

Received August 25, 2020, accepted September 3, 2020, date of publication September 9, 2020, date of current version September 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022866

# Fractal Decomposition Approach for Continuous Multi-Objective Optimization Problems

LÉO SOUQUET<sup>1</sup>, EL GHAZALI TALBI<sup>2</sup>, AND AMIR NAKIB<sup>3</sup> 

<sup>1</sup>Data ScienceTech Institute (DSTI), 75005 Paris, France

<sup>2</sup>Laboratoire CRISTAL, Université de Lille, 59800 Lille, France

<sup>3</sup>Laboratoire LISSI, Université Paris-Est Créteil, 94400 Vitry-sur-Seine, France

Corresponding author: Amir Nakib (nakib@u-pec.fr)

**ABSTRACT** Multi-objective optimization problems (MOPs) have been widely studied during the last decades. In this article, we present a new intrinsically parallel approach based on Fractal decomposition (FDA) to solve MOPs. The key contribution of the proposed approach is to divide recursively the decision space using hyperspheres. Two different methods were investigated: the first one is based on scalarization that has been distributed on a parallel multi-node architecture virtual environments and taking profit from the FDA's properties, while the second method is based on Pareto dominance sorting. A comparison with state of the art algorithms on different well known benchmarks shows the efficiency and the robustness of the proposed decomposition approaches.

**INDEX TERMS** Multi-objective optimization, large-scale optimization, metaheuristics, geometric fractal decomposition, local search, continuous optimization, containers, virtualization, Docker, Kubernetes.


## I. INTRODUCTION

Many problems in science and industry are concerned with multi-objective optimization problems (MOPs). Multi-objective optimization seeks to optimize several components of an objective function vector. Contrary to single-objective optimization, the solution of a MOP is not a single solution, however, a set of solutions known as *Pareto optimal set*, which is called *Pareto front* when it is plotted in the objective space. Any solution of this set is optimal in the sense that no improvement can be made on a component of the objective vector without worsening at least another of its components. The main goal in solving a difficult MOP is to approximate the set of solutions within the Pareto optimal set and, consequently, the Pareto front.

Most of the well-known metaheuristics (e.g. evolutionary algorithms, particle swarm, ant colonies) have been adapted to solve multi-objective problems [2], [3], [13], [30], [31]. For instance, authors in [12] have proposed a new Particle Swarm Optimization algorithm to solve multi-objective problems based on double-archive mechanism and Levy Flight.

Multi-objective metaheuristics can be classified in three main categories:

- **Scalarization-based approaches:** this class of multi-objective metaheuristics contains the approaches which transform a MOP problem into a single-objective one or a set of such problems. Among these methods one can find the aggregation methods, weighted metrics, Tchebychev method, goal programming methods, achievement functions, goal attainment methods and the  $\epsilon$ -constraint methods [14], [19], [20], [22].
- **Dominance-based approaches:** the dominance-based approaches<sup>1</sup> use the concept of dominance and Pareto optimality to guide the search process. Since the beginning of the nineties, interest concerning MOPs area with Pareto approaches always grows. Most of Pareto approaches use EMO (Evolutionary Multi-criterion Optimization) algorithms. Population-based metaheuristics seem particularly suitable to solve MOPs, because they deal simultaneously with a set of solutions which allows to find several members of the Pareto optimal set in a single run of the algorithm. Moreover, they are less sensitive to the shape of the Pareto front (continuity, convexity). The well known NSGA-II algorithm [6] belongs to this category. It uses a crowded comparison method to select suitable individuals. An improved version of the crowding distance has

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Yetgin .

<sup>1</sup>Also named Pareto approaches.

been proposed in [10] and shows to improve NSGA-II performances. The main differences between the various proposed approaches in this category arise in the following search components: fitness assignment, diversity management, and elitism [18], [32], [36].

- **Decomposition-based approaches:** most of decomposition based algorithms in solving MOPs operate in the objective space. One of the well-known frameworks for MOEAs using decomposition is MOEA/D [8], [16], [33]. It uses scalarization to decompose the MOP into multiple scalar optimization subproblems and solve them simultaneously by evolving a population of candidate solutions. Subproblems are solved using information from the neighboring subproblems [21]. Many variations have been proposed such as GWASFGA [29], an evolutionary algorithm based on achievement scalarizing function using the Tchebychev method. A recent approach called CDG [1] is also a decomposition-based MOEA. Instead of using a traditional scalarization method such as Tchebycheff, CDG-MOEA uses a constrained decomposition with grids.

We are interested in tackling MOPs using a decomposition of the decision space. In our previous work, we proposed a new metaheuristic for single objective optimization called *Fractal Decomposition Algorithm (FDA)* [23] which is based on hyperspheres Fractals. It is a deterministic metaheuristic developed to solve large-scale continuous optimization problems. As pointed out, the main principle of the approach consists of dividing the feasible search space into sub-regions with the same geometrical pattern. Hyperspheres were chosen as geometrical form because it has the benefit of scaling easily when the dimension of the problem increase. In this article we propose two algorithms that extend FDA to solve MOPs:

- **Scalarization-based approach (Mo-FDA-S):** Mo-FDA-S adapts FDA using scalarization techniques. This approach has also been developed to benefit from a multi-node parallel environment to improve the computational time taken to solve MOPs problems. This chosen architecture benefits from containers, light-weight virtual machines that are designed to run a specific task.
- **Dominance-based approach (Mo-FDA-D):** Mo-FDA-D uses the principle of non-dominated sorting to find an approximation of the Pareto front. Mo-FDA-D proposes both a new hypersphere evaluation technique based on the hypervolume indicator and a new Pareto local search algorithm.

The paper is organized as follow. Section II defines multi-objective optimization. Section III recalls the main principles of the Fractal decomposition metaheuristic FDA. Then, the scalarization-based FDA algorithm is presented in section IV. The dominance-based FDA algorithm is detailed in section V. In section VI the experimental settings and computational results against competing methods are detailed

and analyzed. Finally, the section VII concludes and presents some future works.

## II. MULTI-OBJECTIVE OPTIMIZATION

*Definition 1 (MOP):* A multi-objective optimization problem (MOP) may be defined as:

$$(MOP) = \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{s.t. } x \in X \end{cases} \quad (1)$$

where  $k$  ( $k \geq 2$ ) is the number of objectives,  $x = (x_1 \dots, x_n)$  is the vector representing the decision variables, and  $X$  represents the set of feasible solutions associated with equality and inequality constraints, and explicit bounds.  $F(x) = (f_1(x), f_2(x) \dots, f_k(x))$  is the vector of objectives to be optimized. In this article, we considered the case of continuous optimization: the decision variables used in the objective functions are required to be continuous variables ( $X \subseteq R$ ).

The set of all values satisfying the constraints defines the *feasible region*  $X$  and any point  $\vec{x} \in X$  is a *feasible solution*. As mentioned before, we seek for the *Pareto optima*.

*Definition 2 (Pareto):* A point  $\vec{x}^* \in X$  is Pareto Optimal if for every  $\vec{x} \in X$  and  $I = \{1, 2, \dots, k\} \forall i \in I (f_i(\vec{x}) \geq f_i(\vec{x}^*))$  and there is at least one  $i \in I$  such that  $f_i(\vec{x}) > f_i(\vec{x}^*)$ .

This definition states that  $\vec{x}^*$  is Pareto optimal if no feasible vector  $\vec{x}$  exists which would improve some criterion without causing a simultaneous worsening in at least one other criterion.

*Definition 3 (Dominance):* A vector  $\vec{u} = (u_1, \dots, u_n)$  is said to dominate  $\vec{v} = (v_1, \dots, v_n)$  (denoted by  $\vec{u} \preceq \vec{v}$ ) if and only if  $\vec{u}$  is partially less than  $\vec{v}$ , i.e.,  $\forall i \in \{1, \dots, n\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$ .

*Definition 4 (Pareto Set):* For a given MOP  $\vec{f}(\vec{x})$ , the Pareto optimal set is defined as  $\mathcal{P}^* = \{\vec{x} \in X | \neg \exists \vec{x}' \in X, \vec{f}(\vec{x}') \preceq \vec{f}(\vec{x})\}$ .

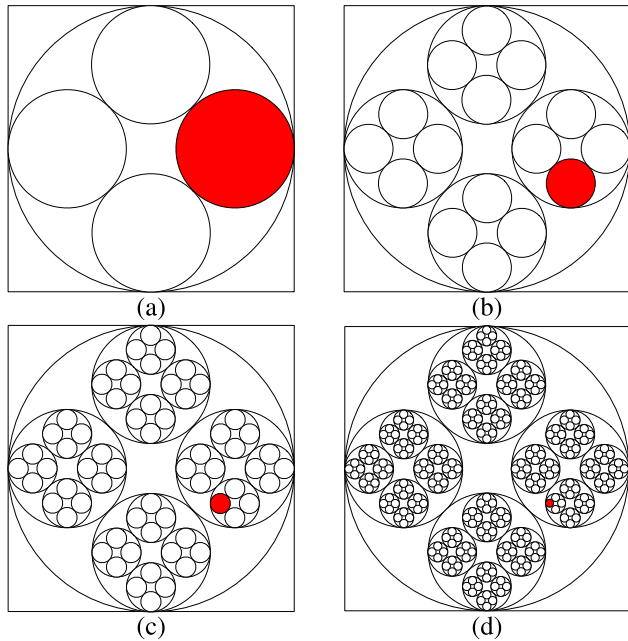
*Definition 5 (Pareto Front):* For a given MOP  $\vec{f}(\vec{x})$  and its Pareto optimal set  $\mathcal{P}^*$ , the Pareto front is defined as  $\mathcal{PF}^* = \{\vec{f}(\vec{x}), \vec{x} \in \mathcal{P}^*\}$ .

*Definition 6 (Reference Point):* A *reference point*  $z^* = [\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n]$  is a vector which defines the aspiration level (or goal)  $\bar{z}_i$  to reach for each objective  $f_i$ .

*Definition 7 (Nadir Point):* A point  $y^* = (y_1^*, y_2^*, \dots, y_n^*)$  is the *nadir point* if it maximizes each objective function  $f_i$  of  $F$  over the Pareto set, i.e.  $y_i^* = \max(f_i(x)), x \in \mathcal{P}^*, i \in [1, n]$ .

## III. FRACTAL DECOMPOSITION ALGORITHM: A RECALL

The Fractal Decomposition Algorithm [23] (FDA) is a divide-and-conquer based algorithm that has been designed to solve large-scale single objective continuous optimization problems. FDA builds a search tree of promising optimum areas of a depth  $k$  (called Fractal depth), by dividing the search space recursively using geometrical hyperspheres. Once FDA divided the search space using hyperspheres as an elementary geometric form, then, a search tree is built. While, navigating through the tree FDA identifies, at each decomposition level, candidate hyperspheres: areas where the global optimum



**FIGURE 1.** Illustration of the Fractal decomposition of the search space: here, the depth of the decomposition is equal to 4. (a) 1<sup>st</sup> level, (b) 2<sup>nd</sup> level, (c) 3<sup>rd</sup> level, (d) 4<sup>th</sup> level.

could be found. This principle is illustrated in Fig. 1 with in case of a four-level decomposition, the red hypersphere being the best one at each level. Once the maximum fractal depth  $k$  is reached, a local search is triggered into promising hyperspheres to find the optimal solution.

Three main phases compose the algorithm: 1) The initialization phase; 2) The exploration phase and 3) The exploitation phase. The first phase aims to initialize the biggest hypersphere possible at the center of the search space and within its boundaries (Upper and lower bound). Once the first hypersphere is set, FDA starts the exploration phase. The algorithm will decompose the first hypersphere into  $2 \times D$  sub-hyperspheres, and apply the *promising hypersphere selection procedure* to all of them. It was designed to find the most promising regions. To do so, their attractiveness is approximated as in [23]. Then, all sub-hyperspheres are sorted by their scores, and the best one is selected to be further decomposed. The whole procedure is called at each level until the maximum fractal depth  $k$  is reached.

The exploitation phase starts when FDA has reached the maximum depth  $k$ . The *Intensive Local Search* (ILS) starts to explore intensively the sub-hyperspheres. ILS starts at the center of each sub-hypersphere and moves along each dimension sequentially, evaluating two solutions  $\vec{x}^{s1}$  and  $\vec{x}^{s2}$  as expressed in (2) and (3), respectively.

$$\vec{x}^{s1} = \vec{x}^s + \omega \times \vec{e}_i \tag{2}$$

$$\vec{x}^{s2} = \vec{x}^s - \omega \times \vec{e}_i \tag{3}$$

where  $\vec{e}_i$  is the unit vector where the  $i^{th}$  element is set to 1, and other elements to 0.  $\omega$  is the step-size in which  $\vec{e}_i$  changes. The best solution among  $\vec{x}^s$ ,  $\vec{x}^{s1}$  and  $\vec{x}^{s2}$  is chosen to be the

next current solution  $\vec{x}^s$ , then, moves to the next dimension. Once all dimensions have been exploited, if no improvement has been found on the current best solution found, the  $\omega$  is reduced by a factor  $\frac{1}{\lambda}$ . ILS stops where either the stopping criterion is reached or the  $\omega$  has reached the tolerance threshold  $\omega_{min}$ . Once all the sub-hyperspheres have been visited by ILS, FDA stops if the stopping criterion was reached or backtracks in the search tree and select the next sub-hypersphere of the level  $k - 1$  creating a new branch in the search tree. The whole algorithm is presented in Algorithm 1. The parameters of FDA are: the Fractal depth  $k = 5$ , the coefficient step size  $\lambda = 0.5$ , the inflation coefficient  $\alpha = 1.75$  and the tolerance threshold  $\omega_{min}$ .

#### IV. SCALARIZATION-BASED FRACTAL DECOMPOSITION

The aggregation (or weighted) method is one of the most popular scalarization method for the generation of Pareto optimal solutions. It consists in using an aggregation function to transform a *MOP* into a single objective problem ( $MOP_\lambda$ ) by combining the various objective functions  $f_i$  into a single objective function  $f$  generally in a linear way:

$$f(x) = \sum_{i=1}^k \omega_i f_i(x), \quad x \in S \tag{4}$$

where the weights  $\omega_i \in [0..1]$  and  $\sum_{i=1}^k \omega_i = 1$ .

The first proposed scalarization approach Mo-FDA-S (Multi-Objective Fractal Decomposition Algorithm Scalarization) uses the Tchebychev function [22]. It introduces the concept of ideal point or reference point  $z_i^*$  as follows:

$$\begin{aligned} & \text{Minimize} \quad \max_{i=1, \dots, k} [\omega_i (f_i(x) - z_i^*)] \\ & \text{Subject to} \quad x \in X \end{aligned} \tag{5}$$

where  $z^* = (z_1^*, \dots, z_k^*)$  is the reference point, and  $\omega = (\omega_1, \dots, \omega_k)$  is the weight vector.

By using  $N$  different weight vectors  $\omega$ , Mo-FDA-S solves  $N$  different problems, each generating one solution composing the final Pareto Front (PF). One of the downsides of using scalarization methods is that the number of solutions composing the PF found by the algorithm will be, at most, the same as the number of different weight vectors  $N$ . In certain cases, if two or more weight vectors  $\omega$  are too close, the algorithm might find the same solution.

#### A. IMPLEMENTATION ON PARALLEL ARCHITECTURE

This approach take profit intrinsically from parallel architectures. Indeed, the algorithm is launched  $N$  times with  $N$  variations of the weight vector  $\omega$ . If we consider that the number of function evaluations (FE) is used as a stopping criterion, even though, each instance only has  $\frac{Max_{fe}}{N}$  FE, the computational time can increase significantly. To overcome this, a multi-node architecture has been developed for Mo-FDA-S. The idea is to have each node finding one solution corresponding to one combination of the weights  $\omega$  and combine all their results to build the final Pareto front.

**Algorithm 1** FDA Algorithm

---

**Input:** Deep of the fractal decomposition:  $k = 5$  and precision threshold:  $\omega_{\min} = 1 \times e^{-20}$

**Input:** Coefficient step-size:  $\lambda = 0.5$ , inflation coefficient:  $\alpha = 1.75$  and dimension of the problem:  $D$

Initialize the center  $\vec{C}$  of the first Hypersphere, at the center of the search space.

**while** *Stopping criterion is not reached* **do**

Partition the current hypersphere  $H$  into  $2 \times D$  sub-hyperspheres

**for**  $2 \times D$  *l-level hypersphere* **do**

Relax hypersphere by  $\alpha$  and compute its quality

**end**

Sort the  $2 \times D$  hyperspheres at the current  $l$ -level by their quality

Replace the current hypersphere  $H$  by the first of the sorted hyperspheres at the current level

**if**  $l == k$  **then**

// last level reached

**for**  $2 \times D$  *hyperspheres at the last level* **do**

Apply the ILS heuristics on each created hypersphere

**end**

**if** *stopping criterion is not reached* **then**

Move up one level ( $l = l - 1$ )

**end**

**end**

**else**

Go to next level:  $l = l + 1$

**end**

**end**

**Result:** the best solution  $BestSol$  and its coordinates

---

The challenge behind this architecture is that the computing resources needed increase with the size of the Pareto-Front. For instance, if  $N = 100$ , it means that 100 nodes would be required, hence 100 different computers (or virtual machines), which can be seen as an oversized architecture. To tackle this important issue we have decided to develop the approach using *containers* and specifically the powerful combination of *docker* as the container technology with *kubernetes* as an orchestrator as shown on Fig. 3. Containers are significantly lighter than virtual machines as they all share the same operating system kernel. This way, a single machine can host more containers than virtual machines. This architecture is significantly lighter than a traditional one and allows to benefit from multi-node approaches while developing it on a limited number of hosts. In addition, containers can be deployed on multiple different physical (or virtual) machines seamlessly, without having to change the structure of our algorithm. Kubernetes is the leading open-source solution for container-orchestration and takes care, in our case, of the creation and deployment of all the containers on the different hosts without changing anything in the algorithm implementation.

Table 1 and Fig. 2 show the obtained computation time. This example considers the time to solve a function in dimension  $D = 30$  with 100 different weights vectors  $\omega$  on different

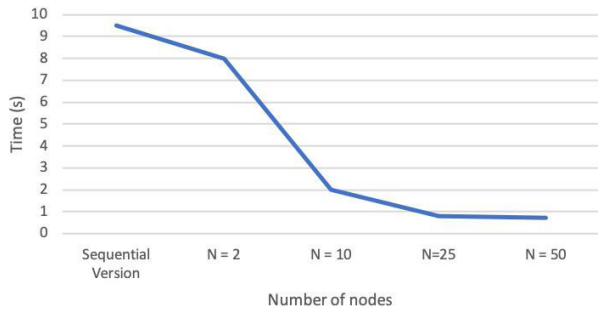
number of virtual hosts  $N$ . It is important to indicate that even on two hosts, computation gain can be observed, however not as important as when the number of hosts increases. This is because each host has to handle, in this case, 50 different containers. Moreover, when  $N$  increases significantly, the gain in time is significant compared to the sequential version but at some point, the increase in compute nodes does not decrease the computational time. This is due to the communication overhead required to synchronise all nodes and gather all points composing the PF. All tests have been done on a cluster of machines with the following characteristics: one 3.1 GHz Intel Xeon® Platinum 817 processor with 16GB of RAM. Moreover, Mo-FDA-S has been developed in Python and uses the library MPI (Message Passing Interface).

## V. DOMINANCE-BASED FRACTAL DECOMPOSITION

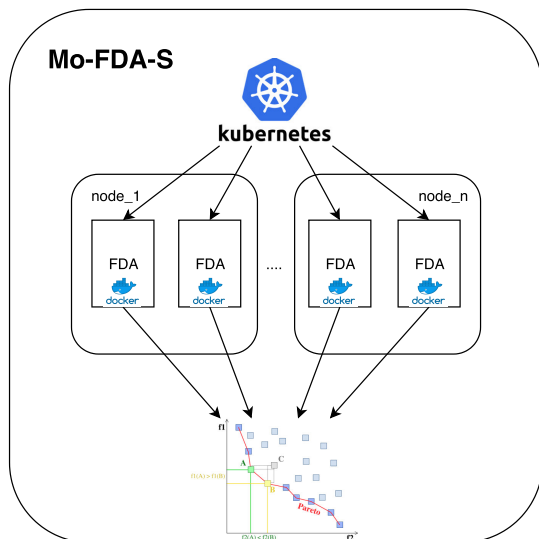
The idea behind Mo-FDA-D (Multi-Objective Fractal Decomposition Algorithm Dominance Based) is to keep the structure of the framework provided by the original version of FDA [23], i.e. the geometric fractal decomposition as well as the geometric form and the different phases composing the original version of FDA. The procedure to evaluate hyperspheres and the ILS heuristic to conduct the local search at the maximal fractal depth  $k$  have been extended to solve MOPs.

**TABLE 1.** An example of computation time required to solve a function with Mo-FDA-S with different number of physical nodes  $N$  for 100 instances of FDA, hence 100 points in the Pareto Front.

Number of Nodes	Time (in seconds)
Sequential Version	9.5
$N = 2$	8
$N = 10$	2
$N = 25$	0.8
$N = 50$	0.6



**FIGURE 2.** Computation time function of the number of physical nodes  $N$  for 100 instances of FDA.



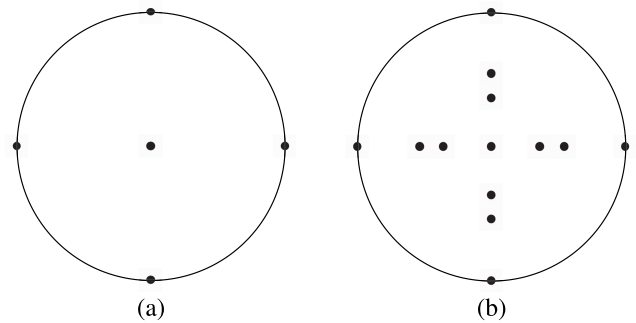
**FIGURE 3.** The architecture of Mo-FDA-S using containers on  $N$  different nodes.

**A. MULTI-OBJECTIVE HYPERSPHERE SELECTION (EXPLORATION STRATEGY)**

This procedure aims to select the most promising region that might contains Pareto solutions. The aim is to find both the most promising region to be further decomposed but also to find potential non-dominated points composing the final Pareto Front (PF). To do so, we evaluated multiple points along each dimension using the following equations:

$$\vec{s} = \vec{C}^l \pm \frac{r_l}{\gamma} \times \vec{e}_i \quad \text{for } i = 1, 2, \dots, D \quad (6)$$

where  $\vec{C}^{(l)}$  is the coordinates of the center of hypersphere being evaluated,  $r$  is its radius,  $\gamma \in [1, 3]$  and  $\vec{e}_i$  is the unit



**FIGURE 4.** Evaluating (a)  $2 \times D$  points plus the center with  $\gamma = 1$  and (b)  $6 \times D$  points plus the center with  $\gamma = \{1, 2, 3\}$ .

vector at the dimension  $i$ . This is illustrated by 4 with a two-dimensional example by two different scenarii:

- (a)  $2 \times D$  points plus the center are evaluated with  $\gamma = 1$ , meaning that the points are on the sphere.
- (b)  $6 \times D$  points plus the center are evaluated with  $\gamma = \{1, 2, 3\}$ . Solutions are on and within the hypersphere.

All evaluated solutions are stored in a temporary list to be sorted. The sorting is based on the Pareto dominance and only non-dominated points are kept, producing a local Pareto front of locally non-dominated solutions within the hypersphere. The sorting algorithm used to sort evaluated points and generate the local PF is called *Simple Cull* [11]. Once the local PF obtained, all points are compared to the *Nadir point* of the objective space and all points above are excluded from the PF.

We have used the hypervolume indicator as a quality indicator to evaluate the sub-hyperspheres. To do so, we compute the hypervolume of the local PF with regards to the *Nadir point*,  $z^{nad}$  [26]. The hypersphere with the highest value is considered better than the other hyperspheres of the same level and will, therefore, be selected to be further decomposed. Once all the hyperspheres of a given level have been evaluated, all the locally non-dominated sets are concatenated and sorted again to find the global PF of non-dominated solutions.

**B. MULTI-OBJECTIVE INTENSIVE LOCAL SEARCH**

Once the fractal depth  $k$  is reached, the intensive local search (ILS) is triggered. In this context, it is important to notice that any multi-objective local search or metaheuristic could be used at this step. Instead of searching locally within hyperspheres of the last level, ILS iterates around each non-dominated solutions found so far during the exploration phase of evaluating hyperspheres. Therefore, the entry point of one ILS instance is one solution of the current global Pareto Set.

ILS starts by creating two empty lists, one for the Pareto Set *listNewPS* (decision space) and one for their corresponding solutions in the Pareto front denoted *listNewPD* (objective space) and insert in the first list the point given as input parameter. Then for each dimension and each point in *listNewPS*, ILS will produce two additional points denoted  $\vec{x}_L$  and  $\vec{x}_R$ . They stand in opposite directions from the current

point being exploited,  $\vec{x}_C$  at equal distance  $\omega$ , also called step size as per the following equations:

$$\vec{x}_L = \vec{x}_C + \omega \times \vec{e}_i \quad (7)$$

$$\vec{x}_R = \vec{x}_C - \omega \times \vec{e}_i \quad (8)$$

where  $\vec{e}_i$  is the unit vector which the  $i^{\text{th}}$  element is set to 1 and the other elements to 0,  $\vec{x}_C$ ,  $\vec{x}_L$  and  $\vec{x}_R$  are then evaluated and inserted in *listNewPS* and their corresponding solutions in the objective space, respectively  $F(\vec{x}_C)$ ,  $F(\vec{x}_L)$  and  $F(\vec{x}_R)$  are added to the *listNewPF* list. Once all the points in *listNewPS* have been exploited for the current dimension, the non-dominated sorting algorithm is applied to the list of all potential solutions (*listNewPF*) and this will generate a new local Pareto Front of non-dominated solutions. Hence *listNewPF* only contains a set of non-dominated solutions and *listNewPS* only contains their equivalent in the decision space. At the end of each iteration, once all dimensions have been searched,  $\omega$  is multiplied by a coefficient defined as a hyperparameter of Mo-FDA-D. This is repeated until either:

- the stopping criterion is reached;
- or  $\omega$  has reached its minimum value  $\omega_{min}$  and therefore ILS moves on to the next point in the Pareto set.

Once ILS has finished searching around each point of the Pareto set, all points found during ILS are sorted and only the non-dominated points will remain and will compose the new global Pareto-Set. Either the stopping criterion is reached and Mo-FDA-D has finished or the backtracking procedure is applied and a new sphere from the level  $k - 1$  is selected to be decomposed. The whole procedure is illustrated in Algorithm 2.

## VI. PERFORMANCE ANALYSIS

In this section, the two proposed algorithms, Mo-FDA-S and Mo-FDA-D, are analyzed and their performance is assessed using different benchmarks (ZDT [34], DTLZ [4], Fonseca-Fleming [9]). In the literature different metrics have been developed to measure the quality of the Pareto sets obtained by different algorithms [28]. Each metric measures a different characteristic of a Pareto front [27]:

- *Convergence* (or accuracy), i.e. the closeness (i.e distance) to the optimal or best known Pareto front;
- *Cardinality*, i.e. the number of points in the Pareto front;
- *Diversity*, i.e. the distribution of the front. The points in a Pareto front should be well spread over the objective space.

We have chosen to focus on the four most commonly used metrics evaluating all aspects of a Pareto front:

- *The Hypervolume* computes the volume of the objective space that is dominated by a reference point [35].
- *The Generational Distance* metric (GD), computes the average distance from the obtained set to the true Pareto front [28].
- *The Inverted generational distance* (IGD), measures both convergence and diversity by computing the

## Algorithm 2 Mo-FDA-D - ILS Procedure

**Input:**  $\omega_{min} = 10^{-5}$ . //precision or tolerance error

**Input:** Coefficient step-size:  $\lambda$

**Input:**  $D$  // the dimension of the problem

**Input:** Number of function evaluations  $NBEval$

**Input:** The first point to search as starting point *startingPoint*

Set the step size  $\omega$  to the radius of a  $k_{th}$  level hypersphere  $H$

Set an empty list for Non-dominated Points Coordinates *listNewPS*

Add *startingPoint* to *listNewPS*

Set an empty for Non-dominated Points Solutions *listNewPF*

**while**  $\omega \geq \omega_{min}$  **do**

**for** Each dimension  $i = 1, \dots, D$  **do**

**foreach** *currentPoint*  $\in$  *listNewPS* **do**

      set  $\vec{x}_C = \text{currentPoint}$

$\vec{x}_L = \vec{x}_C - \omega \times \vec{e}_i$

$\vec{x}_R = \vec{x}_C + \omega \times \vec{e}_i$

      Evaluate the fitness of  $\vec{x}_C$ ,  $\vec{x}_L$  and  $\vec{x}_R$

$NBEval = NBEval + 3$

      Add  $F(\vec{x}_C)$ ,  $F(\vec{x}_L)$  and  $F(\vec{x}_R)$  to *listNewPF*

**end**

    Sort *listNewPF* to leave only the non-dominated solutions

    Modify *listNewPS* so it contains only the coordinates of the non-dominated solutions

**end**

  Decrease the step size  $\omega$ :  $\omega = \omega \times \lambda$ .

**end**

**Output:** *listNewPS* and *listNewPF*

distance from each point known in the true Pareto-Front to each point to the obtained set.

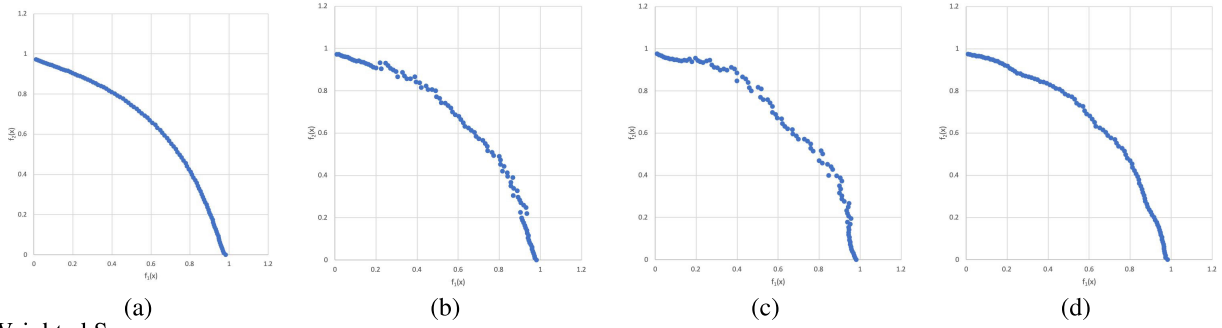
- *The Spread* measures how well spread the non-dominated solutions are over the objective space [6].

The following sections carry out the sensitivity analysis of the proposed multi-objective FDA algorithms to their parameters, and a comparison to some popular multi-objective evolutionary algorithms (e.g. NSGA-II, MOEAD/D).

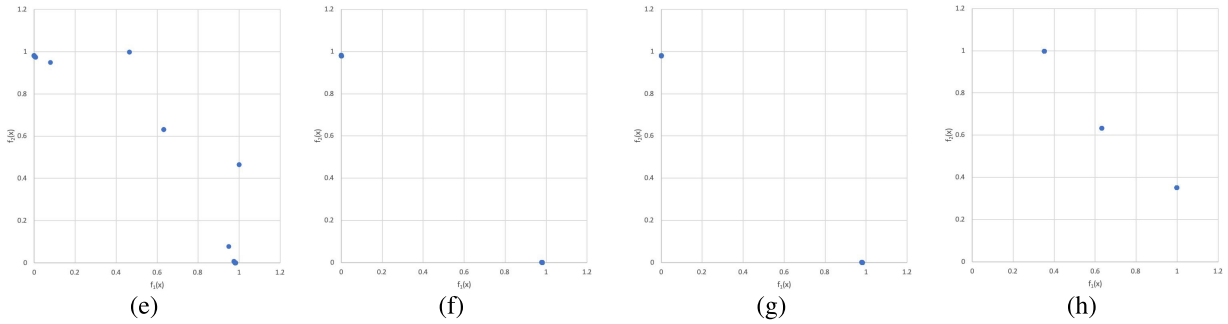
### A. IMPACT OF THE SCALARIZATION METHOD

As Mo-FDA-S is based on the original version of FDA, the sensitivity analysis with regards to its parameters can be found in [24]. However two scalarization methods have been used to study their impact: the Weighted Sum and the Tchebychev methods. The stopping criterion has been set

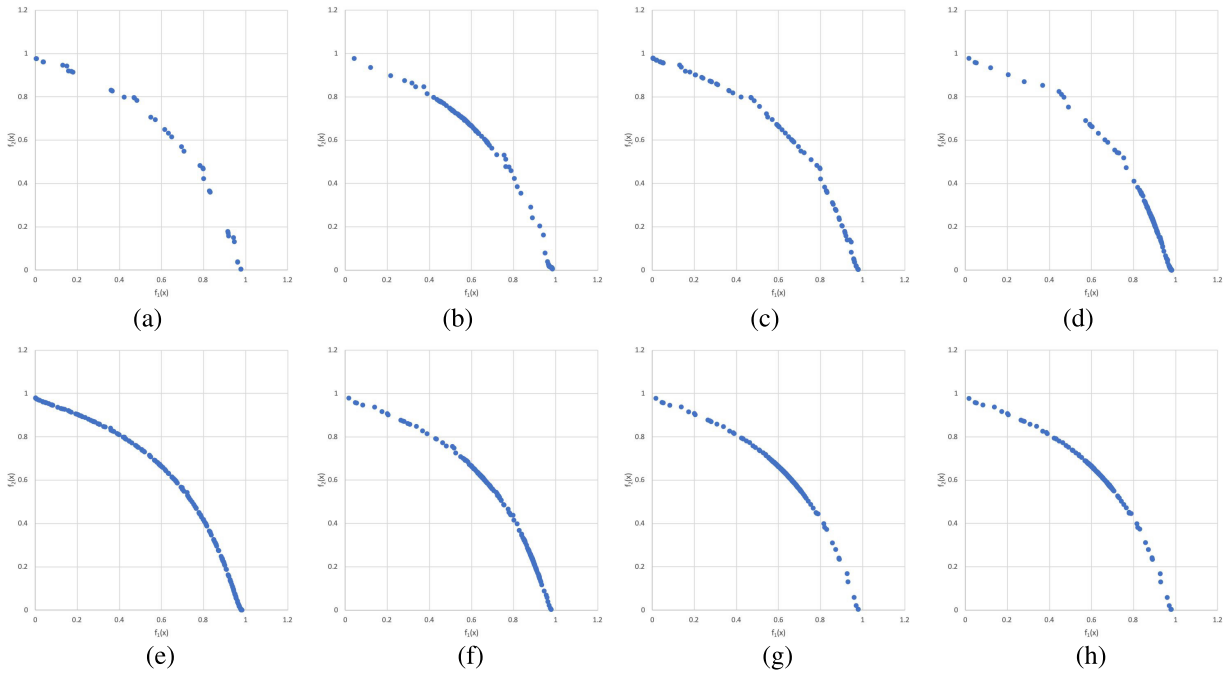
Tchebychev Sum



Weighted Sum



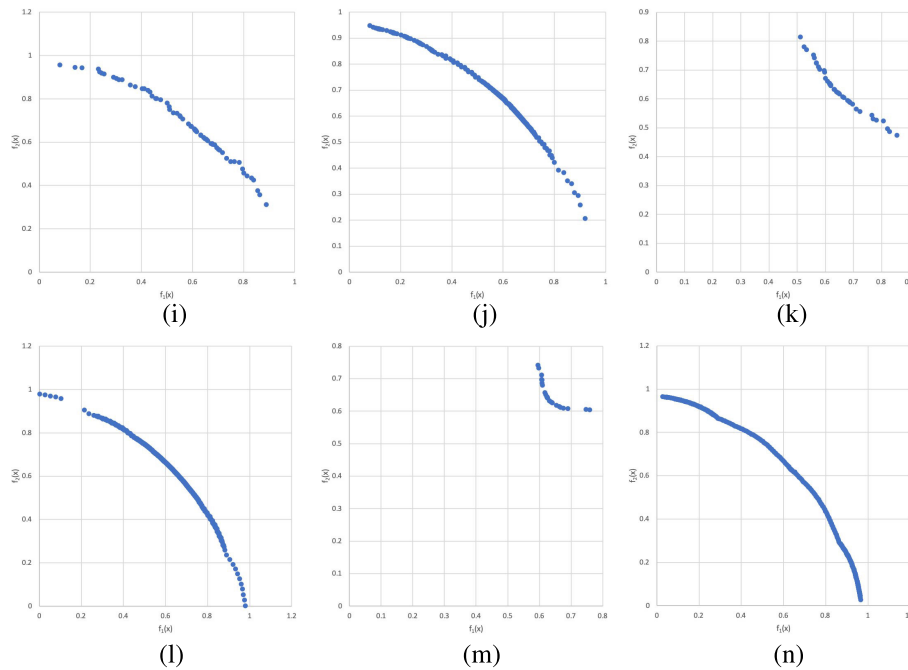
**FIGURE 5.** Pareto fronts for the two studied scalarization methods on 4 different dimensions using Tchebychev function (a)  $D = 2$ , (b)  $D = 5$ , (c)  $D = 10$ , (d)  $D = 30$ , and weighted sum (e)  $D = 2$ , (f)  $D = 5$ , (g)  $D = 10$ , (h)  $D = 30$ .



**FIGURE 6.** Obtained Pareto fronts for different sets of paramters (*nb. k, ILS*): (a) Case 1 (2,5,no ILS), (b) Case 2 (2, 5, ILS), (c) Case 3 (6, 5, no ILS), (d) Case 4 (6, 5, ILS), (e) Case 5 (6, 8, no ILS), (f) Case 6 (6, 8, ILS), (g) Case 7 (6, 16, no ILS), (h) Case 8 (6, 16, ILS).

up to  $5000 \times D$  and  $10^{-5}$  as precision tolerance  $\omega_{min}$ . Using the Hypervolumes in Table 2, it is obvious that the Tchebychev method much more efficient than the aggregation method. The same results has been obtained for all ZDT,

DTLZ, and Fonseca–Fleming benchmarks. As an illustration, Fig. 5 shows the obtained results for the “Fonseca–Fleming” benchmark for different dimensions [9]. Besides, it is interesting to note that Mo-FDA-S works well at low dimensions



**FIGURE 7.** Pareto front for a large scale problem: (i) Case 9 ( $D = 5$ , no ILS), (j) Case 10 ( $D = 5$ , ILS), (k) Case 11 ( $D = 10$ , no ILS), (l) Case 12 ( $D = 10$ , ILS), (m) Case 13 ( $D = 30$ , no ILS), (n) Case 14 ( $D = 30$ , ILS).

**TABLE 2.** Hypervolumes of the two studied scalarization methods on different dimensions.

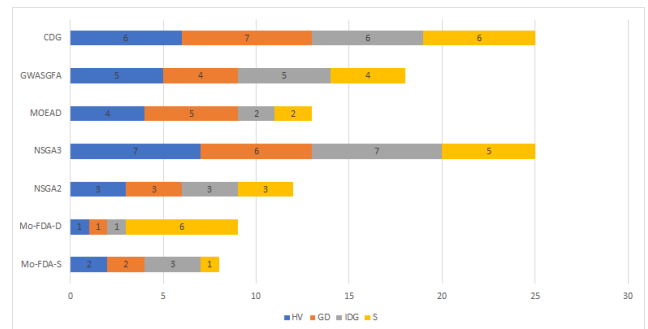
Dimension D	Weighted Sum	Tchebychev
2	0.195	0.336
5	0.042	0.316
10	0.042	0.294
30	0.13	0.315

and high dimensions but is less efficient on intermediate dimensions.

### B. IMPACT OF DOMINANCE BASED EXPLORATION AND EXPLOITATION

The dominance based exploration and exploitation of hyperspheres Fractals is analyzed with regards to three parameters: the number of solutions evaluated in the hyperspheres  $nb$ , the fractal depth  $k$ , and the step size  $\lambda$  by which  $\omega$  is multiplied in ILS. It is important to highlight the fact that in the different cases, the common criteria are the number of function evaluations set up to  $5000 \times D$  as stopping criterion and  $10^{-5}$  as precision tolerance  $\omega_{min}$ .

Using DoE (Design of Experiments) methodology, 8 scenarios of values have been selected for this set of parameters. All those scenarios are illustrated in Fig. 6 and results are shown in Table 3 for a small problem (i.e.  $D = 2$ ). For small problems, ILS tends to concentrate the Pareto front around one some area and therefore penalizes the diversification of the Pareto front. Going to deep hyperspheres (i.e.  $k = 16$ ) will decrease the performances of the algorithm in



**FIGURE 8.** Illustration of the final ranking of the algorithms for 2-Objective functions for each metric.

terms of quality, whether ILS is used or not. However, using ILS increases significantly the computing time. Parameters values maximizing the quality (i.e. hypervolume) lead to an increase in the computing time and vice versa. For small size problems, the best configuration for the set of parameters is ( $nb = 6, k = 8, no ILS$ ).

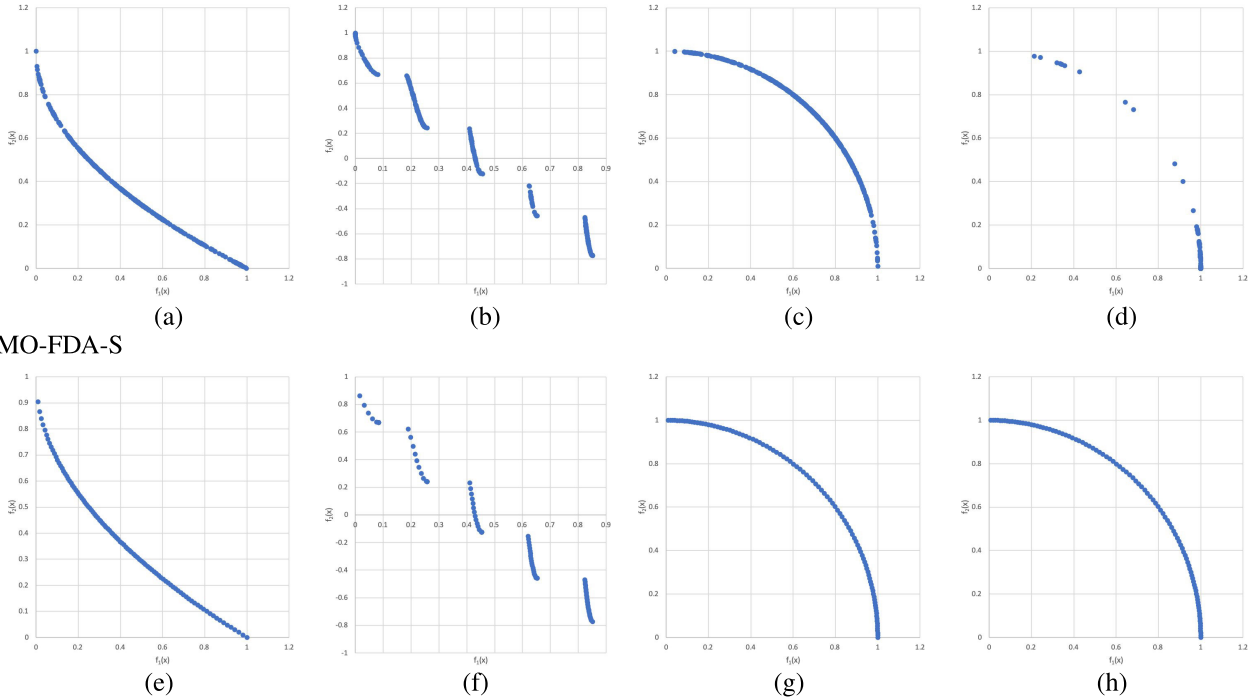
To analyse the results for large scale problems, the selected benchmark ZDT and DTLZ are scaled to dimension  $D = 30$ . The Pareto sets of the different cases are shown in Figure 7 and the quantitative results are shown in Table 4. Those results highlight the important fact that the use of ILS becomes essential when the dimension of the problem increases.

### C. COMPARISON WITH STATE-OF-THE-ART ALGORITHMS

A comparison of our developed approaches has been carried out with some popular multi-objective evolutionary



MO-FDA-D



**FIGURE 9.** Pareto fronts for 4 selected benchmarks for our two approaches. With MO-FDA-D (a) ZDT1, (b) ZDT3, (c) DTLZ2, (d) DTLZ4. With MO-FDA-S (e) ZDT1, (f) ZDT3, (g) DTLZ2, (h) DTLZ4.

**TABLE 3.** Hypervolumes, ranks and computing time for the different studied Case from 1 to 9.

Case	Hypervolume	Rank	Computing Time (in s)
Case 1	0.308078317	8	0.58
Case 2	0.321532937	6	1.24
Case 3	0.326594895	5	1.06
Case 4	0.321305318	7	1.81
Case 5	0.337433922	1	4.95
Case 6	0.332881847	2	1.63
Case 7	0.329591937	3	13.89
Case 8	0.329558587	4	1.82

algorithms (MOEAs). We have considered the 5 following MOEA algorithms:

- NSGA-II: a MOEA based on a non dominated sorting approach [6].
- NSGA-III: an extension of NSGA-II adapted to solve many-objective problems, i.e. more than 3 objectives. It works with a set of supplied or predefined reference points aiming to maintain the diversity among population members [5].
- MOEA/D-DE: a decomposition-based approach which uses scalarization to transform the MOP into a single-objective problem [17]. The different scalar optimization sub-problems are optimized simultaneously. The Tchebychev method is used as a scalarization method.

**TABLE 4.** Hypervolumes, ranks and computing time for the different parameters. The other parameters have been fixed to  $nb = 6$  points evaluation for hypersphere ( $\gamma = \{1, 2, 3\}$ ) and  $k = 8$ .

Dimension D	ILS use	Hypervolume	Time (s)
5	no ILS	0.285	0.84
5	ILS	0.315	9.47
10	no ILS	0.208	1.34
10	ILS	0.328	22.68
30	no ILS	0.155	6.46
30	ILS	0.326	15.13

- GWASFGA: a global weighting achievement scalarizing function genetic algorithm [29]. This algorithm is also based on a scalarization method and uses an achievement scalarizing function which is based on the Tchebychev method but includes the use of the Utopian and the Nadir points.
- CDG: a decomposition-based MOEA. Instead of using a traditional scalarization method such as Tchebychev, CDG-MOEA uses a constrained decomposition with grids [1]. One objective function is selected to be optimized while the other objective functions are converted into constraints by setting up the upper and lower bounds.

All experiments on the competing algorithms have been done using the framework jMetal 5.0 [7], [25]. Settings for the algorithms have been set according to [15] as well as

**TABLE 5.** Mean and Standard deviation values for the hypervolume metric on all 2D functions with ranks in parentheses.

		Mo-FDA-S	Mo-FDA-D	NSGA2	NSGA3	MOEAD	GWASGFA	CDG
ZDT1	Mean	6.62E-01 (2)	<b>6.63E-01 (1)</b>	6.60E-01 (4)	6.24E-01 (7)	6.61E-01 (3)	6.58E-01 (6)	6.60E-01 (5)
	Std.	0.00E+00	0.00E+00	2.53E-04	1.56E-03	3.72E-05	1.02E-04	3.03E-04
ZDT2	Mean	3.29E-01 (2)	<b>3.30E-01 (1)</b>	3.27E-01 (4)	2.95E-01 (7)	3.28E-01 (3)	3.24E-01 (6)	3.27E-01 (5)
	Std.	0.00E+00	0.00E+00	2.64E-04	5.82E-04	5.67E-05	2.38E-05	2.55E-04
ZDT3	Mean	7.94E-01 (4)	<b>7.97E-01 (1)</b>	7.96E-01 (2)	7.77E-01 (7)	7.94E-01 (5)	7.91E-01 (6)	7.94E-01 (3)
	Std.	0.00E+00	0.00E+00	1.55E-04	2.58E-03	1.86E-05	2.14E-04	1.19E-03
ZDT4	Mean	6.62E-01 (2)	<b>6.63E-01 (1)</b>	6.59E-01 (3)	5.69E-01 (6)	6.56E-01 (5)	6.57E-01 (4)	1.79E-02 (7)
	Std.	0.00E+00	0.00E+00	7.84E-04	5.26E-02	2.40E-03	3.33E-04	2.49E-02
ZDT6	Mean	3.19E-01 (4)	3.18E-01 (6)	3.20E-01 (2)	2.98E-01 (7)	<b>3.22E-01 (1)</b>	3.19E-01 (5)	3.20E-01 (3)
	Std.	0.00E+00	0.00E+00	2.22E-04	5.16E-04	6.74E-04	4.89E-05	1.39E-03
DTLZ1	Mean	8.71E-01 (4)	8.71E-01 (3)	<b>8.72E-01 (1)</b>	2.94E-02 (6)	8.70E-01 (5)	8.71E-01 (2)	0.00E+00 (7)
	Std.	0.00E+00	0.00E+00	3.77E-04	1.48E-01	4.92E-03	5.37E-04	0.00E+00
DTLZ2	Mean	2.10E-01 (2)	<b>2.13E-01 (1)</b>	2.09E-01 (4)	1.81E-01 (7)	2.10E-01 (3)	2.06E-01 (6)	2.08E-01 (5)
	Std.	0.00E+00	0.00E+00	1.99E-04	8.10E-04	1.72E-05	1.12E-04	2.68E-04
DTLZ3	Mean	2.10E-01 (2)	<b>2.13E-01 (1)</b>	2.04E-01 (3)	0.00E+00 (6)	1.98E-01 (5)	2.02E-01 (4)	0.00E+00 (6)
	Std.	0.00E+00	0.00E+00	2.52E-03	0.00E+00	8.62E-03	1.17E-03	0.00E+00
DTLZ4	Mean	<b>2.10E-01 (1)</b>	1.70E-01 (5)	1.53E-01 (6)	4.22E-02 (7)	2.10E-01 (2)	1.99E-01 (4)	2.08E-01 (3)
	Std.	0.00E+00	0.00E+00	9.41E-02	7.78E-02	2.59E-05	3.76E-02	3.91E-04

**TABLE 6.** Mean and Standard deviation values for the GD metric on all 2D functions with ranks in parentheses.

		Mo-FDA-S	Mo-FDA-D	NSGA2	NSGA3	MOEAD	GWASGFA	CDG
ZDT1	Mean	4.28E-05 (2)	<b>1.86E-05 (1)</b>	1.97E-04 (4)	3.16E-04 (6)	9.67E-05 (3)	2.24E-04 (5)	2.02E-03 (7)
	Std.	0.00E+00	0.00E+00	4.03E-05	2.90E-04	6.48E-06	4.69E-07	1.38E-03
ZDT2	Mean	4.87E-05 (3)	<b>1.91E-05 (1)</b>	1.25E-04 (5)	2.45E-04 (6)	4.54E-05 (2)	5.76E-05 (4)	2.03E-03 (7)
	Std.	0.00E+00	0.00E+00	3.43E-05	2.90E-04	1.42E-06	7.42E-07	1.60E-03
ZDT3	Mean	2.72E-03 (6)	<b>9.36E-05 (1)</b>	1.94E-04 (4)	1.20E-03 (5)	1.54E-04 (2)	1.73E-04 (3)	8.01E-03 (7)
	Std.	0.00E+00	0.00E+00	1.64E-05	1.99E-03	1.36E-06	1.40E-06	1.87E-03
ZDT4	Mean	4.35E-05 (2)	<b>1.92E-05 (1)</b>	1.40E-04 (3)	1.19E-02 (6)	4.54E-04 (5)	1.41E-04 (4)	4.35E-01 (7)
	Std.	0.00E+00	0.00E+00	4.92E-05	1.13E-02	1.78E-04	1.70E-05	9.43E-01
ZDT6	Mean	3.46E-05 (2)	<b>1.46E-05 (1)</b>	5.72E-05 (3)	1.45E-04 (6)	8.94E-05 (5)	5.97E-05 (4)	6.31E-03 (7)
	Std.	0.00E+00	0.00E+00	3.13E-06	6.08E-06	5.03E-05	2.99E-06	1.99E-02
DTLZ1	Mean	2.06E-05 (2)	<b>7.41E-06 (1)</b>	1.96E-04 (3)	5.83E-01 (6)	5.42E-04 (5)	2.17E-04 (4)	9.28E+00 (7)
	Std.	0.00E+00	0.00E+00	5.21E-05	2.42E-01	6.79E-04	1.04E-04	8.95E-02
DTLZ2	Mean	3.46E-04 (5)	<b>4.23E-05 (1)</b>	2.64E-04 (4)	6.81E-04 (7)	3.49E-04 (6)	2.63E-04 (3)	1.30E-04 (2)
	Std.	0.00E+00	0.00E+00	3.69E-05	3.80E-04	2.23E-06	1.92E-05	1.42E-04
DTLZ3	Mean	2.44E-04 (2)	<b>3.18E-05 (1)</b>	4.46E-04 (3)	1.17E+00 (6)	9.28E-04 (5)	4.29E-04 (4)	1.26E+01 (7)
	Std.	0.00E+00	0.00E+00	1.70E-04	4.32E-01	6.72E-04	1.04E-04	2.85E-01
DTLZ4	Mean	2.44E-04 (7)	2.26E-04 (5)	1.44E-04 (3)	<b>7.97E-05 (1)</b>	2.40E-04 (6)	1.76E-04 (4)	1.45E-04 (2)
	Std.	0.00E+00	0.00E+00	9.41E-05	1.51E-04	9.12E-06	4.67E-05	1.57E-04

default values in jMetal [7]. Population size has been set to  $N = 100$  and the stopping criterion  $Max_{FES} = 300,000$ . As the competing algorithms are stochastic, their results have been averaged over 20 independent runs. As a recall, both Mo-FDA-S and Mo-FDA-D are deterministic algorithms and their results have been obtained after a single run.

As mentioned earlier, we have decided to use a set of 8 functions, 5 from the ZDT family problems and 3 from the DTLZ. The dimension set is  $D = 30$  for both benchmarks. To compare the results obtained by the different algorithms, we used the Friedman Rank sum method to rank the approach based on their performance on each metric and each function.

**TABLE 7.** Mean and Standard deviation values for the IGD metric on all 2D functions with ranks in parentheses.

		Mo-FDA-S	Mo-FDA-D	NSGA2	NSGA3	MOEAD	GWASGFA	CDG
ZDT1	Mean	1.98E-04 (4)	<b>1.19E-04 (1)</b>	1.87E-04 (3)	1.21E-03 (7)	1.60E-04 (2)	2.71E-04 (6)	2.07E-04 (5)
	Std.	0.00E+00	0.00E+00	8.55E-06	3.59E-05	2.94E-07	2.79E-06	1.57E-05
ZDT2	Mean	1.40E-04 (2)	<b>1.39E-04 (1)</b>	1.96E-04 (4)	1.14E-03 (7)	1.41E-04 (3)	2.86E-04 (5)	4.44E-04 (6)
	Std.	0.00E+00	0.00E+00	1.09E-05	1.54E-05	2.22E-08	5.81E-06	3.71E-05
ZDT3	Mean	9.16E-03 (7)	2.78E-04 (2)	<b>2.13E-04 (1)</b>	2.01E-03 (6)	4.71E-04 (3)	7.22E-04 (5)	5.09E-04 (4)
	Std.	0.00E+00	0.00E+00	1.17E-05	4.06E-04	1.30E-06	5.84E-05	1.38E-04
ZDT4	Mean	2.31E-04 (4)	<b>1.24E-04 (1)</b>	1.91E-04 (2)	1.93E-03 (6)	2.23E-04 (3)	2.83E-04 (5)	1.20E-01 (7)
	Std.	0.00E+00	0.00E+00	8.69E-06	1.14E-03	3.69E-05	3.92E-07	2.39E-01
ZDT6	Mean	3.24E-04 (5)	3.72E-04 (6)	1.85E-04 (2)	9.32E-04 (7)	<b>1.18E-04 (1)</b>	2.24E-04 (4)	2.14E-04 (3)
	Std.	0.00E+00	0.00E+00	9.73E-06	1.35E-05	5.79E-06	7.97E-07	4.93E-05
DTLZ1	Mean	6.57E-05 (2)	<b>5.28E-05 (1)</b>	1.13E-04 (3)	6.16E-02 (6)	1.91E-04 (5)	1.41E-04 (4)	2.52E+00 (7)
	Std.	0.00E+00	0.00E+00	9.52E-06	2.60E-02	1.95E-04	9.69E-06	6.46E-04
DTLZ2	Mean	1.72E-04 (2)	<b>8.53E-05 (1)</b>	1.92E-04 (4)	1.41E-03 (7)	1.73E-04 (3)	3.30E-04 (5)	3.32E-04 (6)
	Std.	0.00E+00	0.00E+00	1.18E-05	4.31E-05	1.21E-08	4.98E-06	1.70E-05
DTLZ3	Mean	<b>1.55E-04 (1)</b>	1.93E-04 (2)	2.40E-04 (3)	1.46E-01 (6)	3.32E-04 (4)	3.41E-04 (5)	3.99E+00 (7)
	Std.	0.00E+00	0.00E+00	3.81E-05	5.42E-02	1.97E-04	6.46E-06	9.03E-02
DTLZ4	Mean	<b>1.56E-04 (1)</b>	2.51E-03 (5)	8.58E-03 (6)	2.45E-02 (7)	1.56E-04 (2)	1.36E-03 (4)	7.39E-04 (3)
	Std.	0.00E+00	0.00E+00	1.41E-02	1.31E-02	2.80E-08	5.72E-03	5.83E-05

**TABLE 8.** Mean and Standard deviation values for the Spread metric on all 2D functions with ranks in parentheses.

		Mo-FDA-S	Mo-FDA-D	NSGA2	NSGA3	MOEAD	GWASGFA	CDG
ZDT1	Mean	<b>2.75E-01 (1)</b>	1.15E+00 (7)	3.75E-01 (3)	4.20E-01 (4)	2.82E-01 (2)	8.66E-01 (6)	7.11E-01 (5)
	Std.	0.00E+00	0.00E+00	2.71E-02	2.95E-02	2.17E-04	1.68E-02	1.62E-01
ZDT2	Mean	1.39E-01 (2)	1.10E+00 (7)	3.92E-01 (4)	3.90E-01 (5)	<b>1.35E-01 (1)</b>	2.73E-01 (3)	7.35E-01 (6)
	Std.	0.00E+00	0.00E+00	2.73E-02	1.87E-02	1.40E-04	1.67E-02	1.79E-01
ZDT3	Mean	8.61E-01 (3)	1.11E+00 (5)	<b>5.52E-01 (1)</b>	6.32E-01 (2)	8.74E-01 (4)	1.14E+00(6)	1.20E+00 (7)
	Std.	0.00E+00	0.00E+00	2.64E-02	7.68E-02	2.95E-04	1.84E-02	6.90E-02
ZDT4	Mean	<b>2.75E-01 (1)</b>	1.15E+00 (7)	4.21E-01 (3)	4.90E-01 (4)	2.88E-01 (2)	8.30E-01 (5)	9.40E-01 (6)
	Std.	0.00E+00	0.00E+00	3.71E-02	1.02E-01	4.49E-03	1.03E-01	1.83E-01
ZDT6	Mean	3.01E-01 (2)	1.46E+00 (7)	6.45E-01 (5)	4.71E-01 (3)	<b>1.35E-01(1)</b>	9.30E-01 (6)	6.01E-01 (4)
	Std.	0.00E+00	0.00E+00	4.18E-02	2.05E-02	1.42E-03	3.31E-02	2.62E-01
DTLZ1	Mean	<b>1.07E-02 (1)</b>	1.11E+00 (7)	4.49E-01 (4)	8.41E-01 (5)	3.78E-02 (2)	8.60E-02 (3)	9.38E-01 (6)
	Std.	0.00E+00	0.00E+00	3.44E-02	1.18E-01	5.87E-02	5.73E-02	1.45E-01
DTLZ2	Mean	1.84E-01 (2)	1.11E+00 (7)	3.87E-01 (4)	4.04E-01 (5)	<b>1.81E-01 (1)</b>	2.81E-01 (3)	6.86E-01 (6)
	Std.	0.00E+00	0.00E+00	2.92E-02	2.37E-02	5.74E-05	2.27E-02	3.62E-02
DTLZ3	Mean	<b>1.84E-01 (1)</b>	1.11E+00 (7)	4.13E-01 (4)	8.96E-01 (6)	2.03E-01 (2)	2.24E-01 (3)	8.80E-01 (5)
	Std.	0.00E+00	0.00E+00	2.74E-02	1.25E-01	2.29E-02	4.72E-02	4.42E-02
DTLZ4	Mean	<b>1.83E-01 (1)</b>	1.56E+00 (7)	5.59E-01 (4)	8.94E-01 (6)	1.83E-01 (2)	3.47E-01 (3)	6.76E-01 (5)
	Std.	0.00E+00	0.00E+00	2.72E-01	2.04E-01	2.34E-03	1.29E-01	4.51E-02

1) TWO-OBJECTIVE FUNCTIONS

First we have focused on both the ZDT and DTLZ benchmarks using 2 objective functions. Results from the competing algorithms are shown in Tables 5 to 8. On each table the values in bold highlight the best algorithm for

the given function and the given metric and the value in parentheses represents the rank for the given function. As a recall, the standard deviation for Mo-FDA-S and Mo-FDA-D are equal to zero as they are deterministic algorithms. Mo-FDA-D is regularly ranked first on the first three metrics

**TABLE 9.** Average ranks for each metric and each algorithm over all 2D functions used using the Friedman Rank Sum.

	Mo-FDA-S	Mo-FDA-D	NSGA2	NSGA3	MOEAD	GWASGFA	CDG
HV	2.56	2.22	3.22	6.67	3.56	4.78	4.89
GD	3.44	1.44	3.56	5.44	4.33	3.89	5.89
IDG	3.11	2.22	3.11	6.56	2.89	4.78	5.33
S	1.56	6.78	3.56	4.44	1.89	4.22	5.56

**TABLE 10.** Final ranks based on the Friedman Rank Sum values for the 9 used 2D functions.

	Mo-FDA-S	Mo-FDA-D	NSGA2	NSGA3	MOEAD	GWASGFA	CDG
HV	2	1	3	7	4	5	6
GD	2	1	3	6	5	4	7
IDG	3	1	3	7	2	5	6
S	1	6	3	5	2	4	6
Final Average Rank:	2	2.25	3	6.25	3.25	4.5	6.25

**TABLE 11.** Mean and Standard deviation values for the Hypervolume metric on all 3D functions with ranks in parentheses.

		Mo-FDA-D	NSGA2	NSGA3	GWASGFA	CDG
DTLZ1	Mean	9.46E-01 (3)	9.45E-01 (4)	<b>9.72E-01 (1)</b>	9.67E-01 (2)	5.75E-01 (5)
	Std.	0.00E+00	6.80E-02	8.44E-04	3.27E-04	0.00E+00
DTLZ2	Mean	3.97E-01 (2)	3.74E-01 (5)	<b>4.14E-01 (1)</b>	3.78E-01 (3)	3.77E-01 (4)
	Std.	0.00E+00	6.70E-03	4.47E-04	7.63E-04	1.26E-02
DTLZ3	Mean	<b>3.97E-01 (1)</b>	3.59E-01 (3)	3.92E-01 (2)	3.54E-01 (4)	0.00E+00 (5)
	Std.	0.00E+00	1.11E-02	7.79E-03	3.92E-03	0.00E+00
DTLZ4	Mean	2.08E-01 (5)	3.60E-01 (2)	2.56E-01 (4)	<b>3.61E-01 (1)</b>	3.20E-01 (3)
	Std.	0.00E+00	8.49E-02	1.64E-01	5.56E-02	2.57E-02

**TABLE 12.** Mean and Standard deviation values for the GD metric on all 3D functions with ranks in parentheses.

		Mo-FDA-D	NSGA2	NSGA3	GWASGFA	CDG
DTLZ1	Mean	<b>6.16E-05 (1)</b>	4.74E-03 (4)	7.18E-04 (3)	6.56E-04 (2)	2.70E+01 (5)
	Std.	0.00E+00	1.11E-02	1.89E-04	1.05E-04	1.01E+01
DTLZ2	Mean	<b>1.54E-04 (1)</b>	1.39E-03 (4)	7.62E-04 (2)	1.04E-03 (3)	3.79E-02 (5)
	Std.	0.00E+00	1.47E-04	1.71E-05	4.59E-05	1.74E-02
DTLZ3	Mean	<b>2.93E-04 (1)</b>	1.93E-03 (4)	1.84E-03 (3)	1.66E-03 (2)	6.71E+01 (5)
	Std.	0.00E+00	5.49E-04	3.47E-04	1.19E-04	2.32E+01
DTLZ4	Mean	<b>2.04E-04 (1)</b>	4.82E-03 (3)	4.28E-03 (2)	8.14E-03 (4)	5.10E-02 (5)
	Std.	0.00E+00	1.16E-03	2.24E-03	1.06E-03	2.18E-02

**TABLE 13.** Mean and Standard deviation values for the IGD metric on all 3D functions with ranks in parentheses.

		Mo-FDA-D	NSGA2	NSGA3	GWASGFA	CDG
DTLZ1	Mean	2.48E-04 (2)	6.10E-04 (4)	<b>2.39E-04 (1)</b>	4.01E-04 (3)	2.49E-03 (5)
	Std.	0.00E+00	8.71E-04	3.50E-05	2.41E-06	0.00E+00
DTLZ2	Mean	6.52E-04 (4)	7.77E-04 (3)	<b>5.95E-04 (1)</b>	1.04E-03 (5)	5.95E-04 (2)
	Std.	0.00E+00	6.13E-05	1.02E-06	1.70E-05	1.03E-04
DTLZ3	Mean	1.07E-03 (2)	1.30E-03 (3)	<b>9.94E-04 (1)</b>	1.71E-03 (4)	3.20E+00 (5)
	Std.	0.00E+00	6.62E-05	1.73E-05	8.61E-06	9.11E-16
DTLZ4	Mean	2.91E-03 (4)	<b>1.66E-03 (1)</b>	4.34E-03 (5)	1.93E-03 (3)	1.92E-03 (2)
	Std.	0.00E+00	1.79E-03	3.63E-03	1.90E-03	4.49E-04

but last on the fourth metrics. This means that Mo-FDA-D finds good Pareto fronts, close to the true Pareto front but the solutions are less spread than the other algorithms. Regarding the other approach, Mo-FDA-S shows more stability over all the metrics.

Final ranks are shown in Fig. 8. Table 10 shows the final rank based on the values found in Table 9. This is also illustrated in the Fig. 8. This data shows that Mo-FDA-D is the best algorithm on three metrics, i.e. the Hypervolume, the GD and IGD. However, it performs the worst on the Spread metric. This shows a lack of diversity in the Pareto

**TABLE 14.** Mean and Standard deviation values for the Spread metric on all 3D functions with ranks in parentheses.

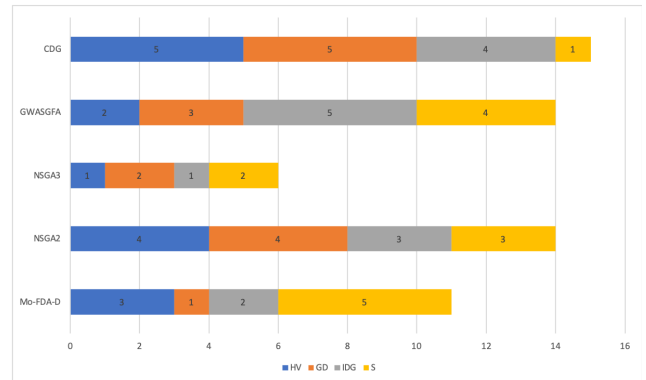
		Mo-FDA-D	NSGA2	NSGA3	GWASGFA	CDG
DTLZ1	Mean	1.20E+00 (5)	8.47E-01 (4)	<b>5.94E-01 (1)</b>	7.88E-01 (3)	6.49E-01 (2)
	Std.	0.00E+00	4.44E-02	6.00E-02	6.37E-02	3.40E-02
DTLZ2	Mean	1.28E+00 (5)	6.94E-01 (3)	<b>5.95E-01 (1)</b>	7.47E-01 (4)	6.33E-01 (2)
	Std.	0.00E+00	5.01E-02	3.13E-02	5.05E-02	4.58E-02
DTLZ3	Mean	1.29E+00 (5)	7.31E-01 (3)	5.89E-01 (2)	8.29E-01 (4)	<b>4.92E-01 (1)</b>
	Std.	0.00E+00	4.00E-02	2.61E-02	6.05E-02	3.90E-02
DTLZ4	Mean	1.71E+00 (5)	6.94E-01 (2)	8.12E-01 (4)	7.65E-01 (3)	<b>6.42E-01 (1)</b>
	Std.	0.00E+00	9.11E-02	2.04E-01	5.47E-02	1.72E-01

**TABLE 15.** Average ranks for each metric and each algorithm over all 3D functions used using the Friedman Rank Sum.

	Mo-FDA-D	NSGA2	NSGA3	GWASGFA	CDG
HV	2.75	3.5	2	2.5	4.25
GD	1	3.75	2.5	2.75	5
IDG	2.75	3	2	3.75	3.5
S	5	3	2	3.5	1.5

**TABLE 16.** Final ranks based on the Friedman Rank Sum values for the 3D functions.

	Mo-FDA-D	NSGA2	NSGA3	GWASGFA	CDG
HV	3	4	1	2	5
GD	1	4	2	3	5
IDG	2	3	1	5	4
S	5	3	2	4	1
Final Average Rank:	2.75	3.5	1.5	3.5	3.75



**FIGURE 10.** Illustration of the final ranking of the algorithms for 3-Objective functions for each metric.

front compared to other algorithms. However, Mo-FDA-S is complementary to Mo-FDA-D where it performs well on the Hypervolume and GD and outperforms the other methods on the Spread. This means that scalarization allows finding a well Pareto front with good diversity. Those conclusions can be seen in Fig. 9 corresponding to the different Pareto sets obtained via our algorithms on 4 selected functions.

## 2) MANY-OBJECTIVE FUNCTIONS

A comparison has been realized for many-objective problems composed of 3 objectives (DTLZ). Results from the competing algorithms are shown in Tables 11 to 14. On each table

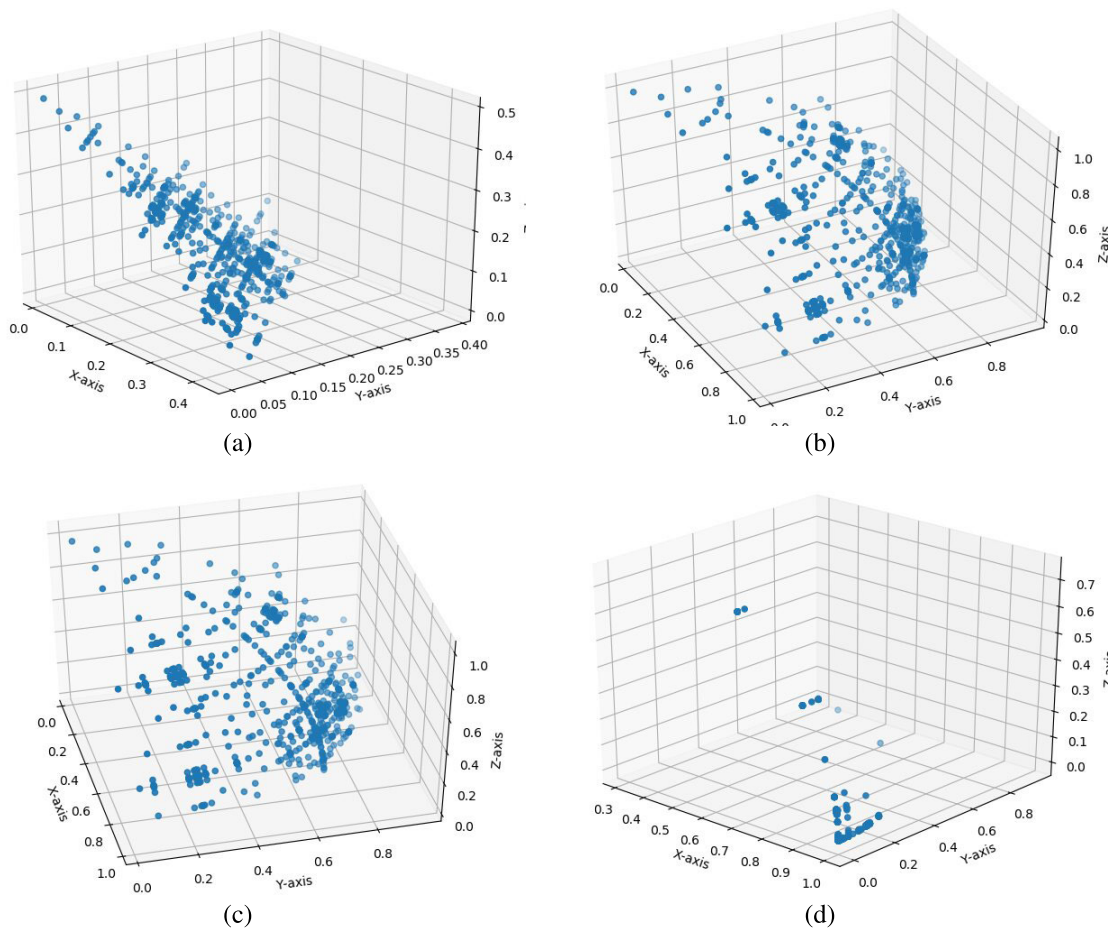


FIGURE 11. Pareto fronts for the 3-Objective functions found by Mo-FDA-D. (a) DTLZ1, (b) DTLZ2, (c) DTLZ3, (d) DTLZ4.

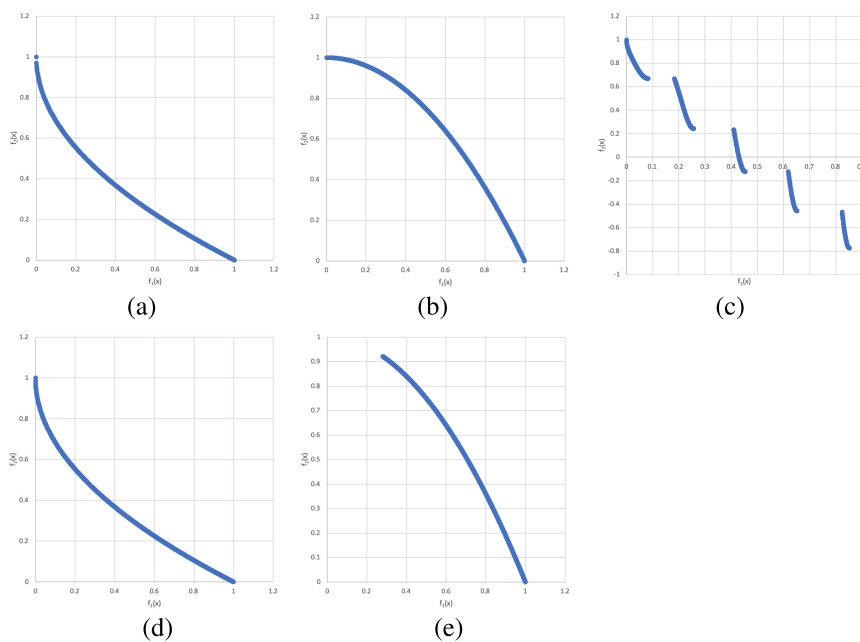


FIGURE 12. The true fronts for the ZDT functions used in our study. (a) ZDT1, (b) ZDT2, (c) ZDT3, (d) ZDT4, (e) ZDT6.

**TABLE 17.** Definitions of the functions used from the ZDT Benchmark.

ZDT1

$$g(x) = 1 + 9 \left( \sum_{i=2}^n x_i \right) / (n - 1)$$

$$F_1(x) = x_1$$

$$F_2(x) = g(x) \left[ 1 - \sqrt{x_1/g(x)} \right]$$

Subject : to :  $x \in [0, 1]$ .

ZDT3

$$g(x) = 1 + 9 \left( \sum_{i=2}^n x_i \right) / (n - 1)$$

$$F_1(x) = x_1$$

$$F_2(x) = g(x) \left[ 1 - \sqrt{x_1/g(x)} - x_1/g(x) \sin(10\pi x_1) \right]$$

Subject : to :  $x \in [0, 1]$ .

ZDT2

$$g(x) = 1 + 9 \left( \sum_{i=2}^n x_i \right) / (n - 1)$$

$$F_1(x) = x_1$$

$$F_2(x) = g(x) \left[ 1 - (x_1/g(x))^2 \right]$$

Subject : to :  $x \in [0, 1]$ .

ZDT4

$$g(x) = 91 + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)]$$

$$F_1(x) = x_1$$

$$F_2(x) = g(x) \left[ 1 - \sqrt{x_1/g(x)} \right]$$

Subject : to :  $x_1 \in [0, 1]$ ,

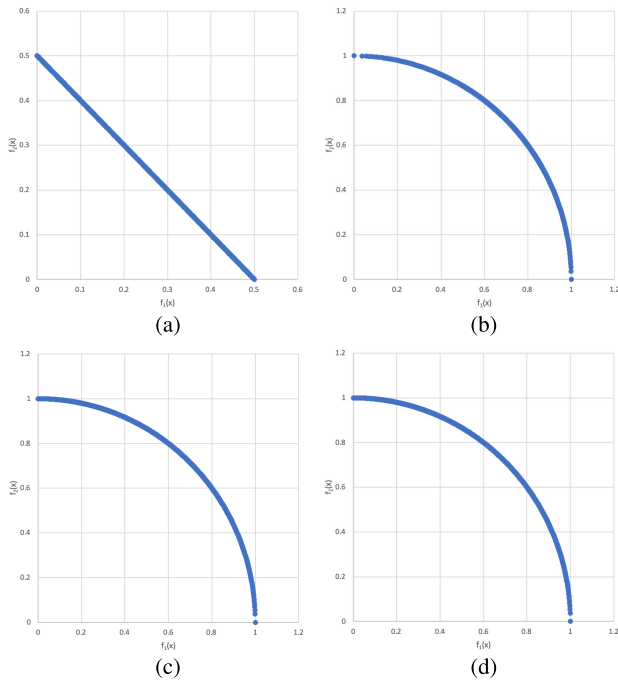
$x_i \in [-5, 5] \ i = 2, \dots, 10$ .

ZDT6

$$g(x) = 1 + 9 \left[ \left( \sum_{i=2}^n x_i \right) / (n - 1) \right]^{0.25}$$

$$F_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$$

$$F_2(x) = g(x) \left[ 1 - (f_1(x)/g(x))^2 \right] \ x \in [0, 1].$$

**FIGURE 13.** The true fronts for the DTLZ functions with 2 Objectives used in our study. (a) DTLZ1, (b) DTLZ2, (c) DTLZ3, (d) DTLZ4.

the values in bold highlight the best algorithm for the given function and the given metric and the value in parenthesis represent the rank. Mo-FDA-SD does not perform as well on 3-Objective functions. However, on each function, it outperforms the other algorithm on at least one metric. Final ranks are shown on Table 10 based on the values found in Table 9. Fig. 10 shows that Mo-FDA-D is the best algorithm for the GD metric. This means that the points found by Mo-FDA-D are closer to the true Pareto front than the other algorithm. It is ranked second on the IGD and similarly to the 2-Objective

function, struggle to perform on the Spread. The best algorithm overall is NSGA-III as it has been adapted to many-objective problems. Our algorithm, Mo-FDA-D is, overall, ranked second in the studied metrics, which shows promising results. Those conclusions can be seen in Fig. 11 representing the Pareto sets of our algorithms the four DTLZ 3-objective functions.

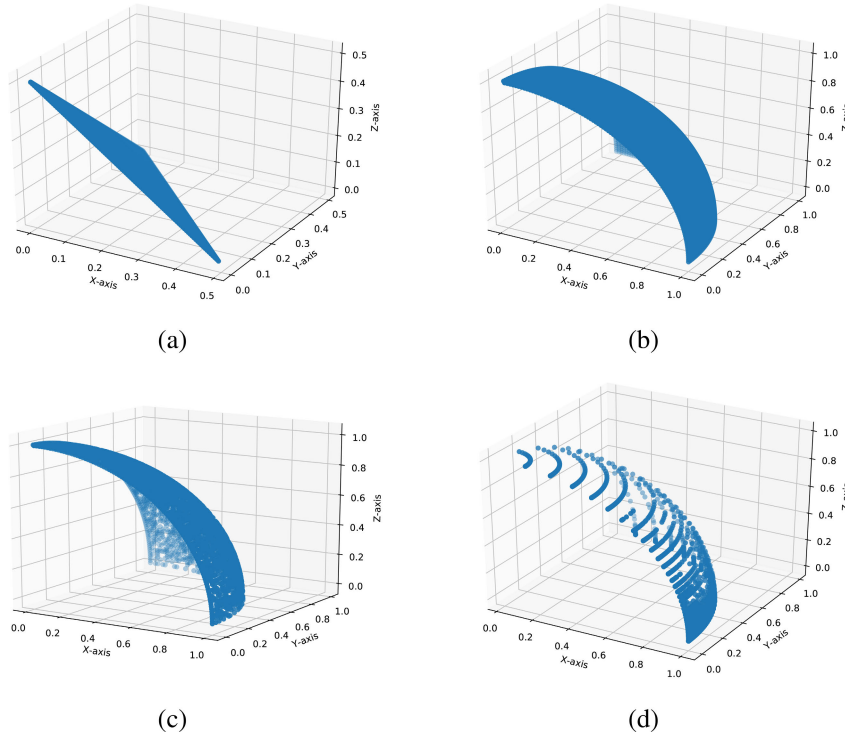
## VII. CONCLUSION AND PERSPECTIVES

In this article, we have proposed new deterministic decomposition approaches to solve MOPs. The decomposition approach is based on geometrical fractal using hyperspheres. The exploration and the exploitation phases of the FDA algorithm have been extended to solve MOPs. Parallel scalarization and dominance based approaches have been developed. A comparison with state-of-the-art MOEAs shows that the proposed algorithms are very competitive in terms of the quality of the obtained Pareto fronts.

The main issues of the Fractal Decomposition Algorithms when handling MOPs are the hypersphere selection (exploration phase) and the Intensive Local Search. In this article, low complex and efficient strategies have been investigated. However, more sophisticated strategies could be designed to deal with those two important issues.

Furthermore, in Mo-FDA-D the hypervolume is used as a selection indicator for the hyperspheres. One can investigate the Spread metric to both improve the hypersphere selection as well as the best solutions to search around using ILS.

A massively parallel implementation on heterogeneous architectures composed of multi-cores and GPUs is under development to take profit from FDA added value: intrinsically parallel. Besides, we will investigate the adaptation of the algorithms to large scale MOPs such as the hyperparameter optimization of deep neural networks.



**FIGURE 14.** The true fronts for the DTLZ functions with 3 Objectives used in our study. (a) DTLZ1, (b) DTLZ2, (c) DTLZ3, (d) DTLZ4.

**TABLE 18.** Definitions of the functions used from the DTLZ Benchmark.

DTLZ1

$$\begin{aligned}
 f_1(\vec{x}) &= \frac{1}{2}x_1(1 + g(\vec{x})) \\
 f_2(\vec{x}) &= \frac{1}{2}(1 - x_1)(1 + g(\vec{x})) \\
 g(\vec{x}) &= 100 \left[ |\vec{x}| + \sum_{x_i \in \vec{x}} (x_i - 0.5)^2 - \cos(20 \cdot \pi(x_i - 0.5)) \right] \\
 0 \leq x_i \leq 1, i &= 1, \dots, n
 \end{aligned}$$

DTLZ3

$$\begin{aligned}
 f_1(\vec{x}) &= (1 + g(\vec{x})) \cos\left(x_1 \frac{\pi}{2}\right) \\
 f_2(\vec{x}) &= (1 + g(\vec{x})) \sin\left(x_1 \frac{\pi}{2}\right) \\
 g(\vec{x}) &= 100 \cdot \left[ |\vec{x}| + \sum_{x_i \in \vec{x}} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right] \\
 0 \leq x_i \leq 1, i &= 1, \dots, n
 \end{aligned}$$

DTLZ2

$$\begin{aligned}
 f_1(\vec{x}) &= (1 + g(\vec{x})) \cos\left(x_1 \frac{\pi}{2}\right) \\
 f_2(\vec{x}) &= (1 + g(\vec{x})) \sin\left(x_1 \frac{\pi}{2}\right) \\
 g(\vec{x}) &= \sum_{x_i \in \vec{x}} (x_i - 0.5)^2 \\
 0 \leq x_i \leq 1, i &= 1, \dots, n
 \end{aligned}$$

DLTZ4

$$\begin{aligned}
 f_1(\vec{x}) &= (1 + g(\vec{x})) \cos\left(x_1^\alpha \frac{\pi}{2}\right) \\
 f_2(\vec{x}) &= (1 + g(\vec{x})) \sin\left(x_1^\alpha \frac{\pi}{2}\right) \\
 g(\vec{x}) &= \sum_{x_i \in \vec{x}} (x_i - 0.5)^2 \\
 \alpha &= 100 \\
 0 \leq x_i \leq 1, i &= 1, \dots, n
 \end{aligned}$$

## APPENDIX DETAILS OF SELECTED BENCHMARK

This appendix contains the functions and the Pareto fronts of the considered benchmarks. Table 17 shows the ZDT problems and Fig. 12 their true Pareto fronts. Table 18 shows the DTLZ problems, Fig. 13 their true Pareto fronts in 2D and Fig. 14 their true Pareto fronts in 3D.

## REFERENCES

- [1] X. Cai, Z. Mei, Z. Fan, and Q. Zhang, "A constrained decomposition approach with grids for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 564–577, Aug. 2018.
- [2] X. Chen, Y. Liu, X. Li, Z. Wang, S. Wang, and C. Gao, "A new evolutionary multiobjective model for traveling salesman problem," *IEEE Access*, vol. 7, pp. 66964–66979, 2019.
- [3] C. A. C. Coello, "Multi-objective optimization," in *Handbook of Heuristics*, R. Martí, P. Pardalos, and M. Resende, Eds. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-07124-4\_17.
- [4] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Comput. CEC*, May 2002, pp. 825–830.
- [5] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using Reference-Point-Based nondominated sorting approach—Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

- [7] J. J. Durillo and A. J. Nebro, "jMetal: A java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, Oct. 2011.
- [8] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. Goodman, "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 74, pp. 621–633, Jan. 2019.
- [9] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, Mar. 1995.
- [10] Y. Gao, B. Song, H. Zhao, X. Hu, Y. Qian, and X. Chen, "An improved selection method based on crowded comparison for multi-objective optimization problems in intelligent computing," *Mobile Netw. Appl.*, pp. 1–11, Dec. 2019, doi: [10.1007/s11036-019-01403-7](https://doi.org/10.1007/s11036-019-01403-7).
- [11] M. Geilen and T. Basten, "A calculator for Pareto points," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, Apr. 2007, pp. 1–6.
- [12] T. Guan, F. Han, and H. Han, "A modified multi-objective particle swarm optimization based on levy flight and double-archive mechanism," *IEEE Access*, vol. 7, pp. 183444–183467, 2019.
- [13] C. Han, L. Wang, Z. Zhang, J. Xie, and Z. Xing, "A multi-objective genetic algorithm based on fitting and interpolation," *IEEE Access*, vol. 6, pp. 22920–22929, 2018.
- [14] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 296–313, Apr. 2018.
- [15] S. Jiang, J. Zhang, and Y.-S. Ong, "Multiobjective optimization based on reputation," *Inf. Sci.*, vol. 286, pp. 125–146, Dec. 2014.
- [16] H. Li, K. Deb, Q. Zhang, P. N. Suganthan, and L. Chen, "Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties," *Swarm Evol. Comput.*, vol. 46, pp. 104–117, May 2019.
- [17] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
- [18] A. Liefvooghe, M. Basseur, L. Jourdan, and E. G. Talbi, "ParadisEO-MOEO: A framework for evolutionary multi-objective optimization," in *Evolutionary Multi-Criterion Optimization. EMO* (Lecture Notes in Computer Science), Matsushima, Japan, vol. 4403, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Germany: Springer, Mar. 2007, pp. 386–400, doi: [10.1007/978-3-540-70928-2\\_31](https://doi.org/10.1007/978-3-540-70928-2_31).
- [19] A. L. Jaimes, S. Z. Martínez, and C. A. C. Coello, "An introduction to multiobjective optimization techniques," in *Optimization in Polymer Processing*, A. Gaspar-Cunha and J. A. Covas, Eds. New York, NY, USA: Nova Science Publishers, 2011, ch. 3, pp. 29–57.
- [20] W. A. Mandal, "Weighted Techebycheff optimization technique under uncertainty," *Ann. Data. Sci.*, 2020, doi: [10.1007/s40745-020-00250-8](https://doi.org/10.1007/s40745-020-00250-8).
- [21] G. Marquet, B. Derbel, A. Liefvooghe, and E. G. Talbi, "Shake them all!" in *Parallel Problem Solving From Nature—PPSN XIII* (Lecture Notes in Computer Science), Ljubljana, Slovenia, vol. 8672, I. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, Eds. Cham, Switzerland: Springer, Sep. 2014, pp. 641–651, doi: [10.1007/978-3-319-10762-2\\_63](https://doi.org/10.1007/978-3-319-10762-2_63).
- [22] K. Miettinen, F. Ruiz, and A. Wierzbicki, *Introduction to Multiobjective Optimization: Interact. Approaches*. Berlin, Germany: Springer, 2008, pp. 27–57.
- [23] A. Nakib, S. Ouchraa, N. Shvai, L. Souquet, and E.-G. Talbi, "Deterministic Metaheuristic based on fractal decomposition for large-scale optimization," *Appl. Soft Comput.*, vol. 61, pp. 468–485, Dec. 2017.
- [24] A. Nakib, B. Thibault, and P. Siarry, "Bayesian based Metaheuristic for large scale continuous optimization," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshop*, May 2015, pp. 314–322.
- [25] A. J. Nebro, J. J. Durillo, and M. Vergne, "Redesigning the jMetal multi-objective optimization framework," in *Proc. Companion Publication Genetic Evol. Comput. Conf. GECCO Companion*, 2015, pp. 1093–1100.
- [26] K. Nowak, M. Mörtens, and D. Izzo, "Empirical performance of the approximation of the least hypervolume contributor," in *Proc. Int. Conf. PPSN Parallel Problem Solving Nature (PPSN XIII)*, in Lecture Notes in Computer Science, Ljubljana, Slovenia, vol. 8672, T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, Eds. Cham, Switzerland: Springer, Sep. 2014, doi: [10.1007/978-3-319-10762-2\\_65](https://doi.org/10.1007/978-3-319-10762-2_65).
- [27] T. Okabe, Y. Jin, and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation," in *Proc. Congr. Evol. Comput. CEC*, Dec. 2003, pp. 878–885.
- [28] N. Riquelme, C. Von Lucken, and B. Baran, "Performance metrics in multi-objective optimization," in *Proc. Latin Amer. Comput. Conf. (CLEI)*, Oct. 2015, pp. 1–11.
- [29] R. Saborido, A. B. Ruiz, and M. Luque, "Global WASF-GA: An evolutionary algorithm in multiobjective optimization to approximate the whole Pareto optimal front," *Evol. Comput.*, vol. 25, no. 2, pp. 309–349, Jun. 2017.
- [30] Z. Sun, M. Wei, Z. Zhang, and G. Qu, "Secure routing protocol based on multi-objective Ant-colony-optimization for wireless sensor networks," *Appl. Soft Comput.*, vol. 77, pp. 366–375, Apr. 2019.
- [31] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Hoboken, NJ, USA: Wiley, 2009.
- [32] J. Yi, J. Bai, H. He, J. Peng, and D. Tang, "Ar-MOEA: A novel preference-based dominance relation for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 788–802, Oct. 2019.
- [33] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [34] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Jun. 2000.
- [35] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving From Nature—PPSN V*, E. E. Agoston, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer, 1998, pp. 292–301.
- [36] J. Zou, Q. Yang, S. Yang, and J. Zheng, "Ra-dominance: A new dominance relationship for preference-based evolutionary multiobjective optimization," *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106192.



**LÉO SOUQUET** received the B.Eng. degree (Hons.) from Edinburgh Napier University, the master's degree from Université Côté d'Azur, and the Ph.D. degree from the Université Paris-Est. He is currently a Co-Founder and the CEO of DSTI, a private higher education institution.



**EL GHAZALI TALBI** received the master's and Ph.D. degrees in computer science from the Institut National Polytechnique de Grenoble, France. He is currently a Full Professor with the University of Lille and the Head of the DOLPHIN Research Group from the Lille's Computer Science Laboratory (LIFL, Université Lille 1) and INRIA Lille Nord Europe. His current research interests include multiobjective optimization, parallel algorithms, metaheuristics, combinatorial optimization, cluster and grid computing, hybrid and cooperative optimization, and applications to logistics/transportation, bioinformatics, and networking.



**AMIR NAKIB** received the M.D. degree in electronic and image processing from University Paris 6, in 2004, the Ph.D. degree (Hons.) in computer science from the Université Paris 12, France, in December 2007, and the Habilitation degree in computer science from University Paris East, in December 2015. Then, he has got a position as the Research Head at Logxlabs Company, where he worked on the design of innovative methods for solving different real world problems. Since September 2010, he has been an Associate Professor with the Laboratoire LISSI, University Paris Est Créteil, where he does research on learning-based optimization for computer vision, and network problems. He has published more than 150 publications in evolutionary computation and optimization.

•••