Multi-Objective Optimization using Differential Evolution: A Survey of the State-of-the-Art

Efrén Mezura-Montes[1], Margarita Reyes-Sierra[2], and
Carlos A. Coello Coello[3*]

[1] Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa, Veracruz, 91000 MEXICO
emezura@lania.mx

[2] Universidad Veracruzana
Facultad de Estadística e Informática
Avenida Xalapa s/n esquina Ávila-Camacho, Xalapa, Veracruz, 91020 MEXICO
reyes_sierra@hotmail.com

[3] CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07360, MEXICO
ccoello@cs.cinvestav.mx

Summary. Differential Evolution is currently one of the most popular heuristics to solve single-objective optimization problems in continuous search spaces. Due to this success, its use has been extended to other types of problems, such as multi-objective optimization. In this chapter, we present a survey of algorithms based on differential evolution which have been used to solve multi-objective optimization problems. Their main features are described and, based precisely on them, we propose a taxonomy of approaches. Some theoretical work found in the specialized literature is also provided. To conclude, based on our findings, we suggest some topics that we consider to be promising paths for future research in this area.

1 Introduction

Nowadays, evolutionary algorithms (EAs) are considered a very effective alternative to solve complex search problems, including either global (i.e., single-objective) [39] or multi-objective optimization problems [10]. The first attempts to use EAs to solve multi-objective problems relied mainly on genetic algorithms (GAs) [16] and evolution strategies (ES) [6]. A comprehensive review of these approaches can be found in [10].

* Corresponding Author

In 1995, Storn and Price proposed the most recent evolutionary algorithm called Differential Evolution (DE) [45] to solve real-parameter optimization problems. DE uses a simple mutation operator based on differences between pairs of solutions (called vectors) with the aim of finding a search direction based on the distribution of solutions in the current population. DE also utilizes a steady-state-like replacement mechanism, where the newly generated offspring (called trial vector) competes only against its corresponding parent (old object vector) and replaces it if the offspring has a higher fitness value. In the remainder of this chapter, we will use trial vector and offspring as synonymous. The same applies for parent and old object vector.

DE shares some characteristics with previous EAs and also has some differences. The similarities are the following: DE is a population-based approach, recombination and mutation are the variation operators used to generate new solutions and a replacement mechanism provides capabilities to maintain a fixed size in the population.

However, unlike GAs, where binary encoding can be used, solutions in DE are coded with real values as in ES. But, DE does not use a fixed distribution (as the Gaussian distribution adopted in ES) to control the behavior of the mutation operator; instead, the current distribution of the solutions in the search space determines the stepsize and the search direction for each individual. This last feature seems to be one of its main advantages.

Due to the multicriteria nature of most real-world problems, multi-objective optimization problems are very common, particularly in engineering applications. As the name indicates, multi-objective optimization problems involve multiple objectives, which should be optimized simultaneously and that often are in conflict with each other. This results in a group of alternative solutions which must be considered equivalent in the absence of information concerning the relevance of the others.

Since Evolutionary Algorithms (EAs) deal with a group of candidate solutions, it seems natural to use them in multi-objective optimization problems to find a group of optimal solutions. Indeed, EAs have proved very efficient in solving multi-objective optimization problems [10, 11].

With the rise of new bio-inspired heuristics for numerical optimization, like Particle Swarm Optimization (PSO) [27] and also DE, it is important to analyze how they are adapted to solve different types of problems, like, in our case, multi-objective optimization problems. This work focuses on a review of the state-of-the-art in multi-objective optimization with DE as a search engine.

This chapter is organized as follows: In Section 2, DE is explained in detail and its main variants are presented. Section 3, provides the statement of the multi-objective optimization problems and also some related definitions. In Section 4 some multi-objective issues included in evolutionary multi-objective optimization are addressed. After that, in Section 5 we show our proposed taxonomy of DE-based approaches for multi-objective optimization. Some theoretical results regarding DE for multi-objective optimization are summarized in Section 6. Finally, Section 7 includes our conclusions and future paths of research.

2 Differential Evolution variants

There are some variants of the DE algorithm. They vary on (1) the type of the criterion to select one of the individuals to be used in the mutation operator (called donor vector), (2) the number of differences computed also in the mutation operator and, finally, (3) in the recombination operator chosen.

The most popular variant is called *"DE/rand/1/bin"*, where "DE" refers to the name of the algorithm, the word "rand" indicates that the donor vector selected to compute the mutation values is chosen at random, "1" is the number of pairs of solutions chosen (most of the time chosen at random) to calculate the mutation differential and finally "bin" means that a binomial recombination is used. The corresponding algorithm of this variant is presented in Figure 1.

Besides typical parameters used in EAs (number of individuals and number of generations), two parameters are adopted in DE: "CR" and "F". "CR" controls the influence of the parent in the generation of the offspring. Higher values mean less influence of the parent in the features of its offspring. "F" scales the influence of the set of pairs of solutions selected to calculate the mutation value (one pair in the case of the algorithm in Figure 1).

In Figure 2 the effect of the DE mutation and recombination operator in its most popular variant (*DE/rand/1/bin*) is explained. $\mathbf{x}_{r_3}$ is the donor solution which can be chosen either at random or it can be the best solution in the population. $\mathbf{x}_{r_1}$ and $\mathbf{x}_{r_2}$ are the pair of solutions chosen always at random and used to compute the difference between them in order to define a search direction. This difference is scaled with the "F" parameter. After that, it is added to $\mathbf{x}_{r_3}$ to define the location of the "mutation vector" (black square in Figure 2). This "mutation vector" is combined with the original parent with a binomial (discrete) recombination and the location of the mutation vector plus the two filled squares in the figure represent the possible positions of the offspring generated. Finally, this offspring will compete against its parent (based on fitness) and the best one will remain in the population for the next generation.

As it was mentioned before, the difference among the different DE variants are mainly on the way the donor solution (from the set chosen to compute the "mutation vector") is selected, the number of pairs of randomly chosen solutions and the type of recombination operator adopted. Among the main variations we distinguish the following:

- Variants with discrete recombination operator (either binomial or exponential):
  - *DE/rand/1/bin*

```
   Begin
      G=0
      Create a random initial population x_{i,G} ∀i, i = 1,...,NP
      Evaluate f(x_{i,G}) ∀i, i = 1,...,NP
      For G=1 to MAX_GEN Do
         For i=1 to NP Do
⇒           Select randomly r_1 ≠ r_2 ≠ r_3 :
⇒           j_rand = randint(1, D)
⇒           For j=1 to D Do
⇒              If (rand_j[0,1) < CR or j = j_rand) Then
⇒                 u_{i,j,G+1} = x_{r_3,j,G} + F(x_{r_1,j,G} − x_{r_2,j,G})
⇒              Else
⇒                 u_{i,j,G+1} = x_{i,j,G}
⇒              End If
⇒           End For
            If (f(u_{i,G+1}) ≤ f(x_{i,G})) Then
               x_{i,G+1} = u_{i,G+1}
            Else
               x_{i,G+1} = x_{i,G}
            End If
         End For
         G = G + 1
      End For
   End
```

Fig. 1. "DE/rand/1/bin" algorithm. randint(min,max) is a function that returns an integer number between min and max. rand$[0, 1)$ is a function that returns a real number between 0 and 1. Both are based on a uniform probability distribution. "NP", "MAX_GEN", "CR" and "F" are user-defined parameters. "D" is the dimensionality of the problem. Steps pointed with arrows change from variant to variant.

  – *DE/rand/1/exp*
  – *DE/best/1/bin*
  – *DE/best/1/exp*

  The "rand" variants select the donor solution ($\mathbf{x}_{r_3}$) and the pair of solutions to calculate the mutation differential ($\mathbf{x}_{r_1}$ and $\mathbf{x}_{r_2}$) at random. In contrast, the "best" variants use the best solution in the population as the donor solution and the pair of solutions are chosen at random.

- Variants with arithmetic recombination:
  – *DE/current-to-rand/1*
  – *DE/current-to-best/1*

  The only difference between them is that the first one selects the donor solution ($\mathbf{x}_{r_3}$) and the pair of solutions to calculate the differential mutation ($\mathbf{x}_{r_1}$ and $\mathbf{x}_{r_2}$) at random. The second one uses the best solution in

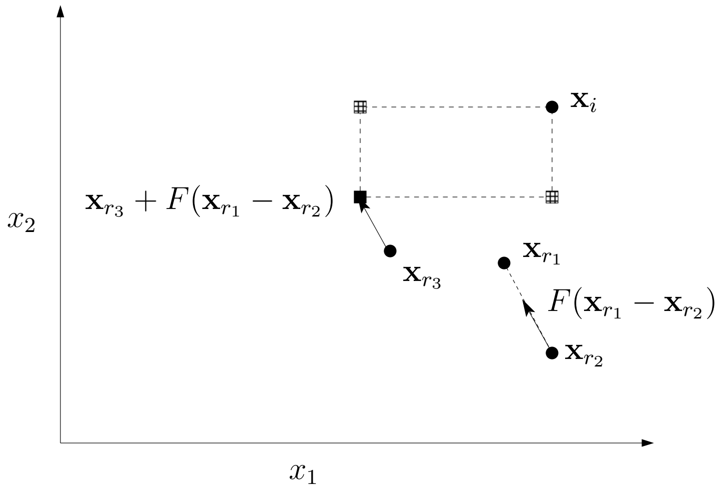

Fig. 2. DE/rand/1/bin recombination and mutation operators example. $\mathbf{x}_i$ is the current parent, $\mathbf{x}_{r_3}$ is the donor individual chosen at random (but it can be the best solution in the population in other variants), $\mathbf{x}_{r_1}$ and $\mathbf{x}_{r_2}$ are the individuals chosen at random to calculate the scaled difference between them and to define a search direction. The black square represents the mutation vector which can be the location of the only offspring generated after performing recombination. Additionally, the filled squares are the other two possible locations for the only offspring after recombination.

the population as the donor solution and, again, the pair of solutions to calculate the differential mutation are chosen randomly.
- Variants with combined arithmetic-discrete recombination:
  - *DE/current-to-rand/1/bin*

The implementation details of each DE variant are summarized in Table 1.

3 Multi-objective Optimization

We are interested in solving problems of the type[4]:

$$\text{minimize } \mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})] \tag{1}$$

subject to:

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{2}$$

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, p \tag{3}$$

[4] Without loss of generality, we will assume only minimization problems.

Variant
rand/p/bin: $u_{i,j} = \begin{cases} x_{r_3,j} + F \cdot \sum_{k=1}^{p} (x_{r_1^p,j} - x_{r_2^p,j}) & \text{if } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
rand/p/exp: $u_{i,j} = \begin{cases} x_{r_3,j} + F \cdot \sum_{k=1}^{p} (x_{r_1^p,j} - x_{r_2^p,j}) & \text{from } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
best/p/bin: $u_{i,j} = \begin{cases} x_{best,j} + F \cdot \sum_{k=1}^{p} (x_{r_1^p,j} - x_{r_2^p,j}) & \text{if } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
best/p/exp: $u_{i,j} = \begin{cases} x_{best,j} + F \cdot \sum_{k=1}^{p} (x_{r_1^p,j} - x_{r_2^p,j}) & \text{from } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
current-to-rand/p: $\mathbf{u}_i = \mathbf{x}_i + K \cdot (\mathbf{x}_{r_3} - \mathbf{x}_i) + F \cdot \sum_{k=1}^{p} (\mathbf{x}_{r_1^p} - \mathbf{x}_{r_2^p})$
current-to-best/p: $\mathbf{u}_i = \mathbf{x}_i + K \cdot (\mathbf{x}_{best} - \mathbf{x}_i) + F \cdot \sum_{k=1}^{p} (\mathbf{x}_{r_1^p} - \mathbf{x}_{r_2^p})$
current-to-rand/p/bin: $u_{i,j} = \begin{cases} x_{i,j} + K \cdot (x_{r_3,j} - x_{i,j}) + F \cdot \sum_{k=1}^{p} (x_{r_1^p,j} - x_{r_2^p,j}) & \text{if } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$

Table 1. DE basic variants. j_r is a random integer number generated between $[0, n]$, where n is the number of variables of the problem. $U_j(0,1)$ is a real number generated at random between 0 an 1. Both numbers are generated using a uniform distribution. p is the number of pairs of solutions used to compute the differences in the mutation operator. $\mathbf{u}_i$ is the offspring (or trial vector), $\mathbf{x}_{r_3}$ is the donor solution chosen at random, $\mathbf{x}_{best}$ is the best solution in the population as donor solution, $\mathbf{x}_i$ is the current parent (old object vector) and $\mathbf{x}_{r_1^p}$ and $\mathbf{x}_{r_2^p}$ are the "pth" pair to compute the mutation differential.

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., m$, $j = 1, ..., p$ are the constraint functions of the problem.

To describe the concept of optimality in which we are interested, we will introduce next a few definitions.

Definition 1. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$, we say that $\mathbf{x} \leq \mathbf{y}$ if $x_i \leq y_i$ for $i = 1, ..., k$, and that $\mathbf{x}$ **dominates** $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$) if $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{x} \neq \mathbf{y}$.

Figure 3 shows a particular case of the **dominance relation** in the presence of two objective functions.

Definition 2. We say that a vector of decision variables $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ is **nondominated** with respect to $\mathcal{X}$, if there does not exist another $\mathbf{x}' \in \mathcal{X}$ such that $\mathbf{f}(\mathbf{x}') \prec \mathbf{f}(\mathbf{x})$.

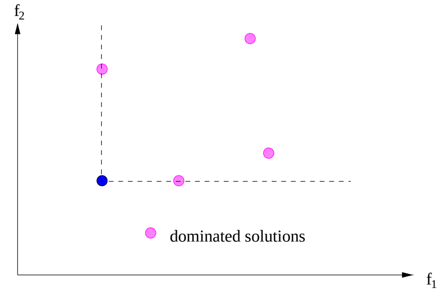

Fig. 3. Dominance relation in a bi-objective space.

Definition 3. We say that a vector of decision variables $\mathbf{x}^* \in \mathcal{F} \subset \mathbb{R}^n$ ($\mathcal{F}$ is the feasible region) is **Pareto-optimal** if it is nondominated with respect to $\mathcal{F}$.

Definition 4. The **Pareto Optimal Set** $\mathcal{P}^*$ is defined by:

$$\mathcal{P}^* = \{\mathbf{x} \in \mathcal{F} | \mathbf{x} \text{ is Pareto-optimal}\}$$

Definition 5. The **Pareto Front** $\mathcal{PF}^*$ is defined by:

$$\mathcal{PF}^* = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^k | \mathbf{x} \in \mathcal{P}^*\}$$

Figure 4 shows a particular case of the **Pareto front** in the presence of two objective functions.

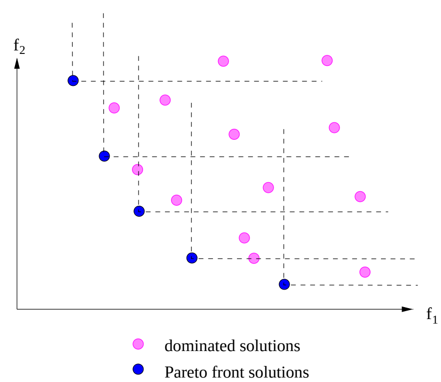

Fig. 4. The Pareto front of a set of solutions in a two objective space.

We thus wish to determine the Pareto optimal set from the set $\mathcal{F}$ of all the decision variable vectors that satisfy (2) and (3). Note however that in practice, not all the Pareto optimal set is normally desirable (e.g., it may not be desirable to have different solutions that map to the same values in objective function space) or achievable.

4 Differential Evolution for Multi-Objective Problems

In order to apply the DE strategy for solving multi-objective optimization problems, the original scheme has to be modified since the solution set of a problem with multiple objectives does not consist of a single solution (as in global optimization). Instead, in multi-objective optimization, we aim to find a set of different solutions (the so-called Pareto optimal set), as mentioned in Section 3.

Two are the main aspects that have been considered by researchers who have extended the DE approach to multi-objective optimization:

1. How to promote diversity into the population?
2. How to select and/or retain the best individuals? That is, how to perform *elitism*?

We briefly discuss these two design aspects in the following Sections.

4.1 Promoting Diversity

Promoting diversity may be done through the selection process by means of mechanisms based on some *quality* measures that indicate the closeness of the individuals within the population. In order to help understanding the specific approaches that are going to be described later on, we present here two of the most important density measures used in the area of multi-objective optimization:

- **Crowding distance** [14]. The crowding distance factor gives us an idea of how crowded are the closest neighbors of a given individual, in objective function space. This measure estimates the perimeter of the cuboid formed by using the nearest neighbors as the vertices. See Figure 5.

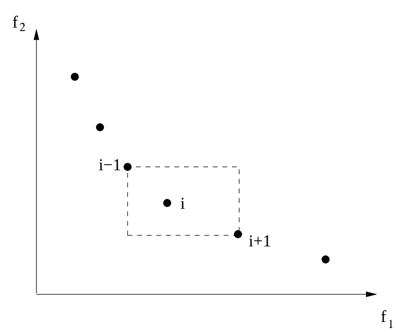

Fig. 5. The crowding distance factor for an example with two objective functions. Individuals with a larger value of this factor are preferred.

- **Fitness sharing** [17, 13]: When an individual is sharing resources with others, its fitness is degraded in proportion to the number and closeness to individual that surround it within a certain perimeter. A neighborhood of an individual is defined in terms of a parameter called σ_{share} that indicates the radius of the neighborhood. Such neighborhoods are called *niches*. See Figure 6.

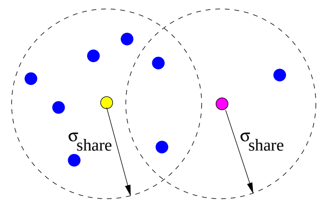

Fig. 6. For each individual, a niche is defined. Individuals whose niche is less crowded are preferred.

4.2 Performing Elitism

In evolutionary multi-objective optimization, elitism is usually implemented through an external archive, also called secondary population, which stores the nondominated individuals found along the search. Such an archive will allow the entrance of a solution only if: (a) it is nondominated with respect to the contents of the archive or (b) it dominates any of the solutions within the archive (in this case, the dominated solutions have to be deleted from the archive).

Besides, elitism can also be implemented through the use of $(\mu + \lambda)$-selection (also called *plus* selection), by which, at each generation, parents and children are compared in order to select the best of them to conform the next population.

One of the most popular mechanisms used to select the best individuals from the combined population of parents and children is the so-called *nondominated sorting* approach. This approach is based on the Pareto ranking mechanism firstly proposed by Goldberg in 1989 [16]. The nondominated sorting mechanism ranks the individuals of the population in different levels in the following way. All nondominated individuals are classified into one category with rank 1 (level 1), then, this group of individuals with rank 1 is ignored and the process is repeated. This time, the nondominated individuals will have rank 2 (level 2). The process continues until all individuals are classified. Individuals with lower rank are always preferred for selection. Figure 7 shows the ranking process of the nondominated sorting approach.

According to Goldberg [16], to maintain appropriate diversity, the nondominated sorting procedure should be used in conjunction with niching techniques as, for example, the fitness sharing mechanism previously mentioned.

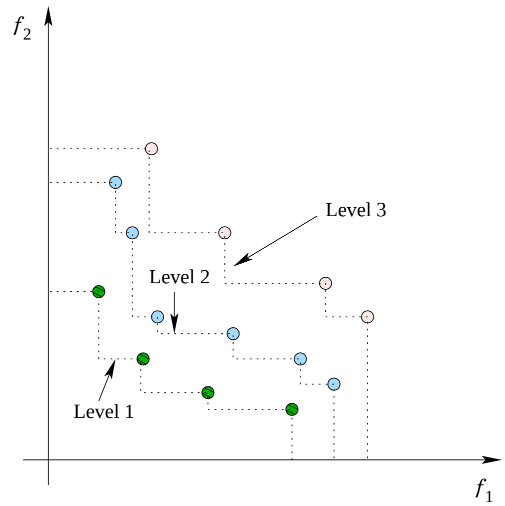

Fig. 7. Ranking process of the nondominated sorting approach.

The first multi-objective evolutionary algorithm (MOEA) which used the nondominated sorting approach proposed by Goldberg was the Nondominated Sorting Genetic Algorithm (NSGA) proposed by Srinivas and Deb in 1994 [54]. The NSGA algorithm combined the nondominated sorting approach with fitness sharing in its corresponding fitness assignment process. Later, the improved version of the NSGA, called NSGA-II [14], incorporated the nondominated sorting approach in order to perform a $(\mu+\lambda)$-selection. In the NSGA-II, the population for the next generation is obtained by first introducing all individuals with rank 1 (level 1), then all individuals with rank 2 (level 2) and the process continues until the population is complete. However, when there is not enough space to include all individuals with rank i (level i), a crowding distance mechanism is applied in order to select the best individuals from such level.

5 Taxonomy

In this section, we propose a classification of approaches, based on common features to adapt DE for multi-objective optimization. The proposed classes are enumerated as follows:

1. Non-Pareto-based approaches.
2. Pareto-based approaches.
   a) Using Pareto dominance
   b) Using Pareto ranking.
3. Combined approaches.

In the following Subsections, we will show the approaches located in each category, by describing their main features, regarding the DE variant used and their companion mechanisms to deal with multi-objective problems.

5.1 Non-Pareto-Based Approaches

In this class, we consider those approaches that use multi-objective concepts like combination of functions, problem transformation, etc.

Babu and Jehan [5] propose the Differential Evolution for Multi-Objective Optimization approach. This algorithm uses the *DE/rand/1/bin* variant with two different mechanisms to solve bi-objective problems: (1) incorporating one objective function as a constraint, and (2) using an aggregating function. A single optimal solution is obtained after N iterations using the *Penalty Function Method* [11] to handle the objective treated as a constraint in the first case. On the other hand, a set of optimal solutions is obtained after N iterations using the *Weighting Factor Method* [11] to provide the importance of each objective from the user's perspective, in the second case. The authors present results for two bi-objective problems and compare them with respect to a simple GA. The authors indicate that the DE algorithm provides the exact optimum with a lower number of evaluations than the GA.

Li and Zhang [36] propose a multi-objective differential evolution algorithm based on decomposition (MODE/D) for continuous multi-objective optimization problems with variable linkages. The authors use the weighted Tchebycheff approach to decompose a multi-objective optimization problem into several scalar optimization subproblems. The differential evolution operator based on the *DE/rand/1/bin* variant is used for generating new trail solutions, and a neighborhood relationship among all the subproblems generated is defined, such that they all have similar optimal solutions. For validating their approach, the authors adopt test problems with variable linkages [41] and propose variants of some of the Zitzler-Deb-Thiele (ZDT) test problems [59]. Results are compared with respect to the NSGA-II [14], the Nondominated Sorting Differential Evolution (NSDE) [25] and GD3 [30]. The authors report that MODE/D clearly outperformed the other approaches with respect to which it was compared.

5.2 Pareto-Based Approaches

In this group we classify those methods that use Pareto concepts to deal with multiple objectives. We divided this class into two subclasses, because we detected two different ways to apply them: (1) as a criterion to select the best solution in the DE selection mechanism and (2) as a ranking procedure.

Pareto dominance

In this subclass, we describe those approaches where Pareto Dominance was used as a criterion to select the best solution between the old population vector and the trial vector. One of the features that distinguishes each approach is the decision made when both solutions are nondominated between each other. Furthermore, other authors use Pareto Dominance as a filter to get only nondominated solutions.

Apparently, Chang et al. [9] constitutes the first reported attempt to extend differential evolution for multi-objective problems. In this paper, the authors use *DE/rand/1/bin* with an external archive (called "Pareto optimal set" by the authors) to store the nondominated solutions obtained during the search. The approach also incorporates fitness sharing to maintain diversity. An interesting aspect of this approach is that the selection mechanism of the differential evolution algorithm is modified in order to enforce that the members of the new generation are nondominated not only regarding their objective values but also regarding a set of distance metric values (one assigned to each objective) which ensure the new solutions are at certain minimum distance from the previously found nondominated solutions. This approach is adopted to fine-tune the fuzzy automatic train operation (ATO) for a typical mass transit system, in which three objectives are considered: (1) punctuality (least deviation from scheduled arrival time), (2) least energy consumption and (3) maximum passenger comfort. This application is discussed in further detail in [8].

Abbass et al. [4, 3, 51] propose the Pareto Differential Evolution (also abbreviated as PDE) algorithm. The authors use an special case of the *DE/-current_to_rand/1/bin* variant with $K = 0$ (see last row in Table 1), because the old population vector (the parent) is used in the calculation of the trial vector (combined with the difference vector) and also in the discrete recombination. The algorithm works as follows. The initial population is initialized using a Gaussian distribution with mean 0.5 and standard deviation 0.15. Only the nondominated solutions are retained in the population for recombination (all dominated solutions are removed). Three parents are randomly selected (one as the main parent and also trial solution) and a child is generated with them. The offspring is placed in the population only if it dominates the main parent; otherwise, a new selection process takes place. This process continues until the population is completed. If the number of nondominated solutions exceeds a certain threshold (50 was adopted in [4]), a distance metric is adopted to remove parents which are too close from each other (this can be seen as a niching procedure in which this distance metric is the niche radius). In this approach, the step-length parameter F is generated from a Gaussian distribution $N(0, 1)$ and the boundary constraints are preserved either by reversing the sign if the variable is ≤ 0 or by repetitively subtracting 1 if it is ≥ 0, until the variable is within the allowable boundaries. This algorithm also incorporates a mutation operator which is applied with certain probability (after the crossover operator), by adding to each variable a small random perturbation. PDE is compared with respect to SPEA [60] in [3] (without mutation) and also with respect to many other approaches (including PAES [28], the NSGA [54] and the NPGA [22]) in [4] (including mutation).

In [2], a new version of PDE is introduced. This version is called Self-adaptive Pareto Differential Evolution (SPDE) algorithm, because it self-adapts its crossover and its mutation rates.

In [1], Abbass proposes an approach called Memetic Pareto Artificial Neural Networks (MPANN). This approach consists of a version of Pareto Differential Evolution (PDE) [3] enhanced with the Back-Propagation (BP) local search algorithm, in order to speed up convergence. MPANN is used to evolve neural networks in which an attempt is made to obtain a trade-off between the architecture and generalization ability of the network. So, two objectives are minimized: (1) error and (2) the number of hidden units. MPANN is validated using two benchmark data sets: the Australian credit card assessment problem and the diabetes problem (both were taken from the UCI Machine Learning Repository [40]). Results are compared with respect to 23 algorithms, which include decision trees, rule-based methods, neural networks and statistical algorithms. MPANN was able to outperform the traditional backpropagation approach and obtained results competitive against the other 23 algorithms with respect to which it was compared.

Kukkonen and Lampinen extended *DE/rand/1/bin* to solve multi-objective optimization problems in their approach called Generalized Differential Evolution (GDE). In fact, GDE is able to solve global and multi-objective optimization problems (either constrained or unconstrained). The first version of their approach [32] modified the original DE selection operation by introducing Pareto Dominance as a selection criterion between the old population member and the trial vector. Also, Pareto dominance in the constraint space is considered to handle the constraints of the problem.

To promote a better distribution of the nondominated solutions, a second version of the approach, called GDE2 [29] was introduced. In this version, a crowding distance measure was used to select the best solution when the old population vector and the trial vector are feasible and nondominated with respect to each other, in such a way that the vector located in the less crowded region will be part of the population of the next generation. The authors acknowledge that GDE2 was sensitive to its initial parameters and that the modified selection mechanism slows down convergence.

Santana-Quintero and Coello Coello [50] propose the ϵ-MyDE. This approach keeps two populations: the main population (which is used to select the parents) and a secondary (external) population, in which the concept of ϵ-dominance [35] is adopted to retain the nondominated solutions found and to distribute them in an uniform way. The concept of ϵ-dominance does not allow two solutions with a difference less than ϵ_i in the i-th objective to be nondominated with respect to each other, thereby allowing a good spread of solutions. ϵ-MyDE uses real numbers representation, and incorporates a constraint-handling mechanism that allows infeasible solutions to intervene during recombination. *DE/rand/1/bin* variant is used to evolve the main population. However, after a user-defined number of generations, the three random solutions used in the mutation operator are selected from the secondary population in such a way that they are close among them in the objective function space. If none of the solutions satisfies this condition, a random solution from this secondary population is chosen. Finally, to improve exploration capabilities, a uniform mutation operator is added.

Portilla Flores [44] proposes a multi-objective version of differential evolution, which is used for concurrent design of pinion–rack continuously variable transmission (CVT). This mechatronic design problem is formulated as a dynamic multi-objective optimization problem in which two objectives are considered: (1) maximize the mechanical CVT efficiency, and (2) minimize the controller energy. The *DE/rand/1/bin* variant is used in this approach and Pareto dominance between a parent and its offspring works like the selection criterion. Also, this technique incorporates a secondary population to retain the nondominated solutions found during the evolutionary process. Finally, it uses the feasibility rules from [38] to handle the constraints of the problem.

However, the approach does not include an explicit mechanism to maintain diversity (although a set of diverse solutions is actually generated). An interesting aspect of this work is that results are compared with respect to a mathematical programming technique: the goal attainment method. The comparison of results indicated that, as expected, the goal attainment method was very sensitive to its initial search point. Also, in several runs, it was not able to converge to a feasible solution. In contrast, the differential evolution algorithm was able to converge to feasible solutions in all the runs performed. However, the solutions generated by the goal attainment method were nondominated with respect to the solutions produced by differential evolution. Additionally, the CPU time required by differential evolution was about twice the time required by the goal attainment method.

Pareto Ranking

This subsection includes those approaches where a Pareto ranking procedure was added to them. The aim is to perform a $(\mu + \lambda)$-selection after the set of trial vectors have been generated from the current population.

In [37], Madavan proposes the Pareto-Based Differential Evolution (PBDE) approach. In this algorithm, Differential Evolution is extended to multi-objective optimization by incorporating Pareto-based mechanisms proposed previously by Deb et al. [12, 14]. It is interesting to note that this approach uses the same special case of the *DE/current_to_rand/1/bin* variant used by Abbass [4, 3, 51] where $K = 0$. The PBDE algorithm modifies the selection procedure in the basic DE algorithm by incorporating the key elements of the NSGA-II algorithm: the nondominated sorting and ranking selection procedure. In this way, once the new candidate vectors are obtained using DE operators (where the basic crossover operator is applied using the trial vector as the main parent), the new population is combined with the existing parents population and then the best members of the combined population (parents plus offspring) are chosen. As in the NSGA-II algorithm, the population for the next generation is filled by taking the individuals from the best nondominated rank down and discarding individuals with the same rank based on the diversity measure (crowding distance). This algorithm is not compared with respect to any other approach and is tested on 10 different unconstrained problems performing 250,000 evaluations. The authors indicate that the approach has difficulties to converge to the true Pareto front in two problems (Kursawe's test function [31] and ZDT4 [59]).

Xue et al. [56, 55] propose the Multi-Objective Differential Evolution (MODE) approach. This algorithm uses a variant of DE created by the authors in which the best individual is incorporated to create the offspring. This variant has some similarities with the traditional *DE/best/1/bin*. A Pareto-based approach is introduced to implement the selection of the best individual: if the trial solution is dominated, a set of nondominated individuals can be identified, and the "best" turns out to be any individual (randomly picked) from this set. On the other hand, if the trial solution is nondominated, it will be the "best" solution itself. The formula used by Xue et al. to create the offspring is the following:

$$p_i' = \gamma \cdot p_{best} + (1-\gamma)p_i + F \cdot \sum_{k=1}^{K} \left(p_{i_{a^k}} - p_{i_{b^k}}\right)$$

where p_{best} is the best individual selected, $\gamma \in [0,1]$ represents the greediness of the operator, and K is the number of perturbation vectors (they use $K = 2$). It is worth noting that the previous formula is applied with certain mutation probability (p_m). Also, the authors adopt $(\mu + \lambda)$-selection, Pareto ranking (according to Goldberg [16]) and crowding distance [14] in order to produce and maintain well-distributed solutions. Actually, the authors incorporate a new parameter, called σ_{crowd}, which is used to penalize the fitness of the individuals, based on the crowding distance values, in order to improve the $(\mu + \lambda)$-selection approach. MODE is used to solve five high dimensionality unconstrained problems with 250,000 evaluations and the results are compared only to those obtained by SPEA [61].

Iorio and Li [25] propose the Nondominated Sorting Differential Evolution (NSDE). This approach is a simple modification of the NSGA-II [14]. The only difference between this approach and the NSGA-II is in the method for generating new individuals. The NSGA-II uses a real-coded crossover and mutation operator, but in the NSDE, these operators are replaced with the operators of Differential Evolution. New candidates are generated using the *DE/current-to-rand/1* variant, which is known to be rotationally invariant. A number of experiments are conducted on a uni-modal rotated problem from the literature. NSDE is used to solve a uni-modal rotated problem with a certain degree of rotation on each plane. The results of the NSDE outperformed those produced by the NSGA-II, and thus, it is shown that Differential Evolution can provide rotationally invariant behavior on a multi-objective optimization problem. In further work, Iorio and Li [26] propose three new versions of

NSDE that incorporate directional information, by selecting parents for the generation of new individuals according to measures of both convergence and spread. For convergence, the authors modify the selection process (of the main parent) of NSDE in order to calculate differential vectors that point towards regions where better ranked individuals are located. For spread, the authors modify NSDE so that it favors the selection process (of the supporting parents) from different regions of decision variable space, but with the same rank. The modified approach is called NSDE-DCS (DCS stands for "directional convergence and spread") and is compared with respect to the NSGA-II, the original NSDE [25], NSDE-DC (NSDE only with the directional convergence mechanism), and NSDE-DS (NSDE only with the directional spread mechanism). Results indicate that all the NSDE versions outperform the NSGA-II, but NSDE-DS practically provides the same results as NSDE-DCS. This is a very interesting outcome that indicates that improving spread may, in some cases, also improve convergence.

Robič and Filipič [48] propose an approach called Differential Evolution for Multi-Objective Optimization (DEMO). They used the *DE/rand/1/bin* variant. DEMO modifies the mechanism followed to decide when a new vector replaces the parent: if the new vector dominates the parent, the new vector replaces the parent; if the parent dominates the new vector, the new vector is discarded; otherwise, the new vector is added in the population. In this way, the population can be extended and the newly created vectors take part immediately in the creation of the subsequent vectors. After the creation process of new vectors has finished, DEMO applies a nondominated sorting mechanism (combined with the use of the crowding distance measure) in order to truncate the population and maintain a fixed number of vectors at each iteration. This enables a fast convergence towards the true Pareto front, while the use of nondominated sorting and crowding distance (derived from the NSGA-II [14]) of the extended population promotes the uniform spread of solutions. Robič and Filipič also propose two additional versions of DEMO in which the newly created vector is not compared against the parent, but against the most similar individual in either the decision variable space or the objective space. The three DEMO variants are compared in five high-dimensionality unconstrained problems outperforming in some problems to the NSGA-II, PDE [2], PAES [28], SPEA [61] and MODE [56]. However, the authors didn't find any variant of DEMO to be significantly better than another, so they recommend to use the original version of DEMO (which compares the new vector against the parent), since it is the most efficient one (computationally speaking).

To deal with the shortcomings of GDE2 (described in the previous group) regarding slow convergence, Kukkonen and Lampinen proposed an improved version called GDE3 [30] (a combination of the earlier GDE versions and the Pareto-Based Differential Evolution algorithm [37]). This version added a growing population size and nondominated sorting (as in the NSGA-II [14]) to improve the distribution of solutions in the final Pareto front and to decrease the sensitivity of the approach to its initial parameters. In GDE3, when the old population vector and the trial vector are feasible and nondominated with respect to each other, both of them are maintained. Hence, the population size will grow. To maintain a fixed population size for the next generation, nondominated sorting is performed after each generation to prune the population size. GDE3 is compared with respect to the NSGA-II in several test functions, including some from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [15].

5.3 Combined Approaches

Finally, this class considers those approaches where a set of schemes have been mixed in the DE-based multi-objective algorithm. There are approaches which consider either Pareto concepts and also population-based concepts in the same approach, or techniques where, besides global search, local search is considered.

In [42], Parsopoulos et al. introduce a parallel multi-population DE called the Vector Evaluated Differential Evolution (VEDE) approach, for multi-objective optimization. VEDE is inspired by the Vector Evaluated Genetic Algorithm (VEGA) [52] approach. A number M of sub-populations are considered in a ring topology. Each population is evaluated using one of the objective functions of the problem, and there is an exchange of information among the populations through the migration of the best individuals. In this way, only the versions of DE that use the best individual to create new vectors can take full advantage of this information exchange. Also, the algorithm incorporates a domination selection procedure to enhance its performance by favoring non-dominated individuals in the population. The selection mechanism introduced by Parsopoulos et al. is similar to that used by Abbass et al. [4], in which the new vector is introduced in the population if it dominates the main parent. Finally, VEDE uses an external archive for the maintenance of the Pareto optimal set. VEDE is validated using four bi-objective unconstrained problems and is compared with respect to VEGA. Furthermore, VEDE was tested on three versions, using different DE variants: *DE/best/1/bin*, *DE/best/2/bin* and *DE/current_to_best/1*. The authors indicate that the proposed approach outperformed VEGA in all cases, however, among the three DE variants, none of the them was clearly superior to the other two.

Santana-Quintero's approach (ϵ-MyDE) was further hybridized with rough sets to give raise to a new approach called DEMORS (Differential Evolution for Multiobjective Optimization with Rough Sets) [20]. DEMORS operates in two phases. During the first phase, an improved version of ϵ-MyDE is applied for 2000 fitness function evaluations. The main improvement on ϵ-MyDE is the incorporation of the so-called Pareto-adaptive ϵ-grid [21] for the secondary population. The concept of Pareto-adaptive ϵ-dominance eliminates several of the drawbacks of ϵ-dominance [35]. During the second phase, a local search procedure based on rough sets theory [43] is applied for 1000 fitness function evaluations, in order to improve the solutions produced at the previous phase.

The idea is to combine the high convergence rate of differential evolution with the high local search capabilities of rough sets. DEMORS is able to converge to the true Pareto front (or very close to it) in test problems with up to 30 decision variables, while only performing 3000 fitness function evaluations. Results are compared with respect to the NSGA-II.

Landa-Becerra and Coello Coello [34] propose the use of the ε-constraint technique [18] hybridized with a single-objective evolutionary optimizer: the cultured differential evolution [33]. The variant used in this approach is *DE/rand/1/bin*, however, the influence of the knowledge of the problem during the process, allows to change the variant to *DE/best/1/bin*. In fact, some modifications to the original *DE/rand/1/bin* are used (e.g. using the absolute value of the differences, adding another scaling factor besides "F" and using historical values of the best solution during the evolutionary process). The ε-constraint method transforms a multi-objective optimization problem into several single-objective optimization problems (each of these optimizations leads to a single Pareto optimal point). This method has been normally disregarded in the evolutionary multi-objective optimization literature due to its high computational cost [53, 46]. However, the authors argue that, if care is placed in the single-objective optimizer, this sort of hybrid can generate the true Pareto front of very difficult multi-objective optimization problems at a reasonable computational cost. Such a hypothesis is validated by solving DTLZ8 and DTLZ9 from the benchmark proposed in [15] together with several other test problems from the benchmark proposed in [23, 24]. All of these test functions are considered very hard to solve by current MOEAs, and this is illustrated by showing the results obtained by the NSGA-II in them. In most cases, even when performing a very high number of fitness function evaluations, the NSGA-II is unable to reach the true Pareto front. In contrast, the hybrid algorithm proposed in this paper is able to converge to the true Pareto front (or very close to it) of all the problems.

6 Convergence Properties of Multi-Objective Differential Evolution

Some theoretical studies about multi-objective extensions of differential evolution have been done recently. In [56, 57], Xue et al. perform a mathematical modeling and convergence analysis of a *continuous* Multi-Objective Differential Evolution (C-MODE) algorithm. The convergence properties of C-MODE are studied in a similar manner to the work presented by Hanne in [19], where convergence has been defined as follows:

Definition 6. A MOEA is said to **converge to the entire set of Pareto optimal solutions $\mathcal{P}^*$ with probability one** if

$$\mathrm{d}(\mathcal{P}^*,\mathcal{P}^t) \rightarrow 0 \text{ with probability one as } t \rightarrow \infty,$$

where $\mathrm{d}(\mathcal{P}^*,\mathcal{P}^t)$ is a distance function between $\mathcal{P}^*$ and $\mathcal{P}^t$, and it is defined as:

$$\mathrm{d}(\mathcal{P}^*, \mathcal{P}^t) = \mid \mathcal{P}^* \cup \mathcal{P}^t \mid - \mid \mathcal{P}^* \cap \mathcal{P}^t \mid$$

The approach of Xue et al. employs underlying geometric structures (*cones*) based primarily on convex sets, to prove the convergence of the population of the C-MODE to the Pareto optimal set with probability one. Readers are directed to the associated references for a detailed description and associated theorem proof details.

On the other hand, Xue et al. study the C-MODE operators and their effects on the convergence properties of the algorithm, under the Gaussian initial population assumption. They show that the limiting properties of C-MODE depend on the factor $(2KF^2 + (1-\gamma)^2)$, where K, F and γ are the parameters associated to the approach. If this factor is greater than 1, the population variance matrix explodes, and C-MODE successfully identifies the optimal solution set; otherwise, the population variance matrix vanishes.

Xue et al. confirm the mathematical results developed by simulation results obtained by applying C-MODE to numerical examples with different parameter settings. Also, they conduct simulation results on complicated continuous benchmark functions and show that the C-MODE performs better when the parameters are set to meet the obtained conditions. In this way, the results obtained by Xue et al. can also be used to guide the parameter setting of the C-MODE when applied in real world applications.

In [56, 58], Xue et al. extend their theoretical work by modeling a *discrete* version of MODE, D-MODE, in the framework of Markov processes and develop the corresponding convergence properties. They study the Markov model for the D-MODE with finite population size. Two situations are considered: one with a population large enough to contain all the Pareto optimal solutions while the other is the opposite. In the second situation, an external archive is needed to store all visited Pareto optimal solutions. In both cases, Xue et al. prove the convergence with probability one of D-MODE to the set of Pareto optimal solutions in a similar manner to the work presented by Rudolph in [49].

7 Conclusions and Future Research Paths

In this chapter, we have presented a survey of Differential Evolution approaches modified to solve multi-objective optimization problems. We found that the techniques can be categorized in three classes: (1) Non-Pareto-based, (2) Pareto-based and (3) combined approaches. In fact, Pareto-based approaches were divided into two sub-classes: Using Pareto dominance and Using Pareto ranking. Combined approaches, as the name indicates, combines different schemes (e.g. global and local search, Pareto dominance and ranking) into one single approach.

In other heuristics to solve multi-objective optimization problems, such as particle swarm optimization (PSO) [47], key features of the heuristic itself have been adapted as to get some benefit in the way the problem is being solved (e.g. leader selection for creating new solutions). In contrast, from our findings, we observed that in the case of DE, the selection of the individuals used for the generation of new solutions has not been modified in most cases, with the exception of a few proposals [26]. We also found that the most popular schemes added to DE for multi-objective optimization were Pareto dominance for the selection mechanism between parent and offspring and Pareto ranking after the whole set of offspring have been generated.

Based on the aforementioned findings, we enumerate the topics we consider as promising paths for future research:

- **Diversity**: DE has shown a high convergence rate, like other metaheuristics such as PSO [27], but with a higher degree of robustness. However, DE present problems to actually reach the true Pareto front (it gets trapped in local optimum fronts). Furthermore, DE has some problems to spread solutions along the obtained front. This seems to indicate that multi-objective DE-based approaches require alternative (i.e., more effective) diversity maintenance schemes.
- **Variants**: Most approaches included in this survey use the most popular variant (*DE/rand/1/bin*) [9, 5, 32, 29, 30, 48, 50, 44, 20, 34, 36]. Despite the fact that other authors have used others variants like *DE/current_to_best/1* [42], *DE/current_to_rand/1* [26], special cases of *DE/current_to_rand/1/bin* [4, 1, 3, 37] and new variants [56, 55], it is not clear which variant is more suited for multi-objective optimization (i.e., which type of mutation and recombination operator is able to bias the search in a better way as to reach the true Pareto front in a more effective manner).
- **DE mutation operator**: In DE for global optimization, it is common to assume that the vectors that will be used to calculate the differences when computing the trial vector are chosen at random. However, as it was shown by Iorio and Li [26], in multi-objective optimization, some additional criteria might be taken into account for the selection of the pairs of solutions to participate in the mutation operator.
- **Parameter adaptation**: Online and self-adaptation attempts are still scarce in multi-objective differential evolution. Novel schemes to adapt key parameters like “CR”, “F” or even the number of differences for the mutation operator are promising topics for future research.
- **Alternative encodings**: DE was proposed for continuous search spaces. Thus, one topic of interest is to develop alternative encodings that allow the use of differential evolution in problems requiring alternative encodings (e.g., combinatorial optimization problems). The use of encodings such as the random keys [7] or other proposals may be alternatives worth exploring in such cases.

- **Theory**: Studies about convergence of different DE variants, and runtime analysis, among other topics, will improve the current DE theory.
- **Applications**: Another path of research is the application of previously proposed DE-based approaches to the solution of real-world multi-objective optimization problems. Interesting behaviors may be found when applying DE in real-world multi-objective search spaces.

Acknowledgments

The first author acknowledges support from CONACyT through project number 52048-Y. The third author acknowledges support from CONACyT through project number 45683-Y.

References

1. Hussein A. Abbass. A Memetic Pareto Evolutionary Approach to Artificial Neural Networks. In *The Australian Joint Conference on Artificial Intelligence*, pages 1–12, Adelaide, Australia, December 2001. Springer. Lecture Notes in Artificial Intelligence Vol. 2256.
2. Hussein A. Abbass. The Self-Adaptive Pareto Differential Evolution Algorithm. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 831–836, Piscataway, New Jersey, May 2002. IEEE Service Center.
3. Hussein A. Abbass and Ruhul Sarker. The Pareto Differential Evolution Algorithm. *International Journal on Artificial Intelligence Tools*, 11(4):531–552, 2002.
4. Hussein A. Abbass, Ruhul Sarker, and Charles Newton. PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 971–978, Piscataway, New Jersey, May 2001. IEEE Service Center.
5. B.V. Babu and M. Mathew Leenus Jehan. Differential Evolution for Multi-Objective Optimization. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2696–2703, Canberra, Australia, December 2003. IEEE Press.
6. Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.
7. James C. Bean. Genetics and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.
8. C.S. Chang and D.Y. Xu. Differential Evolution Based Tuning of Fuzzy Automatic Train Operation for Mass Rapid Transit System. *IEE Proceedings of Electric Power Applications*, 147(3):206–212, May 2000.
9. C.S. Chang, D.Y. Xu, and H.B. Quek. Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system. *IEE Proceedings on Electric Power Applications*, 146(5):577–583, September 1999.

10. Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.
11. Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.
12. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
13. Kalyanmoy Deb and David E. Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.
14. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
15. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.
16. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
17. David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum Associates, 1987.
18. Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297, July 1971.
19. Thomas Hanne. On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, 117(3):553–564, 1999.
20. Alfredo G. Hernández-Díaz, Luis V. Santana-Quintero, Carlos Coello Coello, Rafael Caballero, and Julián Molina. A New Proposal for Multi-Objective Optimization using Differential Evolution and Rough Sets Theory. In Maarten Keijzer et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 675–682, Seattle, Washington, USA, July 2006. ACM Press. ISBN 1-59593-186-4.
21. Alfredo G. Hernández-Díaz, Luis V. Santana-Quintero, Carlos A. Coello Coello, and Julián Molina. Pareto-adaptive ϵ-dominance. *Evolutionary Computation*, 15(4):493–517, Winter 2007.
22. Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.

23. Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A Scalable Multi-objective Test Problem Toolkit. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 280–295, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.
24. Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.
25. Antony W. Iorio and Xiaodong Li. Solving rotated multi-objective optimization problems using differential evolution. In *AI 2004: Advances in Artificial Intelligence, Proceedings*, pages 861–872. Springer-Verlag, Lecture Notes in Artificial Intelligence Vol. 3339, 2004.
26. Antony W. Iorio and Xiaodong Li. Incorporating Directional Information within a Differential Evolution Algorithm for Multi-objective Optimization. In Maarten Keijzer et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 691–697, Seattle, Washington, USA, July 2006. ACM Press. ISBN 1-59593-186-4.
27. James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.
28. Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
29. Saku Kukkonen and Jouni Lampinen. An Extension of Generalized Differential Evolution for Multi-objective Optimization with Constraints. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 752–761, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.
30. Saku Kukkonen and Jouni Lampinen. GDE3: The third Evolution Step of Generalized Differential Evolution. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 443–450, Edinburgh, Scotland, September 2005. IEEE Service Center.
31. Frank Kursawe. A Variant of Evolution Strategies for Vector Optimization. In Hans-Paul Schwefel and Reinhard Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, pages 193–197, Dortmund, Germany, October 1991. Springer-Verlag. Lecture Notes in Computer Science No. 496.
32. Jouni Lampinen. De's selection rule for multiobjective optimization. Technical report, Lappeenranta University of Technology, Department of Information Technology, 2001.
33. Ricardo Landa Becerra and Carlos A. Coello Coello. Cultured differential evolution for constrained optimization. *Computer Methods in Applied Mechanics and Engineering*, 195(33–36):4303–4322, July 1 2006.
34. Ricardo Landa Becerra and Carlos A. Coello Coello. Solving Hard Multiobjective Optimization Problems Using ε-Constraint with Cultured Differential Evolution. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund Burke, Juan J. Merelo-Guervós, L. Darrell Whitley, and Xin Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, pages 543–552. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland, September 2006.

35. Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.
36. Hui Li and Qingfu Zhang. A Multiobjective Differential Evolution Based on Decomposition for Multiobjective Optimization with Variable Linkages. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund Burke, Juan J. Merelo-Guervós, L. Darrell Whitley, and Xin Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, pages 583–592. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland, September 2006.
37. Nateri K. Madavan. Multiobjective Optimization Using a Pareto Differential Evolution Approach. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1145–1150, Piscataway, New Jersey, May 2002. IEEE Service Center.
38. Efrén Mezura-Montes and Carlos A. Coello Coello. A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 9(1):1–17, February 2005.
39. Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, third edition, 1996.
40. D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
41. Tatsuya Okabe, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. On Test Functions for Evolutionary Multi-objective Optimization. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 792–802, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.
42. K.E. Parsopoulos, D.K. Taoulis, N.G. Pavlidis, V.P. Plagianakos, and M.N. Vrahatis. Vector Evaluated Differential Evolution for Multiobjective Optimization. In *2004 Congress on Evolutionary Computation (CEC'2004)*, volume 1, pages 204–211, Portland, Oregon, USA, June 2004. IEEE Service Center.
43. Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11(1):341–356, Summer 1982.
44. Edgar Alfredo Portilla Flores. *Integración Simultánea de Aspectos Estructurales y Dinámicos para el Diseño Óptimo de un Sistema de Transmisión de Variación Continua*. PhD thesis, Departamento de Ingeniería Eléctrica, Sección de Mecatrónica, CINVESTAV-IPN, México, D.F., México, June 2006. (In Spanish).
45. Kenneth V. Price. An Introduction to Differential Evolution. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pages 79–108. McGraw-Hill, London, UK, 1999.
46. S. Ranji Ranjithan, S. Kishan Chetan, and Harish K. Dakshima. Constraint Method-Based Evolutionary Algorithm (CMEA) for Multiobjective Optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 299–313. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
47. Margarita Reyes-Sierra and Carlos A. Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

48. Tea Robič and Bodgan Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 520–533, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.
49. Günter Rudolph. Some Theoretical Properties of Evolutionary Algorithms under Partially Ordered Fitness Values. In Cs. Fabian and I. Intorsureanu, editors, *Proceedings of the Evolutionary Algorithms Workshop (EAW-2001)*, pages 9–22, Bucharest, Romania, January 2001.
50. Luis Vicente Santana-Quintero and Carlos A. Coello Coello. An Algorithm Based on Differential Evolution for Multi-Objective Problems. *International Journal of Computational Intelligence Research*, 1(2):151–169, 2005.
51. Ruhul Sarker, H. Abbass, and C. Newton. Solving Two Multi-objective Optimization Problems using Evolutionary Algorithm. In M. Mohammadian, R. Sarker, and X. Yao, editors, *Computational Intelligence in Control*. Idea Group Publishing, USA, 2002.
52. J. David Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, New Jersey, 1985. Lawrence Erlbaum.
53. Kishan Chetan Srigiriraju. Noninferior Surface Tracing Evolutionary Algorithm (NSTEA) for Multi Objective Optimization. Master's thesis, North Carolina State University, Raleigh, North Carolina, August 2000.
54. N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, fall 1994.
55. Feng Xue. *Multi-Objective Differential Evolution: Theory and Applications*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, September 2004.
56. Feng Xue, Arthur C. Sanderson, and Robert J. Graves. Pareto-based Multi-Objective Differential Evolution. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 862–869, Canberra, Australia, December 2003. IEEE Press.
57. Feng Xue, Arthur C. Sanderson, and Robert J. Graves. Modeling and convergence analysis of a continuous multi-objective differential evolution algorithm. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 228–235, Edinburgh, Scotland, September 2005. IEEE Service Center.
58. Feng Xue, Arthur C. Sanderson, and Robert J. Graves. Multi-objective differential evolution - algorithm, convergence analysis, and applications. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 743–750, Edinburgh, Scotland, September 2005. IEEE Service Center.
59. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
60. Eckart Zitzler, Jürgen Teich, and Shuvra S. Bhattacharyya. Evolutionary Algorithm Based Exploration of Software Schedules for Digital Signal Processors. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vassant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, volume 2, pages 1762–1769, San Francisco, California, July 1999. Morgan Kaufmann.

61. Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

Index