IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 17, 2019, accepted December 26, 2019, date of publication December 30, 2019, date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962906

Proposal and Comparative Study of Evolutionary Algorithms for Optimum Design of a Gear System

MÁXIMO MÉNDEZ[1], DANIEL A. ROSSIT[2,3], BEGOÑA GONZÁLEZ[1], AND MARIANO FRUTOS[2,4]

[1]Instituto Universitario SIANI, Universidad de Las Palmas de Gran Canaria (ULPGC), 35001 Las Palmas de Gran Canaria, Spain
[2]Department of Engineering, Universidad Nacional del Sur, Bahía Blanca 8000, Argentina
[3]INMABB-CONICET, Bahía Blanca 8000, Argentina
[4]IIESS-CONICET, Bahía Blanca 8002, Argentina

Corresponding author: Máximo Méndez (maximo.mendez@ulpgc.es)

This work was supported by CYTED (Grant P318RT0165) and the Instituto Universitario SIANI-ULPGC.

ABSTRACT This paper proposes a novel metaheuristic framework using a Differential Evolution (DE) algorithm with the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Both algorithms are combined employing a collaborative strategy with sequential execution, which is called DE-NSGA-II. The DE-NSGA-II takes advantage of the exploration abilities of the multi-objective evolutionary algorithms strengthened with the ability to search global mono-objective optimum of DE, that enhances the capability of finding those extreme solutions of Pareto Optimal Front (POF) difficult to achieve. Numerous experiments and performance comparisons between different evolutionary algorithms were performed on a referent problem for the mono-objective and multi-objective literature, which consists of the design of a double reduction gear train. A preliminary study of the problem, solved in an exhaustive way, discovers the low density of solutions in the vicinity of the optimal solution (mono-objective case) as well as in some areas of the POF of potential interest to a decision maker (multi-objective case). This characteristic of the problem would explain the considerable difficulties for its resolution when exact methods and/or metaheuristics are used, especially in the multi-objective case. However, the DE-NSGA-II framework exceeds these difficulties and obtains the whole POF which significantly improves the few previous multi-objective studies.

INDEX TERMS Differential evolution, evolutionary computation, gear train optimization, genetic algorithms, mechanical engineering, multi-objective evolutionary algorithms, non-dominated sorting genetic algorithm-II.

I. INTRODUCTION

Stochastic in nature, evolutionary algorithms (EAs) in their version of genetic algorithms (GAs) [1]–[3] have been applied effectively in science, engineering and engineering design [4]–[7]. The evolution of a population of solutions, using a parent selection process first, and reproduction operations (crossower and mutation) on the selected parents later, provides the GAs with great skill to find an approximate optimal solution to problems of high computational complexity. In these algorithms, to ensure the improvement of solutions quality, the selection operator is decisive and must be carefully chosen [8], [9]. Differential evolution (DE) is a simple yet powered stochastic real-parameter global optimization algorithm [10]–[12]. It is not inspired by natural evolution but, it uses computer operators similar to those employed by a standard EA. Each population consists of individuals called parameter vectors or genomes. Based on the concept of vector difference, DE combines with certain probability the components of randomly selected and distinct existing individuals to generate new individuals. Particle swarm optimisation (PSO) is an efficient population-based search method inspired by the behaviour of flocks of birds or schools of fish [13]–[16]. Each population consists of particles that move over the search space. In each iteration of the method, each of the particles in the population moves through the search space at a certain speed according to its own experience (the best individual solution of the particle in the search history) and with the experience provided by the best global solution found so far. In order to speed up the PSO convergence, a simplified PSO, that uses the global best only, called an accelerated PSO (APSO), was proposed in [17]. Also, to increase the

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

convergence even further, a simpler version of APSO with the same order of convergence was proposed in [18].

In engineering, many real optimization problems require meeting, simultaneously, multiple objectives in conflict. In this multi-objective context, the multi-objective evolutionary algorithms (MOEAs) [19], [20] have demonstrated excellent skills to generate, not a single solution, but yes an approximate set of non-dominated solutions called Pareto optimal front (POF). The non-dominated sorting genetic algorithm-II (NSGA-II) [21], the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [22] and the global weighting achievement scalarizing function genetic algorithm (GWASF-GA) [23] are well-known and state-of-the-art MOEAs. NSGA-II is based mainly on two mechanisms: Pareto dominance as a criterion to converge to the Pareto front and crowding-distance operator as augmenting diversification in the population. MOEA/D uses a strategy of breaking down the multi-objective problem (MOP) into a number of scalar subproblems that are solved simultaneously through the evolution of a population of solutions. For the scalarization of subproblems MOEA/D can use different approaches, such as the Tchebycheff approach which works well for combinatorial problems. GWASF-GA, similar to NSGA-II, classifies solutions on Pareto fronts but based on the achievement scalarizing function of Wierzbicki [24]. Despite the advances made by these and other state-of-the-art algorithms, it is still difficult to incorporate diversity in MOPs, specially, when MOPs have many decision variables [25] or many objectives to considers [26], [27]. In addition, in [28] the difficulties of convergence of MOEAs when they are applied to problems that present areas with poor solution densities are pointed out, as will be verified in this paper.

Hybrid techniques combine heuristics and/or metaheuristics (MH) in order to combine their advantages and obtain better yields than those obtained by applying each algorithm separately. Several classifications have been proposed for hybrid metaheuristics [29], [30], [32]. In fact, a significant number of hybrid combinations MH + MH have been formulated. Among the DE + MOEA hybrids, that are an extension of DE to solve MOPs, are those based on decomposition strategy such as MOEA/D-DE [33] and MOEA/D-I [34], and those that use Pareto-based ranking assignment and crowding distance such as NSDE [35], DEMO [36] and MODE [37]. MOEA/D-DE uses a DE operator and a polynomial mutation operator to produce new solutions, and also uses two measures to maintain population diversity. Hybrid-MOEA/D-I optimizes each subproblem by combining GA and DE operators as a mixed reproduction operator, with the aim of diversifying the search. In MODE the best individual is added to the population to generate offspring. In DEMO the new solutions are immediately inserted in the population allowing them to be candidates to be selected as parents in the next generation. NSDE is a simple modification for real representation of the NSGA-II where crossover and mutation operators are replaced by those adopted in DE.

Another approach that is based on the synergy between different algorithms are frameworks methods [38]. Unlike hybrid methods, frameworks maintain the complete structure of the intervening algorithms. This allows the frameworks to alternate, in the same optimization process (i.e. a single run), different optimization strategies achieving very good results. These methods have shown their potential in various optimization problems. For example, in [39] the evolutionary scatter search and PSO are associated to solve a routing problem with time windows. While for the same routing problem but in a multi-objective version, a framework based on MOEA/D and heuristics is proposed [40]. In [41] a framework based on GA and PSO is developed to solve data mining problems. Also in the area of detection of cancer genes framework based methods have been used, as in the case of [42]. In this case the authors propose a framework based on harmony search and GA. As well there have been problems in the chemical industry addressed by frameworks methods, like in [43], where the framework uses APSO and SQP algorithms. Note that in this latter case an EA is associated with an exact method.

This work solves the optimal design of a double reduction gear train, which consists of sprockets that engage each other to transmit power. Double reduction gears are generally used in applications that require very high speeds. A good number of MH have been proposed for the resolution of this type of systems. The problem considered here (multi-objective case) has the special feature of having very low density of solutions in some of the areas near to the POF. For tackling this problem, a novel metaheuristic framework called DE-NSGA-II is proposed, which uses a collaborative combination with sequential execution [30] of the DE and NSGA-II algorithms. This metaheuristic framework allows to diversify the search of solutions, especially in areas with low solution density, and obtain a better approximation of the POF. An original enumeration study of the problem is also presented, which reveals why its resolution is so difficult using both exact and metaheuristic algorithms. Finally, a broad comparison of performance was made between different EAs (mono-objective and multi-objectives), showing the advantages of DE-NSGA-II for the multi-objective case.

This paper has been structured as follows. Section II introduces the problem of optimizing the design of a double reduction gear train and presents a small review of the literature as well as a brief description of some existing algorithms used in the literature to solve this problem. Section III details the proposed metaheuristic framework. The experiments and results achieved are discussed in section IV. Finally, section V describes the conclusions and future work.

II. GEAR TRAIN DESIGN PROBLEM. PRELIMINARY STUDY

The gear train design problem addressed in this work is a problem well-studied in the literature and was proposed in [44]. The problem consists in the design of a gear train to reduce the entry angular speed of the train, to a lower

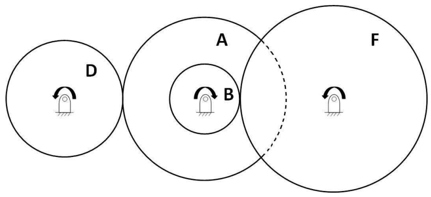

FIGURE 1. Double reduction gear train.

departure speed. The angular velocity variation, i.e., the two-gear transfer ratio, n, is usually described as:

$$n = \frac{\omega_0}{\omega_i} = \frac{t_i}{t_0} \tag{1}$$

where ω_0 is the angular velocity of the output gear, ω_i is the angular velocity of the input gear, and t_0 and t_i are the number of teeth of the output and input gears, respectively. Therefore, the transmission ratio is inversely proportional to the number of teeth of the gears. Fig. 1 illustrates the problem considered for this study, which contemplates 2 pairs of gears (4 gears in total) and aims to bring the transmission ratio as close as possible to the value $1/6.931$. Then, the transmission equation (1) can be rewritten for this problem as:

$$n = \frac{1}{6.931} = \frac{t_D t_B}{t_A t_F} = \frac{x_1 x_2}{x_3 x_4} \tag{2}$$

On the other hand, Sandgren [44] proposed that none of the gears had less than 12 teeth or more than 60. Thus, the gear train design problem seeks to obtain a gear set (x_1, x_2, x_3 and x_4) so that the double reduction gear train is as close as possible to $1/6.931$ and respect the feasibility conditions, i.e., each design variables x_i is an integer in the range $[12, 60]$. Formally, this problem can be defined as follows:

$$\begin{aligned} &min\, f(x) = [\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}]^2 \\ &s.t.\ 12 \leq x_i \leq 60 \quad i = 1, 2, 3, 4 \end{aligned} \tag{3}$$

The problem (3) was first proposed and solved in [44] using the branch and bound method (BB) and, later, Kannan and Kramer [45] addressed it using the augmented Lagrange multiplier (AL) method. Since then, a good number of meta-heuristic algorithms have been used to solve this problem, for example, the combined genetic adaptive search (Gene AS) [46], [47] (the principal difference between Gene AS and GA is the mode the variables are coded and the mode crossover and mutation operators are applied), the mine blast algorithm (MBA) [48] (whose idea emerges from the analysis of the landmine explosion), the biogeography-based optimization algorithm [49] (inspired by the way in which biological species are distributed in time and space), the electromagnetism optimization algorithm [50] (where a hybridization of the electromagnetism-like mechanism with a descent search is performed) and intelligent swarm-inspired algorithms such as the cuckoo search (CS) algorithm [51] (which combines the parasitic behaviour of breeding some species of cuckoo with Lévy's flight strategy of some birds and fruit flies), the locust search II algorithm [52] (inspired by the gregarious behaviour observed in swarms of desert lobsters), the unified particle swarm optimization (UPSO) algorithm [53] (basically, this algorithm articulates the local and global variant of the standard PSO as a unified mechanism), the APSO algorithm [17], the improved accelerated PSO (IAPSO) algorithm [54] (in essence, this algorithm is characterized by the replacement of the position of a particle by the best position, giving a memory to the APSO algorithm, and the introduction of two selected functions to ensure a balance of exploration and exploitation, during search process) and the hybrid PSO-GA algorithm [55] (PSO and GA techniques are combined, in particular, crossover and mutation operators, are included in the standard PSO algorithm).

Using multi-objective optimization concepts, the problem (3) can be redefined as a multi-objective optimization problem as in (4). To this end, a new objective function, that minimizes the maximum size of any of the four gears, has been added. Initially, this problem was defined in [56]. The interest now, is not to have a single solution but rather a more balanced set of solutions that could interest a potential decision maker, since the new objective tends to reduce costs. Only a few authors, and with relative success, have solved this problem. In [56] NSGA-II is applied and the Inverted and shrinkable Pareto archived evolutionary strategies (ISPAES) algorithm, which modifies the Pareto archived evolution strategy (PAES) algorithm [57], is used in [58].

$$\begin{aligned} &min\, f_1(x) = [\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}]^2 \\ &min\, f_2(x) = max(x_1, x_2, x_3, x_4) \\ &s.t.\ 12 \leq x_i \leq 60 \quad i = 1, 2, 3, 4 \end{aligned} \tag{4}$$

The problem (4) can be solved by explicit enumeration using 6 bits (which allow to represent $2^6 = 64$ different binary solutions) for the binary coding of the values of each variable $x_i, i = 1, 2, 3, 4$. This implies an exploration space of $64^4 = 16, 777, 216$ solutions. Considering that the values of each variable are restricted to the range [12, 60] and reducing the numerous solutions of the decision space that produce identical solutions in the objective space (overlapping solutions [59]), a total of 662,165 feasible non-overlapping solutions are obtained. Fig. 2 shows the feasible non-overlapping solutions of the problem. A poor density of solutions can be seen in the area near the Pareto front with low values of $f_1(x)$. This low density may cause important difficulties to find good solutions in this area for MOEAs, solutions that may be of potential interest to a decision maker. The 28 solutions that constitute the true POF are also shown in Fig. 2. Their numerical values and their respective desings (i.e., the value of the decision variables) are shown in Table 1. The maximum values found for f_1 and f_2 were, respectively, $f_1 = 617.807$ and $f_2 = 60$. These values are used for the estimation of the reference point of the hypervolume metric [60] used in Section IV to make comparisons.

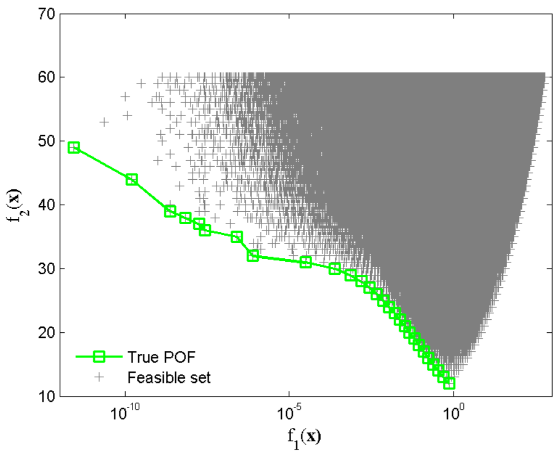

FIGURE 2. Pareto optimal front and feasible set of solutions for the double reduction gear train design problem obtained by explicit enumeration.

TABLE 1. Numerical values of the Pareto optimal front solutions and their respective design variables, for the double reduction gear train design problem, obtained by explicit enumeration.

$f_1(x)$	$f_2(x)$	x_1 x_2 x_3 x_4
2.700857148886513e-12	49	19 16 43 49
1.545045049989077e-10	44	21 13 43 44
2.357640658024884e-09	39	12 15 32 39
6.654885712880334e-09	38	16 12 35 38
1.827380235299579e-08	37	12 12 27 37
2.726450597715287e-08	36	13 12 30 36
2.505232028972835e-07	35	13 12 31 35
7.778632310703362e-07	32	12 12 31 32
3.096463758747167e-05	31	12 12 31 31
2.471396296003555e-04	30	12 12 30 30
7.260549955046195e-04	29	12 12 29 29
1.551898629947143e-03	28	12 12 28 28
2.835726451107918e-03	27	12 12 27 27
4.724971306088980e-03	26	12 12 26 26
7.416770701595737e-03	25	12 12 25 25
1.117686117050354e-02	24	12 12 24 24
1.636669780661988e-02	23	12 12 23 23
2.348290717683538e-02	22	12 12 22 22
3.321553178525184e-02	21	12 12 21 21
4.653540972049632e-02	20	12 12 20 20
6.482759745921461e-02	19	12 12 19 19
9.009909906751659e-02	18	12 12 18 18
1.253093246549743e-01	17	12 12 17 17
1.749085331875285e-01	16	12 12 16 16
2.457389878477507e-01	15	12 12 15 15
3.485893441323532e-01	14	12 12 14 14
5.009690638972156e-01	13	12 12 13 13
7.322578740113634e-01	12	12 12 12 12

III. PROPOSED METAHEURISTIC FRAMEWORK

For addressing the problem described in the previous section, a novel metaheuristic framework using a DE algorithm with the NSGA-II is proposed. First, the DE and NSGA-II algorithms are presented and then the proposed metaheuristic framework.

A. DIFFERENTIAL EVOLUTION ALGORITHM

DE searches for a global optimum in a s-dimensional real parameter space $D \subseteq R^s$. It begins with a randomly initiated population of M s-dimensional real-valued parameter vectors. Each vector forms a candidate solution to the optimization problem. Once the initial population is generated, the mutation, crossover and selection operators follow each other iteratively until a stop criterion is met.

Mutation (*DE/rand/1*). For each i-th target vector from the current population, P^g, three other distinct parameter vectors, say $\mathbf{x}^g_{r_1 i}$, $\mathbf{x}^g_{r_2 i}$ and $\mathbf{x}^g_{r_3 i}$ are sampled randomly. The indices r_{1i}, r_{2i} and r_{3i} are mutually different integers randomly chosen from the range $[1, M]$, which are also different from the index i. Then Eq. (5) is applied:

$$\mathbf{m}^g_i = \mathbf{x}^g_{r_1 i} + F(\mathbf{x}^g_{r_2 i} - \mathbf{x}^g_{r_3 i}), \quad \text{for} \quad i = 1, 2, \ldots, M \qquad (5)$$

where $F \in (0, 2]$ is a real constant factor determined by the user, which controls the amplification of the differential variation $(\mathbf{x}^g_{r_2 i} - \mathbf{x}^g_{r_3 i})$. The higher the value of F, the more the search space is explored, while the smaller the value of F, the more the search space is exploited. Finally, if any component of the mutant vector is outside its definition range, we match it to the nearest bound.

Crossover. In this step, the trial vectors $\mathbf{t}^g_i$ are generated by randomly mixing the components of the mutant vectors $\mathbf{m}^g_i$ with those of the target individuals $\mathbf{x}^g_i$, as follows, for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, s$:

$$t^g_{i,j} = \begin{cases} m^g_{i,j} & \text{if } rand_{i,j}[0,1] < Cr \\ x^g_{i,j} & \text{if } rand_{i,j}[0,1] \geq Cr \end{cases} \qquad (6)$$

where $Cr \in (0, 1)$ is the crossover constant also determined by the user and $rand_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1. A value of Cr close to the unit gives rise to trial vectors constructed mainly from the components of the mutant vectors, while a value close to zero will generate trial vectors constructed mainly from the components of the target individuals.

Selection. To keep the population size constant in all generations, selection is applied to determine, between the trial vectors and the target ones, which vector goes to the next generation. Let be $f(\mathbf{x})$ the objective function to be minimized, the selection operation is described as

$$\mathbf{x}^{g+1}_i = \begin{cases} \mathbf{t}^g_i & \text{if } f(\mathbf{t}^g_i) \leq f(\mathbf{x}^g_i) \\ \mathbf{x}^g_i & \text{if } f(\mathbf{t}^g_i) > f(\mathbf{x}^g_i) \end{cases}, \quad \text{for} \quad i = 1, 2, \ldots, M \qquad (7)$$

B. NON-DOMINATED SORTING GENETIC ALGORITHM II

Of recognized efficiency, the general operation of NSGA-II [21] follows the following scheme: (i) random creation of a first population of parents P^0 of size N, (ii) selection by tournament (best rank and crowding) and application of crossover and mutation operators to create a population of descendants Q^g of size N, (iii) form a combined population $R^g = P^g \cup Q^g$, (iv) classify R^g on non-dominated fronts, (v) then add each front, in increasing order until a front F_k cannot be fully integrated into the new population P^{g+1} of size N, then classifying the F_k front by crowding-distance to add solutions until the size P^{g+1} equals N, (vi) if the stop condition is not met return to step (ii).

C. DE-NSGA-II FRAMEWORK

The proposed framework improves the ability to search for the POF of a MOEA when highly complex problems are solved and, in particular, the one presented in section II. The basic idea is to take advantage of the skills of a MOEA and enrich them with the global mono-objective search skills of DE, in order to find those extreme POF solutions that are difficult to achieve. In this work, NSGA-II and DE are used, however another metaheuristic could be used, although the performance could be different. The general flow chart of the proposed algorithm (see Fig. 3) consists of the following steps:

Step 0. Define the initial parameters of both NSGA-II ($N, PCr, PMut, G$) and DE (M, F, Cr, G_{DE}) *Step 1*. Randomly generate an initial population P^0 (NSGA-II population) of size N.
Step 2. Assign $P^g = P^0$.
Step 3. Classify by dominance P^g in Pareto fronts, $\{F_i\}$.
Step 4. Generate the initial population of DE, U^0, from the $M(\leq N)$ best solutions of P^g (best rank and crowding)
Step 5. Run DE up to stop criterion is met.
Step 6. Include in P^g the elitist solutions from DE.
Step 7. Obtain P^{g+1}.
Step 8. Assign $P^g = P^{g+1}$.
Step 9. If NSGA-II stop criterio is not met, go to Step 3.

Fig. 3 shows how DE and NSGA-II exchange the best solutions from the mono-objective point of view. Although in the problem considered in this work it is enough to make this exchange with only one of the objectives, it can be easily extended to more or even all the objectives of the problem considered. Thus, if k is the number of objectives to optimize and r is the number of objectives represented by the hard-to-find POF extremes, the complexity of DE-NSGA-II is $O(kN^2 + rMG_{DE})$. Consequently, the greater the number of objectives, the greater the complexity of the algorithm. One way to overcome this limitation is to use parallel programming.

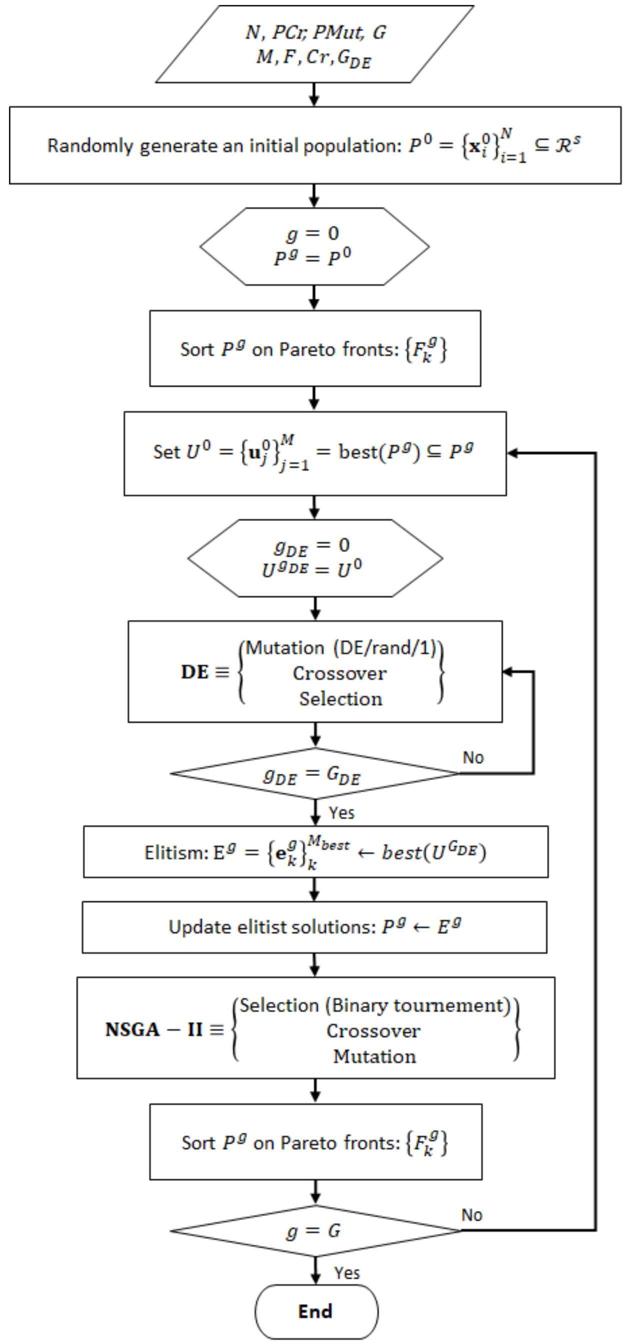

FIGURE 3. Flow chart of DE-NSGA-II.

IV. COMPARATIVE RESULTS

In the following subsections, a comparative analysis is performed between some of the most representative algorithms of both the mono-objective EA family and the MOEA family, and the DE-NSGA-II framework proposed in this work. All experiments are carried out with the problem of designing a double reduction gear train defined in section II. Each algorithm has been independently run 100 times for each test instance in the same initial conditions, from a randomly generated population. The settings and metrics used to evaluate the algorithms are presented in subsection IV-A. The mono-objective problem defined by eq. (3) is solved in subsection IV-B and the multi-objective problem raised in eq. (4) is solved in both subsections IV-C and IV-D.

A. SETTINGS AND METRICS

The APSO, DE and NSDE algorithms were implemented with real coding. Comparing with many PSO variants, APSO uses only two parameters (α and β) and the mechanism is simple to understand. In this work a variant of APSO proposed in [18] has been implemented, which reduces randomness as iterations progress. In particular, $\alpha = \delta^t$ is considered where $\delta \in (0, 1)$ and t represents the current iteration of the algorithm. DE also uses two parameters (F and Cr). In all the experiments, we will use the same settings of APSO and DE parameters, that is, we have used the learning parameters $\beta = 0.6$ and $\delta = 1.0$, the mutation factor $F = 0.3$ and the crossover factor $Cr = 0.9$. Prior to the selection of these parameters, 100 runs of the APSO and DE algorithms

TABLE 2. APSO ($N = 50$ and $G = 100$): the best fitness values over 100 independent runs for different values of β and δ. The global optimum is highlighted in bold.

Best fitness	$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$	$\beta = 1.0$
$\delta = 0.0$	6.19e-09	2.73e-08	2.73e-08	7.13e-08	6.19e-09	1.31e-08	1.18e-09	2.36e-09	8.32e-09	2.06e-08	2.18e-08
$\delta = 0.1$	2.33e-07	1.09e-07	3.37e-08	6.07e-08	1.56e-07	2.73e-08	2.36e-09	2.36e-09	2.31e-11	2.36e-09	8.92e-08
$\delta = 0.2$	2.36e-09	1.18e-09	2.86e-08	5.52e-08	1.46e-08	2.36e-09	2.36e-09	2.36e-09	9.52e-09	2.73e-08	2.36e-09
$\delta = 0.3$	5.50e-08	4.50e-09	1.39e-08	1.17e-10	1.83e-08	1.55e-10	2.18e-08	1.36e-09	2.36e-09	2.31e-11	2.36e-09
$\delta = 0.4$	2.73e-08	2.18e-08	4.20e-07	2.36e-09	1.36e-09	1.18e-09	4.47e-08	**2.70e-12**	1.31e-08	9.92e-10	2.31e-11
$\delta = 0.5$	2.36e-09	3.30e-09	2.36e-09	3.07e-10	2.36e-09	5.35e-08	**2.70e-12**	2.36e-09	2.36e-09	4.92e-09	1.55e-10
$\delta = 0.6$	2.73e-08	1.09e-07	6.19e-09	1.17e-10	3.45e-09	2.31e-11	2.36e-09	2.36e-09	2.36e-09	1.12e-08	**2.70e-12**
$\delta = 0.7$	2.46e-08	1.79e-07	5.52e-08	2.73e-08	1.16e-07	1.55e-10	9.52e-09	**2.70e-12**	1.18e-09	8.32e-09	2.36e-09
$\delta = 0.8$	4.63e-07	2.31e-11	2.31e-11	2.36e-09	1.18e-09	2.73e-08	2.36e-09	9.92e-10	1.17e-10	9.92e-10	2.31e-11
$\delta = 0.9$	1.12e-08	2.36e-09	**2.70e-12**	**2.70e-12**	1.36e-09	1.18e-09	9.92e-10	1.18e-09	1.17e-10	1.55e-10	6.92e-10
$\delta = 1.0$	1.17e-10	**2.70e-12**	2.31e-11	1.55e-10	2.31e-11	2.31e-11	**2.70e-12**	2.31e-11	2.31e-11	2.31e-11	2.31e-11

TABLE 3. APSO ($N = 50$ and $G = 100$): mean values over 100 independent runs for different values of β and δ. Mean values less than 1.e-04 are highlighted in bold.

Mean	$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$	$\beta = 1.0$
$\delta = 0.0$	2.54e-03	2.11e-03	1.58e-03	2.04e-03	2.15e-03	3.01e-03	1.42e-03	2.11e-03	1.68e-03	1.68e-03	1.63e-03
$\delta = 0.1$	2.49e-03	2.87e-03	2.38e-03	2.20e-03	2.26e-03	1.74e-03	1.27e-03	1.17e-03	1.16e-03	2.18e-03	2.27e-03
$\delta = 0.2$	2.49e-03	1.91e-03	2.17e-03	1.87e-03	1.49e-03	1.74e-03	2.09e-03	2.48e-03	1.52e-03	2.65e-03	1.88e-03
$\delta = 0.3$	3.21e-03	2.19e-03	2.22e-03	2.37e-03	2.91e-03	1.95e-03	2.11e-03	2.78e-03	1.51e-03	2.65e-03	1.39e-03
$\delta = 0.4$	2.99e-03	2.68e-03	1.90e-03	1.74e-03	3.78e-03	2.36e-03	1.48e-03	2.08e-04	2.20e-03	2.62e-03	2.31e-03
$\delta = 0.5$	2.84e-03	3.53e-03	2.65e-03	2.08e-03	1.53e-03	2.51e-03	1.95e-03	2.40e-03	2.02e-03	2.02e-03	1.46e-03
$\delta = 0.6$	3.71e-03	1.96e-03	2.78e-03	1.62e-03	2.88e-03	2.06e-03	1.44e-03	2.21e-03	1.56e-03	1.76e-03	1.51e-03
$\delta = 0.7$	1.54e-03	2.20e-03	1.88e-03	2.65e-03	2.26e-03	1.27e-03	2.24e-03	1.41e-03	1.81e-03	1.19e-03	1.53e-03
$\delta = 0.8$	2.59e-03	2.61e-03	2.57e-03	1.91e-03	9.79e-04	1.28e-03	2.04e-03	1.35e-03	2.24e-03	1.87e-03	7.38e-04
$\delta = 0.9$	2.10e-03	1.66e-03	1.24e-03	1.41e-03	1.29e-03	1.69e-03	1.14e-03	1.04e-03	1.25e-03	6.08e-04	6.75e-04
$\delta = 1.0$	7.12e-04	**5.69e-06**	**4.74e-06**	**1.16e-05**	2.52e-04	**5.99e-05**	**5.10e-06**	**5.27e-06**	**6.93e-05**	**2.33e-05**	**7.47e-06**

TABLE 4. APSO ($N = 50$ and $G = 100$): standard deviation values over 100 independent runs for different values of β and δ. Standard deviation values less than 1.e-04 are highlighted in bold.

SD	$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$	$\beta = 1.0$
$\delta = 0.0$	4.57e-03	4.30e-03	2.97e-03	3.95e-03	4.17e-03	7.42e-03	3.51e-03	4.49e-03	3.20e-03	4.07e-03	3.77e-03
$\delta = 0.1$	3.91e-03	6.63e-03	5.94e-03	3.60e-03	4.31e-03	3.96e-03	2.90e-03	2.47e-03	2.36e-03	4.77e-03	4.62e-03
$\delta = 0.2$	4.95e-03	3.78e-03	4.87e-03	3.25e-03	3.17e-03	4.00e-03	5.40e-03	4.70e-03	2.74e-03	6.82e-03	3.08e-03
$\delta = 0.3$	6.04e-03	3.72e-03	5.15e-03	4.37e-03	6.16e-03	4.51e-03	3.58e-03	5.79e-03	2.95e-03	3.66e-03	2.76e-03
$\delta = 0.4$	5.19e-03	5.76e-03	3.84e-03	3.02e-03	9.90e-03	5.79e-03	3.63e-03	4.57e-03	4.35e-03	5.24e-03	5.07e-03
$\delta = 0.5$	5.97e-03	9.44e-03	5.12e-03	3.77e-03	3.20e-03	5.82e-03	4.44e-03	8.97e-03	4.14e-03	4.94e-03	2.90e-03
$\delta = 0.6$	6.54e-03	4.01e-03	5.85e-03	3.75e-03	5.70e-03	3.99e-03	3.72e-03	3.98e-03	2.87e-03	3.96e-03	3.35e-03
$\delta = 0.7$	2.66e-03	4.90e-03	3.93e-03	4.29e-03	4.94e-03	2.73e-03	5.93e-03	3.43e-03	3.65e-03	2.64e-03	5.41e-03
$\delta = 0.8$	5.36e-03	6.80e-03	5.24e-03	4.57e-03	2.18e-03	2.89e-03	5.10e-03	3.10e-03	6.25e-03	3.80e-03	2.51e-04
$\delta = 0.9$	3.90e-03	3.17e-03	3.12e-03	3.03e-03	3.09e-03	4.55e-03	3.19e-03	3.44e-03	3.23e-03	1.50e-03	2.18e-03
$\delta = 1.0$	1.92e-03	**5.35e-05**	**2.71e-05**	1.12e-04	1.69e-03	4.95e-04	**3.50e-05**	**2.11e-05**	4.92e-04	1.38e-04	**5.46e-05**

with a population size $N = 50$ and a maximum number of generations $G = 100$ were performed to compare the different effects of β, δ, F, and Cr in order to find the best results. The experimental results are recorded in Tables 2–7. From Tables 2–4, it can be clearly seen that the best values of mean and standard deviation correspond to $\delta = 1.0$. However, the results seem insensitive to parameter β but as, in general, APSO performs slightly differently with different β, we set $\beta = 0.6$. From Tables 5–7, it can be clearly seen that the best values of mean and standard deviation correspond to $Cr = 0.9$ and $F = 0.3$ or $F = 0.4$. We set $F = 0.3$.

The GA, NSGA-II, GWASF-GA and MOEA/D algorithms were implemented with binary coding, and binary tournament selection operator, uniform crossover rate of 0.8 and bitwise mutation rate of $1/L$ were used, where $L = 24$ is the string length (each variable uses 6 bits). These parameters values were used with good performance in [21]. In addition, for the GA the size of the elitist population was 3 and a probabilistic binary tournament selection (the best solution is chosen with probability [0, 0.7]) was adopted [9]. The parameters used for the DE-NSGA-II are a combination of those specified above for both the DE and the NSGA-II.

The population sizes, N and/or M, and the maximum number of generations, G and/or G_{DE}, used in each of the algorithms are indicated in the corresponding subsection for each experiment. As recommended by the authors, the value

TABLE 5. DE ($N = 50$ and $G = 100$): the best fitness values over 100 independent runs for different values of F and C_r. The global optimum is highlighted in bold.

Best fitness	$F = 0.0$	$F = 0.1$	$F = 0.2$	$F = 0.3$	$F = 0.4$	$F = 0.5$	$F = 0.6$	$F = 0.7$	$F = 0.8$	$F = 0.9$	$F = 1.0$
$Cr = 0.0$	8.58e-07	4.50e-09	1.40e-07	1.88e-07	9.75e-10	4.47e-08	2.36e-09	3.85e-07	1.61e-07	2.73e-08	4.76e-07
$Cr = 0.1$	2.31e-11	2.31e-11	**2.70e-12**	**2.70e-12**	**2.70e-12**	2.31e-11	**2.70e-12**	2.31e-11	8.89e-10	1.17e-10	**2.70e-12**
$Cr = 0.2$	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	2.31e-11	1.17e-10	9.94e-11
$Cr = 0.3$	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	2.31e-11	2.31e-11	2.31e-11
$Cr = 0.4$	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	2.31e-11	2.31e-11	6.60e-10
$Cr = 0.5$	9.94e-11	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**
$Cr = 0.6$	2.31e-11	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**
$Cr = 0.7$	9.94e-11	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**
$Cr = 0.8$	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**
$Cr = 0.9$	2.31e-11	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**
$Cr = 1.0$	1.36e-09	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**	**2.70e-12**

TABLE 6. DE ($N = 50$ and $G = 100$): mean values over 100 independent runs for different values of F and C_r. Mean values less than 1.e-10 are highlighted in bold.

Mean	$F = 0.0$	$F = 0.1$	$F = 0.2$	$F = 0.3$	$F = 0.4$	$F = 0.5$	$F = 0.6$	$F = 0.7$	$F = 0.8$	$F = 0.9$	$F = 1.0$
$Cr = 0.0$	1.63e-03	1.82e-03	3.28e-03	2.09e-03	2.73e-03	2.70e-03	2.16e-03	3.14e-03	3.28e-03	1.98e-03	1.80e-03
$Cr = 0.1$	1.69e-08	6.50e-09	6.57e-09	6.52e-09	1.03e-08	1.69e-08	1.48e-08	1.54e-08	6.57e-09	2.29e-08	1.99e-08
$Cr = 0.2$	1.04e-08	2.31e-09	3.62e-09	4.28e-09	3.61e-09	4.50e-09	6.58e-09	7.14e-09	3.62e-09	9.36e-09	1.14e-08
$Cr = 0.3$	2.11e-08	1.63e-09	1.74e-09	2.55e-09	3.18e-09	4.17e-09	4.72e-09	3.92e-09	1.74e-09	6.05e-09	7.33e-09
$Cr = 0.4$	2.06e-08	1.29e-09	1.14e-09	1.64e-09	1.87e-09	2.61e-09	3.32e-09	3.27e-09	1.14e-09	5.06e-09	6.74e-09
$Cr = 0.5$	5.90e-08	7.55e-10	7.05e-10	1.64e-09	1.54e-09	2.24e-09	2.28e-09	2.39e-09	7.05e-10	4.52e-09	6.04e-09
$Cr = 0.6$	9.59e-08	6.41e-10	5.42e-10	5.13e-10	9.26e-10	1.10e-09	1.82e-09	2.39e-09	5.42e-10	3.40e-09	3.67e-09
$Cr = 0.7$	1.76e-07	9.80e-10	3.29e-10	4.24e-10	5.74e-10	9.05e-10	9.83e-10	1.50e-09	3.29e-10	2.22e-09	3.89e-09
$Cr = 0.8$	3.95e-07	1.77e-09	2.79e-10	2.04e-10	2.70e-10	5.45e-10	6.51e-10	9.68e-10	2.79e-10	1.49e-09	2.62e-09
$Cr = 0.9$	3.18e-06	4.93e-09	3.43e-10	**9.40e-11**	**9.96e-11**	1.94e-10	4.09e-10	8.44e-10	3.43e-10	1.31e-09	1.44e-09
$Cr = 1.0$	1.69e-03	2.20e-04	3.34e-09	1.26e-10	1.17e-10	1.08e-10	1.59e-10	3.47e-10	3.34e-09	7.59e-10	1.59e-09

TABLE 7. DE ($N = 50$ and $G = 100$): standard deviation values over 100 independent runs for different values of β and δ. Standard deviation values less than 2.5e-10 are highlighted in bold.

SD	$F = 0.0$	$F = 0.1$	$F = 0.2$	$F = 0.3$	$F = 0.4$	$F = 0.5$	$F = 0.6$	$F = 0.7$	$F = 0.8$	$F = 0.9$	$F = 1.0$
$Cr = 0.0$	2.95e-03	3.33e-03	6.49e-03	3.86e-03	5.82e-03	5.85e-03	5.10e-03	7.26e-03	5.35e-03	4.90e-03	3.84e-03
$Cr = 0.1$	3.39e-08	1.09e-08	1.37e-08	9.34e-09	1.43e-08	4.85e-08	2.80e-08	2.73e-08	2.14e-08	3.40e-08	2.58e-08
$Cr = 0.2$	1.42e-08	3.19e-09	5.73e-09	7.32e-09	5.35e-09	6.39e-09	9.45e-09	8.48e-09	9.30e-09	1.32e-08	1.36e-08
$Cr = 0.3$	6.23e-08	2.05e-09	2.31e-09	4.30e-09	5.02e-09	6.48e-09	6.16e-09	4.98e-09	9.15e-09	8.13e-09	8.27e-09
$Cr = 0.4$	4.11e-08	2.88e-09	1.18e-09	2.82e-09	2.56e-09	4.39e-09	4.16e-09	5.03e-09	6.01e-09	7.22e-09	8.47e-09
$Cr = 0.5$	1.59e-07	7.86e-10	6.45e-10	2.82e-09	3.03e-09	2.99e-09	2.77e-09	3.78e-09	5.72e-09	5.71e-09	7.30e-09
$Cr = 0.6$	1.77e-07	8.62e-10	6.80e-10	6.71e-10	1.06e-09	1.37e-09	2.87e-09	3.47e-09	3.70e-09	5.69e-09	5.38e-09
$Cr = 0.7$	3.95e-07	1.03e-09	4.87e-10	6.22e-10	6.74e-10	8.28e-10	1.05e-09	1.33e-09	3.18e-09	3.32e-09	5.25e-09
$Cr = 0.8$	8.57e-07	3.76e-09	4.76e-10	4.04e-10	4.65e-10	7.04e-10	6.84e-10	9.69e-10	1.97e-09	1.25e-09	3.94e-09
$Cr = 0.9$	8.30e-06	1.45e-08	5.33e-10	**2.32e-10**	**2.24e-10**	3.52e-10	6.08e-10	1.79e-09	8.29e-10	1.79e-09	1.56e-09
$Cr = 1.0$	2.80e-03	1.25e-03	2.29e-08	2.75e-10	2.71e-10	2.62e-10	3.34e-10	5.94e-10	7.42e-10	7.71e-10	5.51e-09

of the T parameter in GWASF-GA [23] was taken equal to N and in MOEA/D [22] $T = 10\%N$. In order to obtain a comparison in equal conditions between the DE-NSGA-II framework proposed in this work and NSGA-II, it was considered as stopping criterion to reach a maximum number of objective function evaluations. In all other comparisons the stopping criterion was to reach a maximum number of generations, G and/or G_{DE}.

As explained in section II, the gear train design problems defined by eq. (3) and eq. (4) have four integer variables $x_i \in [12, 60], i = 1, 2, 3, 4$. A 6-bit binary encoding allow us to define $2^6 = 64$ different integer values for each variable, but since these values must be within a certain range, unfortunately, unviable designs or unfeasible solutions are generated, which demand repairing methods. This paper compares two simple methods to repair the unfeasibility of the designs, namely repair I and repair II, in order to preserve the viability of the solutions. In repair I method (see Algorithm 1), half of the unfeasible values of the $\mathbf{x}$ variables are setted to the lower bound value, i.e. 12, and the rest of unfeasible values are setted to the upper bound value, i.e. 60. On the other hand, in repair II method (see Algorithm 2), all unfeasible values of the variables $\mathbf{x}$ are reassigned with the upper bound, which is located in the most conflictive region of the feasible space.

Mono-objective algorithms were compared in terms of statistical results and number of function evaluations (NFEs). In this paper, the computational cost, which is considered as

Algorithm 1 Function Repair I: Evaluate Feasibility of $\mathbf{x}_i$ $i = 1, 2, 3, 4$

Input: Lower bound of $\mathbf{x}$, a $= 12$, upper bound of $\mathbf{x}$, b $= 60$
Output: Value of $\mathbf{x}$
for i $= 1, \ldots, 4$
  if $\mathbf{x}_i \geq 8$ & $\mathbf{x}_i \leq 56$
    $\mathbf{x}_i = \mathbf{x}_i + 4$
  elseif $\mathbf{x}_i \leq 7$
    $\mathbf{x}_i =$ a
  elseif $\mathbf{x}_i \geq 57$
    $\mathbf{x}_i =$ b
  end if
end for

Algorithm 2 Function Repair II: Evaluate Feasibility of $\mathbf{x}_i$ $i = 1, 2, 3, 4$

Input: Lower bound of $\mathbf{x}$, a $= 12$, upper bound of $\mathbf{x}$, b $= 60$
Output: Value of $\mathbf{x}$
for i $= 1, \ldots, 4$
  if $\mathbf{x}_i \leq 48$
    $\mathbf{x}_i = \mathbf{x}_i +$ a
  else
    $\mathbf{x}_i =$ b
  end if
end for

the best NFEs corresponding to the obtained best solution, is calculated by the product of the population size (swarm particle number in APSO) and the maximum number of iterations or generations (i.e. $NFEs = N \times G$).

The hypervolume (HV) indicator suggested in [60] is used as a comparison measure between multi-objective algorithms. The reference point considered for the calculation of HV was (620, 65), which guarantees that it is dominated by all the solutions generated during the evolution of the algorithms. In addition, the attainment surface concept introduced in [61] is used considering the approach suggested by Knowles [62].

Moreover, a Wilcoxon rank-sum test [63] has been performed to make pairwise comparisons between the algorithms to study the significance of the results obtained. A significance level of 5% has been taken and the null hypothesis was ''the two algorithms have the same average indicator'', that is, the Wilcoxon test was used to determine if the difference between both means was statistically significant. If the p-value is less than 0.05, the null hypothesis is rejected and, then, we can conclude that the means of the two algorithms are comparable. In this case, a new Wilcoxon rank-sum test with one-sided alternative ''greater'' (or ''less'') and the same significance level is performed, that allows us to conclude which algorithm obtains better results than the other.

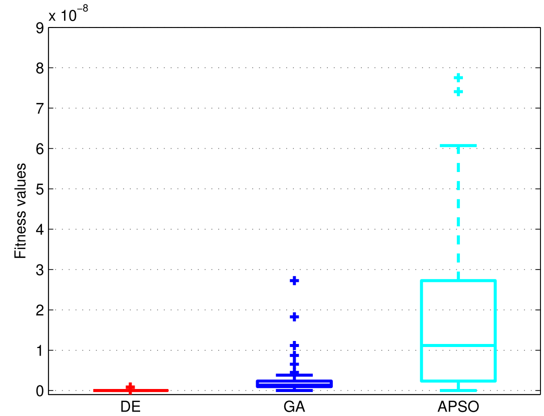

FIGURE 4. DE vs GA vs APSO ($N = 100$ and $G = 100$): box plots of the fitness values.

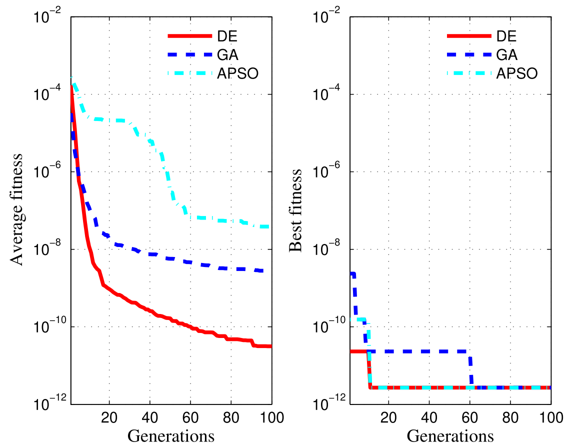

FIGURE 5. DE vs GA vs APSO ($N = 100$ and $G = 100$): evolution of the average fitness (left), evolution of the best fitness (right).

B. DE VS GA VS APSO

In this subsection, the DE, GA and APSO algorithms are compared for the mono-objective problem (3). Mean, standard deviation (in subscript), and the best and worst fitness values achieved by each algorithm (over 100 independent runs) are shown in Table 8. It can be seen that in all the experiments performed DE obtains better results than GA, and GA obtains better results than APSO.

Figs. 4 and 5 extend the comparative results of the experiment with $N = 100$ and $G = 100$ from Table 8. Fig. 4 shows the box plots of the distribution of fitness values and it is observed that DE solutions are concentrated very close to the global optimum, since its median is 2.3078e-11, GA solutions are farther away from the global optimum with a median equal to 1.3616e-09 and APSO solutions are even further away from the global optimum with a greater dispersion and a median equal to 1.1173e-08. The advantage of the DE algorithm over GA and APSO is also shown in Fig. 5 (left), that presents the evolution of the average fitness of the population per generation for the three algorithms, and in Fig. 5 (right), which shows that DE finds the best fitness value in the first generations while GA needs at least 60. In this case, APSO performs better than GA in the first generations.

TABLE 8. DE vs GA vs APSO: mean, standard deviation (in subscript), and the best and worst fitness values, respectively, over 100 independent runs.

G		$N = 20$	$N = 30$	$N = 40$	$N = 50$	$N = 100$
20	DE	1.309e-08$_{5.848e-08}$ 2.700e-12 5.788e-07	5.979e-09$_{1.239e-08}$ 2.700e-12 9.467e-08	2.707e-09$_{4.705e-09}$ 2.700e-12 2.726e-08	2.159e-09$_{3.056e-09}$ 2.700e-12 1.827e-08	1.235e-09$_{1.937e-09}$ 2.700e-12 1.827e-08
	GA	4.044e-07$_{6.687e-07}$ 8.887e-10 4.231e-06	1.538e-07$_{3.369e-07}$ 2.307e-11 2.022e-06	1.247e-07$_{2.397e-07}$ 8.887e-10 1.381e-06	5.599e-08$_{9.861e-07}$ 8.887e-10 6.409e-07	2.292e-08$_{5.765e-08}$ 3.068e-10 4.755e-07
	APSO	6.386e-03$_{1.345e-02}$ 2.357e-09 7.213e-02	9.460e-04$_{2.589e-03}$ 1.545e-10 1.401e-02	1.243e-03$_{2.913e-03}$ 1.166e-10 1.290e-02	5.955e-04$_{1.793e-03}$ 2.700e-12 1.061e-02	6.187e-05$_{4.199e-04}$ 2.700e-12e-3.753e-03
30	DE	2.943e-09$_{5.588e-09}$ 2.700e-12 2.726e-08	1.171e-09$_{1.130e-09}$ 2.700e-12 6.193e-09	1.002e-09$_{1.388e-09}$ 2.700e-12 1.117e-08	1.098e-09$_{1.410e-09}$ 2.700e-12 8.700e-09	4.247e-10$_{6.304e-10}$ 2.700e-12 2.357e-09
	GA	1.991e-07$_{3.605e-07}$ 9.921e-10 2.022e-06	7.915e-08$_{1.406e-07}$ 9.939e-11 7.798e-07	4.870e-08$_{8.159e-08}$ 2.307e-11 4.755e-07	3.960e-08$_{8.269e-08}$ 2.307e-11 5.193e-07	1.145e-08$_{1.356e-08}$ 2.308e-11 7.802e-08
	APSO	1.571e-03$_{5.205e-03}$ 1.545e-10 3.691e-02	1.902e-03$_{5.200e-03}$ 1.545e-10 3.241e-02	5.726e-04$_{2.175e-03}$ 2.700e-12 1.715e-02	1.788e-04$_{9.213e-04}$ 1.166e-10 7.857e-03	1.521e-05$_{1.076e-04}$ 2.307e-11 9.007e-04
40	DE	2.035e-09$_{3.558e-09}$ 2.700e-12 2.726e-08	9.158e-10$_{1.331e-09}$ 2.700e-12 8.700e-09	7.330e-10$_{8.495e-10}$ 2.700e-12 4.503e-09	5.585e-10$_{7.672e-10}$ 2.700e-12 3.299e-09	2.049e-10$_{3.484e-10}$ 2.700e-12 1.361e-09
	GA	1.276e-07$_{3.086e-07}$ 3.067e-10 2.022e-06	4.560e-08$_{8.427e-08}$ 2.700e-12 5.041e-07	2.548e-08$_{5.175e-08}$ 8.887e-10 4.04e-07	2.545e-08$_{5.707e-08}$ 2.307e-11 4.755e-07	7.242e-09$_{8.986e-09}$ 2.308e-11 3.881e-08
	APSO	2.423e-03$_{6.485e-03}$ 1.545e-10 3.605e-02	6.609e-04$_{2.604e-03}$ 2.307e-11 2.075e-02	1.359e-04$_{6.159e-04}$ 2.700e-12 3.920e-03	1.833e-04$_{8.609e-04}$ 2.700e-12 6.440e-03	1.929e-05$_{1.844e-04}$ 2.307e-11 1.844e-03
50	DE	1.639e-09$_{2.865e-09}$ 2.700e-12 1.827e-08	5.827e-10$_{8.516e-10}$ 2.700e-12 3.824e-09	5.181e-10$_{6.762e-10}$ 2.700e-12 3.299e-09	3.140e-10$_{4.721e-10}$ 2.700e-12 2.357e-09	1.213e-10$_{2.687e-10}$ 2.700e-12 1.263e-09
	GA	1.226e-07$_{3.128e-07}$ 2.307e-11 2.022e-06	2.634e-08$_{3.952e-08}$ 1.545e-10 2.663e-07	1.448e-08$_{.826e-08}$ 2.307e-11 7.802e-08	1.215e-08$_{1.506e-08}$ 2.307e-11 1.155e-07	5.365e-09$_{7.207e-09}$ 9.940e-11 2.726e-08
	APSO	1.720e-03$_{4.172e-03}$ 2.700e-12 2.641e-02	3.702e-04$_{1.228e-03}$ 2.700e-12 9.528e-03	2.021e-04$_{8.394e-04}$ 2.307e-11 4.988e-03	1.417e-04$_{5.991e-04}$ 2.307e-11 4.302e-03	3.377e-08$_{7.810e-08}$ 2.700e-12 4.359e-07
70	DE	1.656e-09$_{3.592e-09}$ 2.700e-12 2.726e-08	5.013e-10$_{6.933e-10}$ 2.700e-12 2.357e-09	2.333e-10$_{4.297e-10}$ 2.700e-12 2.357e-09	2.103e-10$_{4.846e-10}$ 2.700e-12 2.357e-09	6.742e-11$_{1.973e-10}$ 2.700e-12 1.183e-09
	GA	5.650e-08$_{1.443e-07}$ 2.307e-11 1.116e-06	3.376e-08$_{8.734e-08}$ 1.166e-10 7.778e-07	1.209e-08$_{.949e-08}$ 2.307e-11 1.234e-07	1.312e-08$_{3.501e-08}$ 9.939e-11 3.005e-07	4.004e-09$_{6.022e-09}$ 2.701e-12 2.726e-08
	APSO	1.416e-03$_{5.358e-03}$ 2.307e-11 4.746e-02	2.491e-04$_{1.121e-03}$ 2.307e-11 9.010e-03	5.288e-05$_{2.470e-04}$ 2.307e-11 1.773e-03	3.828e-06$_{3.491e-05}$ 2.307e-11 3.493e-04	6.853e-07$_{5.630e-06}$ 2.700e-12 5.634e-05
100	DE	1.857e-09$_{3.871e-09}$ 2.700e-12 2.726e-08	3.135e-10$_{4.697e-10}$ 2.700e-12 2.357e-09	2.782e-10$_{5.002e-10}$ 2.700e-12 2.357e-09	8.238e-11$_{2.118e-10}$ 2.700e-12 9.921e-10	3.154e-11$_{1.239e-10}$ 2.700e-12 8.887e-10
	GA	3.573e-08$_{6.524e-08}$ 2.700e-12 4.359e-07	1.412e-08$_{2.341e-08}$ 2.307e-11 1.761e-07	8.421e-09$_{1.415e-08}$ 2.307e-11 1.155e-07	5.681e-09$_{1.006e-08}$ 2.307e-11 7.802e-08	2.704e-09$_{4.024e-09}$ 2.701e-12 2.726e-08
	APSO	3.080e-03$_{2.119e-02}$ 1.545e-10 2.107e-01	9.024e-05$_{5.171e-04}$ 1.545e-10 4.102e-03	3.054e-04$_{1.465e-03}$ 2.307e-11 1.114e-02	7.665e-05$_{4.336e-04}$ 2.307e-11 3.844e-03	3.377e-08$_{7.810e-08}$ 2.700e-12 4.359e-07

A Wilcoxon rank-sum test has been performed to determine whether the difference between the average fitness values achieved by the DE, GA and APSO algorithms is statistically significant. The results are shown in Table 9, where a symbol $\triangle$ indicates that the algorithm in the row has reached a better average fitness value than the algorithm in the column. From Table 9, it can be concluded that DE obtains better average fitness in all cases and AG improves to APSO in solving the problem discussed in this paper.

In addition, another statistical comparison was developed, in which the number of times the global optimum is found in 100 runs is analyzed. This experiment was repeated 100 times, where each experiment consisted of performing 100 runs of DE or GA or APSO. In Fig. 6 the average number of times the global optimum appears in 100 trials of each experiment with DE and GA is shown. For example, for values $N = 100$ and $G = 100$, the average number of times the global optimum appears with DE is 48.24 and with GA is only 0.96. All values shown in Fig. 6 confirm that better results are obtained with DE. In all cases, APSO found the global optimum with an average number of times lower than the one obtained with GA.

TABLE 9. DE vs GA vs APSO: Wilcoxon rank-sum test to compare average fitness.

		GA $N = 20, 30, 40, 50, 100$	APSO $N = 20, 30, 40, 50, 100$
DE	$G = 20$	△ △ △ △ △	△ △ △ △ △
	$G = 30$	△ △ △ △ △	△ △ △ △ △
	$G = 40$	△ △ △ △ △	△ △ △ △ △
	$G = 50$	△ △ △ △ △	△ △ △ △ △
	$G = 70$	△ △ △ △ △	△ △ △ △ △
	$G = 100$	△ △ △ △ △	△ △ △ △ △
GA	$G = 20$		△ △ △ △ △
	$G = 30$		△ △ △ △ △
	$G = 40$		△ △ △ △ △
	$G = 50$		△ △ △ △ △
	$G = 70$		△ △ △ △ △
	$G = 100$		△ △ △ △ △

The problem (3) has also been used as a reference in other research papers. Table 10 (extended from [54] and [55]) shows the results obtained in this work with GA (NFEs = 800) and DE (NFEs = 800) together with those achieved by other metaheuristic methods. It can be seen that DE (NFEs = 800) has the best average fitness value followed by MBA (NFEs = 1120) and IAPSO (NFEs = 800).

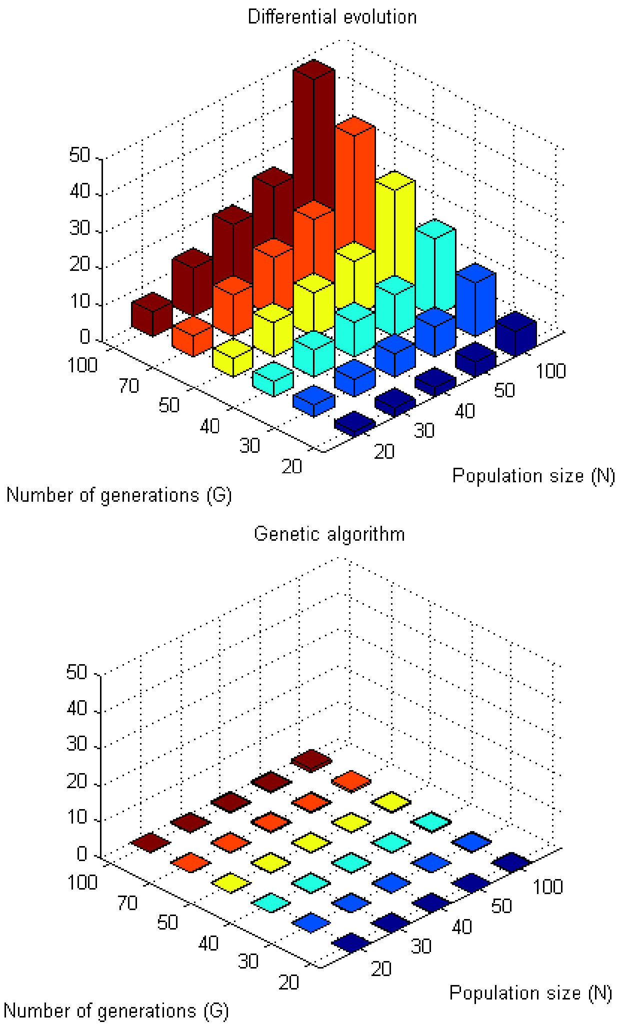

FIGURE 6. **DE vs GA: average number of times the best appears in 100 trials.**

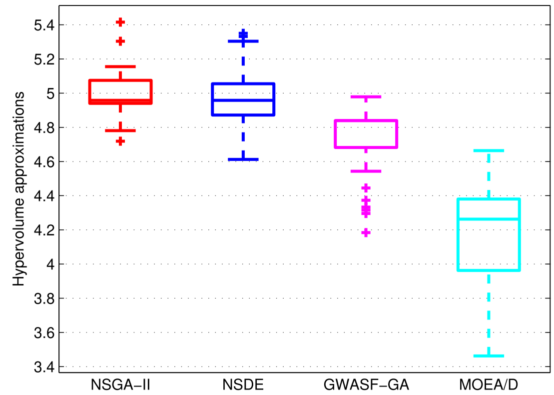

FIGURE 7. **NSGA-II vs MOEA/D vs GWASF-GA vs NSDE ($N = 100$ and $G = 100$, repair II): box plots based on the hypervolume metric.**

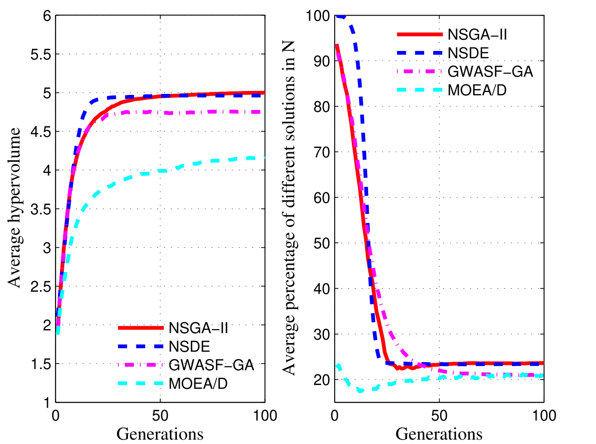

FIGURE 8. **NSGA-II vs MOEA/D vs GWASF-GA vs NSDE ($N = 100$ and $G = 100$, repair II): evolution of the average hypervolume (left), evolution of the average percentage of different solutions in the population (right).**

Although PSO + GA outperforms the previous ones, its reference paper does not clearly indicate the NFEs value, which can be up to a maximum of $20 \times 4 \times 5000 = 40000$. In which case, it should be noted that DE with NFEs = 900 ($N = 30, G = 30$) (see Table 8) already improves the average fitness of the PSO-GA. In any case, in Table 8 we can see that DE with NFEs = 10000 ($N = 100, G = 100$) reaches the best average fitness value (3.154605e-11).

C. NSGA-II VS MOEA-D VS GWASF-GA VS NSDE

Four popular MOEAs are compared in this subsection. Table 11 shows the mean, standard deviation (in subscript), and the best and worst hypervolume (HV) indicators achieved from 100 independent runs. It can be seen that, in all experiments, the NSGA-II algorithm achieves the best performance. Also, it is noted that all the algorithms implemented with binary representation, case of NSGA-II, GAWASF-GA and MOEA/D, achieve better results when use repair II as repairing method for unfeasible solutions.

Figs. 7 and 8 show a more detailed comparison between the four algorithms for the repair II experiment with $G = 100$ and $N = 100$ from Table 11. Fig. 7 shows the box plots based on the hypervolume approximation metric. It is observed that the best median and the lowest dispersion is obtained with NSGA-II. The advantage of NSGA-II over the other algorithms is also shown in Fig. 8 (left) that presents the average hypervolume evolution per generation. NSGA-II is followed, from best to worst result, by NSDE, GWASF-GA and MOEA/D. Fig. 8 (right) shows the evolution of the average percentage of different solutions in the population (of size N) and would explain why the different returns of the algorithms, i.e. the presence of overlapping solutions. For example, from Fig. 8 (right) it is extracted that at the end of the evolution, NSGA-II and NSDE have similar values of the average percentage of different solutions although with slightly better hypervolume value for NSGAII (see Table 11), however in the first generations this value is better for NSDE than for NSGA-II, which justifies that in the first generations NSDE has better average hypervolume value than NSGA-II (see Fig. 8 (left)). The same analysis can be done with the results achieved by GWASF-GA and MOEA/D.

TABLE 10. Comparison and statistical results for gear train model found by different mono-objective methods (NA - not available).

Algorithms	NFEs	Best	Worst	Mean	Std. dev.	x_1 x_2 x_3 x_4
BB [44]		5.712e-06	NA	NA	NA	18 22 45 60
AL [45]		2.146e-08	NA	NA	NA	13 15 33 41
Gene AS [46]	5000	2.701e-12	NA	NA	NA	19 16 49 43
PSO-GA [55]	NA	2.70085e-12	3.2999e-09	1.2149e-09	8.7787e-10	19 16 43 49
MBA [48]	1120	2.70085e-12	2.0629e-08	2.4716e-09	3.9400e-09	19 16 43 49
UPSO [53]	100000	2.70085e-12	8.94899e-07	3.80562e-08	1.09631e-07	NA
CS [51]	5000	2.701e-12	2.3576e-09	1.9841e-09	3.5546e-09	19 16 43 49
APSO [17]	8000	2.700857e-12	7.072678e-06	4.781676e-07	1.44e-06	19 16 43 49
IAPSO [54]	800	2.700857e-12	1.827380e-08	5.492477e-09	6.36e-09	19 16 43 49
Present study (by enumeration)		2.7008571488865e-12				19 16 43 49
GA Present study	800	3.067556e-10	2.022601e-06	1.276162e-07	3.086869e-07	19 16 43 49
DE Present study	**800**	**2.700857e-12**	**2.726451e-08**	**2.035909e-09**	**3.558971e-09**	**19 16 43 49**

TABLE 11. NSGA-II vs MOEA/D vs GWASF-GA vs NSDE ($G = 100$): mean, standard deviation (in subscript), and the best and worst hypervolume, respectively, over 100 independent runs.

repair I	$N = 50$	$N = 100$
NSGA-II	$4.7528_{1.4580e-01}$ 5.0248 - 4.5302	$4.8452_{1.4114e-01}$ 5.0984 - 4.5419
MOEA/D	$3.8965_{3.8264e-01}$ 4.6404 - 3.2060	$3.9666_{3.7465e-01}$ 4.6565 - 3.4188
GWASF-GA	$4.5532_{1.9674e-01}$ 4.9561 - 4.1200	$4.6160_{2.0598e-01}$ 4.8397 - 3.6302
repair II	$N = 50$	$N = 100$
NSGA-II	$4.9595_{1.1066e-01}$ 5.4335 - 4.5985	$5.0021_{1.0550e-01}$ 5.4155 - 4.7188
MOEA/D	$3.9267_{3.6317e-01}$ 4.6605 - 3.2935	$4.172997_{3.3397e-01}$ 4.6640 - 3.4624
GWASF-GA	$4.7223_{1.9471e-01}$ 5.0000 - 4.1962	$4.7465_{1.5993e-01}$ 4.9783 - 4.1840
	$N = 50$	$N = 100$
NSDE	$4.8946_{1.1465e-01}$ 5.2960 - 4.5418	$4.9632_{1.3941e-01}$ 5.3502 - 4.6120

Again, a Wilcoxon rank-sum test was performed to determine whether the difference between the average hypervolume values reached by each pair of algorithms is statistically significant. The results are shown in Table 12 where a symbol $\triangle$ indicates that the algorithm in the row has reached a better average hypervolume than the algorithm in the column, a symbol ∇ indicates that the algorithm in the row has reached a worse average hypervolume than the algorithm in the column and again a symbol – indicates that the comparative is not statistically significant. The results obtained by applying repair I indicate that NSDE obtains better average hypervolume than MOEA/D, GWASF-GA and NSGA-II (between NSDE and NSGA-II only in one case the difference is not significant), NSGA-II achieves better average hypervolume than MOEA/D and GWASF-GA and, lastly, GWASF-GA achieves better average hypervolume than MOEA/D. Also, from Table 12 (repair II) it is conclude that all tests are statistically significant, and NSGA-II wins in all cases, NSDE wins MOEA/D and GWASF-GA, and GWASF-GA wins MOEA/D.

TABLE 12. NSGA-II vs MOEA/D vs GWASF-GA vs NSDE: Wilcoxon rank-sum test to compare the hypervolume indicator.

repair I: $N = 50, 100$			
	MOEA/D	GWASF-GA	NSDE
NSGA-II	$\triangle$ $\triangle$	$\triangle$ $\triangle$	∇ –
MOEA/D		∇ ∇	∇ ∇
GWASF-GA			∇ ∇
repair II: $N = 50, 100$			
	MOEA/D	GWASF-GA	NSDE
NSGA-II	$\triangle$ $\triangle$	$\triangle$ $\triangle$	$\triangle$ $\triangle$
MOEA/D		∇ ∇	∇ ∇
GWASF-GA			∇ ∇

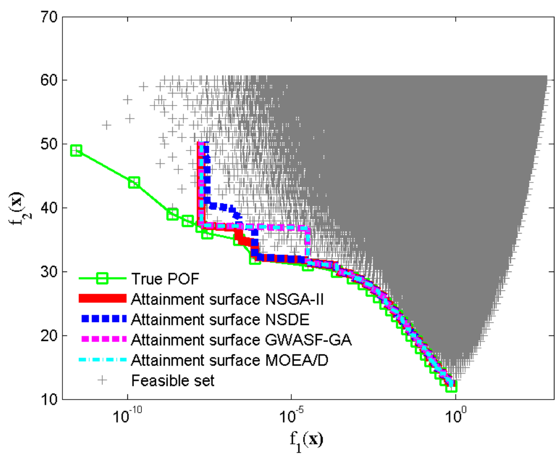

FIGURE 9. NSGA-II vs MOEA/D vs GWASF-GA vs NSDE ($N = 100$ and $G = 100$, repair II): 50% attainment surface.

In addition, Fig. 9 shows the 50% attainment surface over 100 independent runs of NSGA-II, NSDE, GWASF-GA and MOEA/D. For a more complete comparison, the true Pareto front and the feasible solution set are also shown. According to the results shown in Table 11 and Figs. 7 and 8, in Fig. 9 it is observed that NSGA-II achieves the best 50% attainment surface. However, the 50% attainment surface stagnates for all algorithms in the Pareto front zone with low solution density, highlighting the difficulty of the problem when it is solved using MOEAs.

TABLE 13. **DE-NSGA-II (repair II): mean, standard deviation (in subscript), the best and worst hypervolume indicator, over 100 independent runs, with $G = 100$ and $(M \times G_{DE} + N) \times G$ evaluations.**

	$M = 20$ $N = 80$	$M = 30$ $N = 70$	$M = 40$ $N = 60$	$M = 50$ $N = 50$
$G_{DE} = 20$	$5.559273_{3.2220e-02}$ 5.582365 - 5.368282	$5.557192_{2.7718e-02}$ 5.582365 - 5.428426	$5.538939_{4.0896e-02}$ 5.582365 - 5.338458	$5.507893_{1.1221e-01}$ 5.574693 - 5.167925
$G_{DE} = 30$	$5.557878_{1.7452e-02}$ 5.582365 - 5.496172	$5.548601_{2.8543e-02}$ 5.582365 - 5.412035	$5.529370_{4.3369e-02}$ 5.580352 - 5.327332	$5.490075_{7.0215e-02}$ 5.5693347- 5.259644
$G_{DE} = 40$	$5.549170_{2.2785e-02}$ 5.582365 - 5.437964	$5.534972_{3.7976e-02}$ 5.578644 - 5.314085	$5.529370_{4.3369e-02}$ 5.580352 - 5.327333	$5.490075_{7.0215e-02}$ 5.569335 - 5.259644
$G_{DE} = 50$	$5.543342_{2.4314e-02}$ 5.580535 - 5.461478	$5.535731_{2.4159e-02}$ 5.580352 - 5.455241	$5.516490_{4.6994e-02}$ 5.579308 - 5.229564	$5.491366_{5.0352e-02}$ 5.567614 - 5.342674
$G_{DE} = 70$	$5.540403_{3.5937e-02}$ 5.582365 - 5.332845	$5.531438_{2.8599e-02}$ 5.578845 - 5.445423	$5.513164_{4.6737e-02}$ 5.573443 - 5.339896	$5.467308_{6.1254e-02}$ 5.562015 - 5.253888
$G_{DE} = 100$	$5.543236_{2.5624e-02}$ 5.580352 - 5.438514	$5.515572_{4.6801e-02}$ 5.577416 - 5.350312	$5.501916_{4.3828e-02}$ 5.567423 - 5.370499	$5.448379_{6.5730e-02}$ 5.561375 - 5.218822

TABLE 14. **NSGA-II (repair II): mean, standard deviation (in subscript), the best and worst hypervolume indicator, over 100 independent runs, with $N = 100$ and $N \times G$ evaluations.**

$G = 480$	$G = 680$	$G = 880$	$G = 1,080$	$G = 1,480$	$G = 2,080$
$5.1076_{1.0541e-01}$ 5.4973 - 4.9524	$5.1302_{0.8246e-01}$ 5.3327 - 4.9524	$5.1321_{0.8009e-01}$ 5.3152 - 4.9639	$5.1450_{0.6805e-01}$ 5.3638 - 4.9638	$5.1703_{0.6257e-01}$ 5.3617 - 4.9639	$5.1805_{0.5359e-01}$ 5.4398 - 4.9638

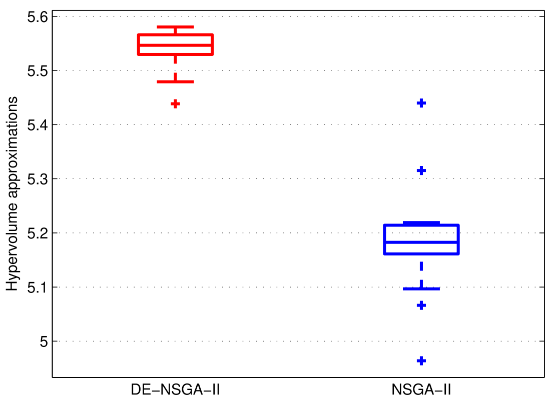

FIGURE 10. **DE-NSGA-II vs NSGA-II (208,000 evaluations): box plots based on the hypervolume metric.**

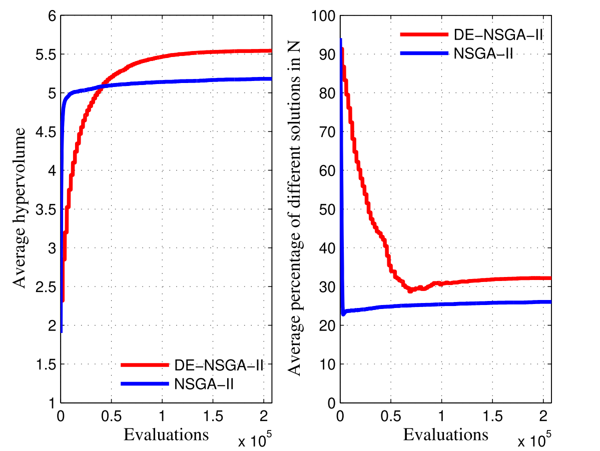

FIGURE 11. **DE-NSGA-II vs NSGA-II (208,000 evaluations): average of hypervolume metric over 100 runs (left), average percentage of different solutions in the population of size N over 100 runs (right).**

D. DE-NSGA-II VS NSGA-II (REPAIR II)

In this subsection, the proposed DE-NSGA-II is compared with the NSGA-II (repair II), since this algorithm proved to be the one that obtained the best results in the previous subsection IV-C. In order to have an equitable comparison, the stopping criterion for both algorithms was defined as the maximum number of evaluations of the objective functions.

Table 13 presents the values of the means, standard deviations (in subscript), best and worst hypervolume (HV) indicator reached by DE-NSGA-II for different values of the number of generations of DE, G_{DE}, and different population sizes, N for NSGA-II and M for DE. This table shows that the results obtained with $M = 20$ and $N = 80$ do not differ greatly from those achieved in the rest of experiments. In addition, the number of evaluations of the objective functions, $(M \times G_{DE} + N) \times G$, is smaller. For this reason, only the results of the experiments performed with $M = 20$ and $N = 80$ have been compared with NSGA-II (repair II). The results obtained with NSGA-II (repair II) are shown in Table 14. It is observed that DE-NSGA-II obtains better results in all cases.

For more detail, the results of the experiment with 208,000 objective function evaluations ($M = 20$, $N = 80$ and $G_{DE} = 100$ for DE-NSGA-II (see Table 13) and $N = 100$ and $G = 2,080$ for NSGA-II (see Table 14)) are shown in Figs. 10 and 11. Fig. 10 shows that DE-NSGA-II achieves better results than NSGA-II (repair II). Fig. 11 (left) shows that although initially NSGA-II obtains better values of the average hypervolume, then DE-NSGA-II improves those values. This improvement can be explained because DE-NSGA-II achieves, throughout the evolution of the algorithm, the best average percentage of different solutions in the population of size N (see Fig. 11 (rigth)).

TABLE 15. **DE-NSGA-II vs NSGA-II: Wilcoxon rank-sum test to compare the hypervolume indicator.**

DE-NSGA-II vs NSGA-II	$M = 20$ $N = 80$
48,000 evaluations	△
68,000 evaluations	△
88,000 evaluations	△
108,000 evaluations	△
148,000 evaluations	△
208,000 evaluations	△

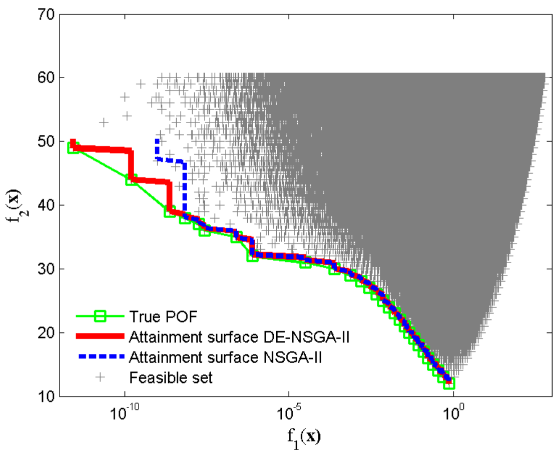

FIGURE 12. **DE-NSGA-II vs NSGA-II (208,000 evaluations): 50% attainment surface.**

TABLE 16. **DE-NSGA-II vs NSGA-II: Average number of times that the hard-to-reach extreme solution of the POF appears in 30 trials and standard deviation (in subscript).**

Evaluations	DE-NSGA-II	NSGA-II
48,000	$24.47_{5.90}$	$0,33_{0.61}$
68,000	$39.20_{5.82}$	$0,27_{0.45}$
88,000	$45.43_{5.74}$	$0,30_{0.60}$
108,000	$54.17_{5.52}$	$0,30_{0.65}$
148,000	$63.37_{4.56}$	$0,17_{0.38}$
208,000	$76.00_{4.93}$	$0,33_{0.67}$

Table 15 shows the results obtained with the Wilcoxon rank-sum test. It can be seen that the proposed algorithm DE-NSGA-II obtains better average hypervolume than NSGA-II in all cases (△).

In Fig. 12 the 50% attainment surface obtained by the DE-NSGA-II and NSGA-II algorithms are shown. It is interesting to remark that in the area of Pareto's optimal front with low density of solutions, the 50% attainment surface obtained by DE-NSGA-II wholly dominates the attainment surface achieved by NSGA-II.

A statistical comparison was made in which each experiment, which consisted of executing each algorithm 100 times, was performed 100 times. The results are shown in Fig. 13 and Table16. Specifically, Fig. 13 shows the average number of times that the hard-to-reach extreme solution of the POF (2.700857148886513e-12, 49) was achieved in the 100 independent runs of each experiment. For example, for 208,000 evaluations of the objective functions,

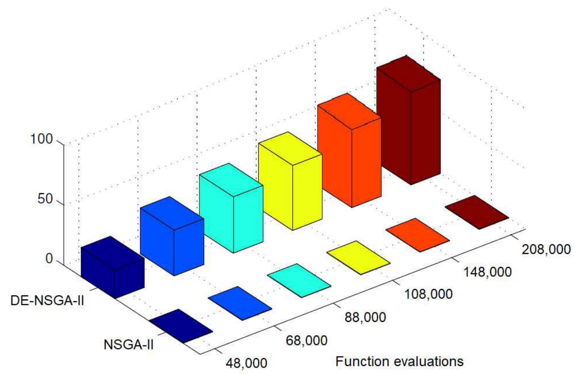

FIGURE 13. **DE-NSGA-II vs NSGA-II: Average number of times that the hard-to-reach extreme solution of the POF appears in 100 trials.**

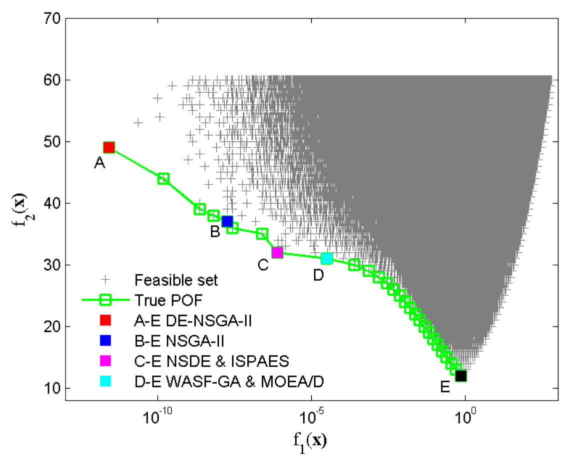

FIGURE 14. **POF obtained by DE-NSGA-II, NSGA-II, ISPAES, NSDE, WASF-GA and MOEA/D algorithms.**

the average number of times that this solution was found with DE-NSGA-II was 76.00 and with NSGA-II only 0.33 times.

Finally, for the problem (4), Fig. 14 shows the POFs obtained with DE-NSGA-II, NSGA-II, WASF-GA and MOEA/D in this work, with NSGA-II in [56] (coincide with those obtained in this work) and with ISPAES in [58], as far as we know all the multi-objective methods that have been applied to this problem (4). It can be seen that DE-NSGA-II finds the whole POF A-E, while NSGA-II finds only section B-E, NSDE and ISPAES section C-E and WASF-GA and MOEA/D section D-E.

V. CONCLUSION AND FUTURE WORK

This paper proposes a MH + MH framework (DE + NSGA-II) (called DE-NSGA-II) with sequential cooperation. An extensive experimental comparative study between EAs is carried out using mono-objective and multi-objective approaches, considering as test-bed a double reduction gear system design problem from the literature.

A preliminary analysis of the problem (solved by enumeration) reveals that the zone where the optimal solution for the mono-objective problem is found, which coincides with one extreme zone of the Pareto optimal front for the multi-objective problem, is scarcely populated. This explains why exact and metaheuristic algorithms get poor returns when they address this problem.

Regarding the mono-objective formulation of the problem, the results obtained when comparing the algorithms DE, GA and APSO show that in all the experiments DE obtains the best fitness values. Also, for the same number of fitness function evaluations, DE obtains the best yields compared to other metaheuristic techniques of the literature, followed by the IAPSO algorithm.

For the multi-objective problem, a rigorous comparative analysis was made between state-of-the-art algorithms (NSGA-II, GWASF-GA, MOEA/D and NSDE) using two quality indicators: the hypervolume metric measure and the attainment surface concept. In addition, the algorithms programmed with binary representation were compared using two different repair strategies for non-feasible solutions. NSGA-II obtained the best results in all experiments performed, followed by NSDE, GWASF-GA and MOEA/D, in that order.

Finally, the DE-NSGA-II algorithm proposed in this work was compared with NSGA-II (repair II) using the number of evaluations of the objective functions as stop criterion in both algorithms. Clearly, DE-NSGA-II obtained in all performed experiments the best average hypervolume value. In addition, in the low density zone of the POF solutions, the 50% attainment surface obtained by DE-NSGA-II wholly dominates the attainment surface achieved by NSGA-II. Also, in the statistical comparison of the number of times that the hard-to-reach extreme solution of the POF is found, DE-NSGA-II yielded the best average value. In addition, DE-NSGA-II finds the whole POF while other multi-objective metaheuristic techniques only find a section of it, and this section corresponds to the area with the highest solution density. Therefore, our framework proposal constitutes a state-of-the-art method for this problem.

Extending this research to other real engineering problems with difficulties similar to those of the problem studied in this work, as well as studying other framework designs to combine multiple metaheuristics and incorporation of knowledge of the problems in the algorithms, are lines of interest to study in future works.

REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1992.

[3] Z. Wang, J. Li, K. Fan, W. Ma, and H. Lei, "Prediction method for low speed characteristics of compressor based on modified similarity theory with genetic algorithm," *IEEE Access*, vol. 6, pp. 36834–36839, 2018.

[4] P. Rai and A. G. Barman, "An approach for design optimization of helical gear pair with balanced specific sliding and modified tooth profile," *Struct. Multidisciplinary Optim.*, vol. 60, no. 1, pp. 331–341, Jul. 2019.

[5] J. Xiong, B. Chen, Y. Chen, Y. Jiang, and Y. Lu, "Route network design of community shuttle for metro stations through genetic algorithm optimization," *IEEE Access*, vol. 7, pp. 53812–53822, 2019.

[6] Q. Zhou, P. Jiang, X. Huang, F. Zhang, and T. Zhou, "A multi-objective robust optimization approach based on Gaussian process model," *Struct. Multidisciplinary Optim.*, vol. 57, no. 1, pp. 213–233, Jan. 2018.

[7] C. Dong, Z. Xiong, X. Liu, Y. Ye, Y. Yang, and W. Guo, "Dual–search artificial bee colony algorithm for engineering optimization," *IEEE Access*, vol. 7, pp. 24571–24584, 2019.

[8] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," *Evol. Comput.*, vol. 4, no. 4, pp. 361–394, Dec. 1996.

[9] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," *Found. Genetic Algorithms*, vol. 1, pp. 69–93, Jan. 1991.

[10] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[11] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[12] X. Li, H. Zhang, and Z. Lu, "A differential evolution algorithm based on multi–population for economic dispatch problems with valve–point effects," *IEEE Access*, vol. 7, pp. 95585–95609, 2019.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[14] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Bristol, U.K.: Luniver Press, 2008.

[15] M. Masood, M. M. Fouad, R. Kamal, I. Glesk, and I. U. Khan, "An improved particle swarm algorithm for multi–objectives based optimization in MPLS/GMPLS networks," *IEEE Access*, vol. 7, pp. 137147–137162, 2019.

[16] S. Sengupta, S. Basak, and R. A. Peters, II, "Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 1, pp. 157–191, 2019.

[17] X.-S. Yang, *Engineering Optimization: An Introduction With Metaheuristic Applications*. Hoboken, NJ, USA: Wiley, 2010.

[18] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, "Chaos-enhanced accelerated particle swarm optimization," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 2, pp. 327–340, 2013.

[19] C. A. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Berlin, Germany: Springer-Verlag, 2006.

[20] C. Han, L. Wang, Z. Zhang, J. Xie, and Z. Xing, "A multi–objective genetic algorithm based on fitting and interpolation," *IEEE Access*, vol. 6, pp. 22920–22929, 2018.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA–II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[22] M. Liu and W.-H. Zeng, "Memory enhanced dynamic multi–objective evolutionary algorithm based on decomposition," *J. Softw.*, vol. 24, no. 7, pp. 1571–1588, Jan. 2014.

[23] R. Saborido, A. B. Ruiz, and M. Luque, "Global WASF–GA: An evolutionary algorithm in multiobjective optimization to approximate the whole Pareto optimal front," *Evol. Comput.*, vol. 25, no. 2, pp. 309–349, Jun. 2017.

[24] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making Theory and Applications* (Lecture Notes in Economics and Mathematical Systems), vol. 177, G. Fandel and T. Gal, Eds. Berlin, Germany: Springer, 1980, pp. 468–486.

[25] H. Ishibuchi, Y. Nojima, K. Narukawa, and T. Doi, "Incorporation of decision maker's preference into evolutionary multiobjective optimization algorithms," in *Proc. 8th Annu. Conf. Genetic Evol. Comput. (GECCO)*, Seattle, WA, USA, 2006, pp. 741–742.

[26] K. Deb and H. Jain, "An evolutionary many–objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[27] H. Jain and K. Deb, "An evolutionary many–objective optimization algorithm using reference–point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[28] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, Sep. 1999.

[29] L. Jourdan, M. Basseur, and E.-G. Talbi, "Hybridizing exact methods and metaheuristics: A taxonomy," *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 620–629, Dec. 2009.

[30] J. Puchinger and G. R. Raidl, "Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification," in *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach* (Lecture Notes in Computer Science), vol. 3562, J. Mira and J. R. Álvarez, Eds. Berlin, Germany: Springer, 2005, pp. 41–53.
[31] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.
[32] E.-G. Talbi, "Hybrid metaheuristics for multi-objective optimization," *J. Algorithms Comput. Technol.*, vol. 9, no. 1, pp. 41–63, 2015.
[33] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA–II," *IEEE Trans. Evol. Computat.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
[34] Y. Xu, O. Ding, R. Qu, and K. Li, "Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization," *Appl. Soft Comput.*, vol. 68, no. 2, pp. 268–282, Jul. 2018.
[35] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 3339, G. I. Webb and X. Yu, Eds. Berlin, Germany: Springer, 2004, pp. 861–872.
[36] T. Robič and B. Filipič, "DEMO: Differential evolution for multiobjective optimization," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 3410, C. A. C. Coello, A. H. Aguirre and E. Zitzler, Eds. Berlin, Germany: Springer, 2005, pp. 520–533.
[37] F. Xue, A. C. Sanderson, and R. J. Graves, "Pareto-based multi-objective differential evolution," in *Proc. Congr. Evol. Comput. (CEC)*, Canberra, ACT, Australia, 2003, pp. 862–869.
[38] J. A. Parejo, A. Ruiz-Cortés, S. Lozano, and P. Fernandez, "Metaheuristic optimization frameworks: A survey and benchmarking," *Soft Comput.*, vol. 16, no. 3, pp. 527–561, Mar. 2012.
[39] J. Zhang, F. Yang, and X. Weng, "An evolutionary scatter search particle swarm optimization algorithm for the vehicle routing problem with time windows," *IEEE Access*, vol. 6, pp. 63468–63485, 2018.
[40] Y. Qi, Z. Hou, H. Li, J. Huang, and X. Li, "A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 62, pp. 61–77, Oct. 2015.
[41] F. Moslehi, A. Haeri, and F. Martínez-Álvarez, "A novel hybrid GA-PSO framework for mining quantitative association rules," *Soft Comput.*, vol. 2019, pp. 1–22, 2019.
[42] K. Das, D. Mishra, and K. Shaw, "A metaheuristic optimization framework for informative gene selection," *Informat. Med. Unlocked*, vol. 4, pp. 10–20, Jan. 2016.
[43] F. Dong, J. Zhang, L. Xie, H. Zhao, and X. He, "An improved APSO–SQP with adaptive transition strategy for dynamic optimization," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 8152–8157, 2014.
[44] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *J. Mech. Des.*, vol. 112, no. 2, pp. 223–229, Jun. 1990.
[45] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *J. Mech. Des.*, vol. 116, no. 2, pp. 405–411, Jun. 1994.
[46] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inf.*, vol. 26, no. 4, pp. 30–45, Aug. 1996.
[47] K. Deb and M. Goyal, "A flexible optimization procedure for mechanical component design based on genetic adaptive search," *J. Mech. Des.*, vol. 120, no. 2, pp. 162–164, Jun. 1998.
[48] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.
[49] S. Mirjalili, "Biogeography-based optimisation," in *Evolutionary Algorithms and Neural Networks*, vol. 780. Cham, Switzerland: Springer, 2019, pp. 57–72.
[50] A. M. A. C. Rocha and E. M. G. P. Fernandes, "Hybrid optimization coupling electromagnetism and descent search for engineering problems," in *Proc. Int. Comput. Math. Methods Sci. Eng. (CMMSE)*, Murcia, Spain, 2008, pp. 533–545.
[51] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.
[52] O. Camarena, E. Cuevas, M. Pérez-Cisneros, F. Fausto, A. González, and A. Valdivia, "LS–II: An improved locust search algorithm for solving optimization problems," *Math. Problems Eng.*, vol. 2018, Oct. 2018, Art. no. 4148975.
[53] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," in *Proc. Int. Conf. Natural Comput., Adv. Natural Comput. (ICNC)* (Lecture Notes in Computer Science), vol. 3612, L. Wang, K. Chen, and Y. S. Ong, Eds. Berlin, Germany: Springer, 2005, pp. 582–591.
[54] N. Ben Guedria, "Improved accelerated PSO algorithm for mechanical engineering optimization problems," *Appl. Soft Comput.*, vol. 40, pp. 455–467, Mar. 2016.
[55] H. Garg, "A hybrid PSO–GA algorithm for constrained optimization problems," *Appl. Math. Comput.*, vol. 274, pp. 292–305, Feb. 2016.
[56] K. Deb, A. Pratap, and S. Moitra, "Mechanical component design for multiple objectives using elitist non-dominated sorting GA," in *Parallel Problem Solving from Nature PPSN VI* (Lecture Notes in Computer Science), vol. 1917, M. Schoenauer, Eds. Berlin, Germany: Springer, 2000, pp. 859–868.
[57] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, Jun. 2000.
[58] A. H. Aguirre, S. B. Rionda, and G. Lizarraga, "ISPAES: Evolutionary multi-objective optimization with constraint handling," in *Proc. 4th Mex. Int. Conf. Comput. Sci. (ENC)*, Tlaxcala, Mexico, 2003.
[59] Y. Nojima, K. Narukawa, S. Kaige, and H. Ishibuchi, "Effects of removing overlapping solutions on the performance of the NSGA-II algorithm," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 3410, C. A. C. Coello, A. H. Aguirre and E. Zitzler, Eds. Berlin, Germany: Springer, 2005, pp. 341–354.
[60] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving From Nature* (Lecture Notes in Computer Science), vol. 1498, A. E. Eiben, T. Bäck, M. Schoenauer, and H. P. Schwefel, Eds. Berlin, Germany: Springer, 1998, pp. 292–301.
[61] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Parallel Problem Solving From Nature—PPSN IV* (Lecture Notes in Computer Science), vol. 1141, H. M. Voigt, W. Ebeling, I. Rechenberg, and H. P. Schwefel, Eds. Berlin, Germany: Springer, 1996, pp. 584–593.
[62] J. Knowles, "A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers," in *Proc. 5th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Warsaw, Poland, 2005.
[63] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

MÁXIMO MÉNDEZ received the M.Sc. degree in industrial engineering from the School of Industrial Engineering of Las Palmas, Spain, and the Ph.D. degree in computer science from the Universidad de Las Palmas de Gran Canaria (ULPGC), Spain. He is currently an Associate Professor with the Department of Informatics and Systems, ULPGC, where he is also a member and Research Fellow of the Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI). His research interests include genetic algorithms, metaheuristics for multiobjective optimization, decision analysis, machine learning and real applications in mechanical engineering, renewable energies, transportation planning, and weather prediction. He has published more than 30 technical articles in international journals, conferences, and book chapters.

DANIEL A. ROSSIT received the Industrial Engineer degree and the Ph.D. degree in engineering. He is currently a Postdoctoral Fellow of CONICET (National Research Council of Argentina) and a Teaching Assistant with the Engineering Department, Universidad Nacional del Sur, Bahía Blanca, Argentina. His research has focused on production problems, operations' research, and engineering systems' optimization. He has published in journals, such as *Omega*, *International Journal of Production Research*, *The International Journal of Advanced Manufacturing Technology*, *Computers and Electronics in Agriculture*, and *International Journal of Computer Integrated Manufacturing*, among others.

BEGOÑA GONZÁLEZ studied mathematics (with a specialization in statistics and operations research) at the University of Santiago de Compostela (USC) and received the Ph.D. degree in mathematics from the University of Las Palmas de Gran Canaria (ULPGC), in 2001. She is currently an Associate Professor with ULPGC, where she has been collaborating in research publications, since 1999, focusing primarily on numerical simulation in environmental engineering (computational fluid dynamics) and optimal design in engineering by using evolutionary algorithms, with more than 50 published works, including JCR journals, book chapters, and presentations in national and international conferences. She has participated in national and international research projects and contracts, as well as she has been a member of scientific committees and/or programs of several international conferences.

MARIANO FRUTOS received the bachelor's degree in industrial engineering and the master's and Ph.D. degrees in engineering from his home university. He is currently an Associate Researcher of CONICET (National Research Council of Argentina) and a Teaching Assistant with the Department of Engineering, Universidad Nacional del Sur, Bahía Blanca, Argentina. His research is centered on scheduling problems in production and its treatment through metaheuristic methods. He has published in *Omega*, *Operational Research*, *Annals of Operations Research*, *Dyna*, and *American Journal of Operations Research*, among others. He participates actively in the Operations Research Community in Latin-America.