_9$	F_3	0.25	$[-100, 100]^D$	0
F14	$NS - F_9$	F_4	0.25	$[-5, 5]^D$	0
F15	$NS - F_{10}$	$NS - F_7$	0.25	$[-10, 10]^D$	0
F16	$NS - F_9$	F_1	0.75	$[-100, 100]^D$	0
F17	$NS - F_9$	F_3	0.75	$[-100, 100]^D$	0
F18	$NS - F_9$	F_4	0.75	$[-5, 5]^D$	0
F19	$NS - F_{10}$	$NS - F_7$	0.75	$[-10, 10]^D$	0

5.3. Intensive level search (ILS)

Different local searches, or even metaheuristics can be used at this level. As in this work, our goal is to design a deterministic, simple and efficient metaheuristic adapted to the large scale problems. A simple local search was considered.

In this local search, two candidate solutions are evaluated per dimension of the search space, denoted by $\vec{x}^{s_1}$ and $\vec{x}^{s_2}$. They stand in opposite directions from the current solution $\vec{x}^s$ along an axis of the search space at equal distance ω, also called step size:

$$\vec{x}^{s_1} = \vec{x}^s + \omega \times \vec{e}_i \tag{37}$$

$$\vec{x}^{s_2} = \vec{x}^s - \omega \times \vec{e}_i \tag{38}$$

where $\vec{e}_i$ is the unit vector which the ith element is set to 1 and the other elements to 0,

Then, the best solution among $\vec{x}^s, \vec{x}^{s_1}$ and $\vec{x}^{s_2}$ is selected to be the next current solution $\vec{x}^s$. The adaptation of the step size ω is performed through the procedure described in Algorithm 5. Depending on the situation, the step size is adapted using the following rules:

- if there is any better candidate solution was found in the neighborhood of $\vec{x}^s$, then, ω is halved (using the reduction factor $\varphi = 0.5$),
- the step size is decreased until a given value $\omega_{\min}$as the tolerance or the precision need.

Algorithm 5. ILS Procedure

```
$\omega_{\min} = 1 \times e^{-20}$,    ▷ $\omega_{\min}$ is the tolerance threshold,
$\varphi = 0.5$,    ▷ $\varphi$ is the step-size reduction factor,
$NBEval = 1$,    ▷ the counter of the evaluations of the objective function,
Compute $\vec{C}$    ▷ the barycenter of the search space,
$BestSol = f(\vec{C})$,    ▷ $BestSol$ is the best objective function (fintess) value
found so far,
$bestPosition = \vec{C}$.    ▷ $bestPosition$ the corresponding position,
Set $D$    ▷ $D$ is the dimension of the problem.
Set the solution $\vec{x}_C$ to the center $\vec{C}$ of the current hypersphere $H$
Evaluate the objective function of the solution $\vec{x}_C$
Set the step size $\omega$ to the radius of the current hypersphere $H$
while $\omega \ge \omega_{\min}$ do
  for Each dimension $i = 1, \ldots, D$ do
$\vec{x}_L = \vec{x}_C - \omega \times \vec{e}_i$
$\vec{x}_R = \vec{x}_C + \omega \times \vec{e}_i$
Evaluate the fitness of $\vec{x}_L$ and $\vec{x}_R$
NBEval = NBEval + 2
Update $\vec{x}_C$ by the best solution among $\{\vec{x}_C, \vec{x}_L, \vec{x}_R\}$.
  end for
  if No improvment of the fitness $\vec{x}_C$ then
Decrease the step size $\omega$: $\omega = \omega \times \varphi$.
  end if
end while
if $f(\vec{x}_C) < BestSol$ then
  Update the best solution $BestSol = f(\vec{x}_C)$ and $bestPosition = \vec{x}_C$
end if
return $\vec{x}_C$
```

Table 4
Sensitivity analysis with respect to the fractal depth (k). The average error is computed for $D = 200$, $D = 500$ and $D = 1000$.

	$D = 200$				$D = 500$				$D = 1000$			
k	3	4	5	6	3	4	5	6	3	4	5	6
$F2$	**5.39E−11**	6.44E−11	1.23E−10	1.31E−10	**1.43E−04**	1.85E−04	4.30E−04	1.23E−03	**1.28E−01**	1.56E−01	3.11E−01	9.28E−01
$F3$	1.75E+03	7.58E+02	**2.51E+02**	8.34E+03	8.55E+02	5.33E+02	5.82E+02	5.30E+02	1.14E+03	**9.37E+02**	1.13E+03	1.15E+03
$F4$	1.30E+02	7.49E+02	**0.00E+00**	1.50E+03	3.50E+02	2.02E+03	**0.00E+00**	3.55E+03	7.12E+02	4.28E+03	0.00E+00	7.02E+03
$F5$	**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**	**0.00E+00**	**0.00E+00**	**0.00E+00**
$F6$	3.06E−13	3.27E−13	**2.52E−13**	1.90E+01	9.20E−13	**7.85E−13**	8.74E−13	1.90E+01	1.78E−12	**1.71E−12**	1.91E−12	1.92E+01
$F13$	1.63E+02	1.18E+02	**7.07E+01**	2.19E+02	4.07E+02	8.30E+02	**3.74E+02**	4.11E+02	2.07E+04	**6.99E+02**	7.69E+02	8.53E+02
$F14$	9.85E+01	5.49E+02	**0.00E+00**	1.20E+03	2.65E+02	1.43E+03	**0.00E+00**	2.65E+03	5.32E+02	3.13E+03	**0.00E+00**	5.16E+03
$F17$	**6.06E+01**	4.26E+03	9.31E+01	8.13E+01	**1.46E+02**	5.46E+03	3.96E+02	1.55E+02	2.33E+03	2.74E+02	**1.95E+02**	2.81E+02
$F18$	2.79E+01	1.97E+02	**0.00E+00**	4.71E+02	8.16E+01	4.70E+02	**0.00E+00**	1.00E+03	1.67E+02	9.35E+02	**0.00E+00**	1.81E+03

Values in bold represent the best result for a given function and dimension across all different values of k.

Table 5
Complexity of methods of the FDA.

Step	Asymptotic complexity
Fractal decomposition	$\log_k(D)$
Quality evaluation of a hypersphere	1
ILS	$\log_2(r/\alpha_{min})$

6. Results and discussions

In this section, the proposed algorithm (FDA) is analyzed and its performance is exposed in the following problems.

6.1. Benchmark functions

The experimental tests were performed on 19 functions (F_1–F_{19}) for large-scale continuous optimization taken from the special issue of soft computing on scalability of evolutionary algorithms (SOCO 2011). The first six functions F_1–F_6 are described in [13], whereas the function F_9 is detailed in [14]. While functions F_{12}–F_{19} are obtained by hybridizing a non-separable function F_{ns} with other functions from the benchmark. This hybridization consists of splitting via the parameter m_{ns} that defines the ratio of variables that are evaluated by F_{ns}. All these functions are exposed in Table 1 and their properties are detailed in Tables 2 and 3. Tests were done for the set of dimensions $D = 50$, 100, 200, 500 *and* 1000 and the stopping criterion was defined by the maximum number of function evaluations (FEs) set to 5000 × D. For the comparison, the stochastic based algorithms were run 25 times for each function of the benchmark.

6.2. Parameters settings

Parameters of FDA are summarized bellow and were fitted empirically:

- The fractal depth (k) is set to 5. This parameter corresponds to the number of decomposition levels to reach.
- The stopping criterion for the ILS is related to a tolerance threshold (ω_{min}) set to $1 \times e^{-20}$. This chosen value is problem dependent. For these experimentations, this value is equal to precision of the shift values in the shifted functions F_1-F_6.
- The step-size reduction factor φ, is set to the standard value 0.5.
- The inflation coefficient α, was set to 1.75.

6.3. Sensitivity analysis of FDA

In this subsection, the sensitivity analysis of FDA against its parameters is presented. The fractal depth k is the parameter that has an impact on the performance of FDA. The rest of the parameters can be set to values presented in Section 6.2.

In these experimentations, the parameter k was varied, while, other parameters were set at their suited values. Table 4 summarizes results on only some functions, because FDA reaches the global optimum for the rest. As it can be seen from obtained results, this parameter is important and the performance of FDA varies against its value. However, for the considered set of functions, the value 5 seems to be the most suited.

6.4. Complexity analysis

The proposed approach includes three distinct parts: the first is the fractal decomposition process; the second consists of the quality's evaluation of the hypersphere, while, the third is the application of ILS.

Their asymptotic complexities are presented in Table 5, respectively, where D represents the problem dimension, r the radius of the current hypersphere and ω_{min} the ILS tolerance threshold.

Using Table 5, the complexity of the FDA is given by (39). Hence, the asymptotic complexity shows that FDA has a logarithmic complexity depending on the fractal depth parameter: $\mathcal{O}_{FDA}(\log_k(D))$:

$$\mathcal{O}(\log_k(D) + 1 + \log_2(r/\omega_{\min})) = \mathcal{O}(\log_k(D)) \tag{39}$$

Besides, the FDA memory complexity is $\Theta(D)$.

6.5. FDA results

The reported results of the Fractal Decomposition based Algorithm are the error values $f(x) - f(x^*)$ obtained for dimensions $D = 50$, 100, 200, 500 and 1000. These results are presented in Tables 6–8 through statistical measures. In our case, the mean and the standard deviation are sufficient to describe the behavior of the FDA for each test function knowing that all the obtained standard deviations are equal to $0.0000E + 00$. Besides, all average errors below

Table 6
Experimental results obtained by FDA on functions $F_1 - F_7$.

Dimension	F_1	F_2	F_3	F_4	F_5	F_6	F_7
50D	0.0000E+00	2.71E−12	9.32E+01	0.00E+00	0.00E+00	6.75E−14	0.00E+00
100D	0.0000E+00	8.48E−12	5.09E+01	0.00E+00	0.00E+00	1.35E−13	0.00E+00
200D	0.0000E+00	1.23E−10	2.51E+02	0.00E+00	0.00E+00	2.52E−13	0.00E+00
500D	0.0000E+00	4.30E−04	5.82E+02	0.00E+00	0.00E+00	8.74E−13	0.00E+00
1000D	0.0000E+00	3.11E−01	1.13E+03	0.00E+00	0.00E+00	1.91E−12	0.00E+00

Table 7
Experimental results obtained by FDA on functions $F_8 - F_{14}$.

Dimension	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
50*D*	0.0000E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.50E+01	0.00E+00
100*D*	0.0000E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.68E+02	0.00E+00
200*D*	0.0000E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.07E+01	0.00E+00
500*D*	0.0000E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.74E+02	0.00E+00
1000*D*	0.0000E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.69E+02	0.00E+00

Table 8
Experimental results obtained by FDA on functions $F_{15} - F_{19}$.

Dimension	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}
50*D*	0.00E+00	0.00E+00	6.09E−04	0.00E+00	0.00E+00
100*D*	0.00E+00	0.00E+00	6.22E+00	0.00E+00	0.00E+00
200*D*	0.00E+00	0.00E+00	9.31E+01	0.00E+00	0.00E+00
500*D*	0.00E+00	0.00E+00	3.96E+02	0.00E+00	0.00E+00
1000*D*	0.00E+00	0.00E+00	1.95E+02	0.00E+00	0.00E+00

Table 9
Number of evaluations to find the best solution for F_3, F_4 and F_{16} for dimensions $D = 50$ and $D = 1000$.

	$D = 50$	$D = 1000$
F_3	250,000	5,000,000
F_4	7802	160,002
F_{16}	12,802	268,002

Table 10
Number of spheres visited for F_3, F_4 and F_{16} for dimensions $D = 50$ and $D = 1000$.

	$D = 50$	$D = 1000$
F_3	5	5
F_4	29	29
F_{16}	23	23

Table 11
Comparison of FDA and DIRECT algorithms for dimensions $D = 50$ and $D = 100$.

Dimensions	$D = 50$		$D = 100$	
	DIRECT	FDA	DIRECT	FDA
F1	5.53E+01	**0.00E+00**	1.06E+04	**0.00E+00**
F2	5.53E+01	**2.71E−12**	7.45E+01	**8.48E−12**
F3	1.79E+04	**9.32E+01**	9.84E+07	**5.09E+01**
F4	1.26E+02	**0.00E+00**	6.64E+02	**0.00E+00**
F5	1.06E+00	**0.00E+00**	5.06E+01	**0.00E+00**
F6	1.59E−01	**6.75E−14**	1.87E−01	**1.35E−13**

Values in bold represent the best value found between DIRECT and FDA.

$1.0000E-14$ are considered equal to $0.0000E+00$ as suggested in [15] where the benchmark is detailed.

As it was expected, the proposed approach seems to have difficulties in solving the Shifted Rosenbrock's function F_3, and the hybrid composition functions involving F_3, F_{13} and F_{17}, because our choice of ILS is more suited for separable and weakly separable problems. One can use another heuristic or metaheuristic rather than ILS. However, the FDA was able to reach the global optimum for 14 out of the 19 tested functions and that, for all the dimensions presented.

On the other hand, the fact that the standard deviations are always equal to $0.0000E+00$ denotes that the algorithm reaches always the same optimum.

6.6. *Analysis of FDA's behavior*

This section focuses on illustrating the way in which FDA behave in terms of number of spheres visited, fitness convergence and function evaluation consumption. The aim is to understand the behaviour of our proposed algorithm on the three main functions types: (1) separable function, (2) weakly separable function and (3) non-separable functions.

To illustrate its behavior, three functions (one of eatch type) taken from the benchmark SOCO 2011 were considered. The Shifted Rosenbrock's Function (F_3) and the Shifted Rastrigin's Function (F_4) are defined in Table 1. For the weakly separable function, the composite function F_{16} has been selected (defined in Table 3). For each function, FDAs behaviour is presented for dimensions $D = 50$ and $D = 1000$ (being the smaller and the bigger on the benchmark).

Table 9 and Fig. 3 show the number of evaluation consumed to find the best solution possible for both mentioned dimensions. For the separable function (F_3) all the function evaluations allowed are consumed without reaching the global optimum in both dimensions $D = 50$ and $D = 1000$. As pointed out, this is due to the intrinsic nature of the ILS. However, for the two other functions, weakly non-separable and separable functions, the global optimum is reached. For F_3, in dimension $D = 50$, the optimum is reached in 7803 evaluations over 25,000 allowed, representing around 31% of the stopping criteria. In $D = 1000$, the optimum is reached in 160,002 over 500,000 allowed, representing around 32% of the allowed function evaluations. This highlights the performance stability and scalability of FDA. In the case of the weakly separable function, F_{16}, FDA reached the optimum after 12,802 function evaluations, in dimension $D = 50$, (over the 25,000 allowed), representing 51% of the permitted evaluation and 268,002 evaluations over 500,000 in $D = 1000$, around 54%. This confirms the stability and scalability of FDA.

Furthermore, the total number of hyperspheres visited, meaning all hyperspheres focused by FDA at all *l* levels, in both exploration and intensification phases is given in Table 10 and illustrated in Fig. 4. In each function, the number of visited spheres is the same for both dimensions ($D = 50$ and $D = 1000$) showing the stability and scalability of the algorithm and our proposed decomposition approach regardless the dimension.

Finally, to illustrate FDAs behaviour, the fitness convergence is shown in Fig. 5 illustrating the fitness over the number of function evaluations. A log function has been applied on fitness axis to emphasize the fitnesss evolution. Once again, for each function the behaviour is similar across dimensions. In addition, in this figure we can observe that the slope drops suddenly and significantly to reach the moment when the best solution is found, being the optimum for F_3 and F_4. This sudden change corresponds to the moment when ILS is triggered and as explained, in the case of non-separable function (F_3 in our case) the curve stabilised without improving significantly until stopping criteria is reached.

In summary, FDA has a stable and scalable behaviour across all dimensions in the case of separable and weakly separable functions, keeping the number of visited spheres constant and the percentage of allowed function evaluations constant as well.

6.7. *Comparison with competing algorithms*

In this section, a comparison of our proposed FDA algorithm is conducted with other optimization algorithms from the literature. In first, a comparison with the related algorithm: DIRECT is performed. Then, FDA is compared with other competing meta-

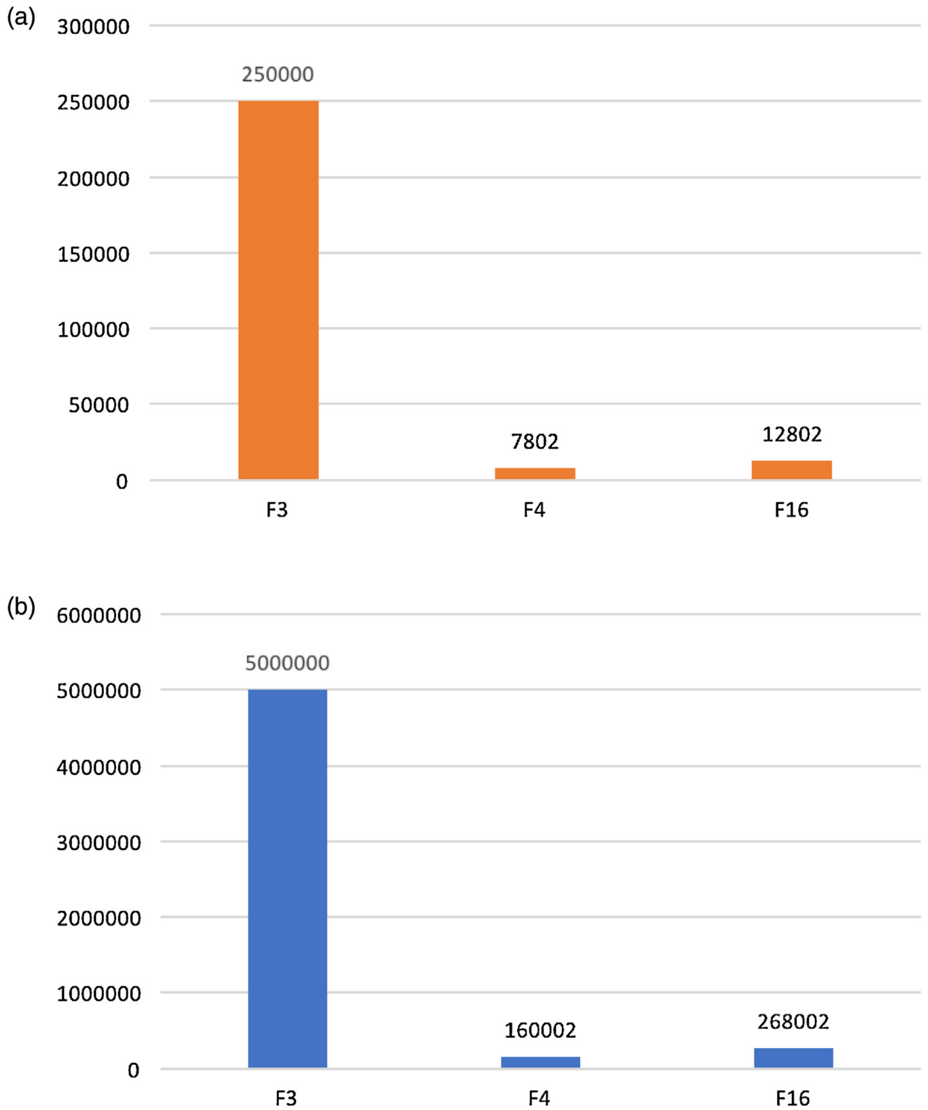


Fig. 3. Illustration number of evaluations to find the best solution for F_3, F_4 and F_{16}. (a) $D = 50$, (b) $D = 1000$.

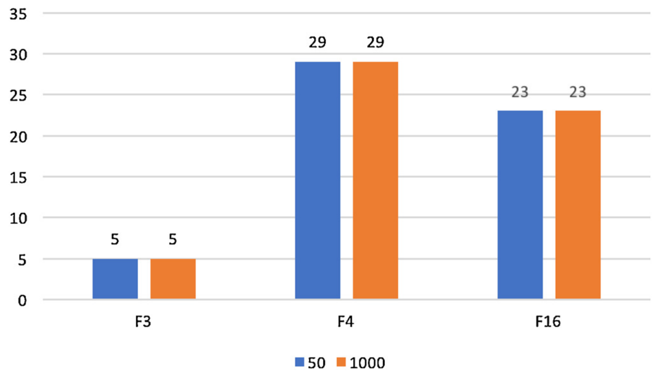


Fig. 4. Illustration the number of hyperspheres visited for F_3, F_4 and F_{16} for dimensions $D = 50$ and $D = 1000$.

heuristics from the literature, theirs results on SOCO 2011 were taken from the corresponding papers.

6.7.1. *Comparison with DIviding RECTangles (DIRECT)*

As pointed out in Section 2, DIviding RECTangles (DIRECT) is a algorithm that decomposes the search domain to find the global optimum. Being one of the most popular in the Multi-Scale Optimization (MSO) category [8], we have decided to compare its results with FDA. However, as its number of expansions grows quadratically with regards to the problem dimension N, we restricted the comparison to functions F_1 to F_6 (Table 1), and to the dimensions $D = 50$ and $D = 100$.

As mentioned in Section 2, DIRECT does not perform well in dimension $D > 10$. This is confirmed by the results showed in Table 11 as FDA outperforms DIRECT on both dimensions and on all functions.

6.7.2. *FDA comparison with SOCO 2011 participants*

As mentioned the first seven metaheuristics considered are presented. However, algorithms based on the hybridization of multiple metaheuristics were excluded because we consider them as a separate class of metaheuristics. Then, the considered algorithms are:

- Differential Evolution Algorithm [16] which uses the exponential crossover ($DE/rand/1/exp$).
- Real-coded Genetic Algorithm (CHC) [17].
- MA-SSW-Chains: Memetic algorithm based on local search chains for large-scale continuous Optimization Problems [18]. In

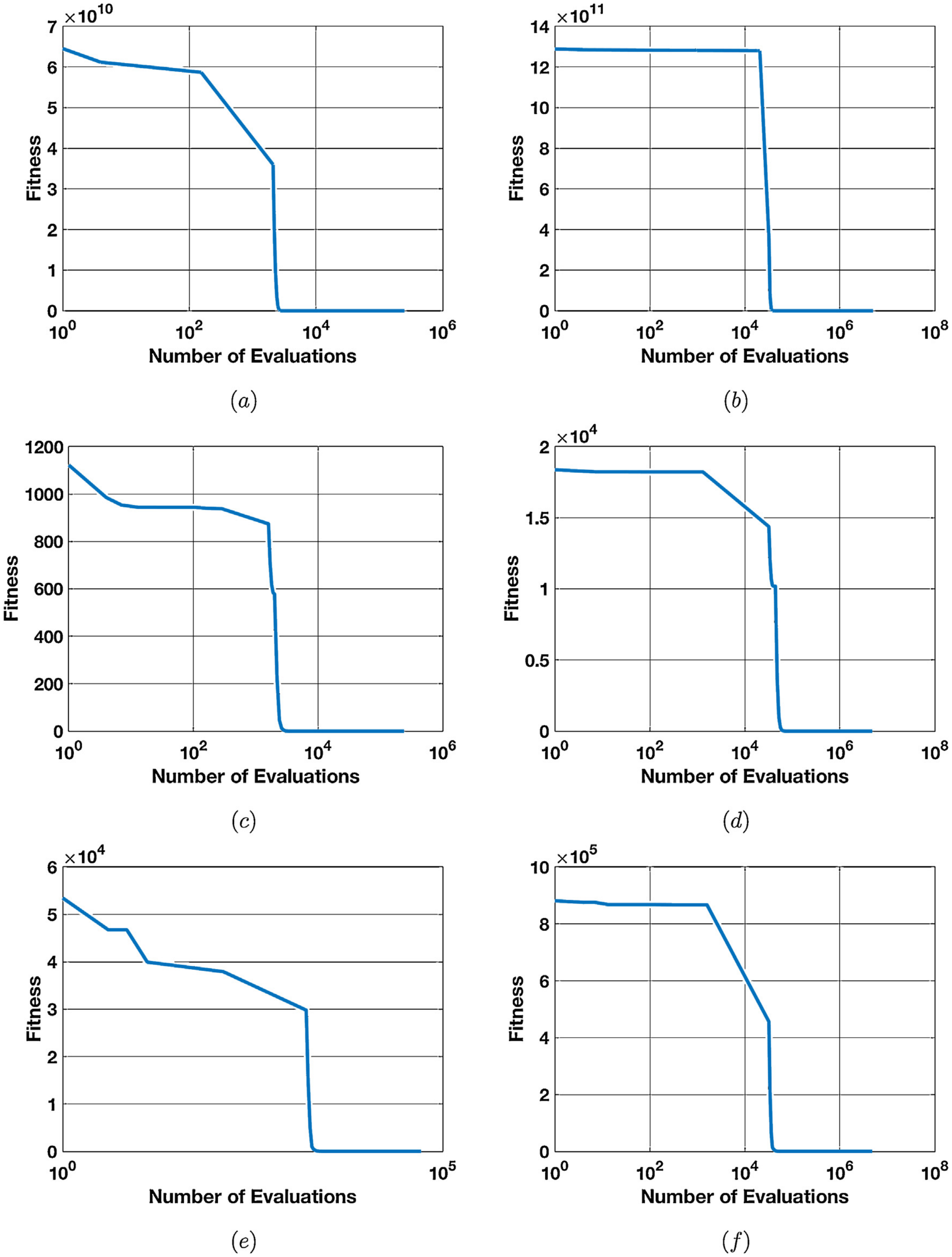


Fig. 5. Diagrams illustrating the log of the fitness over the number of function evaluations. (a) F_3 and $D=50$, (b) F_3 and $D=1000$ (c) F_4 and $D=50$ (d) F_4 and $D=1000$ (e) F_{16} and $D=50$, (f) F_{16} and $D=1000$.

addition to classical memetic algorithm, this version consists of applying a local search, to the last used configuration.

- The first local search method issued from the Multiple Trajectory Search for large-scale optimization [19] presented as MTS-LS1 as in [20]. The algorithm was fitted using the suited values suggested by authors of the original paper [19].
- Self-adaptive Differential Evolution (SaDE) [21] that uses a learning procedure to generate trial vectors strategies with their associated parameters.
- Multi-population Differential Evolution with balanced ensemble of mutation strategies for large-scale optimization (mDE-bES) [22]. This algorithm consists of dividing the population into independent subpopulations using different mutation operators for each one and updating the strategies during the search.

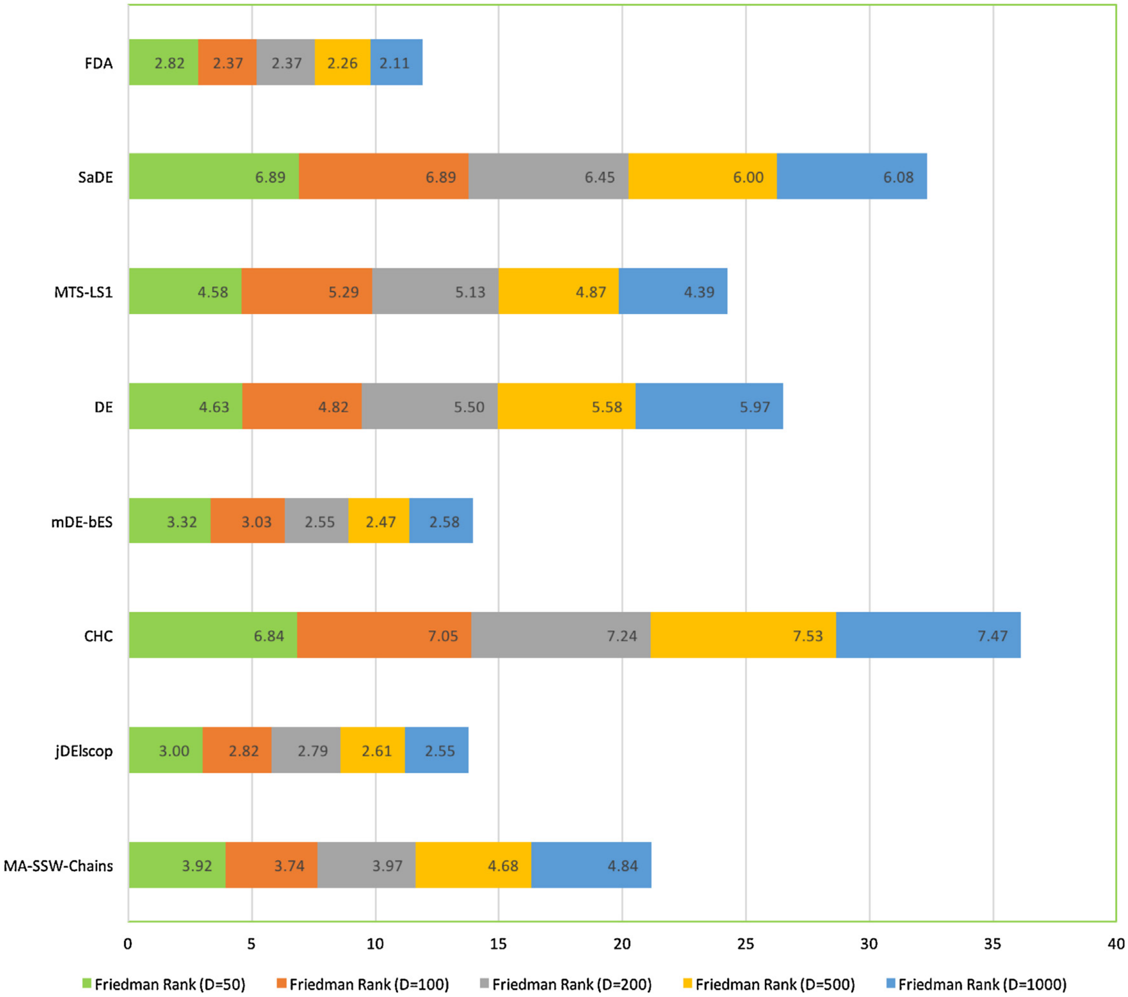


Fig. 6. Illustration of the ranking of the algorithms at dimensions $D = 50$, $D = 100$, $D = 200$, $D = 500$ and $D = 1000$.

- Self-adaptive differential evolution algorithm using population size reduction and three strategies (jDElscop) [23]. jDElscop employs three differential evolution (DE) strategies, a newly proposed population size reduction mechanism, and a mechanism for changing the sign of F control parameter.

Tables 19–23 illustrates the average error obtained by FDA on each algorithm presented above. Table 16 indicates the number of times each algorithm reaches the global optimum for all the benchmarks dimensions, respectively $D = 50$, $D = 100$, $D = 200$, $D = 500$ and $D = 1000$. To perform the performance comparison, the eight algorithms are ranked using the Friedman ranking sum test presented in Tables 13 and illustrated in Fig. 6. In other terms, the average relative rank is computed for each algorithm according to its mean performance for each function and the average ranking computed through all the functions is then reported. It can be observed that our algorithm is ranked first for all dimensions. It is also illustrated in Figs. 8–12 which show the boxplots for the distribution of the average ranks for each algorithm on all functions. In those plots, the circle highlights the outliers, meaning the functions where the algorithm performs surprisingly good or bad. In our case FDA performs surprisingly bad on the function F_{13} on dimension $D = 50$. It is however clear that our work shows better and stable performance among all dimensions on all functions.

To confirm this performance, we have conducted a Wilcoxon pairwise test to FDA and each algorithm presented. The p-values given by the Wilcoxon test have been adjusted using the Holm procedure [24] to control the familiywise error rate. Table 13 presents the Friedman Rank Sum score for each algorithm and shows that FDA is ranked first in all dimensions. Tables 14 and 15 show the resulting p-values (raw and adjusted) of the Wilcoxon test. Thus, algorithms with a p-value < 0.05 are statistically outperformed by our proposed work. Looking at the p-values, FDA statistically outperformed MA-SSW-Chains, CHC, DE, MTS-LS1 and SaDE in all dimensions. Finally, to support our work performance, as shown in Table 16, FDA solved, in average, 14 problems out of 19, and is ranked 1*st* where both mdE-bES a1) the globalnd jDElscop solve,

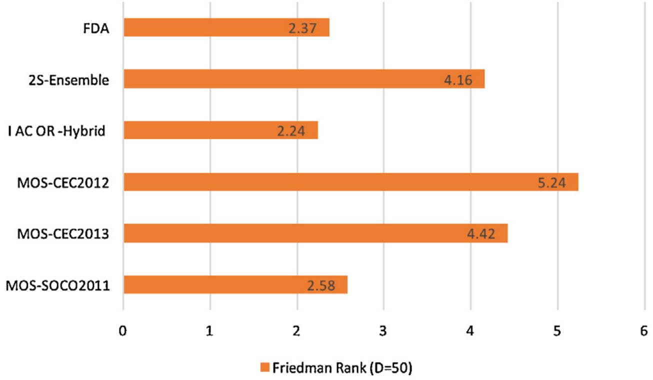


Fig. 7. Illustration of the ranking of the other metaheuristics algorithms at dimensions $D = 50$.

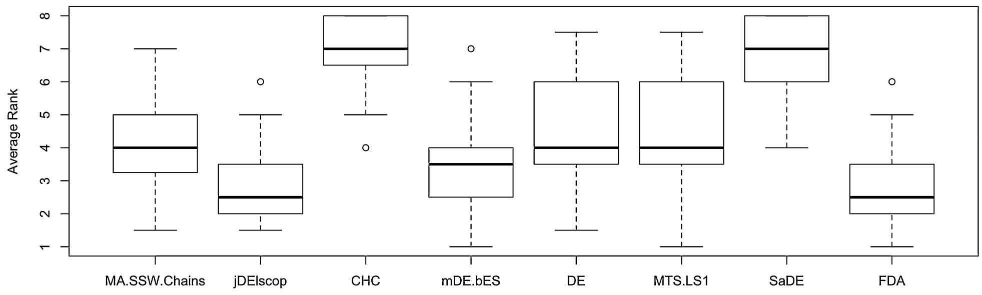


Fig. 8. Boxplots for the distribution of the average ranks for each algorithm on $D = 50$. Circles represent outliers as defined earlier.

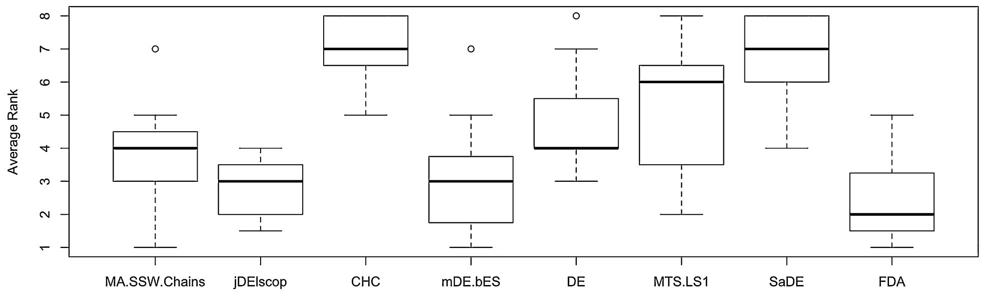


Fig. 9. Boxplots for the distribution of the average ranks for each algorithm on $D = 100$. Circles represent outliers as defined earlier.

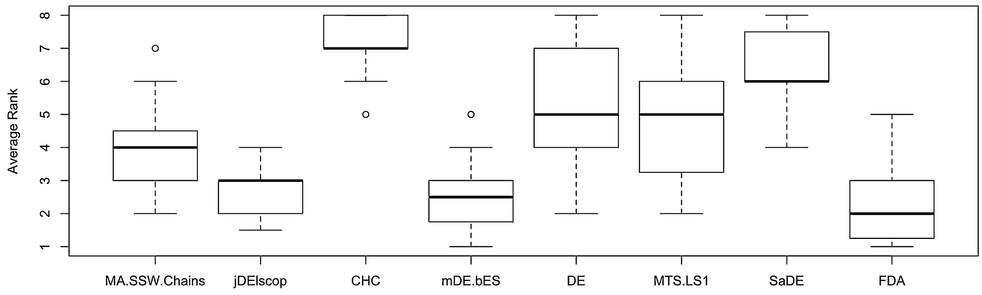


Fig. 10. Boxplots for the distribution of the average ranks for each algorithm on $D = 200$. Circles represent outliers as defined earlier.

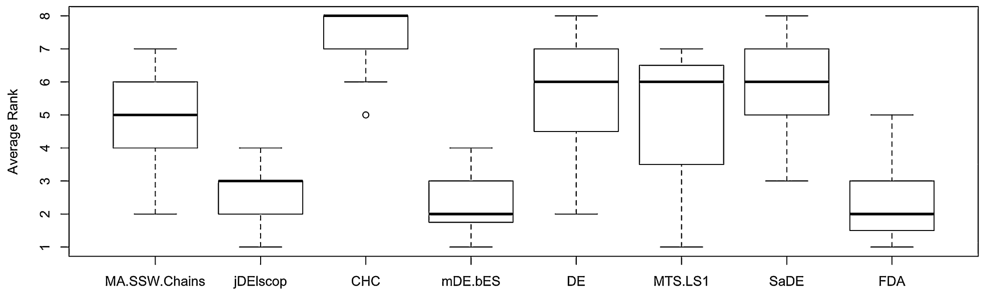


Fig. 11. Boxplots for the distribution of the average ranks for each algorithm on $D = 500$. Circles represent outliers as defined earlier.

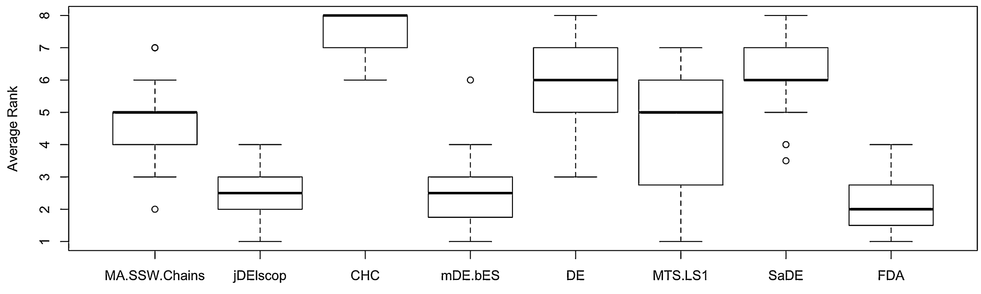


Fig. 12. Boxplots for the distribution of the average ranks for each algorithm on $D = 1000$. Circles represent outliers as defined earlier.

Table 12
The complexity of algorithms being used in the comparison.

Algorithm	Complexity
DE	$\mathcal{O}(D^2)$
CHC	$\mathcal{O}(D^2)$
MTS-LS1	Less than or equal to $\mathcal{O}(D^2)$
SaDE	Less than or equal to $\mathcal{O}(D^3)$
mDE-bES	Less than or equal to $\mathcal{O}(D^2)$
jDElscop	Less than or equal to $\mathcal{O}(D^2)$
MA-SSW-Chains	Greater than or equal to $\mathcal{O}(D^3)$
FDA	Less than or equal to $\mathcal{O}(\log_k(D))$

Table 13
Ranking using Friedman Rank sum of all algorithms at dimensions $D = 50$, $D = 100$, $D = 200$, $D = 500$ and $D = 1000$.

Ranks for dimensions	$D = 50$	$D = 100$	$D = 200$	$D = 500$	$D = 1000$
MA-SSW-Chains	3.92 (4)	3.74 (4)	3.97 (4)	4.68 (4)	4.84 (5)
jDElscop	3.00 (2)	2.82 (2)	2.79 (3)	2.61 (3)	2.55 (2)
CHC	6.84 (7)	7.05 (8)	7.24 (8)	7.53 (8)	7.47 (8)
mDE-bES	3.32 (3)	3.03 (3)	2.55 (2)	2.47 (2)	2.58 (3)
DE	4.63 (6)	4.82 (5)	5.50 (6)	5.58 (6)	5.97 (6)
MTS-LS1	4.58 (5)	5.29 (6)	5.13 (5)	4.87 (5)	4.39 (4)
SaDE	6.89 (8)	6.89 (7)	6.45 (7)	6.00 (7)	6.08 (7)
FDA	2.82 (1)	2.37 (1)	2.37 (1)	2.26 (1)	2.11 (1)

Values in parenthesis represent the algorithm's rank for the given dimension relative to the others.

Table 15
Raw and adjusted (using the Holm procedure) p-values from Wilcoxon test.

FDA vs.	$D = 500$		$D = 1000$	
	Raw	Adjusted	Raw	Adjusted
MA.SSW.Chains	4.88E−05	1.95E−04	1.23E−06	4.90E−06
jDElscop	**1.22E−01**[a]	**2.44E−01**[a]	**8.18E−02**[a]	**1.64E−01**[a]
CHC	7.98E−08	5.58E−07	8.54E−08	5.98E−07
mDE.bES	**4.93E−01**[a]	**4.93E−01**[a]	**2.48E−01**[a]	**2.48E−01**[a]
DE	3.59E−06	1.79E−05	3.30E−07	1.65E−06
MTS.LS1	3.90E−04	1.17E−03	4.97E−04	1.49E−03
SaDE	4.23E−07	2.54E−06	1.68E−07	1.01E−06

[a] p-Value > 0.05, failing to show statistical difference with significan level $\alpha = 0.05$.

Table 16
Number of times global optimum is reached.

Dimensions	$D = 50$	$D = 100$	$D = 200$	$D = 500$	$D = 1000$	Average	Rank
MA-SSW-Chains	9	8	6	3	2	5.6	(4)
jDElscop	12	10	9	8	7	9.2	(3)
CHC	0	0	0	0	0	0	(8)
mDE-bES	9	9	11	9	9	9.4	(2)
DE	7	4	2	1	1	3	(6)
MTS-LS1	7	4	4	2	4	4.2	(5)
SaDE	1	1	1	1	1	1	(7)
FDA	14	14	14	14	14	14	(1)

Values in parenthesis represent the algorithm's rank relative to the others among all dimensions.

in rounded-average, 9 problems and are respectively ranked 2*nd* and 3*rd*. We can conclude that FDA performs well for the considered benchmark problems scalability-wise and is stable among all dimensions.

Regarding the complexity, Table 12 summarizes different complexities of algorithms. It can be noticed that FDA has the lowest complexity. Moreover, the rest of the algorithms in Table 12 have polynomial complexities, while, FDA has a logarithmic complexity. Hence, the theoretical analysis of the proposed approach in addition to the obtained experimental results shows that the FDA can be an efficient alternative to solve large-scale problems.

6.7.3. *Comparison with recent metaheuristics*

In addition to the study conducted in the previous section, a similar experimentation has been performed to compare the performance of FDA with recent metaheuristics. These algorithms were inspired by a comprehensive comparison of large scale global optimizers [24] and other algorithms taken from the literature and are described in the following. While in the previous section, hybrid algorithms using multiple metaheuristics were excluded as part of a different class, we have included some examples in that section as they are state-of-the-art approaches.

- Multiple Offspring Sampling (MSO) based dynamic memetic differential evolution algorithm for continuous optimization refered as MSO-SOCO2011. A hybrid version of MSO combining a differential evolution (DE) algorithm and the first one of the local searches of the MTS algorithm [20].
- Multiple Offspring Sampling in Large Scale Global Optimization (MSO-CEC2012). Another hybrid MSO-based metaheuristic combining the first one of the local searches of the MTS [20] (also used in MSO-SOCO2011) and the Solis and Wets heuristic [25]. Originally applied to the CEC 2005 and CEC 2012, the results for the SOCO-2011 have been found in the litterature [24].
- Large Scale Global Optimization: experimental results with MOS-based hybrid algorithms [26] (MOS-CEC2013). Yet another MSO-based approach. This version also combine the power of the MTS [20] with the Solis Wets algorithm [25] but inovates in integrating a population-based search Genetic Algorithm (GA).
- Two-stage based ensemble optimization for Large-Scale Global Optimization referred as $2S - Ensemble$ in [20] and presented in the original paper [27]. The search procedure is divided into two phases: (1) the global shrinking focusing on finding promising area as fast as possible using a EDA based-on mixed Gaussian and Cachy models (MUEDA) [28] and (2) exploring the selected area using a co-evolution-based algorithm.
- IACO $\mathbb{R}$-Hybrid is an hybridisation method for the exploration phase, based on an Incremental Ant Colony Framework [29] (IACO $\mathbb{R}$) and combining the Multi-Trajectory Local Search (Mtsls1) algorithm and Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [30].

Table 14
Raw and adjusted (using the Holm procedure) p-values from Wilcoxon test.

FDA vs.	$D = 50$		$D = 100$		$D = 200$	
	Raw	Adjusted	Raw	Adjusted	Raw	Adjusted
MA.SSW.Chains	1.62E−02	4.87E−02	2.63E−03	7.90E−03	8.19E−04	2.46E−03
jDElscop	**6.03E−01**[a]	**6.03E−01**[a]	**1.07E−01**[a]	**2.14E−01**[a]	**1.23E−01**[a]	**2.46E−01**[a]
CHC	4.10E−07	2.87E−06	1.25E−07	8.72E−07	1.26E−07	8.82E−07
mDE.bES	**2.56E−01**[a]	**5.12E−01**[a]	**2.16E−01**[a]	**2.16E−01**[a]	**5.64E−01**[a]	**5.64E−01**[a]
DE	4.45E−03	2.22E−02	6.70E−06	3.35E−05	7.48E−06	3.74E−05
MTS.LS1	7.02E−03	2.81E−02	3.18E−05	1.27E−04	5.27E−05	2.11E−04
SaDE	4.62E−07	2.87E−06	1.64E−07	9.81E−07	3.17E−07	1.90E−06

[a] p-value > 0.05, failing to show statistical difference with significan level $\alpha = 0.05$.

Table 17
Ranking using Friedman Rank sum with other algorithms at dimension $D = 50$.

FDA vs.	$D = 50$	Rank
MOS-SOCO2011	2.578947368	(3)
MOS-CEC2013	4.421052632	(5)
MOS-CEC2012	5.236842105	(6)
IACO $\mathbb{R}$-Hybrid	2.236842105	(1)
2S-Ensemble	4.157894737	(4)
FDA	2.368421053	(2)

Values in parenthesis represent the algorithm's rank for the given dimension relative to the others.

Table 18
Raw and adjusted (using the Holm procedure) p-values from Wilcoxon test with other metaheuristics.

FDA vs.	$D = 50$	
	Raw	Adjusted
MOS-SOCO2011	**2.54E−01**[a]	**5.09E−01**[a]
MOS-CEC2013	2.12E−05	8.46E−05
MOS-CEC2012	4.58E−07	2.29E−06
IACO $\mathbb{R}$-Hybrid	**9.16E−01**[a]	**9.16E−01**[a]
2S-Ensemble	5.14E−05	1.54E−04

[a] p-Value > 0.05, failing to show statistical difference with significant level α = 0.05.

The same experimentations, as in the previous section, were performed to compare FDA with these metaheuristics, Table 24 shows the average errors obtained by each algorithm based on 19 functions of the SOCO 2011 benchmark with a focus on the dimension $D = 50$ as results were available in the original papers and in the literature for all algorithms.

Table 17 shows the ranks using the Friedman sum test where FDA is ranked second. This is also illustrated in Fig. 7. Fig. 13 illustrates as a boxplot, the ranks distribution of all algorithms. As mentioned in the previous section, circles represent the outliers. In that case FDA performs surprisingly bad in 3 functions, the first, third and sixth causing the rank to drop second. Overall FDA performance is stable on all other 16 functions.

Table 18 presents the p-values, adjusted using the Holm procedure, resulting from a Wilcoxon pairwise test pairwise to FDA and allows us to highlight the fact that FDA is statistically more efficient than MOS-CEC2013, MOS-CEC2012 and 2S-Ensemble. While MOS-SOCO2011 and IACO $\mathbb{R}$-Hybrid appear to achieve better performance than our work by obtaining respectively similar and better ranks, the adjusted p-values from the Wilcoxon test lead us to ensure that no statistical differences can be found to confirm that impression. In addition, Table 25 shows that FDA solved the same number of problems as the other two most performant algorithms.

Table 19
Average error on 50D functions.

	MA-SSW-Chains	jDElscop	CHC	mDE-bES	DE	MTS-LS1	SaDE	FDA
F_1	0.00E+00	0.00E+00	1.67E−11	0.00E+00	0.00E+00	0.00E+00	2.68E+01	0.00E+00
F_2	7.61E−02	3.15E−02	6.19E+01	1.52E+01	8.84E−11	8.84E−14	1.21E+02	2.71E−12
F_3	4.79E+01	2.28E+01	1.25E+06	4.76E−05	1.63E+02	1.63E+02	7.46E+04	9.32E+01
F_4	1.19E−01	0.00E+00	7.43E+01	1.77E+01	0.00E+00	0.00E+00	1.07E+01	0.00E+00
F_5	0.00E+00	0.00E+00	1.67E−03	0.00E+00	7.68E−03	7.68E−03	1.87E−01	0.00E+00
F_6	4.89E−14	9.55E−14	6.15E−07	3.97E−14	0.00E+00	0.00E+00	4.63E−02	6.75E−14
F_7	0.00E+00	0.00E+00	2.66E−09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_8	3.06E−01	9.97E−03	2.24E+02	1.64E−09	9.56E−12	9.65E−12	6.92E+05	0.00E+00
F_9	2.94E+02	0.00E+00	3.10E+02	0.00E+00	1.03E+02	1.03E+02	3.00E−02	0.00E+00
F_{10}	0.00E+00	0.00E+00	7.30E+00	0.00E+00	0.00E+00	0.00E+00	2.94E−02	0.00E+00
F_{11}	4.49E−03	0.00E+00	2.16E+00	1.15E−08	1.04E+02	1.04E+02	8.35E−02	0.00E+00
F_{12}	0.00E+00	0.00E+00	9.57E−01	0.00E+00	1.34E+01	1.34E+01	4.80E+01	0.00E+00
F_{13}	3.02E+01	1.36E+01	2.08E+06	2.50E−01	2.94E+01	2.94E+01	3.42E+09	5.50E+01
F_{14}	0.00E+00	0.00E+00	6.17E+01	9.60E+00	5.52E+01	5.52E+01	4.22E+03	0.00E+00
F_{15}	0.00E+00	0.00E+00	3.98E−01	0.00E+00	0.00E+00	0.00E+00	8.50E−03	0.00E+00
F_{16}	4.06E−03	0.00E+00	2.95E−09	0.00E+00	4.06E+01	4.06E+01	1.36E+01	0.00E+00
F_{17}	2.60E+01	7.43E−03	2.26E+04	2.42E−01	2.17E+02	2.17E+02	2.36E+05	6.09E−04
F_{18}	0.00E+00	2.41E−14	1.58E+01	5.65E−05	5.65E+01	5.65E+01	2.72E+01	0.00E+00
F_{19}	0.00E+00	0.00E+00	3.59E+02	0.00E+00	0.00E+00	0.00E+00	1.15E−01	0.00E+00

Table 20
Average error on 100D functions.

	MA-SSW-Chains	jDElscop	CHC	mDE-bES	DE	MTS-LS1	SaDE	FDA
F_1	0.00E+00	0.00E+00	3.56E−11	0.00E+00	3.79E+00	1.09E−12	3.13E+01	0.00E+00
F_2	7.01E+00	1.21E+00	8.58E+01	4.00E+01	7.58E+01	4.66E−10	1.26E+02	8.48E−12
F_3	1.38E+02	6.13E+01	4.19E+06	4.90E−01	1.27E+02	2.32E+02	1.11E+05	5.09E+01
F_4	1.19E−01	0.00E+00	2.19E+02	1.87E+01	2.85E+00	1.05E−12	1.58E+01	0.00E+00
F_5	0.00E+00	0.00E+00	3.83E−03	0.00E+00	3.05E−01	6.70E−03	3.53E−01	0.00E+00
F_6	6.03E−14	2.00E−13	4.10E−07	1.44E−13	4.34E−01	1.20E−12	8.32E−02	1.35E−13
F_7	0.00E+00	0.00E+00	1.40E−02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_8	3.48E+01	5.57E+00	1.69E+03	2.32E−03	4.74E+02	1.43E−03	2.83E+05	0.00E+00
F_9	5.63E+02	7.18E−09	5.86E+02	0.00E+00	3.71E−03	2.20E+02	3.00E−02	0.00E+00
F_{10}	0.00E+00	0.00E+00	3.30E+01	0.00E+00	0.00E+00	0.00E+00	4.73E−02	0.00E+00
F_{11}	1.09E−01	8.17E−09	7.32E+01	0.00E+00	8.58E−04	2.10E+02	3.05E−01	0.00E+00
F_{12}	3.28E−03	0.00E+00	1.03E+01	5.36E−04	2.71E+00	3.91E+01	3.79E+01	0.00E+00
F_{13}	8.35E+01	5.11E+01	2.70E+06	8.50E+00	5.87E+01	1.75E+02	3.42E+09	1.68E+02
F_{14}	0.00E+00	0.00E+00	1.66E+02	1.16E+01	2.21E+00	2.04E+02	3.92E+03	0.00E+00
F_{15}	0.00E+00	0.00E+00	8.13E+00	0.00E+00	0.00E+00	0.00E+00	3.99E−02	0.00E+00
F_{16}	1.61E−02	0.00E+00	2.23E+01	0.00E+00	3.52E+00	1.04E+02	1.96E+01	0.00E+00
F_{17}	9.92E+01	3.21E−01	1.47E+05	6.65E−03	1.58E+01	4.17E+02	2.34E+05	6.22E+00
F_{18}	0.00E+00	6.33E−14	7.00E+01	4.46E−01	8.76E−01	1.22E+02	3.05E+01	0.00E+00
F_{19}	0.00E+00	0.00E+00	5.45E+02	0.00E+00	0.00E+00	0.00E+00	2.71E−01	0.00E+00

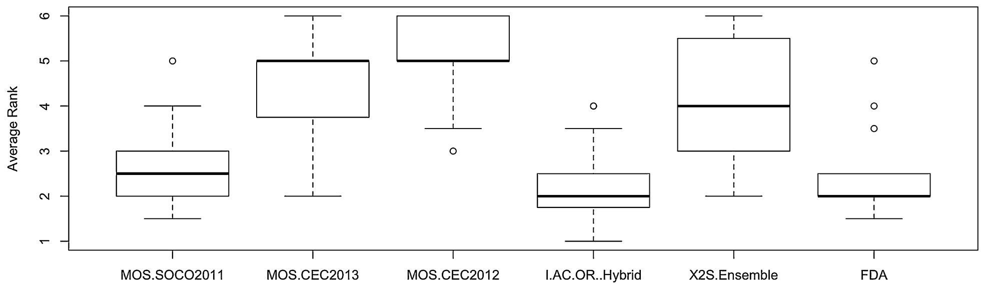


Fig. 13. Boxplots for the distribution of the average ranks for each algorithm at dimensions $D = 50$. Circles represent the outliers as explained earlier.

Table 21
Average error on 200*D* functions.

	MA-SSW-Chains	jDElscop	CHC	mDE-bES	DE	MTS-LS1	SaDE	FDA
F_1	0.00E+00	0.00E+00	8.34E−01	0.00E+00	8.55E+00	2.29E+00	2.03E+01	0.00E+00
F_2	3.36E+01	7.54E+00	1.03E+02	4.15E+01	1.05E+02	4.54E−09	1.03E+02	1.23E−10
F_3	2.50E+02	1.40E+02	2.01E+07	1.35E+02	3.32E+05	1.69E+02	4.82E+04	2.51E+02
F_4	4.43E+00	0.00E+00	5.40E+02	9.27E−13	6.98E+00	2.34E−12	6.25E+00	0.00E+00
F_5	0.00E+00	0.00E+00	8.76E−03	0.00E+00	4.05E−01	5.42E−03	6.43E−02	0.00E+00
F_6	1.19E−13	4.52E−13	1.23E+00	0.00E+00	7.14E−01	2.38E−12	2.73E−02	2.52E−13
F_7	0.00E+00	0.00E+00	2.59E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_8	7.23E+02	2.52E+02	9.38E+03	8.71E−01	5.76E+03	1.42E+01	4.47E+05	0.00E+00
F_9	1.17E+03	4.30E−08	1.19E+03	0.00E+00	8.79E−03	4.27E+02	3.00E−02	0.00E+00
F_{10}	0.00E+00	0.00E+00	7.13E+01	0.00E+00	4.19E−02	0.00E+00	1.59E−02	0.00E+00
F_{11}	3.50E−01	9.58E−09	3.85E+02	0.00E+00	5.07E−03	4.28E+02	4.89E−03	0.00E+00
F_{12}	1.75E−02	0.00E+00	7.44E+01	0.00E+00	3.61E+00	8.42E+01	4.63E+01	0.00E+00
F_{13}	1.68E+02	1.10E+02	5.75E+06	9.45E+01	1.49E+02	2.53E+02	3.16E+09	7.07E+01
F_{14}	9.76E−01	4.11E−16	4.29E+02	1.20E+01	4.75E+00	3.98E+02	4.09E+03	0.00E+00
F_{15}	0.00E+00	0.00E+00	2.14E+01	0.00E+00	0.00E+00	0.00E+00	5.38E−03	0.00E+00
F_{16}	6.02E−02	0.00E+00	1.60E+02	0.00E+00	3.70E+00	1.97E+02	9.49E+00	0.00E+00
F_{17}	7.55E+01	2.39E+01	1.75E+05	8.39E−02	2.23E+01	6.07E+02	2.36E+05	9.31E+01
F_{18}	4.29E−04	2.04E−13	2.12E+02	8.93E−11	2.37E+00	2.34E+02	1.69E+01	0.00E+00
F_{19}	0.00E+00	0.00E+00	2.06E+03	0.00E+00	4.19E−02	0.00E+00	1.00E−01	0.00E+00

Table 22
Average error on 500*D* functions.

	MA-SSW-Chains	jDElscop	CHC	mDE-bES	DE	MTS-LS1	SaDE	FDA
F_1	0.00E+00	0.00E+00	2.84E−12	3.92E−13	2.46E+01	5.77E−12	1.34E+01	0.00E+00
F_2	7.86E+01	3.06E+01	1.29E+02	4.56E+01	1.44E+02	5.34E−06	9.23E+01	4.30E−04
F_3	6.07E+02	4.06E+02	1.14E+06	4.16E+02	1.12E+05	2.20E+02	2.62E+04	5.82E+02
F_4	1.78E+02	1.59E−01	1.91E+03	1.91E−11	1.63E+01	5.62E−12	1.31E+00	0.00E+00
F_5	0.00E+00	0.00E+00	6.98E−03	1.83E−13	4.73E−01	4.24E−03	7.48E−03	0.00E+00
F_6	2.63E−13	1.18E−12	5.16E+00	3.56E−14	1.06E+00	6.18E−12	4.63E−01	8.74E−13
F_7	4.69E−14	0.00E+00	1.27E−01	0.00E+00	0.00E+00	1.46E−12	0.00E+00	0.00E+00
F_8	1.32E+04	5.66E+03	7.22E+04	5.48E+02	6.70E+04	6.16E+03	3.21E+05	0.00E+00
F_9	2.53E+03	6.10E−08	3.00E+03	0.00E+00	1.12E−02	1.00E+03	3.00E−02	0.00E+00
F_{10}	2.80E−01	0.00E+00	1.86E+02	0.00E+00	2.93E−01	0.00E+00	8.41E−03	0.00E+00
F_{11}	4.21E+01	4.40E−08	1.81E+03	0.00E+00	2.43E−01	1.00E+03	2.22E−03	0.00E+00
F_{12}	2.55E+01	0.00E+00	4.48E+02	0.00E+00	1.16E+01	2.47E+02	4.61E+01	0.00E+00
F_{13}	4.00E+02	3.14E+02	3.22E+07	3.23E+02	4.02E+02	5.05E+02	2.97E+09	3.74E+02
F_{14}	5.65E+01	8.00E−02	1.46E+03	1.68E+01	1.16E+01	1.10E+03	3.91E+03	0.00E+00
F_{15}	5.53E+00	0.00E+00	6.01E+01	0.00E+00	4.19E−02	1.08E−12	2.84E−03	0.00E+00
F_{16}	1.08E−01	0.00E+00	9.55E+02	0.00E+00	1.32E+01	4.99E+02	5.82E+00	0.00E+00
F_{17}	1.38E+02	7.65E+01	8.40E+05	6.65E+01	6.94E+01	7.98E+02	2.38E+05	3.96E+02
F_{18}	2.41E−03	1.11E−12	7.32E+02	0.00E+00	3.87E+00	5.95E+02	9.43E+00	0.00E+00
F_{19}	0.00E+00	0.00E+00	1.76E+03	0.00E+00	8.39E−02	0.00E+00	1.00E−01	0.00E+00

Finally, it is crucial to emphasize the fact that, as mentioned earlier, FDA uses a deterministic approach with a single solution, hence and does not benefit from neither a population solution nor a stochastic approach unlike the other state-of-the-art algorithms. Failing to find significant differences with FDA highlights the power of FDA and its potential margin for improvement and, as detailed in the future work section, the FDA approach will lead to great results by incorporating elements which can be found in state-of-the-art algorithms.

6.8. Real problem: Lennard-Jones atomic cluster

In this section we have applied FDA to a real problem, *the Lennard-Jones atomic cluster*.

This problem considers a cluster of K atoms in a 3-dimensional search space. The aim is to minimize the potential energy of this cluster whose equation is:

$$MinV_k(x) = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \left(\frac{1}{||x_i - x_j||^{12}} - \frac{1}{||x_i - x_j||^{6}} \right) \tag{40}$$

Table 23
Average error on 1000*D* functions.

	MA-SSW-Chains	jDElscop	CHC	mDE-bES	DE	MTS-LS1	SaDE	FDA
F_1	0.00E+00	0.00E+00	1.36E−11	8.24E−13	3.71E+01	1.15E−11	3.49E+01	0.00E+00
F_2	1.39E+02	6.14E+01	1.44E+02	5.97E+01	1.63E+02	2.25E−02	1.43E+02	3.11E−01
F_3	1.22E+03	8.48E+02	8.75E+03	9.00E+02	1.59E+05	2.10E+02	1.62E+05	1.13E+03
F_4	1.58E+03	1.99E−01	4.76E+03	4.03E+01	3.47E+01	1.15E−11	3.21E+01	0.00E+00
F_5	5.92E−04	0.00E+00	7.02E−03	0.00E+00	7.36E−01	3.55E−03	6.33E−01	0.00E+00
F_6	1.46E−09	2.67E−12	1.38E+01	1.28E−12	8.70E−01	1.24E−11	4.28E−01	1.91E−12
F_7	6.23E−13	0.00E+00	3.52E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_8	7.49E+04	3.21E+04	3.11E+05	7.98E+03	3.15E+05	1.23E+05	3.08E+05	0.00E+00
F_9	5.99E+03	4.40E−03	6.11E+03	0.00E+00	6.26E−02	1.99E+03	3.00E−02	0.00E+00
F_{10}	2.09E−05	0.00E+00	3.83E+02	0.00E+00	1.67E−01	0.00E+00	1.47E−01	0.00E+00
F_{11}	5.27E+01	8.58E−04	4.82E+03	0.00E+00	4.42E−02	1.99E+03	4.56E−01	0.00E+00
F_{12}	9.48E−02	0.00E+00	1.05E+03	0.00E+00	2.58E+01	5.02E+02	3.43E+01	0.00E+00
F_{13}	1.02E+03	6.57E+02	6.66E+07	6.34E+02	8.24E+04	8.87E+02	3.27E+09	7.69E+02
F_{14}	7.33E+02	3.98E−02	3.62E+03	2.45E+01	2.39E+01	2.23E+03	3.71E+03	0.00E+00
F_{15}	1.16E−13	0.00E+00	8.37E+01	0.00E+00	2.11E−01	0.00E+00	1.11E−01	0.00E+00
F_{16}	2.19E+00	8.04E−01	2.32E+03	0.00E+00	1.83E+01	1.00E+03	2.37E+01	0.00E+00
F_{17}	3.26E+02	1.72E+02	2.04E+07	1.88E+02	1.76E+05	1.56E+03	1.62E+05	1.95E+02
F_{18}	2.58E+01	1.65E−01	1.72E+03	2.49E−01	7.55E+00	1.21E+03	3.54E+01	0.00E+00
F_{19}	0.00E+00	0.00E+00	4.20E+03	0.00E+00	2.51E−01	0.00E+00	9.32E+02	0.00E+00

Table 24
Average error on 50*D* functions for other metaheuristics.

	MOS-SOCO2011	MOS-CEC2013	MOS-CEC2012	IACO ℝ-Hybrid	2S-Ensemble	FDA
F_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_2	5.88E−01	1.10E+02	1.03E+02	0.00E+00	4.31E+01	2.71E−12
F_3	7.09E+01	7.39E+00	9.38E+02	0.00E+00	1.34E+03	9.32E+01
F_4	0.00E+00	0.00E+00	1.90E+02	0.00E+00	8.58E−01	0.00E+00
F_5	0.00E+00	0.00E+00	1.18E−03	0.00E+00	3.00E−03	0.00E+00
F_6	0.00E+00	0.00E+00	1.03E+00	0.00E+00	0.00E+00	6.75E−14
F_7	0.00E+00	2.56E−12	1.03E−13	0.00E+00	Inf.	0.00E+00
F_8	1.66E+05	5.98E+03	1.09E+03	0.00E+00	1.93E+05	0.00E+00
F_9	0.00E+00	2.51E+03	5.95E+03	0.00E+00	2.68E+00	0.00E+00
F_{10}	0.00E+00	1.58E+00	1.79E+02	0.00E+00	0.00E+00	0.00E+00
F_{11}	0.00E+00	2.54E+03	5.88E+03	0.00E+00	3.23E+00	0.00E+00
F_{12}	0.00E+00	9.99E+02	1.12E+03	0.00E+00	0.00E+00	0.00E+00
F_{13}	1.69E+02	1.23E+03	2.03E+03	8.77E−01	1.25E+03	5.50E+01
F_{14}	0.00E+00	3.37E+03	4.32E+03	2.90E−02	4.40E−02	0.00E+00
F_{15}	0.00E+00	1.93E−12	2.04E+01	0.00E+00	Inf.	0.00E+00
F_{16}	0.00E+00	8.02E+03	2.33E+03	1.12E−03	0.00E+00	0.00E+00
F_{17}	6.71E+01	3.55E+11	3.71E+03	1.84E−06	3.39E+01	6.09E−04
F_{18}	0.00E+00	2.03E+03	2.29E+03	9.20E−01	5.51E−01	0.00E+00
F_{19}	0.00E+00	2.05E+03	5.25E+01	0.00E+00	7.99E−17	0.00E+00

Table 25
Number of times global optimum is reached for other metaheuristics.

Dimensions	*D* = 50	Rank
MOS-SOCO2011	14	(1)
MOS-CEC2013	4	(5)
MOS-CEC2012	1	(6)
IACO ℝ-Hybrid	14	(1)
2S-Ensemble	5	(4)
FDA	14	(1)

Values in parenthesis represent the algorithm's rank relative to the others among all dimensions.

where $i = 1, 2, \ldots, K$. The population size is 4, the dimension is equal to 3 × number of atoms, $\Delta = 20$ generations, the domain of definition is $[-2; 2]$, and results are computed on 100 runs. The stoping criteria is set to 65, 000 function evaluations. Tests are made for 8, 9 and 10 atoms, global optima are respectively −19.821489, −24.113360 and −28.422532. In that section we compare FDA with other algorithms SPSO2007, PSO-2S [31] and BEA [32].

Table 26 shows the results for the different compared algorithms. We can clearly see that FDA outperformed the other algorithms.

7. Conclusion and future work

This paper proposed a deterministic metaheuristic to solve large-scale optimization problems. The proposed approach includes a *divide and conquer* mechanism to explore the search space. Indeed, the geometric fractal decomposition uses the hypersphere as a geometrical form to represent the search space and its subregions to be visited. Then, a heuristic with a minimum cost in terms of complexity is applied to lead the search to a smaller promising region allowing the ILS to intensify the search in order to find the best solution in a reduced area at the last level. The Fractal Decomposition based Algorithm was tested on a set of test functions issued from the benchmark provided for the soft computing special issue on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems. The obtained results show the efficiency of the proposed approach and the comparisons with other state-of-the-art algorithms taken from the literature prove that its performance is very competitive for all considered dimensions.

As it was pointed out, the procedure used in ILS does not allow solving of problems that are fully non-separable structure, in near future, our work consists of proposing new heuristics at ILS level to deal with these problems. Then, we expect to introduce a method that is population based and parallel to enhance the performance.

Table 26
Results on the Lennard-Jones atomic cluster for all algorithms.

Number of atoms	8		9		10	
	Min. value	(Std.)	Min. value	(Std.)	Min. value	(Std.)
FDA	−4.96E+00	(0.00E+00)	−6.03E+00	(0.00E+00)	−6.63044	(0.00E+00)
SPSO2007	1.17E+00	(1.80E+00)	2.57E+00	(2.99E+00)	4.66E+00	(4.22E+00)
BEA	−2.31E+00	(4.26E−01)	−2.55E+02	(4.06E−01)	−2.76E+00	(4.85E−01)
PSO-2S	1.47E+00	(9.12E−02)	2.31E+00	(6.06E−02)	1.52E+00	(2.28E−01)

References

[1] M. Dorigo, V. Maniezzo, A. Colorni, 1991. Positive feedback as a search strategy, Technical report 91-016.

[2] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.

[3] J.H. Holland, Adaptation in Natural and Artificial Systems, 1975, 1991.

[4] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st ed., Addison-Wesley Longman Publishing Co., Inc., 1989.

[5] S. Mahdavi, M.E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: a survey, Inf. Sci. 295 (2015) 407–428.

[6] E.-G. Talbi, Metaheuristics: From Design to Implementation, Vol. 74, John Wiley & Sons, 2009.

[7] F. Almeida, D. Gimenez, J.J. Lopez-Espin, M. Perez-Perez, Parameterized schemes of metaheuristics: basic ideas and applications with genetic algorithms, scatter search and GRASP, IEEE Trans. Syst. Man Cybern. Syst. 43 (3) (2013) 82–117.

[8] A. Al-Dujaili, S. Suresh, N. Sundararajan, MSO: a framework for bound-constrained black-box global optimization algorithms, J. Global Opt. 66 (4) (2016) 811–845.

[9] D.R. Jones, C.D. Perttunan, B.E. Stuckman, Lipshitzian optimization without the Lipshitz coefficient, J. Opt. Theory Appl. 79 (1) (1993) 157–181.

[10] M. Demirhan, L. Özdamar, L. Helvacğlu, C.I. Birbil, Fractop: a geometric partitioning metaheuristic for global optimization, J. Global Opt. 14 (4) (1999) 415–436.

[11] S. Kirkpatrick, Optimization by simulated annealing: quantitative studies, J. Stat. Phys. 34 (1984) 975–986.

[12] D. Ashlock, J. Schonfeld, A fractal representation for real optimization, in: 2007 IEEE Congress on Evolutionary Computation, 2007, pp. 87–94.

[13] K. Tang, X. Yáo, P.N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, Z. Yang, Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization, Nature Inspired Computation and Applications Laboratory, USTC, China, 2007, pp. 153–177.

[14] D. Whitley, R. Beveridge, C. Graves, K. Mathias, Test driving three 1995 genetic algorithms: new test functions and geometric matching, J. Heuristics 1 (1) (1995) 77–104.

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

[16] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces, vol. 3, ICSI Berkeley, 1995.

[17] L.J. Eshelman, J.D. Schaffer, in: L.D. Whitley (Ed.), Real-coded Genetic Algorithms and Interval-schemata, FOGA, Morgan Kaufmann, 1992, pp. 187–202.

[18] D. Molina, M. Lozano, A.M. Sánchez, F. Herrera, Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains, Soft Comput. 15 (11) (2011) 2201–2220.

[19] L.-Y. Tseng, C. Chen, Multiple trajectory search for Large Scale Global Optimization, in: 2008 IEEE Congress on Evolutionary Computation, 2008, pp. 3052–3059.

[20] A. LaTorre, S. Muelas, J.-M. Peña, A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test, Soft Comput. 15 (11) (2011) 2187–2199.

[21] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: 2005 IEEE Congress on Evolutionary Computation, Vol. 2, 2005, pp. 1785–1791.

[22] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization, Appl. Soft Comput. 33 (2015) 304–327.

[23] J. Brest, M.S. Maučec, Self-adaptive differential evolution algorithm using population size reduction and three strategies, Soft Comput. 15 (11) (2011) 2157–2174.

[24] A. LaTorre, S. Muelas, J.-M. Pe na, A comprehensive comparison of large scale global optimizers, Inf. Sci. 316 (2014) 517–549.

[25] F.J. Solis, R.J.-B. Wets, Minimization by random search techniques, Math. Oper. Res. 6 (1) (1981) 19–30.

[26] A. LaTorre, S. Muelas, J.-M. Pena, Large scale global optimization: experimental results with MOS-based hybrid algorithms, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2742–2749.

[27] Y. Wang, J. Huang, W.S. Dong, J.C. Yan, C.H. Tian, M. Li, W.T. Mo, Two-stage based ensemble optimization framework for large-scale global optimization, Eur. J. Oper. Res. 228 (2) (2013) 308–320.

[28] Y. Wang, B. Li, A self-adaptive mixed distribution based uni-variate estimation of distribution algorithm for large scale global optimization, in: R. Chiong (Ed.), Nature-Inspired Algorithms for Optimisation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 171–198, http://dx.doi.org/10.1007/978-3-642-00267-0_6.

[29] T. Liao, M.A. Montes de Oca, D. Aydin, T. Stützle, M. Dorigo, An incremental ant colony algorithm with local search for continuous optimization, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO '11, ACM, 2011, pp. 125–132.

[30] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, 2006.

[31] A. El Dor, M. Clerc, P. Siarry, A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization, Comput. Opt. Appl. 53 (1) (2012) 271–295.

[32] A. Nakib, B. Thibault, P. Siarry, Bayesian based metaheuristic for large scale continuous optimization, in: 20015 International Parallel and Distributed Processing Symposium Workshops, 2015, pp. 314–322.