In: Book Title
Editor: Editor Name, pp. 1-26

ISBN 0000000000

Chapter 1

AN INTRODUCTION TO MULTIOBJECTIVE OPTIMIZATION TECHNIQUES

***Antonio López Jaimes,*[*] *Saúl Zapotecas Martínez,*[†] *Carlos A. Coello Coello*[‡]
CINVESTAV-IPN
Departamento de Computación
Evolutionary Computation Group (EVOCINV)
Av. IPN No. 2508
Col. San Pedro Zacatenco
México, D.F. 07360, MEXICO

Key Words: multiobjective optimization, evolutionary algorithms

[*] E-mail address: alopez@computacion.cs.cinvestav.mx

[†] E-mail address: zapoteca@computacion.cs.cinvestav.mx

[‡] E-mail address: ccoello@cs.cinvestav.mx

1. Introduction

A wide variety of problems in engineering, industry, and many other fields, involve the simultaneous optimization of several objectives. In many cases, the objectives are defined in incomparable units, and they present some degree of conflict among them (i.e., one objective cannot be improved without deterioration of at least another objective). These problems are called *Multiobjective Optimization Problems* (MOPs). Let us consider, for example, a shipping company which is interested in minimizing the total duration of its routes to improve customer service. On the other hand, the company also wants to minimize the number of trucks used in order to reduce operating costs. Clearly, these objectives are in conflict since adding more trucks reduces the duration of the routes, but increases operation costs. In addition, the objectives of this problem are expressed in different measurement units.

In single-objective optimization, it is possible to determine between any given pair of solutions if one is better than the other. As a result, we usually obtain a single optimal solution. However, in multiobjective optimization there does not exist a straightforward method to determine if a solution is better than other. The method most commonly adopted in multiobjective optimization to compare solutions is the one called *Pareto dominance relation* [1] which, instead of a single optimal solution, leads to a set of alternatives with different trade-offs among the objectives. These solutions are called *Pareto optimal solutions* or *non-dominated solutions*.

Although there are multiple Pareto optimal solutions, in practice, only one solution has to be selected for implementation. For instance, in the example of the shipping company presented above, only one route from several alternatives generated will be selected to deliver the packages for a given day. Therefore, in the multiobjective optimization process we can distinghish two tasks, namely: i) find a set of Pareto optimal solutions, and ii) choose the most preferred solution out of this set. Since Pareto optimal solutions are mathematically equivalent, the latter task requires a Decision Maker (DM) who can provide subjective preference information to choose the best solution in a particular instance of the multiobjective optimization problem.

We can distinguish two main approaches to solve multiobjective optimization problems. The first is called the Multi-Criteria Decision Making (MCDM) approach which can be characterized by the use of mathematical programming techniques and a decision making method in an intertwined manner. In most of the MCDM's methods the decision maker plays a major role in providing information to build a preference model which is exploited by the mathematical programming method to find solutions that better fit the DM's preferences [2]. Evolutionary Multiobjective Optimization (EMO) is another approach useful to solve multiobjective optimization problems. Since evolutionary algorithms use a population based approach, they usually find an approximation of the whole Pareto front in one run. Although in the EMO community the decision making task has not received too much attention in the past, in recent years a considerable number of works have addressed the incorporation of preference in Multi-Objective Evolutionary Algorithms (MOEAs).

In the following, we present some general concepts and notations used in the remainder of this chapter.

Definition 1 (Multiobjective Optimization Problem)**.** *Formally, a Multiobjective Optimization Problem (MOP) is defined as:*

$$\begin{array}{rl} \textit{“Minimize”} & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})]^T \\ \textit{subject to} & \mathbf{x} \in \mathcal{X}. \end{array} \tag{1}$$

The vector $\mathbf{x} \in \mathbb{R}^n$ is formed by n *decision variables* representing the quantities for which values are to be chosen in the optimization problem. The *feasible set* $\mathcal{X} \subseteq \mathbb{R}^n$ is implicitly determined by a set of equality and inequality constraints. The vector function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^k$ is composed by k scalar *objective functions* $f_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \ldots, k; k \geq 2)$. In multiobjective optimization, the sets $\mathbb{R}^n$ and $\mathbb{R}^k$ are known as *decision variable space* and *objective function space*, respectively. The image of $\mathcal{X}$ under the function $\mathbf{f}$ is a subset of the objective function space denoted by $\mathcal{Z} = \mathbf{f}(\mathcal{X})$ and referred to as the *feasible set in the objective function space*.

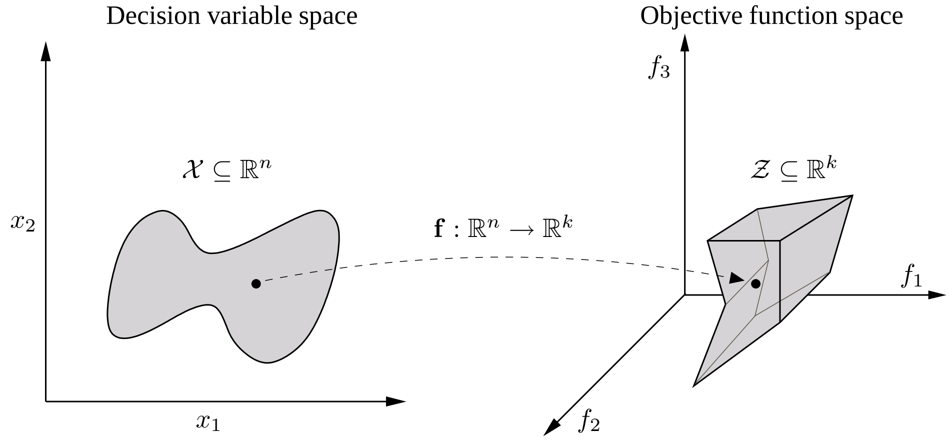
Decision variable space
Objective function space
$\mathcal{X} \subseteq \mathbb{R}^n$
x_2
x_1
$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^k$
f_3
$\mathcal{Z} \subseteq \mathbb{R}^k$
f_1
f_2

Figure 1. Search spaces in multiobjective optimization problems.

2. Notions of Optimality in MOPs

In order to define precisely the multiobjective optimization problem stated in definition (1) we have to establish the meaning of minimization in $\mathbb{R}^k$. That is to say, it is required to define how vectors $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$ have to be compared for different solutions $\mathbf{x} \in \mathbb{R}^n$. In single-objective optimization is used the relation “less than or equal” ($\leq$) to compare the values of the scalar objective functions. By using this relation there may be different optimal solutions $\mathbf{x} \in \mathcal{X}$, but only one optimal value $f_{\min} = \min\{f_i(\mathbf{x}) | \mathbf{x} \in \mathcal{X}\}$, for each function f_i, since the relation $\leq$ induces a total order in $\mathbb{R}$ (i.e., every pair of solutions is comparable, and thus, we can sort solutions from the best to the worst one). In contrast, in multiobjective optimization problems, there is no canonical order on $\mathbb{R}^k$, and thus, we need weaker definitions of order to compare vectors in $\mathbb{R}^k$.

In multiobjective optimization, it is usually adopted the *Pareto dominance relation* originally proposed by Francis Ysidro Edgeworth in 1881 [3], but generalized by the french-italian economist Vilfredo Pareto in 1896 [1].

Definition 2 (Pareto Dominance relation)**.** *We say that a vector* $\mathbf{z}^1$ *Pareto-dominates vector* $\mathbf{z}^2$*, denoted by* $\mathbf{z}^1 \prec_{\text{pareto}} \mathbf{z}^2$*, if and only if*[1]*:*

$$\forall i \in \{1, \ldots, k\} : z_i^1 \leq z_i^2 \tag{2}$$

and

$$\exists i \in \{1, \ldots, k\} : z_i^1 < z_i^2 \tag{3}$$

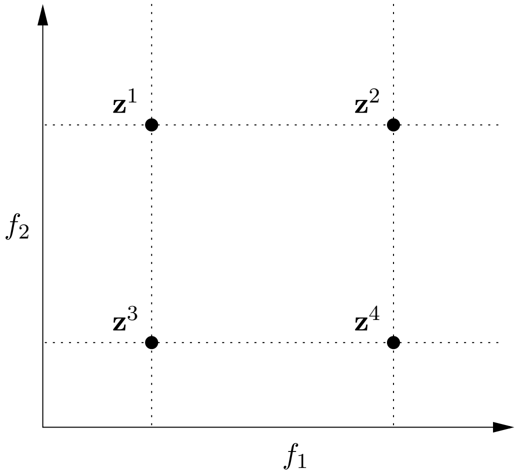

Figure 2. Illustration of the concept of Pareto dominance relation.

Figure 2 illustrates the Pareto dominance relation with an example with four 2-dimensional vectors. Vector $\mathbf{z}^3$ is strictly less than $\mathbf{z}^2$ in both objectives, therefore $\mathbf{z}^3 \prec_{\text{pareto}} \mathbf{z}^2$. Vector $\mathbf{z}^3$ also Pareto-dominates $\mathbf{z}^1$ since with respect to f_1 those vectors are equal, but in f_2, $\mathbf{z}^3$ is strictly less than $\mathbf{z}^1$. Since $\prec_{\text{pareto}}$ is not a total order some elements can be incomparable like is the case with $\mathbf{z}^1$ and $\mathbf{z}^4$, i.e., $\mathbf{z}^1 \not\prec_{\text{pareto}} \mathbf{z}^4$ and $\mathbf{z}^4 \not\prec_{\text{pareto}} \mathbf{z}^1$. Similarly, $\mathbf{z}^3 \prec_{\text{pareto}} \mathbf{z}^4$, $\mathbf{z}^1 \prec_{\text{pareto}} \mathbf{z}^2$, and $\mathbf{z}^4 \prec_{\text{pareto}} \mathbf{z}^2$.

Thus, to solve a MOP we have to find those solutions $\mathbf{x} \in \mathcal{X}$ whose images, $\mathbf{z} = \mathbf{f}(\mathbf{x})$, are not Pareto-dominated by any other vector in the feasible space. In the example shown in Figure 2, no vector dominates $\mathbf{z}^3$, and, therefore, we say that $\mathbf{z}^3$ is *nondominated*.

Definition 3 (Pareto Optimality)**.** *A solution* $\mathbf{x}^* \in \mathcal{X}$ *is Pareto optimal if there does not exist another solution* $\mathbf{x} \in \mathcal{X}$ *such that* $\mathbf{f}(\mathbf{x}) \prec_{pareto} \mathbf{f}(\mathbf{x}^*)$*.*

Definition 4 (Weak Pareto Optimality)**.** *A solution* $\mathbf{x}^* \in \mathcal{X}$ *is weakly Pareto optimal if there does not exist another solution* $\mathbf{x} \in \mathcal{X}$ *such that* $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}^*)$ *for all* $i = 1, \ldots, k$*.*

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

The set of Pareto optimal solutions and its image in objective space is defined in the following.

Definition 5 (Pareto optimal set)**.** *The Pareto optimal set, $\mathcal{P}^*$, is defined as:*

$$\mathcal{P}^* = \{\mathbf{x} \in \mathcal{X} \mid \nexists\, \mathbf{y} \in \mathcal{X} : \mathbf{f}(\mathbf{y}) \preceq \mathbf{f}(\mathbf{x})\}. \tag{4}$$

Definition 6 (Pareto front)**.** *For a Pareto optimal set $\mathcal{P}^*$, the Pareto front, $\mathcal{PF}^*$, is defined as:*

$$\mathcal{FP}^* = \{\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \mid \mathbf{x} \in \mathcal{P}^*\}. \tag{5}$$

Figure 3 illustrates the concept of Pareto optimal set and its image in the objective space, the Pareto front. Darker points denote Pareto optimal vectors. In variable space, these vectors are referred to as Pareto optimal decision vectors, while in objective space, are called Pareto optimal objective vectors. As we can see in the figure, the Pareto front is only composed by nondominated vectors.

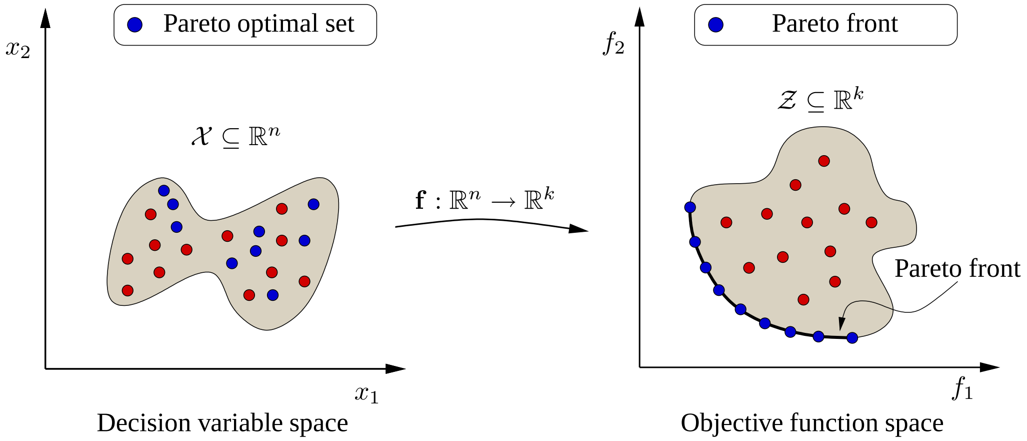
● Pareto optimal set
x_2
$\mathcal{X} \subseteq \mathbb{R}^n$
x_1
Decision variable space
$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^k$
● Pareto front
f_2
$\mathcal{Z} \subseteq \mathbb{R}^k$
Pareto front
f_1
Objective function space

Figure 3. Illustration of the Pareto optimal set and its image, the Pareto front.

On some optimization techniques is useful to know the lower and upper bounds of the Pareto front. The ideal point represents the lower bounds and is defined by $z_i^\star = \min_{z \in \mathcal{Z}} z_i$ for all $i = 1, \ldots, k$. In turn, the upper bounds are defined by the nadir point, which is given by $z_i^{\text{nad}} = \max_{z \in \mathcal{Z}} z_i$ for all $i = 1, \ldots, k$.

As we mentioned before, Pareto dominance is the most common preference relation used in multiobjective optimization. However, it is only one of the possible preference relations available. The interested reader is referred to [4] (Chap. 6) and [5] (Chap. 5), where other preference relations are presented.

As indicated by some authors (see e.g., [5] and [6]), in general, a MOP can be defined completely by $(\mathcal{X}, \mathbb{R}^k, \mathbf{f}, \mathcal{R})$, where $\mathcal{X}$ is the feasible set, $\mathbb{R}^k$ is the objective function space, $\mathbf{f}$ is the objective function vector, and $\mathcal{R}$ is the preference relation, which induces an ordered set on $\mathbb{R}^k$.

3. Mathematical Programming Techniques

The mathematical programming techniques are classified regarding how and when to incorporate preferences from the DM into the search process. A very important issue is the moment at which the DM is required to provide preference information. There are three ways of doing this [4, 7]:

1. Prior to the search (a priori approaches).
2. During the search (interactive approaches).
3. After the search (a posteriori approaches).

In this section we present some of the most popular MCDM's techniques accordign to the above classification.

3.1. A Priori Preference Articulation

3.1.1. Goal Programming

Charnes and Cooper [8] are credited with the devolopment of the goal programming method for a linear model, and played a key role in applying it to industrial problems. In this method, the DM has to assign targets or goals that wishes to archieve for each objective. These values are incorporated into the problem as additional constraints. The objective function then tries to minimize the absolute deviations from the targets to the objectives. The simplest form of this method may be formulated as follows:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{k} |f_i(\mathbf{x}) - T_i| \\ \text{subject to} & \mathbf{x} \in \mathcal{X}, \end{array} \tag{6}$$

where T_i denotes the target or goal set by the decision maker for the ith objective function $f_i(\mathbf{x})$, and $\mathcal{X}$ represents the feasible region. The criterion, then is to minimize the sum of the absolute values of the differences between target values and actually achieved values. A more general formulation of the goal programming objective function is a weighted sum of the pth power of the deviation $|f_i(\mathbf{x}) - T_i|$. Such a formulation has been called *generalized goal programming* [9].

In equation 6, the objective function is nonlinear and the simplex method can be applied only after transforming this equation into a linear form, thus reducing goal programming to a special type of linear programming. In this transformation, new variables δ_i^+ and δ_i^- are defined such that:

$$\delta_i^+ = \frac{1}{2}\{|f_i(\mathbf{x}) - T_i| + [f_i(\mathbf{x} - T_i)]\} \tag{7}$$

$$\delta_i^- = \frac{1}{2}\{|f_i(\mathbf{x}) - T_i| - [f_i(\mathbf{x} - T_i)]\} \tag{8}$$

This means that the absolute value signs can be dropped from problem (6) by introducing the underachievement and the overachievement variables. Adding and subtracting the equations (7) and (8), the resulting equivalent linear formulation may be found:

$$\begin{array}{rl} \text{minimize} & \displaystyle\sum_{i=1}^{k}(\delta_i^+ + \delta_i^-) \\ \text{subject to} & f(\mathbf{x}) - \delta_i^+ + \delta_i^- = T_i, \quad i = 1\ldots,k \\ & \delta_i^+, \delta_i^- \geq 0, \quad i = 1\ldots,k, \\ & \mathbf{x} \in \mathcal{X}, \end{array} \tag{9}$$

Since it is not possible to have both under and overachievements of the goal simultaneously, then at least one of the deviational variables must be zero. In other words:

$$\delta_i^+ \cdot \delta_i^- = 0 \tag{10}$$

Fortunately, this constraint is automatically fulfilled by the simplex method beacuse the objective function drives either δ_i^+ or δ_i^- or both variables simultaneously to zero for all i. Some times it may be desirable to express preference for over or under achievement of a goal. Thus, it may be more desirable to overachieve a targeted reability figure than to underachieve it. To express preference for deviations, the DM can assign relative weights w_i^+ and w_i^- to positive and negative deviations, respectively, for each target T_i. If a minimization problem is considered, choosing the w_i^+ to be larger than w_i^- would be expressing preference for underachievement of a goal. In addition, goal programming provides the flexibility to deal with cases that have conflicting multiple goals. Essentially, the goals may be ranked in order of importance to the problem solver. That is, a priority factor, p_i ($i = 1, \ldots, k$) is assigned to the deviational variables associated with the goals. This is called "*lexicographic ordering*". These factors p_i are conceptually different from weights, as it is explained, for example, in [7]. The resulting optimization model becomes

$$\begin{array}{rl} \text{minimize} & \displaystyle\sum_{i=1}^{k} p_i(w_i^+\delta_i^+ + w_i^-\delta_i^-) \\ \text{subject to} & f(\mathbf{x}) - \delta_i^+ + \delta_i^- = T_i, \quad i = 1\ldots,k \\ & \delta_i^+, \delta_i^- \geq 0, \quad i = 1\ldots,k, \\ & \mathbf{x} \in \mathcal{X}, \end{array} \tag{11}$$

Note that this technique yields a nondominated solutions if the goal point is chosen in the feasible domain. The following theorem is presented and proved in [7]:

Theorem 1. *The solution of a weighted or a lexicographic goal programming problem (9) is Pareto optimal if either the aspiration levels form a Pareto optimal reference point or all the deviational variables δ_i^+ for functions to be minimized and δ_i^- for functions to be maximized have positive values at the optimum.*

3.1.2. Goal-Attainment Method

Similar to the goal programming method, in this approach the decision maker must provide a goal vector $\mathbf{z}^{\text{ref}}$. In addition, the decision maker must provide a vector of weights $\mathbf{w} =$

$[w_1, w_2, \ldots, w_k]$ relating the relative under- or over-attainment of the desired goals. In order to find the best-compromise solution $\mathbf{x}^\star$, the following problem is solved [10, 11]:

$$\begin{aligned} \text{Minimize} \quad & \alpha \\ \text{subject to} \quad & z_i^{\text{ref}} + \alpha \cdot w_i \geq f_i(\mathbf{x}); \quad i = 1, \ldots, k, \\ & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{12}$$

where α is a scalar variable unrestricted in sign and the weights $w_1, w_2, \ldots, w_k$ are normalized so that

$$\sum_{i=1}^{k} |w_i| = 1 \tag{13}$$

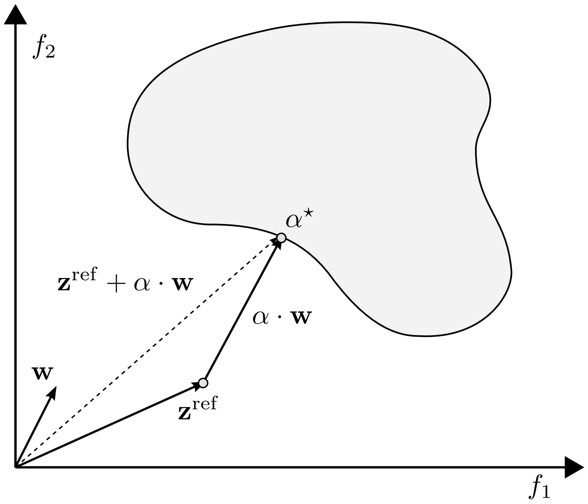
f_2
$\alpha^\star$
$\mathbf{z}^{\text{ref}} + \alpha \cdot \mathbf{w}$
$\alpha \cdot \mathbf{w}$
$\mathbf{w}$
$\mathbf{z}^{\text{ref}}$
f_1

Figure 4. Illustration of the goal attainment method with two objective functions.

If some $w_i = 0$ $(i = 1, \ldots, k)$, it means that the maximum limit of objectives $f_i(\mathbf{x})$ is z_i^{ref}. It can be easily shown [12] that every Pareto optimal solution can be generated by varying the weights, with $w_i \geq 0$ $(i = 1, \ldots, k)$ even for nonconvex problems. The mechanism by which this method operates is illustrated in Figure 4. The vector $\mathbf{z}^{\text{ref}}$ is represented by the decision goal of the DM, who also decides the direction of $\mathbf{w}$. Given vectors $\mathbf{w}$ and $\mathbf{z}^{\text{ref}}$, the direction of the vector $\mathbf{z}^{\text{ref}} + \alpha \cdot \mathbf{w}$ can be determined, and the problem stated by equation (12) is equivalent to finding a feasible point on this vector in objective space which is closest to the origin. It is obvious that the optimal solution of equation (12) is the first point at which $\mathbf{z}^{\text{ref}} + \alpha \cdot \mathbf{w}$ intersects the feasible region in the objective space (denoted by $\mathcal{Z}$ in Figure 4). If this point of intersection exist, then it would clearly be a Pareto optimal solution. It should be pointed out that the optimum value of α informs the DM of whether the goals are attainable or not. A negative value of α implies that the goal of the decision maker is attainable and an improved solution is then to be obtained. Otherwise, if $\alpha > 0$, then the DM's goal is unattainable.

3.1.3. Lexicographic Method

In this method, the objectives are ranked in order of importance by the decision maker (from best to worst). The optimal value $f_i^\star$ ($i = 1, \ldots, k$) is then obtained by minimizing the objective functions sequentially, starting with the most important one and proceeding according to the order of importance of the objectives. Additionally, the optimal value found of each objective is added as a constraint for subsequent optimizations. This way, it is preserved the optimal value of the most important objectives. Only in the case of several optimal solutions in the single optimization of the current objective, the rest of the objectives are considered. Therefore, in the worst case, we have to carry out k single objective optimizations.

Let the subscripts of the objectives indicate not only the objective function number, but also the priority of the objective. Thus, $f_1(\mathbf{x})$ and $f_k(\mathbf{x})$ denote the most and least important objective functions, respectively. Then, the first problem is formulated as

$$\begin{array}{ll} \text{Minimize} & f_1(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X}. \end{array} \tag{14}$$

We have to note that, although only one optimal value $f_1^\star = \min\{f_1(\mathbf{x}) | \mathbf{x} \in \mathcal{X}\}$ is generated for this single-objective problem, it might be possible to obtain many different optimal solutions $\mathbf{x}^\star \in \mathcal{X}$. Nonetheless, regarding the original multiobjective problem, only one of these solutions is Pareto optimal. For this reason, we should consider two situations after the optimization of each objective f_i ($i = 1, \ldots, k$). If we obtain a unique optimal solution, then this solution is the optimal solution of the original multiobjective problem, and, therefore, we stop the optimization process. Otherwise, we have to optimize the next objective. In general, we have to solve the single objective optimization problem ($i = 2, \ldots, k$) given by

$$\begin{array}{ll} \text{Minimize} & f_i(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X}, \\ & f_l(\mathbf{x}) = f_l^\star, \quad l = 1, \ldots, i-1. \end{array} \tag{15}$$

If several optimal solutions were obtained in each optimization problem (15) until objective f_{k-1}, then the unique optimal solution obtained for f_k, i.e., $\mathbf{x}_k^\star$, is taken as the desired solution of the original problem.

In [7, 13] it is proved that the optimal solution obtained by the lexicographic problem is Pareto optimal. For this reason, the lexicographic method is usually adopted as an additional optimization approach in methods that can only guarantee weak optimality by themselves. For example, in the ε-constraint method (see Section 3.2.3.), or in methods based on the Tchebycheff achievement function (see Sections 3.3.2. and 3.3.3.).

3.2. A Posteriori Preference Articulation

3.2.1. Linear Combination of Weights

In this method, the general idea is to associate each objective function with a weighting coefficient and minimize the weighted sum of the objectives. In this way, the multiobjective problem is transformed into a single objective problem. Thus, the new optimization problem is defined as:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{k} w_i f_i(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{16}$$

where $w_i \geq 0$ and is strictly positive for at least one objective, such that $\sum_{i=1}^{k} w_i = 1$. The set of nondominated solutions can be generated by parametrically varying the weights w_i in the objective function. This was initially demonstrated by Gass and Saaty for a two-objective problem. The following implications are consequences of this formulation and their corresponding proofs can be found in [7]:

Theorem 2. *The solution of weighting problem (16) is weakly Pareto optimal.*

Theorem 3. *The solution of weighting problem (16) is Pareto optimal if the weighting coefficients are positive, that is $w_i > 0$ for all $i = 1, \ldots, k$.*

Theorem 4. *The unique solution of the weighting problem (16) is Pareto optimal.*

Theorem 5. *Let the multiobjective optimization problem be convex. If $\mathbf{x}^* \in \mathcal{X}$ is Pareto optimal, then there exists a weighting vector $\mathbf{w}$ $(w_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} w_i = 1)$ which is a solution to the weighting problem (16).*

3.2.2. Normal Boundary Intersection

Das and Dennis [14] proposed this novel method for generating Pareto optimal solutions evenly distributed. The main idea in the Normal Boundary Intersection (NBI) method, is to intersect the feasible objective region with a normal to the convex combinations of the columns of the *pay-off* matrix. For undestanding this method let's see the next definition.

Definition 7. *Let x_i^* be the respective global minimizers of $f_i(\mathbf{x})$, $i = 1, \ldots, k$ over $x \in \mathcal{X}$. Let $F_i^* = F(x_i^*)$, $i = 1, \ldots, k$. Let Φ be the $k \times k$ matrix whose i^{th} column is $F_i^* - F^*$ sometimes known as the pay-off matrix. Then the set of points in R^k that are convex combinations of $F_i^* - F^*$, i.e. $\{\Phi\beta : \beta \in R^n, \sum_{i=1}^{k} \beta_i = 1, \beta_i \geq 0\}$, is referred to as the Convex Hull of Individual Minima (CHIM).*

The set of the attainable objective vectors, $\{F(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ is denoted by $\mathcal{F}$, thus $\mathcal{X}$ is mapped onto $\mathcal{F}$ by F. The space R^k which contains $\mathcal{F}$ is referred to as the *objective space*. The boundary of $\mathcal{F}$ is denoted by $\partial\mathcal{F}$.

Figure 5 illustrates the $CHIM$ for a two-objective problem. In the example we show the shadow minimum or *utopian point* F^* defined by $F^* = \{f_1^*, \ldots, f_k^*\}$, the area in gray describes the objective space $\mathcal{F}$ and the black line describes the complete boundary $\partial\mathcal{F}$ of $\mathcal{F}$.

NBI is a method designed to find the portion of $\partial\mathcal{F}$ which contains the Pareto optimal points. The main idea behind this approach is that the intersection point between the boundary $\partial\mathcal{F}$ and the normal pointing towards the origin emanating from any point in the CHIM is a point on the portion of $\partial\mathcal{F}$ containing the efficient points. This point is guaranteed to be a Pareto optimal point if the trade-off surface in the objective space is convex. Therefore, the original multiobjective problem is traslated into the following new problem.

Given a convex weighting β, $\Phi\beta$ represents a point in the CHIM. Let $\hat{n}$ denote the unit normal to the CHIM simplex towards the origin; then $\Phi\beta + t\hat{n}$ represents the set of points on that normal. The point of intersection of the normal and the boundary of $\mathcal{F}$ closest to the origin is the global solution of the following problem:

$$\begin{aligned} \text{Maximize}_{x,t} \quad & t \\ \text{subject to} \quad & \Phi\beta + t\hat{n} = F(\mathbf{x}), \\ & \mathbf{x} \in \mathcal{X} \end{aligned} \tag{17}$$

The vector constraint $\Phi\beta + t\hat{n} = F(\mathbf{x})$ ensures that the point $\mathbf{x}$ is actually mapped by F to a point on the normal, while the remaining constraints ensure feasibility of $\mathbf{x}$ in $\mathcal{X}$. This approach considers that the shadow minimum F^* is in the origin. Otherwise, the first set of constraints should be $\Phi\beta + t\hat{n} = F(\mathbf{x}) - F^*$.

As many scalarization methods, for various β, a number of points on the boundary of $\mathcal{F}$ are obtained thus, effectively, constructing the Pareto surface.

A quasi-normal direction is used instead of a normal direction, such that it represents an equally weighted linar combination of columns of Φ, multiplied by -1 to ensure that it points towards the origin. That is,

$$\hat{n} = -\Phi\mathbf{v}$$

where $\mathbf{v}$ is a fixed vector with strictly positive components. Commonly, $\hat{n}$ is chosen to be $\hat{n} = -\Phi e$, where e is the column vector of all ones.

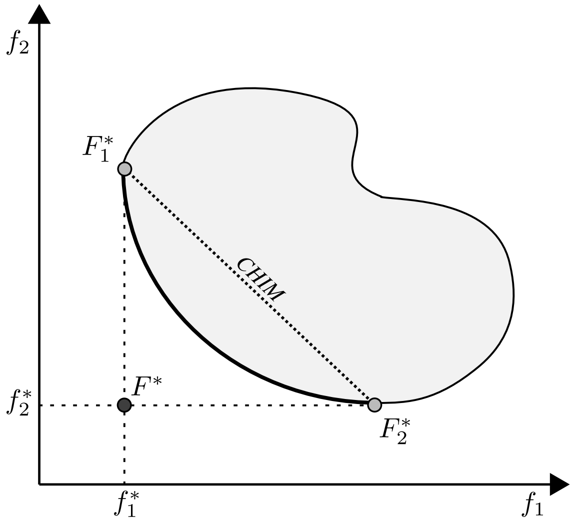

Figure 5. Illustration of the $CHIM$ for a two-objective problem.

3.2.3. ε-Constraint Method

The ε-constraint method is one of best known scalarization techniques to solve multiobjective problems. In this approach one of the objectives is minimized while the others are used as constraints bound by some allowable levels ε_i.

The multiobjective optimization problem is transformed into the following ε-constraint problem

$$\begin{aligned} \text{Minimize} \quad & f_l(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq \varepsilon_i \quad \forall i = 1, \ldots, k \quad i \neq l, \\ & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{18}$$

Figure 6 illustrates the application of the ε-constraint method in a bicriterion problem. In the example we show three different constraint values for f_1 and their respective optimum values for f_2. It is worth noting that for some values of ε_i, the constraint imposed might be active or inactive. For example, the constraint for ε_1 and ε_3 is active, whereas that for ε_2 is inactive.

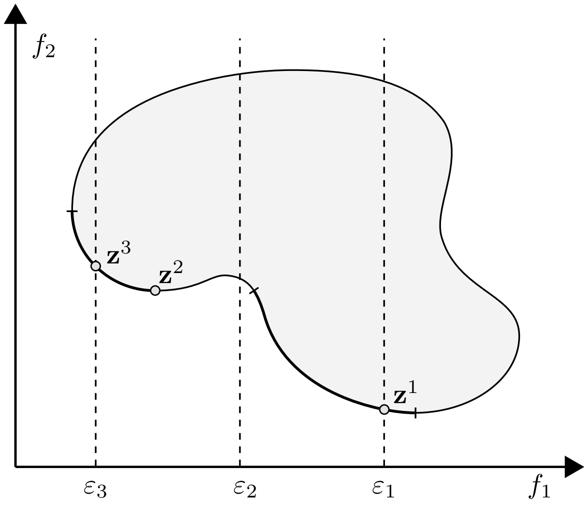

Figure 6. Illustration of the ε-constraint method.

In order to find several Pareto optimal solutions, we need to solve problem (18) using multiple different values for ε_i. In this iterative optimization process the user needs to provide the range of the reference objective, f_l. In addition, it must be provided the increment for the constraints imposed by ε. This increment determines the number of Pareto optimal solutions generated. In Algorithm 3.2.3. it is shown the pseudo code of the iterative ε-constraint optimization for the case of two objectives.

In [13] and [7] are presented an proved the following important theorems related to the optimality of the solutions generated by the ε-constraint problem.

Theorem 6. *The optimal solution of the ε-constraint problem (18) is weakly Pareto optimal.*

Algorithm 1 Pseudocode of an iterative optimization process using the ε-constraint method.

```
Input:
  $f_1^{\min}, f_1^{\max} \in \mathbb{R}$: Lower and upper bounds for objective $f_1$.
  $\delta \in \mathbb{R}$: Increment for constraint $\varepsilon$.

  $PF_{\text{approx}} \leftarrow \emptyset$
  $\varepsilon \leftarrow f_1^{\max}$
  while $\varepsilon \geq f_1^{\min}$ do
      $\mathbf{x} \leftarrow \varepsilon$-MINIMIZE$(\mathbf{f}, \varepsilon)$    ▷ Minimize using problem (18)
      $PF_{\text{approx}} \leftarrow PF_{\text{approx}} \cup \{\mathbf{x}\}$
      $\varepsilon \leftarrow \varepsilon - \delta$
  end while
  Return the approximation of the Pareto front $PF_{\text{approx}}$
```

Theorem 7. *The solution $\mathbf{x}^\star \in \mathcal{X}$ is Pareto optimal if and only if $\varepsilon_i = f_i(\mathbf{x}^\star)$ for all $i = 1, \ldots, k$, $i \neq l$, and $\mathbf{x}^\star$ is an optimal optimal solution of problem (18) for all $l = 1, \ldots, k$.*

Theorem 8. *If $\mathbf{x}^\star$ is the unique optimal solution of problem (18) for some $l = 1, \ldots, k$, then $\mathbf{x}^\star$ is Pareto optimal.*

As pointed out by Ehrgott [13], Theorem 7 only provides a method to check Pareto optimality instead of a method to find Pareto optimal solutions since the values for ε must be equal to the nondominated vector $\mathbf{f}(\mathbf{x}^\star)$.

Therefore, in order to generate Pareto optimal solutions only, we need to solve k single objective problems, or less than k if we obtain a unique optimal solution in one of the problems. One possibility to avoid weakly Pareto optimal solutions is the use of lexicographic optimization (see Section 3.1.3.) in problem (18). That is, if f_1 has multiple optimal solutions, then select the best solution with respect to objective f_2 an so on.

3.2.4. Method of Weighted Metrics

The idea behind this method is to find the closest feasible solution to a reference point, which usually is the ideal point. Some authors, such as Duckstein [15] and Zeleny [16], call this method compromise programming. The most common metrics to measure the distance between the reference point and the feasible region are those derived from the L_p-metric, which is defined by

$$||y||_p = \left(\sum_{i=1}^{p} |y_k|^p \right)^{1/p}, \tag{19}$$

for $1 \leq p \leq \infty$. The value of p indicates the type of metric. For $p = 1$ we obtain the Manhattan metric, while for $p = \infty$ we obtain the so-called Tchebycheff metric. From the L_p-metrics is derived the following compromise problem

$$\begin{aligned} \text{Minimize} \quad & \left(\sum_{i=1}^{p} |f_i(\mathbf{x}) - z_i^\star|^p \right)^{1/p} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{20}$$

In order to obtain different (weakly) Pareto optimal solutions we must allow weights in problem (20). The resulting weighted compromise programming problem is

$$\begin{aligned} \text{Minimize} \quad & \left(\sum_{i=1}^{p} w_i |f_i(\mathbf{x}) - z_i^\star|^p \right)^{1/p} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{21}$$

For $p = 1$, all deviations from $z_i^\star$ are taken into account. Ehrgott [5] shows that the method of linear combination of weights (see Section 3.2.1.) is a special case of the weighted compromise problem with $p = 1$. For $p = \infty$, i.e, using the Tchebycheff metric, the largest deviation is the only one taken into consideration. The resulting weighted Tchebycheff problem is defined by

$$\begin{aligned} \text{Minimize} \quad & \max_{i=1,\ldots,k} \{ w_i |f_i(\mathbf{x}) - z_i^\star| \} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{22}$$

This problem presents the most interesting theoretical result, and is one of the most commonly employed. Depending on the properties of the metric employed, we obtain different results regarding the optimality of the solutions generated.

In [7] and [13] is shown that the solution of the weighted compromise programming problem (21) with $1 \leq p < \infty$ is Pareto optimal if one the following conditions holds:

1. The optimal solution of (21) is unique.
2. $w_i > 0$ for all $i = 1, \ldots, k$.

It is important to note, however, that for $1 \leq p < \infty$, although problem (21) can generate Pareto optimal solutions, it does not necessarily find all of them. In constrast, the weighted Tchebycheff problem is able to generate every Pareto optimal solution [7, 5]. Unfortunately, if the solution of the Tchebycheff problem is not unique, some of the solutions generated are weakly Pareto optimal. In order to identify the Pareto optimal solutions, Miettinen [7] suggests two possible approaches: use lexicographic ordering to solve the Tchebycheff problem, or modify the original problem. In the latter approach, Steuer and Choo [17] suggest aggregating an augmentation term to the original problem. Thus, it is obtained the augmented weighted Tchebycheff problem

$$\begin{aligned} \text{Minimize} \quad & \max_{i=1,\ldots,k} \{ w_i |f_i(\mathbf{x}) - z_i^\star| \} + \rho \sum_{i=1}^{k} |f_i(\mathbf{x}) - z_i^\star| \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{23}$$

where ρ is a sufficiently small positive scalar. However, it is worth noting that using this approach it may be possible that some Pareto optimal solutions cannot be found. Nevertheless, every properly Pareto optimal solution can be obtained by this approach. The set of properly Pareto optimal solutions is a subset of the Pareto optimal solutions in which unbounded tradeoffs are not allowed.

3.3. Interactive Preference Articulation

3.3.1. Method of Geoffrion-Dyer-Feinberg (GDF)

This interactive method developed by Geoffrion et al. [18] is based on the maximization of a value function (utility function) using a gradient-based method. The value function is only implicitly known, but is assumed to be differentiable and concave. The gradient-based method employed is the Frank-Wolfe method [19], however, as indicated by the authors, other methods could be used in an interactive fashion. The Frank-Wolfe method assumes that the feasible set, $\mathcal{X} \subseteq \mathbb{R}^n$, is compact and convex. The direction-finding problem of the Frank-Wolfe method is the following:

$$\begin{array}{rl} \text{Maximize} & \nabla_x U(\mathbf{f}(\mathbf{x}^h)) \cdot \mathbf{y} \\ \text{subject to} & \mathbf{y} \in \mathcal{X}, \end{array} \tag{24}$$

where $U : \mathbb{R}^k \to \mathbb{R}$ is the value function, $\mathbf{x}^h$ is the current point, and $\mathbf{y}$ is the new variable of the problem. Using the chain rule it is obtained

$$\nabla_x U(\mathbf{f}(\mathbf{x}^h)) = \sum_{i=1}^{k} \left(\frac{\partial U}{\partial f_i} \right) \nabla_x f_i(\mathbf{x}^h). \tag{25}$$

Dividing this equation by $\frac{\partial U}{\partial f_1}$ we obtain the following reformulation of the Frank-Wolfe problem

$$\begin{array}{rl} \text{Maximize} & \left(\sum_{i=1}^{k} -m_i^h \nabla_x f_i(\mathbf{x}^h) \right) \cdot \mathbf{y} \\ \text{subject to} & \mathbf{y} \in \mathcal{X}, \end{array} \tag{26}$$

where $m_i^h = (\partial U / \partial f_i)/(\partial U / \partial f_1)$ for all $i = 1, \ldots, k$, $i \neq 1$ are the marginal rates of substitution (or indifference tradeoff) at $\mathbf{x}^h$ between objectives f_1 and f_i. The marginal rate of substitution is the amount of loss on objective f_i that the decision maker is willing to tolerate in exchage of one unit of gain in objective f_1, while the value of the other objectives remain unchanged.

The prodecedure of the GDF method is the following:

Step 0: Provide an initial point $\mathbf{x}^1 \in \mathcal{X}$. Set $h = 1$.

Step 1: The decision maker must provide marginal rates of substitution between f_1 (the reference objective) and the other objectives at the current point $\mathbf{x}^h$.

Step 2: Find the optimal solution $\mathbf{y}^h$ of problem (26). Set the new search direction $\mathbf{d}^h = \mathbf{y}^h - \mathbf{x}^h$. If $\mathbf{d}^h = 0$, go to **Step 5**.

Step 3: The decision maker must determine the best step-size, t^h, to compute the new solution $\mathbf{x}^h$. Then, set $\mathbf{x}^{h+1} = \mathbf{x}^h + t^h \mathbf{d}^h$.

Step 4: If $\mathbf{x}^{h+1} = \mathbf{x}^h$, go to **Step 5**, else set $h = h + 1$ and go to **Step 1**.

Step 5: Return $\mathbf{x}^h$ as the final solution.

The most important steps of this procedure are the steps 1 and 3. One possibility to estimate the marginal rates is to compare the solutions

$$[f_1(\mathbf{x}^h), f_2(\mathbf{x}^h), \ldots, f_j(\mathbf{x}^h), \ldots, f_k(\mathbf{x}^h)],$$

and

$$[f_1(\mathbf{x}^h) - \Delta_1, f_2(\mathbf{x}^h), \ldots, f_j(\mathbf{x}^h) + \Delta_j, \ldots, f_k(\mathbf{x}^h)],$$

where Δ_j is an small amount added to f_j in compensation of a decrement in f_1 by a small amount Δ_1, while the other values remain unaltered. The idea is to modified the quantities Δ_j and Δ_1 until the two solutions are indifferent to the decision maker. Thus, $m_i^h \approx \frac{\Delta_1}{\Delta_j}$. Regarding the selection of the optimal step-size, Geoffrion et al. proposed a graphical procedure that presents to the decision maker several alternative vectors varying t in the interval $[0, 1]$. That is, the vectors $z_i = f_i(\mathbf{x}^h + t\mathbf{d})$ for $i = 1, \ldots, k$ using different values of $t \in [0, 1]$.

3.3.2. Tchebycheff Method

The Tchebycheff method proposed in [17], is an iterative method which was designed to be user-friendly, thus, complicated information is not requiered. This method is based on the minimization of a function value, assuming that the global ideal objective vector (*utopian vector*) is known. The metric to be used for measuring the distances to a utopian objective vector is the weighted Tchebycheff metric. Thus, the multiobjective optimization problem is transformed into a single-objective optimization problem, defined by

$$\begin{array}{ll} \text{Minimize} & \max\limits_{i=1,\ldots,k} [w_i(f_i(\mathbf{x}) - z_i^*)] \\ \text{subject to} & \mathbf{x} \in \mathcal{X}, \end{array} \tag{27}$$

where $\mathbf{w} \in W = \{\mathbf{w} \in \mathbb{R}^k | 0 < w_i < 1, \sum_{i=1}^k w_i = 1\}$ and $\mathbf{z}^*$ is the utopian objective vector.

Theorem 9. *Let $\mathbf{x}^* \in \mathcal{X}$ be Pareto optimal. Then there exists a weighting vector $\mathbf{0} < \mathbf{w} \in \mathbb{R}^k$ such that $\mathbf{x}^*$ is a solution of the weighted Tchebycheff problem (27), where the reference point is the utopian objective vector $\mathbf{z}^*$.*

Thus, from the above theorem, every Pareto optimal solution of any multiobjective optimization problem can be found by solving problem (27). However, with this approach, some of the solutions may be weakly Pareto optimal solutions. For solving this negative aspect, the Tchebycheff method can be stated formulating the distance minimization problem as a *lexicographic weighted Tchbycheff* approach, as follows:

$$\begin{array}{ll} \text{Minimize} & \max\limits_{i=1,\ldots,k} [w_i(f_i(\mathbf{x}) - z_i^*)], \sum\limits_{i=1}^{k} (f_i(\mathbf{x}) - z_i^*) \\ \text{subject to} & \mathbf{x} \in \mathcal{X}, \end{array} \tag{28}$$

The following theorems are consequences of the connection between the lexicographic weighted Tchebycheff problem and the Pareto optimal solutions.

Theorem 10. *The solution of lexicographic weighted Tchebycheff problem (28) is Pareto optimal.*

Theorem 11. *Let $\mathbf{x} \in \mathcal{X}$ be Pareto optimal. Then there exists a weighting vector $\mathbf{0} < \mathbf{w} \in \mathbb{R}^k$ such that $\mathbf{x}$ is a unique solution of lexicographic weighted Tchebycheff problem (28).*

At each iteration, the Tchebycheff method provides different subsets of nondominated solutions. These solutions consist of $P(\approx n)$ representative points, generated by using an augmented weighted Tchebycheff problem (for example, the lexicographic weighted Tchebycheff problem (28)), from which the DM is required to select one as his most preferred. Below, we describe the complete Tchebycheff method.

Step 0: Calculate the ideal point z^* and let $z^{**} = z^* + \epsilon$, where ϵ is a vector of arbitrarily small positive values. Let $W^1 = \{\mathbf{w} \in \mathbb{R}^k : w_i \in [0,1], \sum_{i=1}^{k} w_i = 1\}$ be the initial set of weighting vectors. Set $h = 1$.

Step 1: Generate a large number ($50n$) of weighting vectors from W^h.

Step 2: Find the optimal solutions of problem (28). Filter the $2P$ resulting nondominated points to obtain P solutions.

Step 3: Show the P compromise solutions to the DM and ask him to select the one he most prefers. Let z^h be the selected point.

Step 4: i. If $h = t$ then **Stop** with z^h as the preferred solution (where t is a prespecified number of iterations); else

ii. Let $\mathbf{w}^h$ be the weighting vector which generated z^h in step 2. Its components are given by:

$$w_i^h = \frac{1}{z_i^{**} - z_i^h} \left[\sum_{j=1}^{k} \frac{1}{z_j^{**} - z_j^h} \right] \qquad (i = 1, \ldots, k)$$

Determine the reduced set of weighting vectors:

$$W^{h+1} = \{\mathbf{w} \in \mathbb{R}^k : w_i \in [l_i, u_i], \sum_{i=1}^{k} w_i = 1\}$$

where

$$[l_i, u_i] = \begin{cases} [0, r^h] & \text{if } w_i^h \leq r^h/2, \\ [1 - r^h, 1] & \text{if } w_i^h \geq 1 - r^h/2, \\ [w_i^h - r^h/2, w_i^h + r^h/2] & \text{otherwise,} \end{cases}$$

and r^h is a prespecified “convergence factor” r raised to the hth power.
Set $h = h + 1$ and go to **Step 1**.

3.3.3. Reference Point Methods

The proposed preference relation is based on the reference point method proposed by Wierzbicki [20, 21], and thus, this section presents a summary of this method.

The reference point approach is an interactive multiobjective optimization technique based on the definition of a scalarization achievement function. The basic idea of this technique is the following. First, the DM is asked to give a reference point. This point represents the aspiration levels for each objective. Then, the solutions that better satisfy the aspiration levels are computed using an achievement scalarization function, which is a type of utility function based on a reference point. If the DM is satisfied with the current solution, the interactive process ends. Otherwise, the DM must provide another reference point.

Definition 8 (Achievement scalarizing function)**.** *An achievement scalarizing function is a parameterized function $s_{\mathbf{z}^{ref}}(\mathbf{z}) : \mathbb{R}^k \rightarrow \mathbb{R}$, where $\mathbf{z}^{ref} \in \mathbb{R}^k$ is a reference point representing the decision maker's aspiration levels. Thus, the multiobjective problem is transformed into the following scalar problem:*

$$\begin{array}{rl} \textit{Minimize} & s_{\mathbf{z}^{ref}}(\mathbf{z}) \\ \textit{subject to} & \mathbf{z} \in \mathcal{Z}. \end{array} \tag{29}$$

Most of the achievement scalarization functions are based on the Tchebycheff metric (L_∞ metric). Based on the Tchebycheff distance we can define an appropriate achievement scalarizing function.

Definition 9 (Augmented Tchebycheff scalarizing function)**.** *The augmented weighted Tchebycheff scalarizing function is defined by*

$$s_\infty(\mathbf{z}, \mathbf{z}^{\mathrm{ref}}) = \max_{i=1,\ldots,k} \{\lambda_i(z_i - z_i^{ref})\} + \rho \sum_{i=1}^{k} \lambda_i(z_i - z_i^{ref}), \tag{30}$$

where $\mathbf{z}^{\mathrm{ref}}$ is a reference point, $\rho > 0$ is an augmentation coefficient sufficiently small, and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_k]$ is a vector of weights such that $\forall i \; \lambda_i \geq 0$ and, for at least one i, $\lambda_i > 0$.

The (weighted) Tchebycheff scalarizing function poses some convenient properties over other scalarizing functions. As proved in [7] and [13], by using the augmented version of this function we can find any Pareto optimal solution.

In most of the reference points methods the exploration of the objective space is made by moving the reference point at each iteration. In contrast, the weights are kept unaltered during the interactive optimization process. That is, weights do not define preferences, but they are mainly used for normalizing each objective function. Usually, the weights are set for all $i = 1, \ldots, k$ as

$$\lambda_i = \frac{1}{z_i^{\mathrm{nad}} - z_i^\star}$$

It is important to mention that the DM can provide both feasible and infeasible reference points. On the one hand, if the reference point is infeasible, then the minimum of (30) is the closest feasible point to the aspiration levels. On the other hand, if $\mathbf{z}^{\mathrm{ref}}$ is feasible, the solution generated by (30) improves the aspiration levels.

3.3.4. Light Beam Search

The Light Beam Search (LBS) method proposed by Jaszkiewicz and Slowinski [22], is an iterative method which combines the reference point idea and tools of Multi-Attribute Decision Analysis (MADA). At each iteration, a finite sample of nondominated points is generated. The sample is composed of a current point called *middle point*, which is obtained in previous iteration, and J nondominated points from its neighborhood. A local preference model in the form of an *outranking relation* S is used to define the neighborhood of the middle point. It is said that a outranks b (aSb), if a is considered to be at least as good as b. The outranking relations is defined by DM, which specify three preference thresholds for each objective. They are *indifference threshold*, *preference threshold* and *veto threshold*. The DM has the possibility to scan the inner area of the neighborhood along the objective function trajectories between any two characteristic neighbors or between a characteristic neighbor and the middle point. Below, the general scheme of the LBS procedure is shown.

Step 0: Ask the DM to specify the starting aspiration and reservation points.

Step 1: Compute the starting middle point.

Step 2: Ask the DM to specify the local preferential information to be used to build an outranking relation.

Step 3: Present the middle point to the DM.

Step 4: Calculate the charateristic neighbors of the middle point and present them to the DM.

Step 5: If DM is satisfied then **Stop**, else

i. ask the DM to choose one of the neighboring points to be the new middle point, or

ii. update the preferential information, or

iii. define a new aspiration point and/or a reservation point.

Go to **Step 4**.

4. Evolutionary Algorithms

Currently, there is a large variety of traditional mathematical programming methods (see for example [7, 5]) to solve MOPs. However, some researchers [23, 24, 25, 26] have identified several limitations of traditional mathematical programming approaches to solve MOPs. Some of them are the following:

1. We need to run many times those algorithms to find several elements of the Pareto optimal set.

2. Many of them require domain knowledge about the problem to be solved.

3. Some of those algorithms are sensitive to the shape or continuity of the Pareto front.

These complexities call for alternative approaches to deal with certain types of MOPs. Among these alternative approaches, we can find Evolutionary Algorithms (EAs), which are stochastic search and optimization methods that simulate the natural evolution process. At the end of 1960s, Rosenberg [27] proposed the use of genetic algorithms to solve MOPs. However, it was until 1984, when David Schaffer [28] introduced the first actual implementation of what it is now called a Multi-Objective Evolutionary Algorithm (MOEA). From that moment on, many researchers [29, 30, 31, 32, 33, 34] have proposed a wide variety of MOEAs.

As other stochastic search strategies (e.g., simulated annealing, ant colony optimization, or particle swarm optimization), MOEAs do not guarantee to find the true Pareto optimal set but, instead, aim to generate a good approximation of such set in a reasonable computational time. On the other hand, MOEAs are particularly well-suited to solve MOPs because they operate over a set of potential solutions (i.e., the population). This feature allows them to generate several elements of the Pareto optimal set (or a good approximation of them) in a single run. Furthermore, MOEAs are less susceptible to the shape or continuity of the Pareto front than traditional mathematical programming techniques, require little domain information and are relatively easy to implement and use.

Single objective EAs and MOEAs share a similar structure. The major difference is the fitness assignment mechanism since a MOEA deals with fitness vectors of dimension k ($k \geq 2$). As pointed out by different authors [31, 4], finding an approximation to the Pareto front is by itself a bi-objective problem whose objectives are:

- minimize the distance of the generated vectors to the true Pareto front, and

- maximize the diversity of the achieved Pareto front approximation.

Therefore, the fitness assignment scheme must consider these two objectives. Algorithm 2 describes the basic structure of a MOEA.

Algorithm 2 Pseudocode of a MOEA.

```
$t \leftarrow 0$
Generate an initial population $P(t)$
while the stopping criterion is not fulfilled do
    Evaluate the objective vector f for each individual in $P(t)$
    Assign a fitness for each individual in $P(t)$
    Select from $P(t)$ a group of parents $P'(t)$ preferring the fitter ones
    Recombine individuals of $P'(t)$ to obtain a child population $P''(t)$
    Mutate individuals in $P''(t)$
    Combine $P(t)$ and $P''(t)$ and select the best individuals to get $P(t+1)$
    $t \leftarrow t + 1$
end while
```

Usually, the initial population is generated in a random manner. However, if we have some knowledge about the characteristics of a good solution, it is wise to use this information to create the initial population. The fitness assignment scheme requires a ranking of the individuals according to a preference relation and then, assigning a scalar fitness value to each individual using such rank. The selection for reproduction (line 6) is carried out as in the single objective case, for instance, using tournament selection. In contrast, the selection for survival (line 9), intended to maintain the best solutions so far (i.e., elitism), uses a preference relation to remove some solutions and maintain the population size constant. To ensure diversity of the approximation set, the selection mechanism is also based on a density estimator of the objective function space.

4.1. MOGA

Carlos M. Fonseca and Peter J. Fleming [33] proposed the Multi-Objective Genetic Algorithm (MOGA), which was one of the first in using Pareto dominance to rank individuals. In MOGA, the rank of a certain individual corresponds to the number of individuals in the current population by which it is dominated. That is, the rank of invidual $\mathbf{x}^i$ at generation t is given by rank$(\mathbf{x}^i, t) = 1 + p_i$, where p_i is the number of individuals that dominate $\mathbf{x}^i$ in the current generation. Note that all nondominated individuals in the population receive rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface. N refers to the population size, g is the specific generation, $f_j(\mathbf{x}^k)$ is the j-th objective function, $\mathbf{x}^k$ is the k-th individual, P the population.

Fitness assignment is performed in the following way:

1. Sort population according to rank.

2. Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $n \leq N$) in the way proposed by David E. Goldberg [35] according to some function, usually linear, but not necessarily.

3. Average the fitnesses of individuals with the same rank, so that all of them will be sampled at the same rate. This procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used.

As Goldberg and Deb [36] indicate, this type of blocked fitness assignment is likely to produce a large selection pressure that might produce premature convergence. In order to avoid this, MOGA adopts a fitness sharing scheme [37] that “penalizes” solutions lying too close from others in some space (e.g., objective function space).

4.2. NSGA and NSGA-II

The Nondominated Sorting Genetic Algorithm (NSGA) was proposed by Srinivas and Deb [30] and is another variation of Goldberg’s approach [35]. The NSGA is based on several layers of classifications of the individuals. Before selection is performed, the population is ranked on the basis of nondomination: all nondominated individual are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diveristy of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population are classified. Stochastic remainder proportionate selection is adopted for this technique. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows for a better search of the different nondominated regions and results in convergence of the population toward such regions. Sharing, by its part, helps to distribute the population over this region (i.e. the Pareto front of the problem). As a result, one might think that this MOEA converges rather quickly; however, a computational bottleneck occurs with the fitness sharing mechanism.

An improved version of the NSGA algorithm, called NSGA-II was proposed by Deb et al. [38, 39]. As shown in Figure 7, the NSGA-II builds a population of competing individuals, ranks and sorts each individual according to its nondomination level, it applies Evolutionary Operators (EVOPs) to create a new offspring pool, and then combines the parents and offspring before partitioning the new combined pool into fronts. The NSGA-II then computes a crowding distance for each member and it uses this value in the selecion process in order to spread the solutions along the Pareto front. This is the most popular MOEA used today, and it is frequently adopted to compare the performance of newly introduced MOEAs.

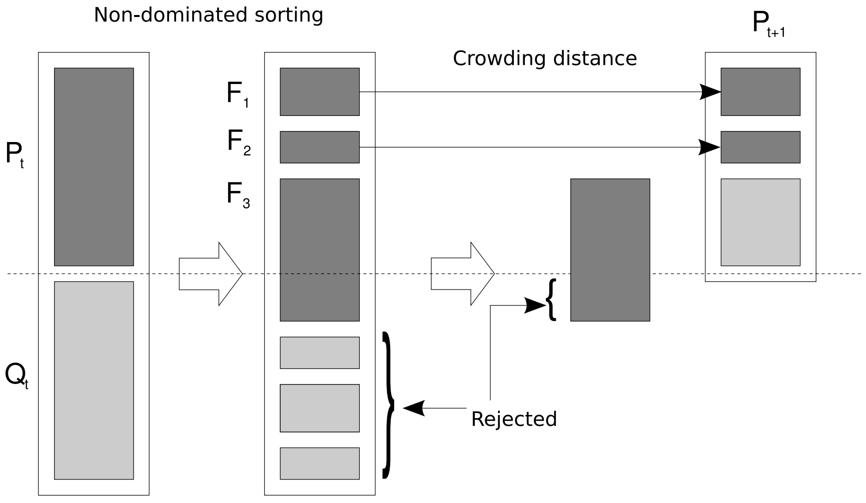

Figure 7. Flow diagram that shows the way in which he NSGA-II works. P_t is the parents population and Q_t is the offspring population at generation t. F_1 are the best solutions from the combined populations (parents and offspring). F_2 are the second best solutions and so on.

4.3. SPEA and SPEA2

The Strength Pareto Evolutionary Algorithm (SPEA) was introduced by Eckart Zitzler and Lothar Thiele [31]. This approach integrates some successful mechanisms from other

MOEAs, namely, a secondary population (external archive) and the use of Pareto dominance ranking. SPEA uses an external archive containing nondominated solutions previously found. At each generation, nondominated individuals are copied to the external nondominated set.

In SPEA, the fitness of each individual in the primary population is computed using the individuals of the external archive. First, for each individual in this external set, a strength value is computed. The strength, s_i, of individual i is determined by $s_i = \frac{n}{N+1}$, where n is the number of solutions dominated by i, and N is the size of the archive. This strength is similar to the ranking value of MOGA, since it is proportional to the number of solutions to which a certain individual dominates. Finally, the fitness of each individual in the primary population is equal to the sum of the strengths of all the external members that dominate it. This fitness assignment considers both closeness to the true Pareto front and even distribution of solutions at the same time. Thus, instead of using niches based on distance, Pareto dominance is used to ensure that the solutions are properly distributed along the Pareto front.

Since the size of the archive may grow too large, the authors employed a technique that prunes the contents of the external nondominated set so that its size remains below a certain threshold.

There is also a revised version of SPEA (called SPEA2) [40]. SPEA2 has three main differences with respect to its predecessor: (1) it incorporates a fine-grained fitness assignment strategy which takes into account for each individual the number of individuals that dominate it and the number of individuals to which it dominates; (2) it uses a nearest neighbor density estimation technique which guides the search more efficiently, and (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions.

4.4. PAES

The Pareto Archived Evolution Strategy (PAES) was designed and implemented by Joshua D. Knowles and David W. Corne [41]. PAES consists of a $(1 + 1)$ evolution strategy (i.e., a single parent that generates a single offspring) in combination with a historical archive that records some of the nondominated solutions previously found. This archive is used as a reference set against which each mutated individual is being compared. PAES also uses a novel approach to keep diversity, which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its “coordinates” or “geographical location”). A map of such grid is maintained, indicating the number of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space). Furthermore, the procedure has a lower computational complexity than traditional niching methods [41]. The adaptive grid of PAES and some other issues related to external archives (also called “elite” archives) have been studied both from an empirical and from a theoretical perspective (see for example [42]). Other implementations of PAES were also proposed, namely $(1 + \lambda)$-ES and $(\mu + \lambda)$-ES. However, these were found not to improve overall performance.

4.5. PESA

The Pareto Envelope-based Selection Algorithm (PESA) is suggested by Corne et al. [43]. PESA consists of a small internal population and a larger external population. A hypergrid division of phenotype space is used to maintain selection diversity (using a crowding measure) as the MOEA runs. Furthermore, this crowding measure is used to allow solutions to be retained in an external archive similar to the one adopted by PAES [41]. A revised version of this MOEA is called PESA-II [44]. The difference between the PESA-I and II is that in the second, selection is region-based and the subject of selection is now a hyperbox, not just an individual (i.e., it first selects a hyperbox, and then it selects an individual within that hyperbox). The motivation behind this approach is to reduce the computational cost associated with Pareto ranking [44].

4.6. New Trends in MOEAs

Today, one of the trends regarding the design of MOEAs is the adoption of performance measures to select individuals (see for example [45]). Also, there is growing interest in dealing with problems having a large number of objectives (see for example [46]) and to deal with expensive objective functions (see for example [47]).

4.7. Incorporation of Preferences in MOEAs

Among the earliest attempts to incorporate preference in a MOEA, we can find Fonseca and Fleming's proposal [33]. This proposal consisted of extending the ranking mechanism of MOGA to accommodate goal information as an additional criterion. They used the goal attainment method, so that the DM could supply new goals at each generation of the MOEA, reducing in consequence the size of the solution set under inspection and learning.

Deb [48] proposed a technique to transform goal programming problems into multiobjective optimization problems which are then solved using a MOEA. In goal programming the DM has to assign targets or goals that wishes to achieve for each objective, and these values are incorporated into the problem as additional constraints. The objective function then attempts to minimize the absolute deviations from the targets to the objectives.

Yun et al. [49] proposed the use of Generalized Data Envelopment Analysis (GDEA) [50] with aspiration levels for choosing desirable solutions from the Pareto optimal set. This is an interactive approach in which a nonlinear aggregating function is optimized by a genetic algorithm in order to generate the Pareto optimal solutions of the multiobjective optimization problem. The decision maker must define aspiration levels for each objective, as well as the ideal values for each of them. Then, the aspiration levels are adopted as constraints during the optimization, so that the Pareto optimal solutions are filtered out and those closest to the aspiration levels are assigned the higher fitness values.

Branke et al. [51] proposed an approach called Guided MOEA also exploiting the concept of utility function. The idea is to express the DM's preferences in terms of maximal and minimal linear weighting functions, corresponding directly to slopes of a linear utility function. The authors determine the optimal solution from a population using both of the previously mentioned weighting functions. Those individuals are given rank one and are considered the borderline solutions (since they represent extreme cases of the DM's preferences). Then all the nondominated vectors are evaluated in terms of these two linear weighting functions. After that, all solutions that have a better fitness than either of the two borderline individuals are assigned the same rank (these are the individuals preferred by the DM). These solutions are removed from the population and a similar ranking scheme is applied to the remaining individuals. The authors used a biased version of fitness sharing, in which the maximum and minimum niche counts are incorporated into a formula assigning each individual a fitness at least as good as that of any other individual with inferior rank.

More recently, Deb and Sundar [52] incorporated a reference point approach into the NSGA-II [38]. They introduced a modification in the crowding distance operator in order to select from the last front the solutions that would take part of the new population. They used the Euclidean distance to sort and rank the population accordingly (the solution closest to the reference point receives the best rank). The proposed method was designed to take into account a set of reference points. The drawback of this scheme is that it does not guarantee weakly Pareto optimality, particularly in MOPs with disconnected Pareto fronts. A similar approach was also proposed by Deb and Kumar [53], in which the light beam search procedure was incorporated into the NSGA-II. Similar to the previous approach, they modified the crowding operator to incorporate DM's preferences. They used a weighted Tchebycheff achievement function to assign the crowding distance to each solution in each front. Thus, the solution with the least distance will have the best crowding rank. Like in the previous approach, this algorithm finds a subset of solutions around the optimum of the achievement function using the usual outranking relation. However, from the three parameters that specify the outranking relation, they only used the veto threshold.

4.8. New Trends in the Incorporation of Preferences in MOEAs

One interesting trend in this area is the integration of mechanisms to define preferences from the user into the selection process in a more natural way by allowing, for example, the use of set preference relations of any kind [54].

5. Conclusion

This chapter has presented several techniques to solve multiobjective optimization problems using both mathematical programming and evolutionary computation approaches.

The choice of the most appropriate approach to be used depends on the nature of the problem to be solved and on the available resources. Since mathematical programming techniques normally emphasize the use of interactive techniques, they are suitable for problems in which the decision maker has considerable knowledge of the problem in order to express his/her preferences accurately. In turn, evolutionary algorithms are not only useful to approximate the Pareto front, but also to gain knowledge about the problem, i.e., to understand the structure of the possible set of solutions, the degree of conflict and the trade-offs among the objectives. In other words, MOEAs are a good choice when little information is available about a certain MOP.

Another important topic that complements the solution of a MOP is the incorporation of user's preferences and, as such, this topic is briefly discussed in this chapter, in the context of their use combined with MOEAs.

The main aim of this chapter has been to provide a general overview of the multiobjective optimization field and to serve as a departing point for those interested in working in this research area.

References

[1] Pareto, V. Cours D'Economie Politique. F. Rouge, 1896.

[2] Miettinen, K. Introduction to Multiobjective Optimization: Noninteractive Approaches. In Branke, J.; Deb, K.; Miettinen, K.; Slowinski, R., eds., Multiobjective Optimization: Interactive and Evolutionary Approaches, 1–26. Springer-Verlag, Berlin, Heidelberg, 2008.

[3] Edgeworth, F. Y. Mathematical Physics. P. Keagan, 1881.

[4] Coello Coello, C. A.; Lamont, G. B.; Van Veldhuizen, D. A. Evolutionary Algorithms for Solving Multi-Objective Problems. Second edn. Springer, New York, 2007. ISBN 978-0-387-33254-3.

[5] Figueira, J.; Greco, S.; Ehrgott, M. Multiple Criteria Decision Analysis: State of the Art Surveys. Springer Verlag, Boston, Dordrecht, London, 2005.

[6] Zitzler, E.; Knowles, J.; Thiele, L. Quality Assessment of Pareto Set Approximations. In Branke, J.; Deb, K.; Miettinen, K.; Slowinski, R., eds., Multiobjective Optimization. Interactive and Evolutionary Approaches, 373–404. Springer. Lecture Notes in Computer Science Vol. 5252, Berlin, Germany, 2008.

[7] Miettinen, K. M. Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston, Massachusetts, USA, 1998.

[8] Charnes, A.; Cooper, W. W. Management Models and Industrial Applications of Linear Programming, vol. 1. John Wiley, New York, 1961.

[9] Ignizio, J. Goal programming and extensions. DC Heath Lexington, Massachusetts, USA, 1976.

[10] Gembicki, F. Vector optimization for control with performance and parameter sensitivity indices. Ph.D. thesis, Case Western Reserve University, 1974.

[11] Gembicki, F.; Haimes, Y. Approach to performance and sensitivity multiobjective optimization: The goal attainment method. IEEE Transactions on Automatic Control 1975. 20, 769–771.

[12] Chen, Y.; Liu, C. Multiobjective VAR planning using the goal-attainment method. IEE Proceedings-Generation, Transmission and Distribution 1994. 141, 227–232.

[13] Ehrgott, M. Multicriteria Optimization. second edn. Springer, Berlin, 2005.

[14] Das, I.; Dennis, J. E. Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. SIAM Journal on Optimization 1998. 8, 631–657.

[15] Duckstein, L. Multiobjective optimization in structural design- The model choice problem. New directions in optimum structural design(A 85-48701 24-39). Chichester, England and New York, Wiley-Interscience, 1984, 1984. 459–481.

[16] Zeleny, M. Compromise programming. In Cochrane, J.; Zeleny, M., eds., Multiple Criteria Decision Making, 262–301. University of South Carolina Press, Columbia, 1973.

[17] Steuer, R. E.; Choo, E.-U. An interactive weighted Tchebycheff procedure for multiple objective programming. Mathematical Programming 1983. 26, 326–344.

[18] Geoffrion, A.; Dyer, J.; Feinberg, A. An interactive approach for multicriteria optimization with an application to the operation of an accademic department. Management Science 1973. 19, 357–369.

[19] Frank, M.; Wolfe, P. An algorithm for quadratic programming. Naval Research Logistics Quarterly 1956. 3, 95–110.

[20] Wierzbicki, A. The use of reference objectives in multiobjective optimisation. In G., F.; T., G., eds., MCDM theory and Application, Proceedings. No. 177 in Lecture notes in economics and mathematical systems, Springer Verlag, Hagen, 1980 468–486.

[21] Wierzbicki, A. A methodological guide to multiobjective optimization. In Iracki, K.; Malanowski, K.; Walukiewicz, S., eds., Optimization Techniques, Part 1, vol. 22 of *Lecture Notes in Control and Information Sciences*. Springer, Berlin, 1980 99–123.

[22] Jaszkiewicz, A.; Slowinski, R. The light beam search approach -an overview of methodology and applications. European Journal of Operational Research 1999. 113, 300–314.

[23] Coello Coello, C. A. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems. An International Journal 1999. 1, 269–308.

[24] Deb, K. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In Miettinen, K.; Mäkelä, M. M.; Neittaanmäki, P.; Periaux, J., eds., Evolutionary Algorithms in Engineering and Computer Science, chap. 8, 135–161. John Wiley & Sons, Ltd, Chichester, Reino Unido, 1999.

[25] Fogel, L. J. Artificial Intelligence through Simulated Evolution. Forty Years of Evolutionary Programming. John Wiley & Sons, Inc., Nueva York, 1999.

[26] Michalewicz, Z.; Fogel, D. B. How to Solve It: Modern Heuristics. Springer, Berlin, 2000.

[27] Rosenberg, R. S. Simulation of genetic populations with biochemical properties. Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, EE. UU., 1967.

[28] Schaffer, J. D. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. Ph.D. thesis, Vanderbilt University, 1984.

[29] Coello Coello, C. A.; Toscano Pulido, G. Multiobjective Optimization using a Micro-Genetic Algorithm. In Spector, L.; Goodman, E. D.; Wu, A.; Langdon, W.; Voigt, H.-M.; Gen, M.; Sen, S.; Dorigo, M.; Pezeshk, S.; Garzon, M. H.; Burke, E., eds., Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001). Morgan Kaufmann Publishers, San Francisco, California, 2001 274–282.

[30] Srinivas, N.; Deb, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation 1994. 2, 221–248.

[31] Zitzler, E.; Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 1999. 3, 257–271.

[32] Horn, J.; Nafpliotis, N. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Tech. Rep. IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, EE. UU., 1993.

[33] Fonseca, C. M.; Fleming, P. J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Forrest, S., ed., Proceedings of the Fifth International Conference on Genetic Algorithms. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers, San Mateo, California, 1993 416–423.

[34] Knowles, J. D.; Corne, D. W. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In 1999 Congress on Evolutionary Computation. IEEE Service Center, Washington, D.C., 1999 98–105.

[35] Goldberg, D. E. Genetic Algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[36] Goldberg, D.; Deb, K. A comparative analysis of selection schemes used in genetic algorithms. Foundations of genetic algorithms 1991. 1, 69–93.

[37] Deb, K.; Goldberg, D. E. An Investigation of Niche and Species Formation in Genetic Function Optimization. In Schaffer, J. D., ed., Proceedings of the Third International Conference on Genetic Algorithms. George Mason University, Morgan Kaufmann Publishers, San Mateo, California, 1989 42–50.

[38] Deb, K.; Agrawal, S.; Pratab, A.; Meyarivan, T. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

[39] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 2002. 6, 182–197.

[40] Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Tech. Rep. 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[41] Knowles, J. D.; Corne, D. W. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation 2000. 8, 149–172.

[42] Knowles, J.; Corne, D. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. IEEE Transactions on Evolutionary Computation 2003. 7, 100–116.

[43] Corne, D. W.; Knowles, J. D.; Oates, M. J. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Schoenauer, M.; Deb, K.; Rudolph, G.; Yao, X.; Lutton, E.; Merelo, J. J.; Schwefel, H.-P., eds., Proceedings of the Parallel Problem Solving from Nature VI Conference. Springer. Lecture Notes in Computer Science No. 1917, Paris, France, 2000 839–848.

[44] Corne, D. W.; Jerram, N. R.; Knowles, J. D.; Oates, M. J. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In Spector, L.; Goodman, E. D.; Wu, A.; Langdon, W.; Voigt, H.-M.; Gen, M.; Sen, S.; Dorigo, M.; Pezeshk, S.; Garzon, M. H.; Burke, E., eds., Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001). Morgan Kaufmann Publishers, San Francisco, California, 2001 283–290.

[45] Beume, N.; Naujoks, B.; Emmerich, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 2007. 181, 1653–1669.

[46] Farina, M.; Amato, P. A fuzzy definition of "optimality" for many-criteria optimization problems. IEEE Transactions on Systems, Man, and Cybernetics Part A—Systems and Humans 2004. 34, 315–326.

[47] Voutchkov, I.; Keane, A. Multiobjective Optimization using Surrogates. In Parmee, I., ed., Adaptive Computing in Design and Manufacture 2006. Proceedings of the Seventh International Conference. The Institute for People-centred Computation, Bristol, UK, 2006 167–175.

[48] Deb, K. Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms. In 1999 Congress on Evolutionary Computation. IEEE Service Center, Washington, D.C., 1999 77–84.

[49] Yun, Y.; Nakayama, H.; Arakawa, M. Multiple criteria decision making with generalized DEA and an aspiration level method. European Journal of Operational Research 2004. 158, 697–706.

[50] Yun, Y.; Nakayama, H.; Tanino, T.; Arakawa, M. A Multi-Objective Optimization Method Combining Generalized Data Envelopment Analysis and Genetic Algorithms. In 1999 IEEE International Conference on Systems, Man, and Cybernetics, vol. 1. IEEE, 1999 671–676.

[51] Branke, J.; Kaußler, T.; Schmeck, H. Guiding Multi Objective Evolutionary Algorithms Towards Interesting Regions. Tech. Rep. 398, Institute für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe, Karlsruhe, Germany, 2000.

[52] Deb, K.; Sundar, J. Reference Point Based Multi-objective Optimization Using Evolutionary Algorithms. In GECCO ’06: Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM, New York, NY, USA, 2006 635–642.

[53] Deb, K.; Kumar, A. Light beam search based multi-objective optimization using evolutionary algorithms. In IEEE Congress on Evolutionary Computation. 2007 2125–2132.

[54] Zitzler, E.; Thiele, L.; Bader, J. SPAM: Set Preference Algorithm for Multiobjective Optimization. In Rudolph, G.; Jansen, T.; Lucas, S.; Poloni, C.; Beume, N., eds., Parallel Problem Solving from Nature–PPSN X, 847–858. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, 2008.