Expert Systems With Applications 120 (2019) 357–371

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Multi-objective grey wolf optimizer based on decomposition

Saúl Zapotecas-Martínez*, Abel García-Nájera, Antonio López-Jaimes

Departamento de Matemáticas Aplicadas y Sistemas, Universidad Autónoma Metropolitana Unidad Cuajimalpa, Avenida Vasco de Quiroga 4871, Col. Santa Fe Cuajimalpa, Ciudad de México, C.P. 05348, México

ARTICLE INFO

Article history:
Received 25 April 2018
Revised 5 September 2018
Accepted 1 December 2018
Available online 3 December 2018

Keywords:
Multi-objective evolutionary algorithms
Multi-objective grey wolf optimizer
Decomposition-based MOEAs

ABSTRACT

An optimization technique aims to find the best solution to an optimization problem. If the problem considers only one objective function, the best solution will provide the optimal value for such objective. However, if the problem considers two or more objectives, the selection of solutions will not be that straightforward since such objective functions are usually in conflict. For this kind of optimization problems, the use of analytical or exact methods becomes impractical. Thus, heuristic or metaheuristic approaches have to be applied for finding the optimal solutions or, at least, approximate solutions to the optimum. As a consequence, a wide variety of metaheuristics inspired by nature has been proposed for solving optimization problems. Among them, the Grey Wolf Optimizer is a metaheuristic of recent creation that in the last few years has attracted the attention of many researchers. Furthermore, a multi-objective extension of this technique was recently introduced proving its high performance comparable to other multi-objective optimization methods. In this paper, a multi-objective grey wolf optimizer based on the decomposition is introduced. Our proposed algorithm approximates Pareto solutions cooperatively by defining a neighborhood relation among the scalarizing subproblems in which the multi-objective problem is decomposed. The performance of our proposed method is compared against those achieved by six state-of-the-art bio-inspired techniques showing its high performance in both, well-known benchmark problems and two real-life engineering problems.

1. Introduction

Optimization problems appear in any industry or sector. In this kind of problems, the aim is to find the values of the decision variables involved which provide the best solution to the problem. In many cases, finding these values is a difficult task, so that exact methods cannot be employed and heuristics have to be adopted. Moreover, real-world applications consider not only one, but several objectives that have to be optimized and which are commonly in conflict. That is, the optimization of one objective leads to the deterioration of others. These sorts of problems are called multi-objective optimization problems. In multi-objective optimization, the goal is to obtain the values of the decision variables involved that yield a set of solutions which represent the best trade-off between the objectives. The fact of having to optimize multiple objectives makes the optimization task even more challenging.

In the last decades, several metaheuristics inspired by natural processes have been proposed. Such techniques can be classified according to the paradigm they are inspired in, being one of these paradigms the Swarm Intelligence, which deals with collective behaviors that result from the local interactions of individual components with each other and with their environment (Bonabeau, Dorigo, & Theraulaz, 1999). In this paradigm we can find, to name a few, the Particle Swarm Optimization (Kennedy & Eberhart, 1995), the Ant Colony Optimization (Dorigo & Caro, 1999), the Artificial Bee Colony (Karaboga & Basturk, 2007), and the Grey Wolf Optimizer (Mirjalili, Mirjalili, & Lewis, 2014).

The Grey Wolf Optimizer (GWO) is one of the most recent Swarm Intelligence metaheuristics proposed for global optimization. It is inspired by the grey wolves in nature that search for the optimal way of hunting preys. GWO algorithm applies the same principle following the pack hierarchy for organizing the different roles in the wolves pack (Faris, Aljarah, Al-Betar, & Mirjalili, 2017). GWO has been employed in a broad variety of optimization problems due to its advantages over other Swarm Intelligence methods: it has fewer parameters, and it does not require derivative information in the initial search. On the other hand, this inspired strategy is also simple, flexible, scalable, and has a proper balance between exploration and exploitation of the search space leading to an appropriate convergence (Faris et al., 2017). These are the reasons

* Corresponding author.
E-mail addresses: szapotecas@correo.cua.uam.mxm (S. Zapotecas-Martínez), agarcian@correo.cua.uam.mx (A. García-Nájera), alopez@correo.cua.uam.mx (A. López-Jaimes).

https://doi.org/10.1016/j.eswa.2018.12.003

why GWO has gained a high research interest with large audiences from several domains in a short time. Currently, this methodology has been applied in many engineering fields including bioinformatics, networking, machine learning, environmental, and image processing (Faris et al., 2017).

Recently, a multi-objective version of the GWO was proposed (MOGWO) which extends the idea of the original GWO for solving optimization problems with multiple and conflicting objectives (Mirjalili, Saremi, Mirjalili, & Coelho, 2016). In MOGWO, a fixed-sized external archive is integrated to the GWO for saving and retrieving the Pareto optimal solutions. This archive is then employed to define the social hierarchy and simulate the hunting behavior of grey wolves in multi-objective search spaces. Even though MOGWO achieved very competitive results against other multi-objective optimizers, there is still room for improvement.

Throughout the headway of multi-objective optimization, different principles have emerged to deal with the so-called multi-objective optimization problems (MOPs). Pareto optimality and performance indicators have been the two approaches traditionally adopted by multi-objective algorithms during the 2000s. However, it is well known that multi-objective algorithms adopting either Pareto optimality or performance indicators become inefficient as their performances decrease as the number of objectives increases (Bringmann & Friedrich, 2009; Ikeda, Kita, & Kobayashi, 2001). This fact has motivated the development of new strategies to deal with MOPs.

In this paper, we introduce a multi-objective algorithm based on the GWO. We employ some strategies to overcome the notable disadvantages of the existing multi-objective algorithms based on swarm intelligence techniques. Mainly, our algorithm posseses the following properties.

- The proposed approach is based on the decomposition of multi-objective problems which has been suggested to deal with complicated problems. Therefore, Pareto optimal solutions are approximated by decomposing a MOP into a number of scalarizing subproblems.
- A dive and conquer strategy is employed by defining a neighborhood relation among the subproblems. Thus, neighboring subproblems are solving by strategic movements of wolves which cooperatively interact with their neighbors.
- In contrast to several multi-objective techniques based on bio-inspired methods, our proposed approach does not adopt an external archive which increases the computational time of the algorithm.

The proposed approach can be considered for general purposes of multi-objective optimization in continuous and unconstrained spaces. Therefore, we study the performance of the resulting algorithm in two well-known test suites. Additionally, we evaluate the performance of our proposed approach in two real-life applications with unknown features. We show that the proposed approach adopting the above-listed components is computationally efficient and produces better results than six state-of-the-art algorithms which were adopted in our comparative study.

The remainder of this paper is organized as follows. Section 2 introduces concepts and approaches on multi-objective optimization. It is followed by Section 3, which reviews the related work. The proposed MOGWO based on decomposition is described in Section 4. The experimental study on well-known multi-objective benchmark problems and on real-world applications is presented in Sections 5 and 6, respectively. Finally, conclusions and some paths for future research are given in Section 7.

2. Multi-objective optimization

2.1. Preliminaries of multi-objective optimization

A multi-objective optimization problem (MOP) can be stated[1] as follows:

$$\min_{\mathbf{x}\in\Omega} \quad \{\mathbf{F}(\mathbf{x})\} \tag{1}$$

where $\Omega \subset \mathbb{R}^n$ defines the decision variable space and $\mathbf{F}$ is defined as the vector of the objective functions:

$$\mathbf{F} : \Omega \to \mathbb{R}^M, \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))^T$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the function to be optimized. In this work, we consider the box-constrained case, i.e., $\Omega = \prod_{i=1}^{n}[a_j, b_j]$. Therefore, each vector $\mathbf{x} = (x_1, \ldots, x_n)^T \in \Omega$ is such that $a_i \le x_i \le b_i$ for all $i \in \{1, \ldots, n\}$.

In single-objective optimization, the relation "less than or equal" ($\le$) is used to compare the scalar objective values. Since this relation induces a total order in $\mathbb{R}$ (i.e., every pair of solutions is comparable), we can sort solutions from the best to the worst one. In contrast, in multi-objective optimization problems, there is no canonical order on $\mathbb{R}^M$, and thus, we need weaker definitions of order to compare vectors in $\mathbb{R}^M$. To describe the concept of optimality in which we are interested, the following definitions are provided (Miettinen, 1999).

Definition 1. Let $\mathbf{x}, \mathbf{y} \in \Omega$. We say that $\mathbf{x}$ *dominates* $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$) if and only if $f_i(\mathbf{x}) \le f_i(\mathbf{y})$ and $\mathbf{F}(\mathbf{x}) \ne \mathbf{F}(\mathbf{y})$.

Definition 2. Let $\mathbf{x}^\star \in \Omega$. We say that $\mathbf{x}^\star$ is a *Pareto optimal* solution, if there is no other solution $\mathbf{y} \in \Omega$ such that $\mathbf{y} \prec \mathbf{x}^\star$.

Definition 3. The *Pareto optimal set* (PS) is defined by:

$$PS = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal solution}\}$$

and its image ($PF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in PS\}$) is called the *Pareto optimal front* (PF).

We are interested in generating as many (but different) elements of the Pareto optimal set as possible, while maintaining a distribution of solutions as uniform as possible along the Pareto front.

2.2. Evolutionary approaches for multi-objective optimization

The resolution of multi-objective optimization problems consists in identifying the best trade-off alternatives among the conflicting objectives. Therefore, instead of obtaining a single optimal solution (as in the single-objective optimization case), a set of solutions showing the best compromises among the objective functions is found. Because of their nature (based on a population), multi-objective evolutionary algorithms (MOEAs) have become the preferred tool to deal with this type of problems. According to their principles, MOEAs can be categorized into three main groups, those based on Pareto, indicators, and decomposition.

Pareto-based Approaches. The most common preference relation to compare objective vectors is the Pareto dominance relation. Although there are different ways to use Pareto dominance within MOEAs, most of the existing evolutionary algorithms rank the population to assess closeness to the PF. This way, the rank of a solution can be used as a criterion for mating or survival selection. Some methods that uses Pareto dominance to rank solutions include *dominance rank* (Goldberg, 1989), *dominance count* (Zitzler, Laumanns, & Thiele, 2002), and the *dominance*

[1] Without loss of generality, we assume continuous minimization problems.

depth (Srinivas & Deb, 1994). However, a good approximation of the Pareto front has to fulfill two goals simultaneously: convergence and diversity. Therefore, to distribute the solutions along the entire trade-off curve, Pareto dominance has to be used in cooperation with a second criterion. Some methods that have been proposed to distribute solutions along the Pareto optimal front include *fitness sharing and niching* (Deb & Goldberg, 1989), *clustering* (Zitzler et al., 2002), *crowding distance* (Deb, Pratap, Agarwal, & Meyarivan, 2002), among many others. In the early days of the 2000s, Pareto-based MOEAs became one of the most used strategies to achieve a relatively good approximation of the real PF. However, their use has decreased because of the difficulty to achieve a proper spread of solutions (Farhang-Mehr & Azarm, 2002; Gee, Tan, Shim, & Pal, 2015; Hallam, Blanchfield, & Kendall, 2005) and the loss of their discriminant property in high-dimensional objective spaces (Brockhoff & Zitzler, 2006; López Jaimes & Coello Coello, 2015). Examples of recent Pareto-based approaches are SPEA2+SDE (Li, Yang, & Liu, 2014), Θ-DEA (Yuan, Xu, Wang, & Yao, 2016), and SPEA/R (Jiang & Yang, 2017) .

Indicator-based Approaches. A strategy widely employed by MOEAs to achieve a relatively good representation of the PF is related to the use of performance indicators (Farhang-Mehr & Azarm, 2002; Gee et al., 2015; Hallam et al., 2005). The indicator-based evolutionary algorithm (IBEA) (Zitzler & Künzli, 2004) posed the possibility to optimize a performance indicator in the evolutionary process of MOEAs. As it is well known, there exists a large number of indicators to assess the performance of MOEAs, see for example the studies reported in (Jiang, Ong, Zhang, & Feng, 2014; Okabe, Jin, & Sendhoff, 2003; Zitzler, Thiele, Laumanns, Fonseca, & da Fonseca, 2003). Such indicators can assess, in different ways, convergence and diversity, or both of them at the same time. In particular, a good representation of the real PF of a MOP can be achieved by using performance indicators such as hypervolume (Zitzler & Thiele, 1998), *R*2 (Hansen & Jaszkiewicz, 1998), *IGD* (Coello Coello & Reyes Sierra, 2004), among others. However, the use of indicator-based MOEAs is limited by the high computational cost to compute a particular indicator, such as the hypervolume which increases as the number of objectives increases. Recent indicator-based approaches are RIB-EMOA (Zapotecas-Martínez, Sosa Hernández, Aguirre, Tanaka, & Coello Coello, 2014), MOMBI-II (Hernández Gómez & Coello Coello, 2015), MOEA/IGD-NS (Tian, Zhang, Cheng, & Jin, 2016), BCE-IBEA (Li, Yang, & Liu, 2016), and LIBEA (Zapotecas-Martínez, López-Jaimes, & García-Nájera, 2018).

Decomposition-based Approaches. In the last decade, scalarizing functions have been employed by several evolutionary approaches, giving rise to the so-called MOEAs based on decomposition. Decomposition-based approaches rely on solving a number of scalarizing functions which are formulated by the same number of weight vectors. This strategy to solve MOPs have been found to be very effective to deal with complicated test problems (see for instance (Li & Zhang, 2009; Zhang & Li, 2007; Zhang, Liu, & Li, 2009)). In addition, by having a well-distributed set of weight vectors, a well representative sample of the real PF can be obtained. For this reason, decomposition-based MOEAs have become an excellent alternative to deal with multi-objective problems. Some variants or improvements of MOEA/D can be found in Li and Zhang (2009), Zhang, Zhou, and Jin (2008a), Li, Deb, Zhang, and Kwong (2015). On the other hand, different metaheuristics based on decomposition are for instance MOPSO/D (Peng & Zhang, 2008), MOABC/D (Medina, Das, Coello, & Ramírez, 2014), and dMOPSO (Zapotecas-Martínez & Coello Coello, 2011).

The MOEAs mentioned above, are a flavor of distinct evolutionary methodologies based on different principles. However, a more comprehensive review of this type of approaches can be found in (Coello Coello, 1999; Nedjah & de Macedo Mourelle, 2015; Trivedi, Srinivasan, Sanyal, & Ghosh, 2017; Zhou et al., 2011).

2.3. Decomposition of a multi-objective problem

It is well-known that a Pareto optimal solution to the problem in Eq. (1) is an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the objective functions f_i's (Miettinen, 1999). Many scalar approaches have been proposed to aggregate the objectives of a MOP. Among them, the Tchebycheff approach (Bowman Jr., 1976) is one of the most used methods. This technique transforms the vector of function values **F** into a scalar minimization problem which is of the form:

$$\begin{aligned} \textbf{minimize}: \quad & g^{tch}(\mathbf{x}|\lambda, \mathbf{z}) = \max_{1\le j\le M}\{\lambda_j|z_j - f_j(\mathbf{x})|\} \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \subset \mathbb{R}^n \end{aligned} \tag{2}$$

where Ω is the feasible region, $\mathbf{z} = (z_1, \dots, z_k)^\intercal$ is a reference point such that $z_j = \max\{f_j(\mathbf{x})|\mathbf{x} \in \Omega\}$ for each $i = \{1, \dots, M\}$, and $\lambda = (\lambda_1, \dots, \lambda_M)^\intercal$ is a weight vector, i.e., $\lambda_l \geq 0$ for all $j = 1, \dots, M$ and $\sum_{j=1}^{M} \lambda_j = 1$.

For each Pareto optimal point $\mathbf{x}^\star$, there exits a weight vector λ such that $\mathbf{x}^\star$ is the optimum solution of Eq. (2) and each optimal solution of Eq. (2) is a Pareto optimal solution of the problem in Eq. (1).

Therefore, an appropriate representation of the Pareto front could be reached by solving different scalarizing problems. Such problems can be defined by a set of well-distributed weight vectors, which establish the search direction in the optimization process. This is the principal idea behind mathematical programming methods for multi-objective problems and current multi-objective evolutionary approaches based on decomposition, see for example the works reported in (Peng & Zhang, 2008; Zhang & Li, 2007).

3. Related work

3.1. Grey wolf optimizer

Mirjalili et al. (2014) proposed the Grey Wolf Optimizer (GWO), which is inspired by the canis lupus (grey wolf). According to Mirjalili et al., grey wolves live in a pack from five to twelve members, and they have a very strict social dominant hierarchy: the alpha wolves, the beta wolves, the delta wolves, and the omega wolves. The alpha is mostly responsible for making decisions about hunting, sleeping place and time to wake. The beta wolves are subordinate wolves that help the alpha in making decisions and other pack activities. Therefore, beta wolves are probably the best candidates to be the alpha in case one of the alpha wolves passes away or becomes very old. Delta wolves have to submit to alphas and betas, but they dominate the omega. Scouts, sentinels, elders, hunters, and caretakers belong to this category. Finally, the lowest ranking grey wolves are the omegas, which play the role of scapegoats and always have to submit to all the other dominant wolves.

In addition to the social hierarchy of wolves, GWO also considers group hunting as a social behavior of grey wolves. The main phases of grey wolf hunting are: (a) tracking, chasing, and approaching the prey, (b) pursuing, encircling, and harassing the prey until it stops moving, and (c) attacking the prey.

The mathematical model of the social hierarchy of grey wolves considered by the GWO is as follows: The fittest solution as the alpha (α). Consequently, the second and third best solutions are named beta (β) and delta (δ), respectively. The rest of the candidate solutions are assumed to be omega (ω). In GWO, the hunting (optimization) is guided by α, β, and δ. The ω wolves follow these three wolves. In this case, the prey is the optimum value that must be found.

The first step is the case when the grey wolves encircle the prey. The mathematical model is:

$$\mathbf{d} = |\mathbf{c} \circ \mathbf{x}_p(t) - \mathbf{x}(t)| \tag{3}$$

$$\mathbf{x}(t+1) = \mathbf{x}_p(t) - \mathbf{A} \circ \mathbf{d} \tag{4}$$

where $\circ$ denotes the entrywise product between two vectors, $|\cdot|$ denotes the entry wise absolute value of the vector, t indicates the current iteration, $\mathbf{x}_p$ is the position vector of the prey, $\mathbf{x}$ is the position vector of a wolf, and $\mathbf{A}$ and $\mathbf{c}$ are coefficient vectors defined as:

$$\mathbf{A} = 2\mathbf{a} \circ \mathbf{r}_1 - \mathbf{a} \tag{5}$$

$$\mathbf{c} = 2 \cdot \mathbf{r}_2 \tag{6}$$

where $\mathbf{a}$ is linearly decreased from 2 to 0 during the iterations, and $\mathbf{r}_1, \mathbf{r}_2 \in [0, 1]^n$ are random vectors.

Grey wolves can recognize the location of the prey and encircle them. The alpha wolf usually guides the hunt. The beta and delta might also participate in hunting. However, in an abstract search space, the location of the optimum (the prey) is unknown. In order to simulate the hunting behavior of grey wolves mathematically, the alpha (best candidate solution), beta, and delta are supposed to have better knowledge about the potential location of the prey. Therefore, the first three best solutions obtained so far are recorded, and the other search agents (including the omegas) are obliged to update their positions according to the location of the best solutions. The following equations are continuously run for each search agent during the optimization process in order to simulate the hunting and find promising regions of the search space:

$$\mathbf{d}_\alpha = |\mathbf{c}_1 \circ \mathbf{x}_\alpha - \mathbf{x}| \tag{7}$$

$$\mathbf{d}_\beta = |\mathbf{c}_2 \circ \mathbf{x}_\beta - \mathbf{x}| \tag{8}$$

$$\mathbf{d}_\delta = |\mathbf{c}_3 \circ \mathbf{x}_\delta - \mathbf{x}| \tag{9}$$

$$\mathbf{x}_1 = \mathbf{x}_\alpha - \mathbf{A}_1 \circ \mathbf{d}_\alpha \tag{10}$$

$$\mathbf{x}_2 = \mathbf{x}_\beta - \mathbf{A}_2 \circ \mathbf{d}_\beta \tag{11}$$

$$\mathbf{x}_3 = \mathbf{x}_\delta - \mathbf{A}_3 \circ \mathbf{d}_\delta \tag{12}$$

$$\mathbf{x}(t+1) = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3}{3} \tag{13}$$

The search process starts with a random population of grey wolves (candidate solutions). Over the course of iterations, alpha, beta, and delta grey wolves estimate the probable position of the prey. Each candidate solution updates its distance from the prey. The parameter $\mathbf{a}$ is decreased from 2 to 0 in order to emphasize exploration and exploitation, respectively. Finally, the GWO algorithm is terminated by the satisfaction of an end criterion. Algorithm 1 presents the GWO.

According to Faris et al. (2017), GWO has been modified to be inline with the search space of complex domains due to the complex nature of real-world optimization problems. These modifications can be categorized according to the type of proposed improvement in the GWO: updating mechanisms, new operators, encoding scheme of the individuals, and population structure and hierarchy.

Regarding the updating mechanisms, some approaches have a different decreasing procedure of parameter $\mathbf{a}$, used in Eq. (5): Mittal, Singh, and Sohi (2016) proposed decreasing the value of $\mathbf{a}$ by using an exponential decay function, Long, Liang, Cai, Jiao, and Zhang (2017) adapted the parameter $\mathbf{a}$ by using a non-linear modulation index, and Rodríguez, Castillo, and Soria (2016) used fuzzy logic to implement a dynamic adaptation of $\mathbf{a}$.

begin
  Initialize grey wolf pack $\mathbf{x}_i$ $(i = 1, 2, \ldots, n)$
  Initialize $\mathbf{a}$, $\mathbf{A}$, and $\mathbf{C}$
  Compute fitness of each grey wolf
  $\mathbf{x}_\alpha \leftarrow$ the best grey wolf
  $\mathbf{x}_\beta \leftarrow$ the second best grey wolf
  $\mathbf{x}_\delta \leftarrow$ the third best grey wolf
  $t \leftarrow 1$ **while** $t <$ *maximum number of iterations* **do**
    for *each grey wolf* **do**
      Update its position by equation (13)
    Update $\mathbf{a}$, $\mathbf{A}$, and $\mathbf{c}$
    Compute fitness of each grey wolf
    Update $\mathbf{x}_\alpha$, $\mathbf{x}_\beta$, and $\mathbf{x}_\delta$
    $t \leftarrow +1$;
  return $\mathbf{x}_\alpha$;

Algorithm 1: GWO algorithm.

Different strategies for computing the positions of wolves have also been proposed. For example, Saremi, Mirjalili, and Mirjalili (2015) adopted a strategy based on the incorporation of a step size that is proportional to the fitness of the individual in the search space in the current generation. Malik, Mohideen, and Ali (2015) adopted an approach that, instead of using the simple average of the best individuals α, β, and δ, it used a weighted average of these three. Rodríguez et al. (2017) proposed three different methods for updating the positions of the omega wolves: the weighted average, based on the fitness, and based on fuzzy logic.

Regarding new operators, in an attempt to increase the diversity among the population in GWO, Kishor and Singh (2016) proposed a modified version of the GWO that incorporates a simple crossover operator between two wolves in order to facilitate sharing information between pack mates. Saremi et al. (2015) investigated the improvement of GWO by means of another type of operators called evolutionary population dynamics (EPD), which was applied in GWO to eliminate the worst individuals in the population and the reposition of them around the leading wolves in the population, i.e. α, β, and δ. Finally, Zhang and Zhou (2015) integrated Powell optimization algorithm into GWO as a local search operator and called it PGWO. Powell's algorithm is a method for finding a minimum of a function which does not need to be differentiable, and no derivatives are taken.

3.2. Multi-objective grey wolf optimizer

More recently, Mirjalili et al. (2016) extended the GWO for solving multi-objective problems and called their proposal as multi-objective grey wolf optimizer (MOGWO). To perform multi-objective optimization, MOGWO integrates two components. The first component is a fixed-size archive, which is responsible for storing non-dominated Pareto optimal solutions obtained so far. The critical module of this archive is the controller, which controls when a solution enters to the archive. During an iteration, new solutions obtained so far are compared against those in the archive. There would be three possible cases:

a) The new solution is dominated by at least one of the archived solutions. In this case, the solution is not allowed to enter the archive.
b) The new solution dominates one or more solutions in the archive. In this case, the dominated solutions in the archive are removed from it and the new solution will be able to enter the archive.

c) If neither the new solution nor archive members dominate each other, the new solution is added to the archive.

If the archive is overfull, the grid mechanism is run to re-arrange the segmentation of the objective space and find the most crowded segment to remove one of its solutions. Then, the new solution should be inserted into the least crowded segment to improve the diversity of the final Pareto approximation.

The probability of deleting a solution is proportional to the number of solutions in the hypercube (segment). If the archive is overfull, the most crowded segments are first selected, and a solution is removed from one of them randomly to provide a space for the new solution. There is a particular case when a solution is inserted outside the hypercubes. In this case, all the segments are extended to cover the new solution. Hence, the segments of other solutions can be changed as well.

The second component is the leader selection mechanism. In GWO, three of the best solutions obtained during the search, play the role of alpha, beta, and delta wolves. In a multi-objective search space, however, the solutions cannot be directly compared due to the Pareto optimality concepts discussed earlier. The leader selection component chooses the least crowded segments of the search space and offers one of its non-dominated solutions as alpha, beta, or delta wolves. The selection is done by the roulette-wheel selection method with the probability $P_i = c/N_i$ for each hypercube i, where c is a constant number greater than one and N_i is the number of Pareto optimal solutions in the i-th segment. Algorithm 2 describes the MOGWO.

begin
  Initialize grey wolf pack $\mathbf{x}_i$ $(i = 1, 2, \dots, n)$
  Initialize **a**, **A**, and **c**
  Compute fitness of each grey wolf
  Find the non-dominated solutions and initialize the archive with them
  $\mathbf{x}_\alpha \leftarrow$ select leader from the archive
  $\mathbf{x}_\beta \leftarrow$ select leader from the archive
  $\mathbf{x}_\delta \leftarrow$ select leader from the archive
  $t \leftarrow 1$;
  while *$t <$ maximum number of iterations* **do**
    for *each grey wolf* **do**
      Update its position by equation (13)
    Update **a**, **A**, and **c**
    Compute fitness of each grey wolf
    Find the non-dominated solutions
    Update the archive with respect to the obtained non-dominated solutions
    if *the archive is full* **then**
      Run the grid mechanism to remove one of the current archive members
      Add the new solution to the archive
    if *any of the new added solutions to the archive is located outside the hypercubes* **then**
      Update the grids to cover the new solutions
    $\mathbf{x}_\alpha \leftarrow$ select leader from the archive
    $\mathbf{x}_\beta \leftarrow$ select leader from the archive
    $\mathbf{x}_\delta \leftarrow$ select leader from the archive
    $t \leftarrow +1$;
  return archive;

Algorithm 2: MOGWO algorithm.

To the best knowledge of the authors, there is no a modified version of the MOGWO. Therefore, the proposed approach can be considered as the first improvement on the MOGWO.

3.3. Approaches based on decomposition

Decomposition approach is a technique for solving MOPs suggested in the early days of multi-criteria optimization (Bowman Jr., 1976; Zadeh, 1963; Zeleny, 1973). This technique relies on solving a number of scalarizing functions in which is decomposed a MOP. The integration of this approach into evolutionary algorithms dates back from the late 1990s and early 2000s (Hughes, 2003; Ishibuchi & Murata, 1998; Jaszkiewicz, 2002; Jin, Okabe, & Sendho, 2001; Paquete & Stützle, 2003). Perhaps, the multi-objective genetic local search (MOGLS) (Ishibuchi & Murata, 1998) algorithm is the first decomposition-based evolutionary approach reported in the specialized literature. With this, Ishibuchi and Murata (1998) showed a different way to solve MOPs following the principles of evolutionary computation. However, the introduction of the multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang & Li, 2007) established the beginning of a new generation of evolutionary approaches capable of solving problems with multiple objectives efficiently. Throughout the development of evolutionary approaches, several methodologies follow the principle of decomposition have been proposed. In this regard, some researchers have reported alternatives or improvements to the original MOEA/D, see for example the works reported in (Li & Zhang, 2009; Zhang et al., 2009). Recent developments in decomposition-based algorithms can be found in (Chen, Li, & Xin, 2017; Cheng, Yen, & Zhang, 2016; Shim, Tan, & Tang, 2015; Zhang, wei Gong, yong Sun, & yang Qu, 2018). On the other hand, decomposition-based approaches applied to solve particular problems have been reported in (Li & Yin, 2012; Sengupta, Nasir, Mondal, & Das, 2011; Trivedi, Srinivasan, Pal, Saha, & Reindl, 2015; Wang, Ong, & Ishibuchi, 2018). However, the approaches of interest in this paper, are decomposition-based techniques inspired by swarm intelligence.

One of the first swarm intelligence methodologies for multi-objective optimization adopting the principle of decomposition is the multi-objective particle swarm optimizer based on decomposition (MOPSO/D) proposed by Peng and Zhang (2008). MOPSO/D adopts the decomposition of a MOP to achieve a relatively good approximation to the real PF. However, the use of particle swarm optimization (PSO)(Kennedy & Eberhart, 1995) operators and an external archive are used to approximate Pareto optimal solutions. In the last decade, the use of PSO in decomposition-based multi-objective approaches has been prevalent. Thus, several approaches adopting PSO can be found in the specialized literature. Moubayed, Petrovski, and McCall (2010) proposed a new smart multi-objective particle swarm optimizer using decomposition (SDMOPSO). Zapotecas-Martínez and Coello Coello (2011) introduced a generational approach namely dMOPSO. This approach employs a mechanism to reinitialize the particles based on a new concept that they called “age”. If a particle does not improve its personal position in a flight cycle, then the particle increases (by one) its age. If a particle exceeds a certain (pre-defined) age threshold, the particle (including, its velocity, its age and its best personal) is reinitialized. Liu and Niu (2013) also proposed a MOPSO based on decomposition (MOPSO-D) which is based on SDMOPSO. Other multi-objective particle swarm optimizers based on the decomposition of MOPs can be seen as modifications or improvements of the pioneering methodologies described above, see for example the works reported in (Dai, Wang, & Ye, 2015; Moubayed, Petrovski, & McCall, 2014). Recently, a multi-objective artificial bee colony algorithm based on decomposition (MOABC/D) (Medina et al., 2014) was introduced. This approach decomposes a MOP into a number of scalarizing subproblems. Thus, the employed, onlooker, and scout phases of the modified artificial bee colony (ABC) introduced by Karaboga and Akay (2011), approximate Pareto optimal solutions by solving the scalar subproblems in which the MOP is decomposed. Medina et al. (2014) also introduced a multi-objective teaching-learning algorithm based on decomposition (MOTLA/D). This approach employs the teaching-learning-based optimization algorithm introduced by Rao, Savsani, and Vakharia (2011). Analogously to the above approaches, MOTLA/D decomposes a MOP into several scalarizing subproblems. Then, the teacher and learner phases approximate Pareto optimal solutions by solving these scalar subproblems. To the best authors' knowledge, there is no multi-objective extension of the grey wolf optimizer based on decomposition, and this is the focus of the investigation reported in this work. In the following section, we detail the proposed GWO for multi-objective optimization based on the principle of decomposition.

4. Multi-objective grey wolf optimizer based on decomposition (MOGWO/D)

The Multi-objective Grey Wolf Optimizer based on Decomposition (MOGWO/D), introduced in this paper, decomposes a MOP into several single-objective subproblems. Therefore, a relatively good approximation to the real PF of a MOP can be achieved by solving a set of N scalarizing subproblems into which the MOP is decomposed. Thus, throughout the search, MOGWO/D finds the best solution to each subproblem defined by each weight vector and the scalarizing function under consideration.

Similar to other decomposition-based MOEAs, our proposed scheme is an *a posteriori* technique since it generates a sample of the entire Pareto front. Then, this set of alternative solutions should be shown to the decision maker in order to select the one (s)he prefers the most. On the other hand, although decomposition-based approaches bear some resemblance with *a posteriori* weighting methods, the main difference stems in the strategy to solve the single-objective problems derived from the scalarizing functions. In weighting methods, each problem is solved independently of each other. In fact, every solution can be found sequentially using different runs of the optimizer. In contrast, in decomposition-based approaches, by its nature, all the single-objective problems must be solved simultaneously. That is, sharing the same population in order that every individual can contribute to optimize any single-objective subproblem.

As pointed out in Section 2.3, there exist several decomposition methods to approximate Pareto optimal solutions. However, boundary intersection-based methods possess certain advantages over other decomposition techniques including the Tchebycheff and weighted sum approaches (see the discussion presented in (Das & Dennis, 1998; Zhang & Li, 2007)). Because of that, MOGWO/D adopts the Penalty Boundary Intersection (PBI) approach (Zhang & Li, 2007) as its decomposition method. Note however that the study of other decomposition approaches (as those presented in (Ehrgott, 2005; Miettinen, 1999)) coupled with MOGWO/D remains as a topic for future research. Formally, the PBI (Zhang & Li, 2007) scalarizing problem is formulated as:

$$\begin{aligned} \textbf{minimize}: &\quad g^{pbi}(\mathbf{x}|\lambda, \mathbf{z}) = d_1 + \theta d_2 \\ \text{s.t.} &\quad \mathbf{x} \in \Omega \subset \mathbb{R}^n \end{aligned} \tag{14}$$

where

$$d_1 = \frac{||(\mathbf{F}(\mathbf{x}) - \mathbf{z})^\intercal \lambda||}{||\lambda||}$$

and $$d_2 = \left|\left|(\mathbf{F}(\mathbf{x}) - \mathbf{z}) - d_1 \frac{\lambda}{||\lambda||}\right|\right|$$

where Ω is the feasible region, $\mathbf{z} = (z_1, \ldots, z_k)^T$, such that: $z_j = \min\{f_j(\mathbf{x})|\mathbf{x} \in \Omega\}$ and $\lambda = (\lambda_1, \ldots, \lambda_M)^\intercal$ is a weight vector, i.e., $\sum_{j=1}^{M} \lambda_j = 1$ and $\lambda_j \geq 0$ for each $j \in \{1, \ldots, M\}$. Since $\mathbf{z} = (z_1, \ldots, z_M)^\intercal$ is unknown, MOGWO/D states each component z_j by the minimum value for each objective f_j ($j \in \{1, \ldots, M\}$) found during the search.

At the beginning of the algorithm, a pack of N wolves $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is randomly initialized. Considering $\Lambda = (\lambda^1, \ldots, \lambda^N)^\intercal$ as a well-distributed set of weight vectors, each wolf $\mathbf{x}_i$ in the pack, aims to optimize one of the subproblems defined by the weight vector $\boldsymbol{\lambda}^i$. Therefore, each wolf moves towards a better position to hunt its prey, i.e., the optimal value to its corresponding scalarizing problem with weight vector λ^i, that is $g^{pbi}(\mathbf{x}_i|\boldsymbol{\lambda}^i, \mathbf{z})$.

In contrast to the existing MOGWO (Mirjalili et al., 2016), the hunting process into the proposed MOGWO/D is carried out cooperatively. To understand this process, let us define the neighborhood π^i of a weight vector $\boldsymbol{\lambda}^i$ as the set of its closest weight vectors in Λ. For convenience, the neighborhood π^i shall contain only the indexes of the T closets weight vectors to $\boldsymbol{\lambda}^i$. In this way, for the ith subproblem ($i \in \{1, \ldots, N\}$), a sub-pack of wolves $\mathbf{x}_j$'s ($\forall j \in \pi^i$) is implicitly defined and considered to optimize the ith subproblem, i.e., the subproblem defined by the weight vector λ^i. However, for increasing the search exploration, MOGWO/D uses a probabilistic selection of neighborhoods. Concretely, π^i is defined by the indexes of the T closet weight vectors to λ^i with probability ρ. Otherwise, the complete set of indexes is considered, that is $\pi^i = \{1, \ldots, N\}$.

The hunting process into MOGWO/D is carried out by strategic movements among the alpha (α), beta (β), and delta (δ) wolves in the ith sub-pack (π^i). Therefore, a new position $\mathbf{x}_{new}$ is computed by considering the $\mathbf{x}_\alpha$, $\mathbf{x}_\beta$, and $\mathbf{x}_\delta$ wolves taken out from the ith sub-pack according to Eq. (13). However, instead of considering a hierarchy order among the wolves (i.e., α (the first), β (the second), and δ (the third) best solutions) that optimize the i^{th} subproblem, MOGWO/D selects the α, β, and δ wolves randomly from the ith sub-pack. To improve the search abilities, the new position $\mathbf{x}_{new}$ is perturbed by performing a polynomial-based mutation (PBM) (Deb & Agrawal, 1999). Thus, the final position is defined by $\mathbf{x}'_{new} = PBM(\mathbf{x}_{new})$. Finally, solution $\mathbf{x}'_{new}$ shall replace at most n_r solutions into the ith sub-pack, if and only if, this solution is better than others in the neighborhood, that is if $g^{pbi}(\mathbf{x}_{new}|\boldsymbol{\lambda}^i, \mathbf{z}) \leq g^{pbi}(\mathbf{x}_j|\boldsymbol{\lambda}^i, \mathbf{z})$ for all $j \in \{1, \ldots, T\}$. To summarize, the proposed MOGWO/D can be read as shown Algorithm 3 .

4.1. Computational time

At each iteration, MOGWO/D computes N new solutions which denote the evolution of wolves' positions throughout the search. The new solutions are used to update the wolves' neighborhood at each iteration of MOGWO/D. Thus, the neighborhood update (lines 14–19 in Algorithm 3) governs the major computational time in MOGWO/D. The cardinality of π^i depends on the probability ρ. Notably, the cardinality of π^i is given by T with a probability ρ or N with probability $1 - \rho$. Thus the computational time employed to update the neighborhood for an m-objective optimization problem is given by $O(m \times T)$ with probability ρ or $O(m \times N)$ otherwise. Hence, the computational time at each iteration of MOGWO/D is $O(m \times T \times N)$ with probability ρ or $O(m \times N^2)$ with probability $1 - \rho$. Considering a high probability for ρ ($\rho = 0.9$) and a low value for T ($T = 20$) (as stated in MOGWO/D, *cf.* Section 5.6) the computational time of MOGWO/D becomes much smaller than traditional MOEAs whose computational times are around $O(m \times N^2)$.

Input:
N: the number of subproblems to be decomposed;
Λ: a well-distributed set of weight vectors $\{\lambda^1, \ldots, \lambda^N\}$;
ρ: a neighborhood selection probability;
T: the sub-pack size;
n_r: a number maximum of replacements into the sub-pack;
Output:
P: the final approximation of the PF;

$\mathbf{z} = (+\infty, \ldots, +\infty)^\intercal$;
Generate random positions $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ in Ω;
for $i \in \{1, \ldots, N\}$ **do**
  $B_i \leftarrow \{i_1, \ldots, i_T\}$, such that: $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i;
  $z_j \leftarrow \min(z_j, f_j(\mathbf{x}_i))$ for all $j \in \{1, \ldots, M\}$.
while *stopping criterion is not satisfied* **do**
  foreach $i \in permutation(\{1, \ldots, N\})$ **do**
    NEIGHBORHOOD SELECTION:
    if $(rand() < \rho)$ **then** $\pi^i = B_i$;
    else $\pi^i = \{1, \ldots, N\}$;
    COMPUTE NEW POSITION: Following equation (13), compute a new position ($\mathbf{x}_{new}$) by considering $\mathbf{x}_\alpha, \mathbf{x}_\beta$, and $\mathbf{x}_\delta$ solutions, with indexes $\alpha, \beta, \delta \in \pi^i$ chosen randomly, such that $\alpha \neq \beta \neq \delta$.;
    PERTURBATION: $\mathbf{x}'_{new} = PBM(\mathbf{x}_{new})$;
    UPDATE $\mathbf{z}$: $z_j \leftarrow \min(z_j, f_j(\mathbf{x}'_{new}))$ for $j = 1, \ldots, M$;
    UPDATE NEIGHBORING SUBPROBLEMS:
    $c = 0$;
    foreach $j \in \pi^i$ **do**
      if $(g^{pbi}(\mathbf{x}'_{new}|\lambda^j, \mathbf{z}) < g^{pbi}(\mathbf{x}_j|\lambda^j, \mathbf{z})$ **and** $c < n_r)$ **then**
        $\mathbf{x}_j = \mathbf{x}'_{new}$;
        $c = c + 1$;
return P;

Algorithm 3: General Framework of the proposed MOGWO/D.

5. Experimental study

5.1. Metaheuristics adopted for performance comparison

As we mentioned before, there exists a wide variety of metaheuristics available in the specialized literature to deal with multi-objective optimization problems. However, we adopted six state-of-the-art multi-objective approaches that, because their working principles, are natural candidates to carry out our comparative study. In particular, we are interested in the comparison of our proposed MOGWO/D against the first extension of the grey wolf optimizer into the multi-objective context. Thus, we adopted the original MOGWO proposed by Mirjalili et al. (2016). In our comparison, we adopted the original version of MOEA/D (Zhang & Li, 2007). The multi-objective particle swarm optimizer based on decomposition (MOPSO/D) (Peng & Zhang, 2008) is adopted in our comparative study. We also adopted the MOABC/D (Medina et al., 2014) which employs a modified version of the artificial bee colony (Karaboga & Akay, 2011). Another multi-objective approach compared in our study is the MOTL/D (Medina et al., 2014). Finally, a decomposition-based multi-objective optimizer (dMOPSO) (Zapotecas-Martínez & Coello Coello, 2011) is also adopted. The above algorithms were performed with the standard parameters given by their respective authors (see Section 5.4).

5.2. Multi-objective test problems

In our comparative study, we adopted the well-established Deb-Thiele-Laummans-Zitzler (DTLZ) (Deb, Thiele, Laumanns, & Zitzler, 2005) test suite. DTLZ test problems consist of seven continuous and box-constrained multi-objective problems scalable in the number of objectives and decision variables. These test problems possess several features including multi-modality, bias, separability, and different Pareto front shapes such as convex, concave, linear, disconnected and degenerate surfaces. Although DTLZ test problems are scalable in the number of objectives, we evaluate the performance of MOGWO/D considering the three-objective formulation. DTLZs test problems were set to their original configuration, that is, for DTLZ1 seven decision variables, for DTLZ2–DTLZ5 12 decision variables, for DTLZ6 22 decision variables, and for DTLZ7 30 decision variables were employed.

For a more comprehensive study, we also consider the MOPs with complicated Pareto sets proposed in (Zhang et al., 2008b). This test suite has been designed to evaluate the performance of MOEAs when dealing with complicated PS topologies (property of continuous MOPs usually observed in real-world problems (Li & Zhang, 2009)). The adopted test problems (namely UFs) present different properties regarding separability, multi-modality, and different PF geometries including convexity, concavity, discontinuities, etc. We consider the continuous box-constrained problems UF1–F10, with UF1–UF7 being two-objective problems and UF8–UF10 being three-objective problems. UFs test problems were set to their original configuration, that is, 30 decision variables. In the following, we detail the experimental study carried out in this work.

5.3. Performance assessment

The performance assessment of MOEAs has been one of the topics widely studied into the EMOO (evolutionary multi-objective optimization) community. To date, there exist numerousness performance indicators to assess, in different ways, the behavior of an MOEA—see the review presented in (Okabe et al., 2003; Zitzler et al., 2003). In a multi-objective problem, an MOEA should reach the two main goals which are related to *proximity* and *distribution* of solutions along the Pareto optimal front. Therefore, it is common to evaluate the performance of an MOEA by using performance indicators of proximity and distribution. However, as pointed out in (Zitzler et al., 2003), a performance indicator should be compliant concerning Pareto optimality. Therefore, when evaluating the performance of MOEAs, it is preferred the use of performance indicators complaint (or at least weakly complaint) regarding Pareto optimality.

In particular, the comparison among the algorithms considered in this study was carried out by following the performance assessment experimental setup recommended by Knowles, Thiele, and Zitzler (2006). Thus, the multi-objective algorithms were evaluated adopting two performance indicators (Pareto and weakly Pareto complaint, respectively) which assess the proximity and distribution of solutions achieved by a multi-objective technique. In the next, we expose the details of the performance indicators adopted in our comparative study.

Normalized Hypervolume ($\mathcal{H}_n$). The hypervolume performance indicator ($\mathcal{H}$) was introduced in (Zitzler & Thiele, 1998) to assess the performance of MOEAs. This performance indicator is Pareto compliant (Zitzler et al., 2003), and quantifies both proximity and distribution of non-dominated solutions along the PF. The hypervolume corresponds to the non-overlapped volume of all the hypercubes formed by a reference point **r** (given by the user) and each solution **a** in the PF approximation (A). Hyper volume indicator is mathematically stated as:

$$\mathcal{H}(A) = \mathcal{L}\left(\bigcup_{\mathbf{a}\in A}\{\mathbf{x}|\mathbf{a} \prec \mathbf{x} \prec \mathbf{r}\}\right) \quad (15)$$

where $\mathcal{L}$ denotes the Lebesgue measure and $\mathbf{r} \in \mathbb{R}^k$ denotes a reference vector being dominated by all solutions in A.

Therefore, the normalized $\mathcal{H}$ indicator (denoted here as $\mathcal{H}_n$) is defined by

$$\mathcal{H}_n(A) = \frac{\mathcal{H}(A)}{\prod_{i=1}^{M}|r_i - u_i|} \quad (16)$$

where $\mathbf{u} = (u_1, \ldots, u_M)^T$ is the known ideal vector and M denotes the number of objectives. Thus, $\mathcal{H}_n$ value is given into the range [0,1].

A high value of this performance indicator means that the set A has a good approximation and distribution along the true PF.

Inverted Generational Distance plus (IGD^+). The inverted generational distance plus (IGD^+) (Ishibuchi, Masuda, Tanigaki, & Nojima, 2015) is a modification of the well-established *IGD* (Coello Coello & Reyes Sierra, 2004). This performance indicator is weakly Pareto compliant and quantifies how far a given Pareto front approximation is from a reference set.

Let R be the real Pareto front, the IGD^+ for a set of approximated solutions A is calculated as:

$$IGD^+(A, R) = \left(\frac{1}{|R|}\sum_{\mathbf{r}\in R}\min_{\mathbf{a}\in A} d^+(\mathbf{r}, \mathbf{a})\right)^{1/p} \quad (17)$$

where $p = 2$ and d^+ is defined by

$$d^+(\mathbf{r}, \mathbf{a}) = \sqrt{\sum_{k=1}^{M}(\max\{a_k - r_k, 0\})^2} \quad (18)$$

where M is the number of objective functions.

A value of zero in this performance measure, indicates that all the solutions obtained by the algorithm are on the true Pareto front.

5.4. Experimental setup

As pointed out before, we validate the results produced by MOGWO/D against those achieved by MOGWO, MOEA/D, MOPSO/D, MOABC/D, MOTLA/D, and dMOPSO on DTLZ (DTLZ1–DTLZ7) and UF (UF1–UF10) test problems. For a fair comparison between algorithms, we adopted the same population size N for all the algorithms, which is implicitly defined by the number of subproblems formulated for the MOEAs based on decomposition. Scalarizing subproblems were defined by a set of weight vectors and the penalty boundary intersection (PBI) approach with a penalty value $\theta = 5$. To deal with different objective scales, we use the normalized PBI such as employed by Zhang and Li (2007). The weight vectors were generated by using the simplex-lattice design (Scheffé, 1958). Therefore, the number of weight vectors is given by $N = C_{H+M-1}^{M-1}$, where M is the number of objective functions. Consequently, the settings of N is controlled by the parameter H. Here, we use $H = 99$ (for two-objective problems) and $H = 21$ (for three-objective problems), i.e., 100 and 210 weight vectors for MOPs having two and three objectives, respectively.

For each MOEA, the search was restricted to perform $N \times 500$ and $N \times 2,000$ fitness function evaluations for DTLZ and UF test problems, respectively. The rest of parameters were set as follows:

- For MOGWO/D, we use a neighborhood size $T = 20$, a neighborhood selection probability $\rho = 0.9$, mutation index $\eta_{pbm} = 20$ (for PBM), perturbation ratio $P_m = 1/n$ (n is the number of variables), and maximum number of replacements $n_r = 2$.
- MOGWO was performed using an external archive size N_{ar} equal to the number of agents N_a, and they were defined by the number of weight vectors in the decomposition approach, that is $N_{ar} = N_a = N$.
- MOEA/D was performed using the parameter settings suggested by its authors (Zhang & Li, 2007). More precisely, $T = 20$, $\eta_{pbm} = 20$, $P_m = 1/n$ (n is the number of variables of the problem), $\eta_{sbx} = 20$, and $P_c = 1$.
- MOPSO/D was performed following the recommendations given by its authors (Peng & Zhang, 2008). $T = 30$, $K = 1,000$, $\alpha = 1e-6$, $\eta_m = 20$, and $P_m = 0.05$. The velocity constraints (c_1, c_2) and the inertia factor (w) were set as recommended in (Moubayed et al., 2010; Sierra & Coello, 2005), i.e, taking uniformly distributed values such that: c_1, $c_2 \in (1.2, 2.0)$ and $w \in (0.1, 0.5)$ for each velocity calculation.
- MOABC/D was performed following the recommendations given by its authors (Medina et al., 2014). $T_n = 30$, $S_r = 3$, $\delta = 0.9$, $MR = 0.5$ and $limit = 15$.
- MOTLA/D was performed following the recommendations given by its authors (Medina et al., 2014). $T_n = 30, S_r = 3, \delta = 0.9$, $\eta_{pbm} = 20$ and $P_m = 1/n$ (n is the number of variables of the problem).
- dMOPSO was performed following the recommendations given by its authors (Zapotecas-Martínez & Coello Coello, 2011). The velocity constraints (c_1, c_2) and the inertia factor (w) were set randomly according to a uniform distribution, such that: c_1, $c_2 \in (1.2, 2.0)$ and $w \in (0.1, 0.5)$ for each velocity calculation. Finally, T_a was set to 2.

For each MOP, 30 independent runs were performed with each MOEA. The algorithms were evaluated using the $\mathcal{H}_n$ and IGD^+ performance indicators described in Section 5.3. Statistical analysis was carried out over all the runs in the test problem and the performance indicator under consideration. Since the characteristics of the DTLZ and UF test problems are known, the $\mathcal{H}_n$ performance indicator was computed by using the reference point $\mathbf{r} = (1.1, \ldots, 1.1)^T$ and the ideal vector $\mathbf{u} = (0, \ldots, 0)^T$[2] With the above considerations, a good measure of approximation and distribution is reported by $\mathcal{H}_n$ indicator when assessing the performance of each multi-objective algorithm. The IGD^+ performance measure was computed by using the reference sets available in the public domain.

5.5. Analysis of results

The results obtained by the MOGWO/D were compared against those produced by MOGWO, MOEA/D, MOPSO/D, MOABC/D, MOTLA/D, and dMOPSO. Tables 1 and 2 show the results achieved by the algorithms in DTLZ and UF test problems, for $\mathcal{H}_n$ and IGD^+ performance indicators, respectively. In each cell, the number on the left is the average indicator value, and the number on the right (in small font size) is the standard deviation. The best values for each performance indicator and test problem are reported in **boldface**. To identify significant differences among the results, we adopt the Mann–Whitney–Wilcoxon (Wilcoxon, 1945) non-parametric statistical test with a p-value of 0.05 and Bonferroni correction (Bonferroni, 1936). Thus, an algorithm statistically better than all others can be considered as the best algorithm in

[2] However, for DTLZ7 (where the objective scales are different) we use $\mathbf{r} = (0.94, 0.94, 6.33)^T$ and $\mathbf{u} = (0, 0, 2.61)^T$.

Table 1
Table of results obtained by the MOEAs using the $\mathcal{H}_n$ performance indicator.

MOP	MOGWO/D	MOGWO	MOEA/D	MOPSO/D	MOABC/D	MOTLA/D	dMOPSO
DTLZ1	0.0000 ± 0.000	0.0000 ± 0.000	**$\underline{0.8539}$** ± 0.000	0.0603 ± 0.133	0.0000 ± 0.000	0.3800 ± 0.374	0.0280 ± 0.074
DTLZ2	0.5561 ± 0.002	0.3032 ± 0.017	**$\underline{0.5763}$** ± 0.000	0.4877 ± 0.010	0.5038 ± 0.015	0.5192 ± 0.006	0.5617 ± 0.001
DTLZ3	0.0000 ± 0.000	0.0000 ± 0.000	**$\underline{0.5666}$** ± 0.007	0.0000 ± 0.000	0.0000 ± 0.000	0.0035 ± 0.019	0.0000 ± 0.000
DTLZ4	**$\underline{0.5597}$** ± 0.004	0.5314 ± 0.006	0.4959 ± 0.160	0.4998 ± 0.009	0.5030 ± 0.015	0.5192 ± 0.014	0.5545 ± 0.004
DTLZ5	**$\underline{0.1961}$** ± 0.001	0.1894 ± 0.007	0.1932 ± 0.004	0.1825 ± 0.005	0.1925 ± 0.004	0.1940 ± 0.002	0.1941 ± 0.000
DTLZ6	0.1548 ± 0.048	**0.1975** ± 0.002	0.0272 ± 0.019	0.1370 ± 0.074	0.1720 ± 0.046	0.1965 ± 0.002	0.1968 ± 0.000
DTLZ7	**$\underline{0.3968}$** ± 0.003	0.3900 ± 0.037	0.1274 ± 0.102	0.3150 ± 0.022	0.3736 ± 0.077	0.2257 ± 0.128	0.3845 ± 0.001
UF1	**$\underline{0.6008}$** ± 0.003	0.5624 ± 0.039	0.4603 ± 0.074	0.5734 ± 0.023	0.5309 ± 0.070	0.6040 ± 0.032	0.59054 ± 0.026
UF2	**0.6723** ± 0.001	0.6567 ± 0.008	0.6362 ± 0.034	0.6241 ± 0.011	0.6187 ± 0.015	0.6651 ± 0.006	0.6715 ± 0.004
UF3	0.4431 ± 0.087	0.3655 ± 0.071	0.3560 ± 0.039	0.3786 ± 0.022	0.3820 ± 0.037	0.3161 ± 0.101	**0.4557** ± 0.035
UF4	0.2984 ± 0.007	0.3582 ± 0.002	0.3480 ± 0.004	0.3420 ± 0.006	0.3085 ± 0.006	0.3253 ± 0.014	**0.3615** ± 0.006
UF5	**0.1416** ± 0.058	0.0075 ± 0.018	0.0520 ± 0.074	0.1369 ± 0.095	0.0058 ± 0.019	0.0404 ± 0.056	0.0012 ± 0.005
UF6	0.1406 ± 0.013	0.1357 ± 0.042	0.1512 ± 0.102	**0.1564** ± 0.063	0.1192 ± 0.080	0.1928 ± 0.091	0.0507 ± 0.031
UF7	**0.5282** ± 0.003	0.4607 ± 0.065	0.2517 ± 0.120	0.4163 ± 0.086	0.3587 ± 0.122	0.5275 ± 0.015	0.5277 ± 0.011
UF8	**$\underline{0.4409}$** ± 0.054	0.0914 ± 0.134	0.3701 ± 0.087	0.3805 ± 0.015	0.2496 ± 0.045	0.3588 ± 0.041	0.2710 ± 0.138
UF9	**$\underline{0.7003}$** ± 0.070	0.4792 ± 0.169	0.5455 ± 0.033	0.6010 ± 0.067	0.4367 ± 0.056	0.5316 ± 0.039	0.5234 ± 0.042
UF10	**$\underline{0.1050}$** ± 0.044	0.0049 ± 0.019	0.0859 ± 0.084	0.0791 ± 0.018	0.0106 ± 0.019	0.0579 ± 0.049	0.0909 ± 0.000

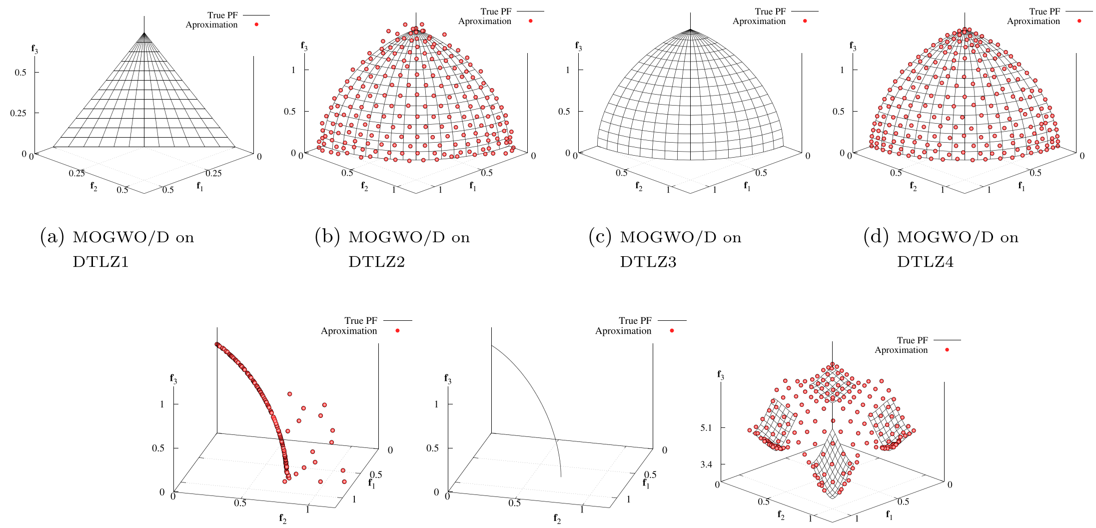
Table 2
Table of results obtained by the MOEAs using the IGD^+ performance indicator.

MOP	MOGWO/D	MOGWO	MOEA/D	MOPSO/D	MOABC/D	MOTLA/D	dMOPSO
DTLZ1	8.2216 ± 3.266	8.0863 ± 3.347	**$\underline{0.0094}$** ± 0.000	1.3945 ± 1.664	4.5932 ± 3.370	0.5338 ± 0.849	2.1872 ± 1.721
DTLZ2	0.0264 ± 0.001	0.2528 ± 0.028	**$\underline{0.0149}$** ± 0.000	0.0561 ± 0.004	0.0527 ± 0.006	0.0483 ± 0.004	0.0237 ± 0.001
DTLZ3	161.6611 ± 18.633	174.7923 ± 20.424	**$\underline{0.0211}$** ± 0.005	38.5011 ± 24.195	86.8645 ± 25.834	35.0184 ± 31.144	45.3955 ± 27.107
DTLZ4	**$\underline{0.0243}$** ± 0.002	0.0314 ± 0.006	0.1061 ± 0.190	0.0528 ± 0.004	0.0514 ± 0.007	0.0472 ± 0.007	0.0277 ± 0.002
DTLZ5	**0.0067** ± 0.001	0.0101 ± 0.004	0.0090 ± 0.003	0.0170 ± 0.005	0.0090 ± 0.004	0.0083 ± 0.002	0.0082 ± 0.000
DTLZ6	0.1331 ± 0.166	**$\underline{0.0022}$** ± 0.000	0.2382 ± 0.053	0.2258 ± 0.385	0.0855 ± 0.139	0.0057 ± 0.001	0.0058 ± 0.000
DTLZ7	**$\underline{0.0468}$** ± 0.002	0.0612 ± 0.089	1.2237 ± 0.411	0.1529 ± 0.131	0.2000 ± 0.341	0.8472 ± 0.515	0.0486 ± 0.000
UF1	**$\underline{0.0766}$** ± 0.002	0.0962 ± 0.031	0.1748 ± 0.064	0.0919 ± 0.011	0.1185 ± 0.050	0.0774 ± 0.014	0.0753 ± 0.013
UF2	**0.0386** ± 0.001	0.0498 ± 0.006	0.0746 ± 0.035	0.0716 ± 0.010	0.0765 ± 0.012	0.0459 ± 0.005	0.0392 ± 0.003
UF3	0.2005 ± 0.062	0.2737 ± 0.060	0.2420 ± 0.031	0.2168 ± 0.017	0.2218 ± 0.026	0.3413 ± 0.141	**0.1763** ± 0.028
UF4	0.1012 ± 0.004	**0.0564** ± 0.001	0.0662 ± 0.004	0.0695 ± 0.004	0.0882 ± 0.003	0.0868 ± 0.011	0.0589 ± 0.004
UF5	**0.3544** ± 0.055	0.8571 ± 0.371	0.4399 ± 0.108	0.4547 ± 0.128	1.0434 ± 0.409	0.5787 ± 0.116	0.7403 ± 0.147
UF6	**0.3291** ± 0.086	0.3294 ± 0.013	0.5258 ± 0.203	0.3325 ± 0.096	0.5066 ± 0.181	0.3592 ± 0.118	0.4831 ± 0.119
UF7	**0.0362** ± 0.002	0.0848 ± 0.060	0.3131 ± 0.137	0.1332 ± 0.093	0.1906 ± 0.137	0.0390 ± 0.013	0.0388 ± 0.007
UF8	**$\underline{0.0844}$** ± 0.038	1.1018 ± 0.764	0.1388 ± 0.117	0.0997 ± 0.006	0.1741 ± 0.048	0.1163 ± 0.018	0.3473 ± 0.211
UF9	**$\underline{0.0771}$** ± 0.053	0.2576 ± 0.193	0.1746 ± 0.033	0.1502 ± 0.055	0.2420 ± 0.028	0.2014 ± 0.028	0.2133 ± 0.027
UF10	**0.4066** ± 0.114	2.0461 ± 1.093	0.5225 ± 0.264	0.5150 ± 0.050	0.6249 ± 0.181	0.6210 ± 0.143	0.6005 ± 0.000

the concerned test problem regarding the performance indicator under consideration, in such case, this value is underlined.

In Table 1, it can be seen that the results obtained by MOGWO/D outperformed those produced by MOGWO, MOEA/D, MOPSO/D, MOABC/D, MOTLA/D, and dMOPSO in most of the test problems (DTLZs and UFs), in terms of $\mathcal{H}_n$ indicator. Besides, the performance of MOGWO/D became significantly better than the other algorithms in six test problems. Table 1 provides a quantitative assessment of the performance of MOGWO/D regarding the $\mathcal{H}_n$ performance indicator. In other words, the solutions obtained by MOGWO/D achieved a better approximation and distribution along the PF than those produced by all the algorithms in most of the test problems. For the particular case of DTLZ test problems, MOGWO/D obtained a low performance in problems DTLZ1–DTLZ3 and DTLZ6 (four test problems). We noticed that for DTLZ1 and DTLZ3 the performance of our proposed approach was bad. In these test problems, MOGWO/D obtained $\mathcal{H}_n = 0$. It means that the non-dominated solutions produced by our proposed algorithm did not outperform the reference point employed to compute the $\mathcal{H}_n$ indicator. The low performance of MOGWO/D in DTLZ1 and DTLZ3 problems could be due to the high multi-modality that these problems have. Regarding UF test problems, it is worth noticing that for UF5, the performance of the proposed MOGWO/D was not significantly better than all the algorithms, even though the MOGWO/D obtained a better $\mathcal{H}_n$ value than the other algorithms. On the other hand, we also noticed that for UF3, UF4, and UF6 the performance of MOGWO/D was outperformed by dMOPSO and MOPSO/D, respectively. Although dMOPSO obtained a better $\mathcal{H}_n$ value in UF3 and UF4, dMOPSO was not significantly better than MOGWO/D in these test problems. Analogously it happens with MOPSO/D, which obtained a better $\mathcal{H}_n$ value in UF6, however, its performance was not significantly better than MOGWO/D.

$\mathcal{H}_n$ performance indicator allows us to assess the proximity and distribution of non-dominated solutions produced by an MOEA. However, it is possible to have a better idea of how far a set of solutions is from the real PF by using performance indicators based on reference sets. In this way, we adopted the IGD^+ indicator to evaluate the performance of the evolutionary approaches. Table 2 shows the results obtained by the proposed MOGWO/D and the adopted MOEAs concerning the IGD^+ indicator. As can be seen, MOGWO/D obtained the best performance according to the IGD^+ indicator in most of the test problems (a lower value a better performance). The results presented in Table 2 corroborate the good or bad performance of MOEAs when solving the adopted test problems. Notably, in DTLZ test problems, MOGWO/D obtained a poor performance regarding DTLZ1–DTLZ3, and DTLZ6. Regarding UF test problems, it can be seen that for UF3 and UF4 test problems, the performance of MOGWO and dMOPSO overcame the performance of MOGWO/D. Note, however, that MOGWO/D outperformed all the other algorithms in eight UF test problems. In particular, we do not expect MOGWO/D to obtain the best results for all the test problems. As Wolpert and Macready proved (Wolpert & Macready, 1997), if an algorithm performs well on a particular class of problems then it necessarily pays with a degraded perfor-



(a) MOGWO/D on DTLZ1 (b) MOGWO/D on DTLZ2 (c) MOGWO/D on DTLZ3 (d) MOGWO/D on DTLZ4

(e) MOGWO/D on DTLZ5 (f) MOGWO/D on DTLZ6 (g) MOGWO/D on DTLZ7

Fig. 1. Pareto front approximations obtained by MOGWO/D on DTLZ1–DTLZ7 test problems.

mance on the set of all remaining problems. It, in fact, holds for the proposed MOGWO/D.

To appreciate the results obtained by our proposed approach, Figs. 1 and 2 show the Pareto front approximations reached by MOGWO/D on the DTLZ and UF test problems. The plots correspond to the best execution produced by MOGWO/D for each test problem. From this plots, it is possible to observe the poor performance of MOGWO/D in DTLZ1, DTLZ3, and DTLZ6 test problems. In general terms, we can see the excellent performance of our proposed MOGWO/D in most of the test problems adopted. From this experimental study, we consider that our proposed MOGWO/D is, in fact, an excellent choice to deal with MOPs with complicated PSs.

5.6. Impact of ρ and T on the performance of MOGWO/D

In this section, we investigate the performance of the proposed MOGWO/D using a list of ten different probabilities $\{0.1, 0.2, \ldots, 1\}$ for ρ and seven different neighborhood sizes $\{3, 6, 9, 12, 15, 18, 21\}$ for T.

We selected the IGD^+ indicator to assess the performance of MOGWO/D under different settings after 500 (for DTLZ problems) and 2000 (for UF problems) generations. In this scenario, Figs. 3 and 4 illustrate the heat map for different values of ρ and T on DTLZ1–DTLZ7 and UF1–UF10 test problems. In each plot, the x-axis shows the different ρ probabilities, while the y-axis represents the neighborhood size T. Black and white colors denote a better and a worse performance of MOGWO/D on the adopted test problems, respectively. As we can see, MOGWO/D achieved a better performance for high values of ρ in most of the DTLZ and UF test problems. According to the results, we noted that using a value $\rho = 0.9$ the performance of MOGWO/D became much better in most of the test problems. On the other hand, the neighborhood size T cannot be utterly generalized as the performance of MOGWO/D (with a particular T value) depends directly on the test problem under consideration. Depending on the problem, the performance of MOGWO/D using a higher value of T became to benefit the performance of MOGWO/D. Thus, we consider $T = 20$ a good choice for this parameter as we noted that, in most of the cases, the performance of MOGWO/D decreased with smaller neighborhood sizes. Nonetheless, the standard values that we state in this study (i.e., $\rho = 0.9$ and $T = 20$) are not universal, and their adjustment should be considered depending on the problem to be solved.

6. Solving two real-world applications

6.1. Case study I: Crash-Worthiness design of vehicles

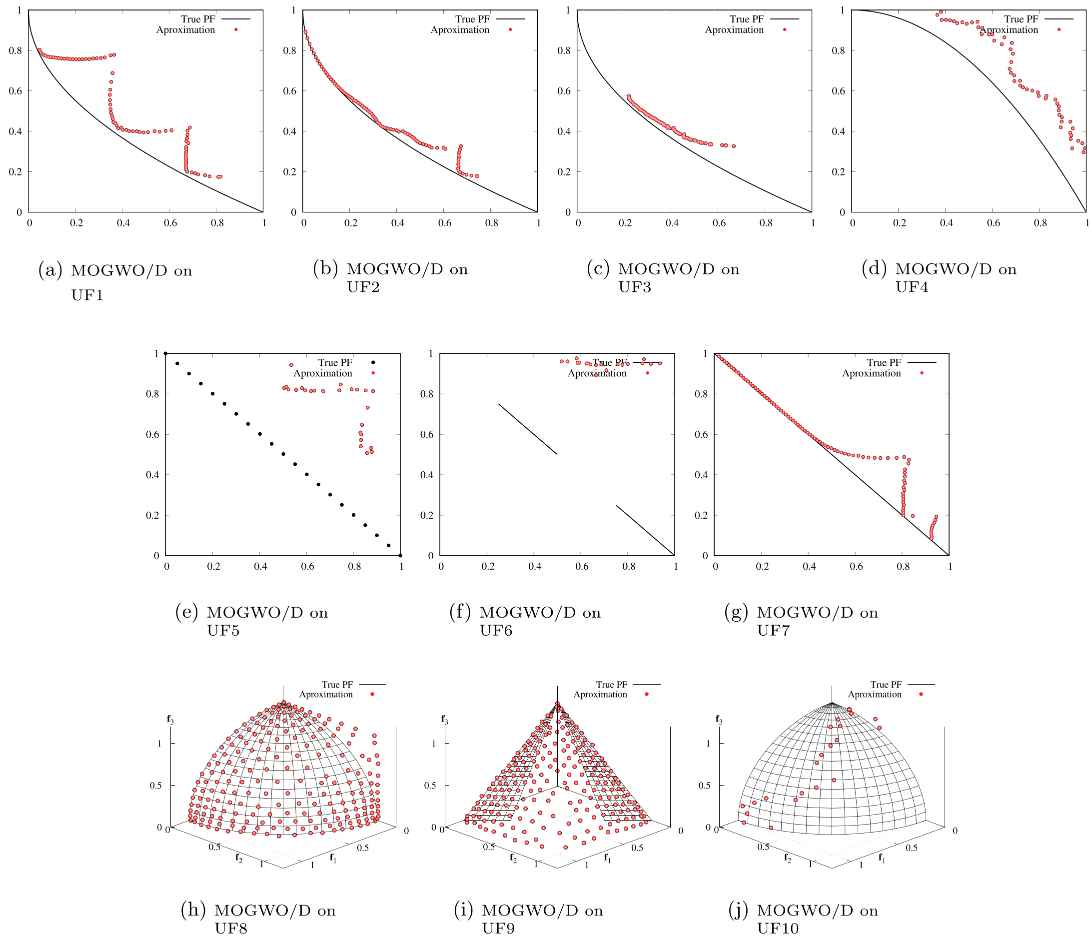
The crash-worthiness design (CWD) problem of vehicles aims to optimize the frontal structure of a vehicle for crash-worthiness (Liao, Li, Yang, Zhang, & Li, 2008). The thickness of five reinforced members $(t_1, \ldots, t_5)$ around the frontal structure is chosen as design variables. The multi-objective formulation of the problem consists of: 1) minimize the *"mass of vehicle"* (*Mass*); 2) minimize the *deceleration during the "full frontal crash"* (A_{in}), which is proportional to biomechanical injuries caused to the occupants; and 3) minimize the *"toe board intrusion"* (*Intrusion*) in the "offset-frontal crash", which accounts for the structural integrity of the vehicle. Thus, the box-constrained multi-objective optimization problem is written as follows:

$$\begin{aligned}\textbf{minimize} &: f_1(\mathbf{x}) = Mass \\ \textbf{minimize} &: f_2(\mathbf{x}) = A_{in} \\ \textbf{minimize} &: f_3(\mathbf{x}) = Intrusion\end{aligned} \tag{19}$$

where $\mathbf{x} = (t_1, t_2, t_3, t_4, t_5)^\intercal$, such that $1mm \leq t_i \leq 3mm$, for $i \in \{1, \ldots, 5\}$. The mathematical formulation for the three objectives can be found in the original study (Liao et al., 2008).

6.2. Case study II: Synthesis gas production

In the chemical industry, the production of methanol, ammonia, hydrogen and higher hydrocarbons require synthesis gas (or syn gas). The main three syn gas production methods are carbon dioxide reforming, steam reforming, and partial oxidation of methane. Larentis, de Resende, Salim, and Pinto (2001) developed a multi-objective formulation for the synthesis gas production (SGP) by combined reforming. Subsequently, a normalization of the combined reforming empirical multi-objective model developed by Larentis et al. (2001) was introduced by Mohanty (2006). The normalized objective functions for the SGP problem (which are methane (CH_4) conversion in (%), carbon monoxide (CO) selectivity in (%) and hydrogen to carbon monoxide (H_2/CO) ra-



(a) MOGWO/D on UF1 (b) MOGWO/D on UF2 (c) MOGWO/D on UF3 (d) MOGWO/D on UF4

(e) MOGWO/D on UF5 (f) MOGWO/D on UF6 (g) MOGWO/D on UF7

(h) MOGWO/D on UF8 (i) MOGWO/D on UF9 (j) MOGWO/D on UF10

Fig. 2. Pareto front approximations obtained by MOGWO/D on UF1–UF10 test problems.

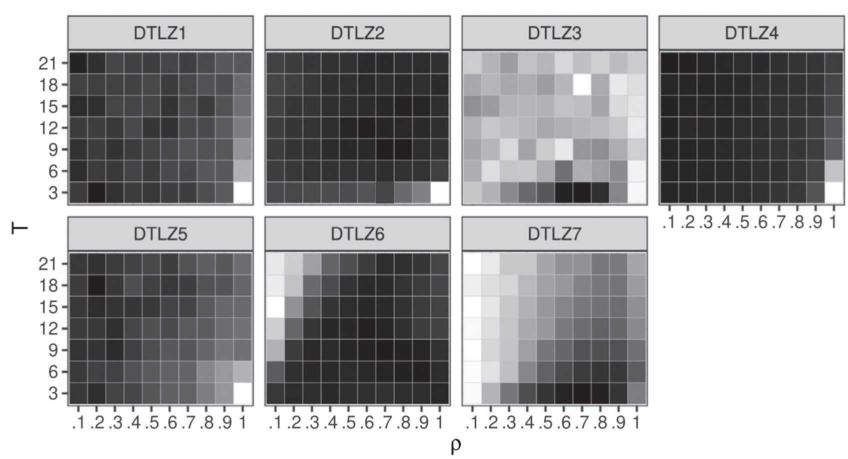


Fig. 3. Heat map for different configurations of ρ (x-axis) and T (y-axis). Black and white colors denote a better and a worse performance of MOGWO/D on DTLZ1–DTLZ7 test problems, respectively.

tio) such as modeled by Larentis et al. (2001), and applied by Mohanty (2006) and Ganesan, Elamvazuthi, Shaari, and Vasant (2013) can be written as follows:

$$\begin{aligned} \textbf{maximize} &: f_1(\mathbf{x}) = CH_4(\%) \text{ conversion} \\ \textbf{maximize} &: f_2(\mathbf{x}) = CO(\%) \text{ selectivity} \\ \textbf{minimize} &: f_3(\mathbf{x}) = H_2/CO \text{ ratio} \end{aligned} \tag{20}$$

where $\mathbf{x} = (O_2/CH_4, GV, T)^\intercal$, such that O_2/CH_4 is the oxygen to methane ratio (*gmol/gmol*), *GV* is the hourly space velocity (h^{-1}) and T is the reaction temperature (°C). The above decision variables are constrained as per the experimental setup described in Larentis et al. (2001). More precisely, $O_2/CH_4 \in [0.25,\ 0.55]$ *gmol/gmol*, $GV \in [10,\ 000,\ 20,\ 000]$ h^{-1}, and $T \in [600,\ 1{,}100]$ °C. The mathematical formulation for the three-objective problem can be found in (Mohanty 2006; Ganesan et al. 2013).

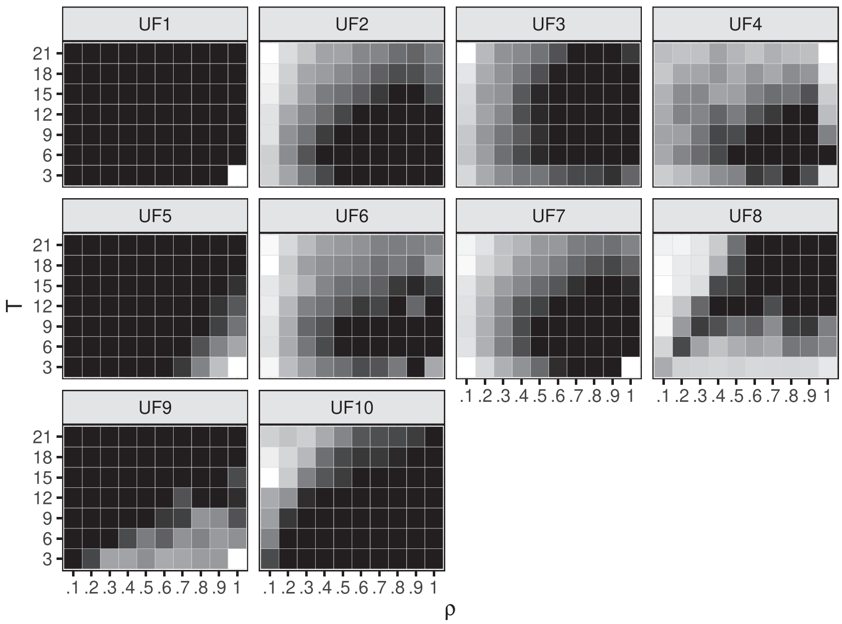


Fig. 4. Heat map for different configurations of ρ (x-axis) and T (y-axis). Black and white colors denote a better and a worse performance of MOGWO/D on UF1–UF10 test problems, respectively.

Table 3
Table of results obtained by the MOEAs using the $\mathcal{H}$ and IGD^+ performance indicators.

MOP	MOGWO/D	MOGWO	MOEA/D	MOPSO/D	MOABC/D	MOTLA/D	dMOPSO
	$\mathcal{H}$						
CWD	**241.2553** $\pm$ 0.285	238.6851 $\pm$ 2.291	227.6605 $\pm$ 0.274	227.1744 $\pm$ 0.616	227.3609 $\pm$ 0.544	229.4346 $\pm$ 1.979	238.3450 $\pm$ 1.177
SGP	**844.5856** $\pm$ 0.272	653.3729 $\pm$ 16.928	844.4173 $\pm$ 0.308	844.2012 $\pm$ 0.174	844.5392 $\pm$ 0.655	842.7222 $\pm$ 0.113	844.4561 $\pm$ 0.186
	IGD^+						
CWD	**0.0072** $\pm$ 0.000	0.0167 $\pm$ 0.018	0.1038 $\pm$ 0.000	0.1100 $\pm$ 0.004	0.1069 $\pm$ 0.004	0.1005 $\pm$ 0.018	0.0101 $\pm$ 0.018
SGP	**0.0449** $\pm$ 0.000	0.3279 $\pm$ 0.279	0.0461 $\pm$ 0.001	0.0461 $\pm$ 0.000	0.0450 $\pm$ 0.001	0.0441 $\pm$ 0.000	0.0450 $\pm$ 0.001

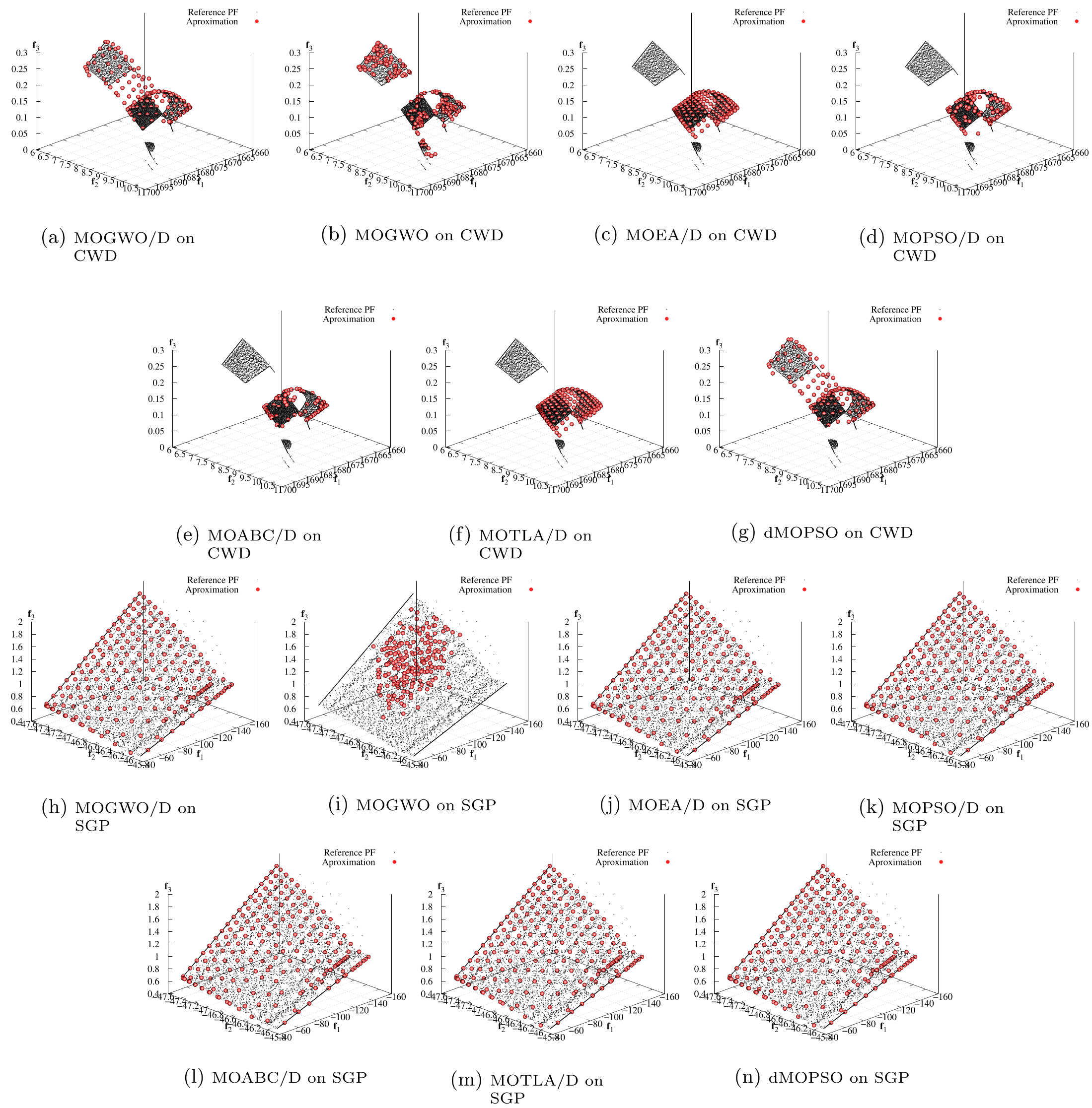
6.3. Analysis of results

In this study, we compare the results achieved by the proposed MOGWO/D against those obtained by MOGWO, MOEA/D, MOPSO/D, MOABC/D, MOTLA/D, and dMOPSO. We followed the experimental setup described in Section 5. However, for the two real-world applications, some minor changes were done. Initially, the number of generations was set to 500. Since the features of the above engineering problems are unknown, the reference PF to compute the IGD^+ performance indicator had to be constructed. In our experimental study, the reference PF for each engineering problem was constructed in two stages. 1) The non-dominated solutions found by all the algorithms over the 30 independent executions were captured; 2) From these solutions, we employed a clustering algorithm to select 6000 non-dominated solutions which define the reference set used to compute the IGD^+ performance measure. In the case of the hypervolume, we adopted the full hypervolume $\mathcal{H}$ indicator Eq. (15). The reference vector $\mathbf{r}$ to compute $\mathcal{H}$, was obtained by finding the maximum value for each objective in the constructed reference PF. In this way, the $\mathcal{H}$ performance indicator shall consider, in a better measure, the extreme portions of the PF approximation found by each MOEA.

In Table 3, we present the results obtained by the proposed MOGWO/D and the adopted MOEAs for the $\mathcal{H}$ and IGD^+ performance indicators over the two real-life problems. As can be seen, the proposed MOGWO/D obtained a higher hypervolume value than all the MOEAs under comparison in both engineering problems. It means that the non-dominated solutions found by MOGWO/D reached a better convergence and spread along the true PF. Moreover, MOGWO/D outperformed all the multi-objective approaches significantly for the CWD test problem according to the Mann–Whitney–Wilcoxon non-parametric statistical test with a p-value of 0.05 and Bonferroni correction. As we said before, $\mathcal{H}$ performance indicator allows us to assess the relative performance of a multi-objective algorithm. However, it is possible to have a better idea of how far a set of solutions is from a reference PF by using performance indicators based on reference sets. In this sense, the MOEAs were evaluated employing the IGD^+ performance indicator and using as the reference set, the constructed approximation such as it was described above.

Thus, the second part of Table 3 shows the results regarding the IGD^+ performance measure. From this table, it is possible to see that the proposed MOGWO/D also obtained the best IGD^+ values in both engineering problems. Corroborating the judgment already done by the $\mathcal{H}$ performance indicator, MOGWO/D became better than all the MOEAs under consideration, but now, regarding the IGD^+ performance indicator. Considering a large number of solutions in the reference set (6,000 solutions for each problem), the use of the IGD^+ performance indicator assesses, in a better way, proximity and distribution of the non-dominated solutions along the PF. Therefore, we can say that our proposed MOGWO/D obtained a better approximation and distribution of solutions along the approximated PF.

Finally, Fig. 5 illustrates the PF approximations obtained by the MOEAs in a single execution. The plots in this figure show the PF reference used in the computation of the IGD^+ performance measure (points in black), and the solutions (points in red) achieved in the last generation by each MOEA. For a better appreciation



(a) MOGWO/D on CWD (b) MOGWO on CWD (c) MOEA/D on CWD (d) MOPSO/D on CWD

(e) MOABC/D on CWD (f) MOTLA/D on CWD (g) dMOPSO on CWD

(h) MOGWO/D on SGP (i) MOGWO on SGP (j) MOEA/D on SGP (k) MOPSO/D on SGP

(l) MOABC/D on SGP (m) MOTLA/D on SGP (n) dMOPSO on SGP

Fig. 5. PF approximations obtained by MOGWO/D, MOGWO, MOEA/D, MOPSO/D, MOABC/D, MOTLA/D, and dMOPSO for the CWD and SGP problems, respectively.

of the plots in the SGP problem (which involves two maximization functions and one minimization function), we multiply by minus one the objectives concerned to maximization. Thus, the set of solutions displayed in these plots corresponds to the experiment (run) that produced a IGD^{+} value closest to the averaged IGD^{+} value reported in Table 3. A notable performance of MOGWO/D, MOGWO, and dMOPSO can be seen for CWD problem, as the entire PF reference was almost covered. On the other hand, the three decomposition-based MOEAs (i.e., MOGWO/D, MOEA/D, MOPSO/D, MOABC/D, MOTLA/D, and dMOPSO) were capable of covering the PF reference almost entirely in the SGP problem. Considering the present study, we conclude that the proposed MOGWO/D is, in fact, a good choice to deal with real-world problems.

7. Conclusions and future research

In this paper, we have introduced a Multi-Objective Grey Wolf Optimizer based on Decomposition (MOGWO/D). Decomposition is a technique suggested in traditional multi-criteria methods, which transforms a multi-objective problem into a number of single-objective subproblems. In contrast to MOGWO, MOGWO/D approximates Pareto optimal solutions by solving simultaneously the scalarizing subproblems, in which the multi-objective problem is decomposed. Each subproblem is then solved by a sub-pack defined from the entire set of wolves destined to approximate the entire PF of a multi-objective problem. In this regard, an important issue to consider in MOGWO/D, is the neighboring relation among the scalarizing subproblems which are cooperatively solved by a sub-pack of wolves. The experimental study showed the effectiveness of the proposed approach when approximating the PF of well-known benchmark problems (DTLZ and UF) and two real-life engineering applications. A more comprehensive study was carried out evaluating the performance of MOGWO/D using different neighborhood sizes (T's) and neighborhood selection probabilities (ρ's). Our results revealed that MOGWO/D had a better performance for higher values of ρ in most of the benchmark problems (in our study $\rho = 0.9$). On the other hand, a specific sub-pack size T cannot be generalized as the performance of MOGWO/D depends on the test problem. Nonetheless, $T = 20$ can be regarded as a good value since, for most of the test functions, the performance of MOGWO/D decayed with lower sub-pack sizes. As we noticed, in most of the multi-objective problems, the performance of MOGWO/D was superior regarding the achievements obtained by the adopted state-of-the-art MOEAs. In a hand-to-hand comparison between MOGWO/D and MOGWO (which adopts Pareto optimality in their environmental selection procedure), we evidenced that the performance of MOGWO/D was significantly better than that one achieved by MOGWO in most of the MOPs adopted. This, in fact, shows the benefits of employing the multi-objective decomposition when solving problems with difficult properties.

As part of our future work, we plan to extend the use of MOGWO/D to deal with constrained multi-objective problems. We are also interested in exploring the inclusion of single-objective mathematical programming techniques into MOGWO/D. We hypothesize that bio-inspired methodologies guided by mathematical methods could improve their performance drastically. Another interesting investigation is regarding the use of MOGWO/D in real-life applications. As we know, the features of real-world problems are unknown. This motivates the use of different heuristics to deal with this type of problems. Being MOGWO/D an algorithm of recent creation, its potential in real-world problems is unknown. On the other hand, the development of a MOGWO based on performance indicators is also part of our desirable future research.

Authors' contributions

Saúl Zapotecas-Martínez: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Writing Original Draft, Writing Review & Editing.

Abel García-Nájera: Investigation, Resources, Writing Original Draft, Writing Review & Editing.

Antonio López-Jaimes: Investigation, Writing Review & Editing.

References

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm intelligence – From natural to artificial systems. *Studies in the sciences of complexity*. Oxford University Press.

Bonferroni, C. E. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 8*, 3–62.

Bowman, V. J., Jr. (1976). Oh the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Lecture notes in economics and mathematical system: 130* (pp. 76–86).

Bringmann, K., & Friedrich, T. (2009). Approximating the least hypervolume contributor: NP-Hard in general, but fast in practice. In M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, & M. Sevaux (Eds.), *Evolutionary multi-criterion optimization. 5th international conference, EMO 2009* (pp. 6–20). Nantes, France: Springer. Lecture Notes in Computer Science Vol. 5467.

Brockhoff, D., & Zitzler, E. (2006). Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. In T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, & X. Yao (Eds.), *Parallel problem solving from nature - PPSNIX, 9th international conference* (pp. 533–542). Reykjavik, Iceland: Springer. Lecture Notes in Computer Science Vol. 4193.

Chen, J., Li, J., & Xin, B. (2017). Dmoea- εC: Decomposition-based multiobjective evolutionary algorithm with the ε -constraint framework. *IEEE Transactions on Evolutionary Computation, 21*(5), 714–730. doi:10.1109/TEVC.2017.2671462.

Cheng, J., Yen, G. G., & Zhang, G. (2016). A grid-based adaptive multi-objective differential evolution algorithm. *Information Sciences, 367–368*, 890–908. doi:10.1016/j.ins.2016.07.009.

Coello Coello, C. A. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems. An International Journal, 1*(3), 269–308.

Coello Coello, C. A., & Reyes Sierra, M. (2004). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, & H. Sossa (Eds.), *Proceedings of the third mexican international conference on artificial intelligence (MICAI'2004)* (pp. 688–697). Springer Verlag. Lecture Notes in Artificial Intelligence Vol. 2972.

Dai, C., Wang, Y., & Ye, M. (2015). A new multi-objective particle swarm optimization algorithm based on decomposition. *Information Sciences, 325*, 541–557.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating pareto optimal points in multicriteria optimization problems. *SIAM Journal on Optimization, 8*(3), 631–657.

Deb, K., & Agrawal, S. (1999). A niched-penalty approach for constraint handling in genetic algorithms. In *International conference on artificial neural networks and genetic algorithms (ICANNGA'99)* (pp. 235–243). Springer-Verlag.

Deb, K., & Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In J. D. Schaffer (Ed.), *Proceedings of the third international conference on genetic algorithms* (pp. 42–50). George Mason University San Mateo, California: Morgan Kaufmann Publishers.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, L. Jain, & R. Goldberg (Eds.), *Evolutionary multiobjective optimization. Theoretical advances and applications* (pp. 105–145). USA: Springer.

Dorigo, M., & Caro, G. D. (1999). Ant colony optimization: A new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (cat. no. 99th8406): 2* (p. 1477Vol. 2).

Ehrgott, M. (2005). *Multicriteria optimization* (2nd). Springer, Berlin.

Farhang-Mehr, A., & Azarm, S. (2002). Diversity assessment of Pareto optimal solution sets: An entropy approach. In *Congress on evolutionary computation (cec'2002): 1* (pp. 723–728). Piscataway, New Jersey: IEEE Service Center.

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2017). Grey wolf optimizer: A review of recent variants and applications. *Neural Computing and Applications*, 1–23.

Ganesan, T., Elamvazuthi, I., Shaari, K. Z. K., & Vasant, P. (2013). Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production. *Applied Energy, 103*, 368–374.

Gee, S. B., Tan, K. C., Shim, V. A., & Pal, N. R. (2015). Online diversity assessment in evolutionary multiobjective optimization: A Geometrical perspective. *IEEE Transactions on Evolutionary Computation, 19*(4), 542–559.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Hallam, N., Blanchfield, P., & Kendall, G. (2005). Handling diversity in evolutionary multiobjective optimisation. In *2005 IEEE congress on evolutionary computation (CEC'2005): 3* (pp. 2233–2240). Edinburgh, Scotland: IEEE Service Center.

Hansen, M. P., & Jaszkiewicz, A. (1998). Evaluating the quality of approximations to the non-dominated set. *Technical Report IMM-REP-1998-7*. Technical University of Denmark.

Hernández Gómez, R., & Coello Coello, C. A. (2015). Improved metaheuristic based on the r2 indicator for many-objective optimization. In *Proceedings of the 2015 annual conference on genetic and evolutionary computation* (pp. 679–686). ACM.

Hughes, E. J. (2003). Multiple single objective Pareto sampling. In *The 2003 congress on evolutionary computation, 2003. CEC '03.: 4* (pp. 2678–2684Vol.4). doi:10.1109/CEC.2003.1299427.

Ikeda, K., Kita, H., & Kobayashi, S. (2001). Failure of pareto-based moeas: does non-dominated really mean near to optimal? In *Proceedings of the 2001 congress on evolutionary computation (IEEE cat. no.01th8546): 2* (pp. 957–962).

Ishibuchi, H., Masuda, H., Tanigaki, Y., & Nojima, Y. (2015). Modified distance calculation in generational distance and inverted generational distance. In A. Gaspar-Cunha, C. H. Antunes, & C. Coello Coello (Eds.), *Evolutionary multi-criterion optimization, 8th international conference, EMO 2015* (pp. 110–125). Guimarães, Portugal: Springer. Lecture Notes in Computer Science Vol. 9019.

Ishibuchi, H., & Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28*(3), 392–403. doi:10.1109/5326.704576.

Jaszkiewicz, A. (2002). On the performance of multiple-objective genetic local search on the 0/1 knapsack problem –A comparative experiment. *IEEE Transactions on Evolutionary Computation, 6*(4), 402–412. doi:10.1109/TEVC.2002.802873.

Jiang, S., Ong, Y.-S., Zhang, J., & Feng, L. (2014). Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Transactions on Cybernetics, 44*(12), 2391–2404.

Jiang, S., & Yang, S. (2017). A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Transactions on Evolutionary Computation, 21*(3), 329–346.

Jin, Y., Okabe, T., & Sendho, B. (2001). Adapting weighted aggregation for multiobjective evolution strategies. In *EMO 2001: Evolutionary multi-criterion optimization* (pp. 96–110). Springer.

Karaboga, D., & Akay, B. (2011). A modified artificial bee colony (abc) algorithm for constrained optimization problems. *Applied Soft Computing, 11*(3), 3021–3031.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Neural networks, 1995. proceedings., IEEEinternational conference on: 4* (pp. 1942–1948 vol.4).

Kishor, A., & Singh, P. K. (2016). Empirical study of grey wolf optimizer. In *Fifth international conference on soft computing for problem solving* (pp. 1037–1049). Springer.

Knowles, J., Thiele, L., & Zitzler, E. (2006). A tutorial on the performance assessment of stochastic multiobjective optimizers. *214*. Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland. Revised version

Larentis, A. L., de Resende, N. S., Salim, V. M. M., & Pinto, J. C. (2001). Modeling and optimization of the combined carbon dioxide reforming and partial oxidation of natural gas. *Applied Catalysis A: General, 215*(1), 211–224.

Li, H., & Zhang, Q. (2009). Multiobjective optimization problems with complicated pareto sets, MOEA/d and NSGA-II. *IEEE Transactions on Evolutionary Computation, 13*(2), 284–302.

Li, K., Deb, K., Zhang, Q., & Kwong, S. (2015). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation, 19*(5), 694–716.

Li, M., Yang, S., & Liu, X. (2014). Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation, 18*(3), 348–365.

Li, M., Yang, S., & Liu, X. (2016). Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 20*(5), 645–665.

Li, X., & Yin, M. (2012). Design of multiobjective reconfigurable antenna array with

discrete phase shifters using multiobjective differential evolution based on decomposition. *International Journal of RF and Microwave Computer-Aided Engineering, 22*(6), 675–681.
Liao, X., Li, Q., Yang, X., Zhang, W., & Li, W. (2008). Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization, 35*, 561–569.
Liu, Y., & Niu, B. (2013). A multi-objective particle swarm optimization based on decomposition. In D.-S. Huang, P. Gupta, L. Wang, & M. Gromiha (Eds.), *Emerging intelligent computing technology and applications: 9th international conference, ICIC 2013, Nanning, China, July 28–31, 2013. Proceedings* (pp. 200–205)). Berlin, Heidelberg: Springer Berlin Heidelberg.
Long, W., Liang, X., Cai, S., Jiao, J., & Zhang, W. (2017). A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. *Neural Computing and Applications, 28*(1), 421–438.
López Jaimes, A., & Coello Coello, C. A. (2015). Many-objective problems: Challenges and methods. In J. Kacprzyk, & W. Pedrycz (Eds.), *Springer handbook of computational intelligence* (pp. 1033–1046). Springer-Verlag Berlin Heidelberg. chapter 51
Malik, M. R. S., Mohideen, E. R., & Ali, L. (2015). Weighted distance grey wolf optimizer for global optimization problems. In *Computational intelligence and computing research (ICCIC), 2015 IEE international conference on* (pp. 1–6). IEEE.
Medina, M. A., Das, S., Coello, C. A. C., & Ramírez, J. M. (2014). Decomposition-based modern metaheuristic algorithms for multi-objective optimal power flow–a comparative study. *Engineering Applications of Artificial Intelligence, 32*, 10–20.
Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Boston, Massachuisetts: Kluwer Academic Publishers.
Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.
Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. d. S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications, 47*, 106–119.
Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing, 2016*, 8.
Mohanty, S. (2006). Multiobjective optimization of synthesis gas production using non-dominated sorting genetic algorithm. *Computers & Chemical Engineering, 30*(6), 1019–1025.
Moubayed, N. A., Petrovski, A., & McCall, J. (2014). *D^2MOPSO*: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evolutionary Computation, 22*(1), 47–77.
Moubayed, N. A., Petrovski, A., & McCall, J. A. W. (2010). A novel smart multi-objective particle swarm optimisation using decomposition. In *Ppsn (2)* (pp. 1–10).
Nedjah, N., & de Macedo Mourelle, L. (2015). Evolutionary multi-objective optimisation: A survey. *International Journal of Bio-Inspired Computation, 7*(1), 1–25.
Okabe, T., Jin, Y., & Sendhoff, B. (2003). A critical survey of performance indices for multi-objective optimization. In *Proceedings of the 2003 congress on evolutionary computation (CEC'2003): 2* (pp. 878–885). Canberra, Australia: IEEE Press.
Paquete, L., & Stützle, T. (2003). A two-phase local search for the biobjective traveling salesman problem. In *Emo 2003: Evolutionary multi-criterion optimization* (pp. 479–493). Springer.
Peng, W., & Zhang, Q. (2008). A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *2008 IEEE international conference on granular computing* (pp. 534–537).
Rao, R., Savsani, V., & Vakharia, D. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303–315.
Rodríguez, L., Castillo, O., & Soria, J. (2016). Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic. In *Evolutionary computation CEC), 2016 IEEEcongress on* (pp. 3116–3123). IEEE.
Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C. I., et al. (2017). A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Applied Soft Computing, 57*, 315–328.
Saremi, S., Mirjalili, S. Z., & Mirjalili, S. M. (2015). Evolutionary population dynamics and grey wolf optimizer. *Neural Computing and Applications, 26*(5), 1257–1263.
Scheffé, H. (1958). Experiments with mixtures. *Journal of the Royal Statistical Society Series B (Methodological, 20*(2), 344–360.
Sengupta, S., Nasir, M., Mondal, A. K., & Das, S. (2011). An improved multi-objective algorithm based on decomposition with fuzzy dominance for deployment of wireless sensor networks. In *Swarm, evolutionary, and memetic computing* (pp. 688–696). Springer Berlin Heidelberg.
Shim, V. A., Tan, K. C., & Tang, H. (2015). Adaptive memetic computing for evolutionary multiobjective optimization. *IEEE Transactions on Cybernetics, 45*(4), 610–621. doi:10.1109/TCYB.2014.2331994.
Sierra, M. R., & Coello, C. A. C. (2005). Improving pso-based multi-objective optimization using crowding, mutation and epsilon-dominance. In C. A. C. Coello, A. H. Aguirre, & E. Zitzler (Eds.), *Evolutionary multi-criterion optimization, third international conference, EMO 2005, Guanajuato, Mexico, march 9–11, 2005, proceedings*. In *Lecture Notes in Computer Science: 3410* (pp. 505–519). Springer.
Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation, 2*(3), 221–248.
Tian, Y., Zhang, X., Cheng, R., & Jin, Y. (2016). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *Evolutionary computation (CEC), 2016 IEEEcongress on* (pp. 5222–5229). IEEE.
Trivedi, A., Srinivasan, D., Pal, K., Saha, C., & Reindl, T. (2015). Enhanced multiobjective evolutionary algorithm based on decomposition for solving the unit commitment problem. *IEEE Transactions on Industrial Informatics, 11*(6), 1346–1357.
Trivedi, A., Srinivasan, D., Sanyal, K., & Ghosh, A. (2017). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation, 21*(3), 440–462.
Wang, Z., Ong, Y. S., & Ishibuchi, H. (2018). On scalable multiobjective test problems with hardly-dominated boundaries. *IEEE Transactions on Evolutionary Computation*. 1–1.
Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*(6), 80–83.
Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82. doi:10.1109/4235.585893.
Yuan, Y., Xu, H., Wang, B., & Yao, X. (2016). A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation, 20*(1), 16–37.
Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control, 8*(1), 59–60. doi:10.1109/TAC.1963.1105511.
Zapotecas-Martínez, S., & Coello Coello, C. A. (2011). A multi-objective particle swarm optimizer based on decomposition. In *Proceedings of the 13th annual conference on genetic and evolutionary computation*. In *GECCO '11* (pp. 69–76). New York, NY, USA: ACM.
Zapotecas-Martínez, S., López-Jaimes, A., & García-Nájera, A. (2018). Libea: A lebesgue indicator-based evolutionary algorithm for multi-objective optimization. *Swarm and Evolutionary Computation*. doi:10.1016/j.swevo.2018.05.004.
Zapotecas-Martínez, S., Sosa Hernández, V. A., Aguirre, H., Tanaka, K., & Coello Coello, C. A. (2014). Using a family of curves to approximate the pareto front of a multi-Objective optimization problem. In T. Bartz-Beielstein, J. Branke, B. Filipič, & J. Smith (Eds.), *Parallel problem solving from nature - ppsn xiii, 13th international conference* (pp. 682–691). Ljubljana, Slovenia: Springer. Lecture Notes in Computer Science Vol. 8672.
Zeleny, M. (1973). Compromise programming. In J. Cochrane, & M. Zeleny (Eds.), *Multiple criteria decision making* (pp. 262–301). Columbia: University of South Carolina Press.
Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation, 11*(6), 712–731.
Zhang, Q., Liu, W., & Li, H. (2009). The performance of a new version of moea/d on cec09 unconstrained mop test instances. In *2009 IEEE congress on evolutionary computation* (pp. 203–208).
Zhang, Q., Zhou, A., & Jin, Y. (2008a). RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation, 12*(1), 41–63.
Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., & Tiwari, S. (2008b). Multiobjective optimization test instances for the CEC 2009 special session and competition. *Technical Report CES-487*. School of Computer Science and Electrical Engineering, University of Essex.
Zhang, S., & Zhou, Y. (2015). Grey wolf optimizer based on powell local optimization method for clustering analysis. *Discrete Dynamics in Nature and Society, 2015*.
Zhang, Y., wei Gong, D., yong Sun, J., & yang Qu, B. (2018). A decomposition-based archiving approach for multi-objective evolutionary optimization. *Information Sciences, 430–431*, 397–413. doi:10.1016/j.ins.2017.11.052.
Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation, 1*(1), 32–49.
Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In X. Y. et al. (Ed.), *Parallel problem solving from nature - ppsn viii* (pp. 832–842). Birmingham, UK: Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.
Zitzler, E., Laumanns, M., & Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, & T. Fogarty (Eds.), *Eurogen 2001. evolutionary methods for design, optimization and control with applications to industrial problems* (pp. 95–100). Athens, Greece.
Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—A comparative study. In A. E. Eiben (Ed.), *Parallel problem solving from nature v* (pp. 292–301). Amsterdam: Springer-Verlag.
Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation, 7*(2), 117–132.