Multi-objective Optimization for Space Mission Design Problems

Alfredo Arias Montaño[1] Carlos A. Coello Coello[2]

Oliver Schütze[2]

[1] Instituto Politécnico Nacional, ESIME Ticoman

Aeronautical Engineering Department

Av. Ticoman No. 600, Col. San José Ticoman

México, D. F., 07340, MEXICO

e-mail: aarias@ipn.mx

[2] CINVESTAV-IPN

Computer Science Department

Av. IPN No. 2508, Col. San Pedro Zacatenco

México D.F. 07360, MEXICO

e-mail: {ccoello,schuetze}@cs.cinvestav.mx

Abstract

Multi-objective optimization is an active research field which is—among others—currently successfully applied in the design of interplanetary space missions. In this chapter, we give a short introduction to multi-objective optimization and give an overview of commonly used methods for the numerical treatment of such problems, addressing both deterministic and stochastic approaches. Further on, we briefly summarize recent works that deal with multi-objective space mission design.

1 Introduction

In a variety of applications in industry and finance the problem arises that several objective functions have to be optimized concurrently leading to a *multi-objective optimization problem* (MOP). For instance, in space mission design, which we address in this chapter, two typcal important objectives are the time of flight (TOF) of the spacecraft to reach its destiny, and the cost of the mission (e.g., measured by the change in velocity Δv which has a direct impact on the fuel consumption and hence also on the overall cost of the mission). Since these objectives are typically contradicting—the 'cheapest' trajectory is certainly not the fastest one and vice versa—it comes as no surprise that the solution set, the so-called Pareto set, does not consist of one single solution (as for 'classical' scalar optimization problems). Instead, it forms a $(k-1)$-dimensional object where k is the number of objectives involved in the MOP.

In this chapter, we introduce the concept of multi-objective optimization (MOO) and state some theoretical background. Further, we present the state-of-the-art of both deterministic and stochastic search methods to compute a finite size representation of the Pareto set, respectively its image, the Pareto front. Furthermore, we discuss scenarios where MOO has been considered for MOPs related to space mission design.

The remainder of this chapter is organized as follows: In Section 2, we give a brief introduction to multi-objective optimization. In Section 3, we state the most commonly used mathematical programming techniques for MOPs and in Section 4 we give an overview of evolutionary multi-objective optimization algorithms. In Section 5, we briefly summarize some research works for which MOO techniques have been used to solve space mission design problems. Finally, in Section 6, we discuss potential future research trends in this field.

2 Multi-objective Optimization

In the following we give a brief introduction to continuous multi-objective optimization. For a more thorough discussion the reader is referred e.g. to [1, 2].

A multi-objective optimization problem (MOP) can in mathematical terms be expressed as follows:

$$\min_{\mathbf{x}\in Q} F(\mathbf{x}), \tag{1}$$

where the map F consists of the objective functions $f_i : Q \to \mathbb{R}$ under consideration, i.e.,

$$F : Q \to \mathbb{R}^k, \qquad F(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})]^T.$$

The set $Q \subset \mathbb{R}^n$ is the domain of F, where n is the dimension of the parameter space (also called decision space). In general, Q can be expressed by constraint functions, i.e.,

$$Q = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, l, \text{ and } h_j(\mathbf{x}) = 0,\ j = 1, \ldots, m\},$$

where the g_i's are called inequality constraints and the h_i's equality constraints. If $Q = \mathbb{R}^n$, then the problem is called unconstrained. In many cases, Q is defined by box constraints, i.e., the domain forms the n-dimensional box

$$Q = B_{l,u} := \{\mathbf{x} \in \mathbb{R}^n \ :\ l_i \leq x_i \leq u_i,\ i = 1, \ldots, n\}, \tag{2}$$

where $\mathbf{l}, \mathbf{r} \in \mathbb{R}^n$ with $l_i \leq u_i, i = 1, \ldots, n$, are the lower and upper bounds, respectively. Next, we have to define optimal solutions of a given MOP. Note that we cannot proceed as for classical scalar optimization problems (i.e., problems of the kind (1) where $k = 1$): While for any two (objective) values $a, b \in \mathbb{R}$ it is either $a = b$ or one of the values is lower than the other one (and hence this value can be considered as 'better' according to the given optimization problem), this does not hold for vectors in $\mathbb{R}^k$. Instead, we

have to use another way to compare solutions. This is usually done using the concept of *dominance* [3].

Definition 1 (dominance relation)**.** (a) Let $\mathbf{v}, \mathbf{w} \in \mathbb{R}^k$. Then the vector $\mathbf{v}$ is *less than* $\mathbf{w}$ (in short: $\mathbf{v} <_p \mathbf{w}$), if $v_i < w_i$ for all $i \in \{1, \ldots, k\}$. The relation $\leq_p$ is defined analogously.

(b) A vector $\mathbf{y} \in Q$ is called *strictly dominated* (or simply *dominated*) by a vector $\mathbf{x} \in Q$ ($\mathbf{x} \prec \mathbf{y}$) with respect to (1) if

$$F(\mathbf{x}) \leq_p F(\mathbf{y}) \quad \text{and} \quad F(\mathbf{x}) \neq F(\mathbf{y}),$$

else $\mathbf{y}$ is called non-dominated by $\mathbf{x}$.

(c) A vector $\mathbf{y} \in Q$ is called *weakly dominated* by a vector $\mathbf{x} \in Q$ ($\mathbf{x} \preceq \mathbf{y}$) with respect to (1) if $F(\mathbf{x}) \leq_p F(\mathbf{y})$.

If a vector $\mathbf{x}$ dominates a vector $\mathbf{y}$, then $\mathbf{x}$ can be considered to be 'better' according to the given MOP. The definition of optimality (i.e., of the 'best' solution) of a given MOP is now straightforward.

Definition 2 (Pareto point, Pareto set, Pareto front)**.** (a) A point $\mathbf{x} \in Q$ is called *(Pareto) optimal* or a *Pareto point* of (1) if there is no $\mathbf{y} \in Q$ which dominates $\mathbf{x}$.

(b) A point $\mathbf{x} \in Q$ is called a *weak Pareto point* or *weakly opimal* if there exists no $\mathbf{y} \in Q$ such that $F(\mathbf{y}) <_p F(\mathbf{x})$.

(c) The set of all Pareto optimal solutions is called the *Pareto set*, i.e.,

$$\mathcal{P} := \{\mathbf{x} \in Q \ : \ \mathbf{x} \text{ is a Pareto point of } (1)\}. \tag{3}$$

(d) The image $F(\mathcal{P})$ of $\mathcal{P}$ is called the *Pareto front*.

To familiarize with the concept of dominance, we consider the following hypothetical setting for a bi-objective problem (i.e., $k = 2$): Assume we are given six candidate solutions $\mathbf{x}_1, \ldots, \mathbf{x}_6$ such that the images are given by $\mathbf{y}_1 = F(\mathbf{x}_1) = [1, 6]^T$, $\mathbf{y}_2 = [1, 4]^T$,

$\mathbf{y}_3 = [2,3]^T$, $\mathbf{y}_4 = [3,2]^T$, $\mathbf{y}_5 = [5,1]^T$, and $\mathbf{y}_6 = [3.5,3.5]^T$ (compare to Figure 1). $\mathbf{x}_6$ is dominated by $\mathbf{x}_3$ as well as by $\mathbf{x}_4$. Further, $\mathbf{x}_1$ is weakly dominated by $\mathbf{x}_2$, but not strictly dominated by $\mathbf{x}_2$ nor any other of the candidate solutions. The solutions $\mathbf{y}_2, \ldots, \mathbf{y}_5$ are mutually non-dominated. Hence, in this example there does not exist one single 'best' solution but rather a set of optimal solutions. We will see shortly that this is typically the case.

If the objectives of the given problem are differentiable one can state a necessary condition for optimality analog to the scalar objective case. Here, we state the theorem for unconstrained problems. For further variants the reader is referred e.g. to [4, 1].

Theorem 1 (Theorem of Kuhn and Tucker [4])**.** *Let $\mathbf{x}^*$ be a Pareto point of (1), then there exists a vector $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0, i = 1, \ldots, k$, and $\sum_{i=1}^k \alpha_i = 1$ such that*

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}^*) = 0. \tag{4}$$

A vector $\alpha \in \mathbb{R}^k$ is called a convex weight if $\alpha_i \geq 0$, $i = 1, \ldots, k$ and $\sum_{i=1}^k \alpha_i = 1$. Equation (4) says that the zero vector can be expressed as a convex combination of the gradients $\nabla f_i(\mathbf{x}^*)$ at each Pareto point $\mathbf{x}^*$. Note that for $k = 1$ the equation reads as $\nabla f_1(\mathbf{x}^*) = 0$ which is a well-known fact from scalar optimization (e.g., [5]). A point $\mathbf{x} \in Q$ satisfying Equation (4) is called a *Karush–Kuhn–Tucker point*[1] or short KKT point.

Theorem 1 can be used to give a qualitative description of the Pareto set. For this, define the following map:

$$\begin{aligned} \tilde{F} :& \mathbb{R}^{n+k} \to \mathbb{R}^{n+1} \\ \tilde{F}(\mathbf{x}, \alpha) =& \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix}. \end{aligned} \tag{5}$$

[1] Named after the works of Karush [6] and Kuhn & Tucker [4].

If $\mathbf{x}^*$ is a Pareto point there exists by the above theorem a vector $\alpha^* \in \mathbb{R}^k$ such that $\tilde{F}(\mathbf{x}^*, \alpha^*) = 0$. Hence, the Pareto set plus the according set of weight vectors are contained in the preimage $\tilde{F}^{-1}(\mathbf{0})$, and we expect by the Implicit Function Theorem (e.g., [7]) that this set forms a set of dimension $k-1$. This is indeed the case under certain assumptions on the MOP, see [8] for a thorough discussion.

Example 1. We consider the following unconstrained MOP [9, 10]:

$$f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$$
$$f_i(\mathbf{x}) = \sum_{j=1}^{n} (x_j - a_j^i)^2,$$

where $\mathbf{a}^i \in \mathbb{R}^n$, $i = 1, \ldots, k$. The minimizer of each objective is given by $\mathbf{a}^i$, and the set of KKT points is given by the $(k-1)$-simplex that contains all the $\mathbf{a}^i$'s as vertices, i.e.,

$$S := S(\mathbf{a}^1, \ldots, \mathbf{a}^k) = \left\{ \sum_{i=1}^{k} \mu_i \mathbf{a}^i \ : \ \mu_1, \ldots, \mu_k \geq 0 \text{ and } \sum_{i=1}^{k} \mu_i = 1 \right\}.$$

One can see this for instance by mutual inclusion: Let $\mathbf{x} \in S$, i.e., there exists a convex weight $\mu \in \mathbb{R}^k$ such that $\sum_{i=i}^{k} \mu_i \alpha^i = \mathbf{x}$. We obtain

$$\sum_{i=1}^{k} \mu_i \nabla f_i(\mathbf{x}) = \sum_{i=1}^{k} 2(\mathbf{x} - \mathbf{a}^i) = 2 \underbrace{\sum_{i=1}^{k} \mu_i \mathbf{x}}_{=\mathbf{x}} - 2 \underbrace{\sum_{i=1}^{k} \mu_i \mathbf{a}^i}_{=\mathbf{x}} = 0$$

Hence, $\mathbf{x}$ is a KKT point with weight vector μ.

Let on the other hand $\mathbf{x}$ be a KKT point, i.e., there exists by Theorem 1 a convex weight $\alpha \in \mathbb{R}^k$ such that

$$0 = \sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}) = 2 \underbrace{\sum_{i=1}^{k} \alpha_i \mathbf{x}}_{=\mathbf{x}} - 2 \sum_{i=1}^{k} \alpha_i \mathbf{a}^i,$$

and we obtain $\mathbf{x} = \sum_{i=1}^{k} \alpha_i \mathbf{a}^i \in S$. Hence, for each KKT point the associated weight vector is simply the weight vector within the simplex. The preimage $\tilde{F}^{-1}(\mathbf{0})$ is thus given by

$$\tilde{F}^{-1}(\mathbf{0}) = \left\{ (\mathbf{x}, \alpha) \ : \ \mathbf{x} = \sum_{i=1}^{k} \alpha_i \mathbf{a}^i, \ \alpha_i \geq 0, i = 1, \ldots, k, \ \sum_{i=1}^{k} \alpha_i = 1 \right\} \tag{6}$$

Since further the Pareto set is equal to the set of KKT points in case the problem is convex (e.g., [2]) we have that the Pareto set of MOP (1) is given by S.

For the special case $n = 1$, $k = 2$, $a^1 = 0$, and $a^2 = 2$ one obtains

$$\begin{aligned} F : \mathbb{R} &\to \mathbb{R}^2 \\ F(x) = \quad & [x^2, (x-2)^2]^T, \end{aligned} \tag{7}$$

which is also known as Schaffer's problem [11]. Figure 2 shows the objective functions and the Pareto front of the problem. The Pareto set is given by the interval $\mathcal{P} = [0, 2]$ and the Pareto front is a curve connecting the points $[0, 4]^T$ and $[4, 0]^T$.

---

3 Mathematical Programming Techniques for MOPs

In this section we present the most common mathematical programming techniques for the numerical treatment of MOPs. Since we assume that the models related to space mission design are non-linear we restrict ourselves to methods for such models.

3.1 Scalarization Methods

One way to to attack MOPs is to use scalarization methods, i.e., to transform the original problem (1) into a scalar optimization problem (SOP) of the form

$$\min_{\mathbf{x}\in Q} f_\alpha(\mathbf{x}), \tag{8}$$

where $f_\alpha : Q \to \mathbb{R}$ and $\alpha \in \mathbb{R}^k$ is an external parameter. Note that for a given value of α the solution of (8) is typically a single point rather than a $(k-1)$-manifold. Finite size Pareto set/front approximations can hence only be obtained by choosing a clever sequence of optimization problems of the form (8) which in turn calls for a suitable set $A := \{\alpha^{(1)}, \ldots, \alpha^{(m)}\} \subset \mathbb{R}^k$ of external parameters. For problems of that kind we refer e.g. to [12, 1, 13, 14, 15, 16].

The general advantage of the use of scalarization methods is that they can be tackled with any solver for SOPs (i.e., ones that 'fit' to be problem at hand, for instance, methods that do not exploit gradient information). On the other hand, it is not always ensured that the resulting set of minimizers forms a suitable approximation of the Pareto set/front (e.g., in terms of the spread along the set of interest).

Weighted Sum Method The Weighted Sum Method is probably the oldest scalarization method [17]. The underlying idea is to assign to each objective a certain weight $\alpha_i \geq 0$, and to minimize the resulting weighted sum. Given problem (1), the Weighted Sum Problem can be stated as follows:

$$\begin{aligned} \min \;\; & f_\alpha(\mathbf{x}) := \sum_{i=1}^{k} \alpha_i f_i(\mathbf{x}) \\ \text{s.t.} \;\; & \mathbf{x} \in Q \end{aligned} \tag{9}$$

The main advantage of the Weighted Sum Method is that one can expect to find Pareto optimal solutions, to be more precise:

Theorem 2 ([17]). *Let $\alpha_i > 0$, $i = 1, \ldots, k$, then a solution of (9) is Pareto optimal.*

For instance, for MOP (1) for a given vector α the solution of (9) is given by $\mathbf{x}^*_\alpha = \sum_{i=1}^{k} \alpha_i \mathbf{a}^i$ as the discussion in Example 1 shows. One can show that for problems where

the Pareto front is convex all points on this set can be reached by solving (9) for a particular value of α (compare also to Figure 3).

On the other hand, the proper choice of α—though it appears to be intuitive at first sight—is in certain cases a delicate problem. Further, the images of (global) solutions of (9) cannot be located in parts of the Pareto front where it is concave. That is, not all points of the Pareto front can be reached when using the Weighted Sum Method which represents a severe drawback. For more details we refer to [1, 18].

Weighted Tchebycheff Method The aim of the Weighted Tchebycheff Method is to find a point those image is as close as possible to a given reference point $\mathbf{Z} \in \mathbb{R}^k$. For the distance assignment the weighted Tchebycheff metric is used: Let $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^{k} \alpha_i = 1$, and let $\mathbf{Z} = [z_1, \ldots, z_k]^T$, then the Weighted Tchebycheff Method [19] reads as follows:

$$\min_{\mathbf{x} \in Q} \max_{i=1,\ldots,k} \alpha_i |f_i(\mathbf{x}) - z_i| \tag{10}$$

Note that the solution of (10) depends on $\mathbf{Z}$ as well as on α. The main advantage of the Weighted Tchebycheff Method is that by a proper choice of these vectors every point on the Pareto front can be reached. See Figure 4 for an example.

Theorem 3. *The solution of (10) is weakly Pareto optimal if* $\alpha \in \mathbb{R}^k_+$.

Theorem 4. *Let* $\mathbf{x}^* \in Q$ *be Pareto optimal. Then there exists* $\alpha \in \mathbb{R}^k_+$ *such that* $\mathbf{x}^*$ *is a solution of (10), where* $\mathbf{Z}$ *is chosen as the utopian vector of the MOP.*

The utopian vector $\mathbf{F}^* = [f_1^*, \ldots, f_k^*]^T$ of a MOP consists of the minimal objective values f_i^* of each function f_i.

On the other hand, the proper choices of $\mathbf{Z}$ and α might also present a delicate problem for a particular problem. More details on this method can e.g. be found in [1, 20].

ϵ-Constraint Method The idea of the ϵ-constraint Method [21] is to select one objective f_i, $i \in \{1, \ldots, k\}$, and to treat all the others as constraints by imposing upper

bounds on the function values. This leads to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x}\in Q} \quad & f_i(\mathbf{x}) \\ \text{s.t.} \quad & f_j(\mathbf{x}) \leq \epsilon_j \quad \forall j \in \{1,\dots,k\}\backslash\{i\} \end{aligned} \tag{11}$$

Theorem 5. *A vector $\mathbf{x}^* \in Q$ is Pareto optimal if and only if it is a solution of the ϵ-constraint problem (11) for every $i = 1,\dots,k$, where $\epsilon_j = f_j(\mathbf{x}^*)$ for $j \in \{1,\dots,k\}\backslash\{i\}$.*

Hence, using the ϵ-constraint Method, it is possible to find every Pareto optimal solution regardless of the form of the Pareto front. Similar to the other scalarization methods presented above, the proper choice of the values of ϵ may get difficult.
Figure 5 shows the solution of

$$\begin{aligned} \min_{\mathbf{x}\in Q} \quad & f_1(\mathbf{x}) \\ \text{s.t.} \quad & f_2(\mathbf{x}) \leq \epsilon_2 \end{aligned} \tag{12}$$

for a bi-objective problem.

Normal Boundary Intersection (NBI) The NBI method [12] computes finite size approximations of the Pareto front in the following two steps (compare to Figure 6): First, the convex hull of individual minima (CHIM) is computed which is the $(k-1)$-simplex connecting the objective values of the minima of each objective. In the second step, points $\mathbf{y}_i$ from the CHIM are selected and the point $\mathbf{z}_i^* \in Q$ is computed such that the image $F(\mathbf{x}_i^*)$ has the maximal distance from $\mathbf{y}_i$ in the direction that is normal to the CHIM and points toward the origen.
To be more precise, let $\mathbf{x}_i^*$ be a global minimizer of the i-th objective, let $\mathbf{F}_i^* := F(\mathbf{x}_i^*)$, and denote

$$\mathbf{\Phi} := [\mathbf{F}_1^*,\dots,\mathbf{F}_k^*] \in \mathbb{R}^{k\times k}. \tag{13}$$

Then the CHIM is defined as

$$CHIM = \{\mathbf{\Phi}\omega \ : \ \omega \in \mathbb{R}^k \ : \ \sum_{i=1}^{k} \omega_i = 1, \ \omega_i \geq 1, \ i = 1,\dots,k\}. \tag{14}$$

The optimization problem in the second step is called the NBI-subproblem. Given an initial value $\mathbf{\Phi}\omega = \sum_{i=1}^{k} \omega_i \mathbf{F}_i^*$ and the direction $\eta \in \mathbb{R}^k$ which is orthogonal to the CHIM and points toward the origin, the NBI-subproblem can be stated in mathematical terms as follows:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{l}} \quad & l \\ \text{s.t.} \quad & F(\mathbf{x}_0) + l\alpha = F(\mathbf{x}) \\ & \mathbf{x} \in Q \end{aligned} \tag{15}$$

The usage of the scalarization (15) can be helpful since there are scenarios where the aim is to steer the search in a certain direction given in objective space (e.g., [22, 23, 24]). On the other hand, solutions of (15) do not have to be Pareto optimal [12].
Further methods that utilize search directions in objective space can be found in [25, 26, 27, 28, 12, 29, 30].

3.2 Pareto Descent Methods

Another prominent way to perform a local search toward the set of optimal solutions is to use *descent directions*. A vector $\nu \in \mathbb{R}^n$ is called a descent direction if a search in this direction leads to an improvement of all objective values. To be more precise, ν is a descent direction of (1) at a point $\mathbf{x} \in \mathbb{R}^n$ if there exists a $\bar{t} \in \mathbb{R}_+$ such that

$$F(\mathbf{x} + t\nu) <_p F(\mathbf{x}), \quad \forall\, t \in (0, \bar{t}). \tag{16}$$

If all objectives of the MOP are differentiable, then (16) is equivalent to

$$\nabla f_i(\mathbf{x})^T \mathbf{v} < 0, \quad i = 1, \ldots, k. \tag{17}$$

Hence, if such a descent direction ν is given at a point $\mathbf{x}$, a further candidate solution $\mathbf{x}_{new}$ that dominates $\mathbf{x}$ can easily be found by a line search, i.e., by setting

$$\mathbf{x}_{new} = \mathbf{x} + t\nu, \tag{18}$$

where $t \in \mathbb{R}_+$ is a (sufficiently small) step size.

In the following we present the approach of Schäffler, Schultz and Weinzierl [31] for the computation of a descent direction for unconstrained MOPs. For further ways to compute descent directions we refer e.g. to [13, 32, 33, 30, 34, 35].

Theorem 6 ([31]). *Let problem (1) be given and the map $q : \mathbb{R}^n \to \mathbb{R}^n$ be defined by*

$$q(\mathbf{x}) = \sum_{i=1}^{k} \hat{\alpha}_i \nabla f_i(\mathbf{x}), \tag{19}$$

where $\hat{\alpha}$ is a solution of

$$\min_{\alpha \in \mathbb{R}^k} \left\{ \left\| \sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}) \right\|_2^2 ; \alpha_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{20}$$

Then the following statements hold.

(a) Either $q(\mathbf{x}) = 0$ or $-q(\mathbf{x})$ is a descent direction.

(b) For each $\hat{\mathbf{x}} \in \mathbb{R}^n$, there exists a neighborhood $N(\hat{\mathbf{x}})$ and a constant $L_{\hat{\mathbf{x}}} \in \mathbb{R}_0^+$ such that

$$\|q(\mathbf{x}) - q(\mathbf{y})\|_2 \leq L_{\hat{\mathbf{x}}} \|\mathbf{x} - \mathbf{y}\|_2, \quad \forall\, \mathbf{x}, \mathbf{y} \in N(\hat{\mathbf{x}}). \tag{21}$$

Note that if $q(\mathbf{x}) = 0$, then $\mathbf{x}$ is a KKT point. Thus, the solution of the initial value problem

$$\begin{aligned} x(0) &= \mathbf{x}_0 \in Q \\ \dot{x}(t) &= -q(x(t)), \quad t > 0 \end{aligned} \tag{22}$$

from an initial solution $\mathbf{x}_0 \in Q$ leads to a KKT point. Note that the computation of q

also contains a test for a first order necessary optimality which can be used as a stopping criterium for the numerical realization of this method.

3.3 Multi-objective Continuation Methods

Since the set of interest, the Pareto set/front, forms at least locally a manifold, it can make sense to perform a search along this set once locally optimal solutions are detected. Such methods are particularly advantageous if the Pareto front is connected.
In the following we describe the core of the predictor-corrector (PC) methods as described in [36] for general implicitly defined manifolds and in [8] in the context of multi-objective optimization. Related methods and further adaptions to the context of multi-objective optimization can e.g. be found in [37, 38, 30, 39].

Crucial for the PC methods presented in [36, 8] is the map $\tilde{F}$ as described in Equation (5) that transforms problem (1) into a root finding problem: The set of interest is now

$$M := \{(\mathbf{x}, \alpha) \in \mathbb{R}^{n+k} \mid \tilde{F}(\mathbf{x}, \alpha) = 0\}. \tag{23}$$

Before we come to the description of PC methods we state some technichal details. The tangent space of M at $(\mathbf{x}, \alpha) \in M$ is given by

$$T_{(\mathbf{x},\alpha)}M = ker\mathbf{J}\tilde{\mathbf{F}}(\mathbf{x}, \alpha) = \{\mathbf{u} \in \mathbb{R}^{n+k} \mid \mathbf{J}\tilde{\mathbf{F}}(\mathbf{x}, \alpha)\mathbf{u} = \mathbf{0}\}, \tag{24}$$

where $\mathbf{J}\tilde{\mathbf{F}}(\mathbf{x}, \alpha)$ denotes the Jacobian of $\tilde{F}$ at $(\mathbf{x}, \alpha)$ and $ker\mathbf{A}$ the kernel of a matrix $\mathbf{A}$. Well-spread tangent vectors can e.g. be obtained by computing a $\mathbf{QR}$-factorization (e.g., [5]) of $\mathbf{J}\tilde{\mathbf{F}}(\mathbf{x}, \alpha)^T$:
Let $\mathbf{Q} = (\mathbf{Q}_1|\mathbf{Q}_2) \in \mathbb{R}^{(n+k)\times(n+k)}$ be an orthogonal matrix where $\mathbf{Q}_1 \in \mathbb{R}^{(n+k)\times(n+1)}$, $\mathbf{Q}_2 \in \mathbb{R}^{(n+k)\times(k-1)}$, and $\mathbf{R} = \left(\frac{\mathbf{R}_1}{0}\right) \in \mathbb{R}^{(n+k)\times(n+1)}$, where $\mathbf{R}_1 \in \mathbb{R}^{(n+1)\times(n+1)}$ is a right

upper triangular matrix such that

$$\mathbf{J}\tilde{\mathbf{F}}(\mathbf{x},\alpha)^T = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla^2 f_i(\mathbf{x}) & 0 \\ \nabla f_1(\mathbf{x})^T & 1 \\ \vdots & \vdots \\ \nabla f_k(\mathbf{x})^T & 1 \end{pmatrix} = \mathbf{QR}. \tag{25}$$

Hereby, $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{n\times n}$ denotes the Hessian of f at $\mathbf{x}$. Then the columns of $\mathbf{Q}_2$ build an orthonormal basis of $T_{(\mathbf{x},\alpha)}M$.

Now we are in the position to formulate the general idea of PC methods together with a common implementation: Given a point $(\mathbf{x}^*, \alpha^*) \in M$, further points along M are computed in the following two steps (compare to Figure 7):

(P) Predict a set $\{\mathbf{p}^1, \ldots, \mathbf{p}^s\} \subset \mathbb{R}^{n+k}$ of distinct and well-distributed points that are near both to $(\mathbf{x}^*, \alpha^*)$ and M. This can e.g. be done by using (25), i.e., by choosing

$$\mathbf{p}^i = (\mathbf{x}^*, \alpha^*) + t_i \mathbf{q}_i, \tag{26}$$

where $t_i \in \mathbb{R}\backslash\{0\}$ is a step size and $\mathbf{q}_i$ the i-th column vector of $\mathbf{Q}_2$.

(C) for $i = 1, \ldots, s$

Starting with the predicted point $\mathbf{p}^i$, compute by (typically few) iterative steps an approximated element $(\mathbf{x}^{(i)}, \alpha^{(i)}) \in M$, i.e., such that $\tilde{F}(\mathbf{x}^{(i)}, \alpha^{(i)}) \approx \mathbf{0}$. This can e.g. be done by applying root finding methods such as the Gauss-Newton method (e.g., [5]) to $\tilde{F}$.

Note that in the above realizations one needs to compute or approximate the Hessians of the objectives. Continuation methods that only require the gradient information can be found in [30, 39, 40].

3.4 Set Oriented Approaches

Next to the point-wise iterative methods described above there exist other approaches that generate an entire set in each iteration step with the aim that the resulting sequence of sets converges to the Pareto set in a suitable sense (e.g., in the Hausdorff sense). Subdivision techniques [41, 42, 43, 44, 45, 46] start with a compact subset Q_0 of the domain, represented by a collection of n-dimensional boxes. Each box gets subdivided into smaller sub-boxes and after certain conditions it is decided if a box is promising – i.e., if it could contain a part of the Pareto set—or not. The 'unpromising' boxes are deleted from the collection while the process—subdivision and selection—is continued successively on the remaining boxes until the desired granularity of the boxes is reached. In this manner, the sequence Q_i of box collections obtained in each iteration step converges toward the set of interest.

Figure 8 shows a numerical result of the algorithm $DS - Subdivision$ [43] applied on the bi-objective problem

$$\begin{aligned} f_1, f_2 :& \mathbb{R}^2 \to \mathbb{R} \\ f_1(\mathbf{x}) &= (x_1 - 1)^4 + (x_2 - 1)^2, \\ f_2(\mathbf{x}) &= (x_1 + 1)^2 + (x_2 + 1)^2 \end{aligned} \tag{27}$$

The Pareto set $\mathcal{P}$ of MOP (27) is a curve connecting the points $[-1, -1]^T$ and $[1, 1]^T$. After 20 iteration steps the resulting box collection Q_{20} already forms a tight covering of $\mathcal{P}$.

Similar in spirit are methods based on interval analysis (e.g., [47, 48, 49, 50]). These interval analysis techniques ensure that the box collection is always a superset of the Pareto set, i.e., that no sub-box that contains a part of $\mathcal{P}$ is rejected wrongly, leading to reliable numerical computations of the Pareto set.

Further, there exist cell mapping techniques [51] that can be seen as the predecessors of the subdivision techniques. Methods of that kind divide the domain into a set of small

n-dimensional cells and perform a cell-to-cell mapping of a given dynamical system g. In doing so, a global view on the dynamics of g on the domain is obtained. If, for instance, multi-objective descent directions are chosen as dynamical systems, one can compute the Pareto set [52] or the set of approximate solutions [53] of a given MOP.
Common to all the methods is that they are restricted to a moderate dimension of the parameter space (say, $n \leq 20$) and to a low number of objectives ($k < 5$).

4 Evolutionary Multi-objective Optimization

Most real-world multi-objective optimization problems are difficult to solve. Specific algorithms to solve this type of problems can be so specialized, that they can only be applied to a small range of problems, or they can be instead, more general, but rather inefficient. Some general search heuristics might require high computational time and will eventually fail if the problem's search space is very large, i.e., large scale problems considering the decision variable space. Also, hill-climbing algorithms face problems when searching for optimal solutions in cases in which the problem's space is multimodal[2], since in such cases, they will get stuck in local optima most of the time.

In order to tackle difficult multi-objective optimization problems with large, multimodal and accidented search spaces, a number of metaheuristics have been proposed. From the many metaheuristics currently available, multi-objective evolutionary algorithms (MOEAs) are perhaps the most popular [54]. MOEAs are inspired by nature and one of their main advantages, as compared to traditional (i.e., mathematical programming) methods, is that they need very little problem specific knowledge and can, therefore, be applied to a broad range of problems. Because of their stochastic nature, MOEAs can be applied to discontinuous, non-differentiable and possibly noisy and/or highly nonlinear search spaces. The following are some of the reasons, for which MOEAs are well suited for complex optimization problems:

- They are improvement-driven. MOEAs are designed to continuously improve the

[2]Multimodal search spaces contain not only one global optimum but many suboptima, which might deceive a simple search algorithm.

fitness function defined in the problem.

- MOEAs are inherently quantitative and, therefore, they are well suited for parameter optimization.
- MOEAs allow the incorporation of a wide variety of extensions and constraints than cannot be provided in traditional methods.
- MOEAs are robust, balancing, at the same time, efficiency and efficacy.
- MOEAs are easily coupled to other optimization techniques. For example, the use of memetic algorithms allows the combination of global and local search processes.

MOEAs are stochastic search techniques which are inspired on nature, namely on Darwin's evolutionary theory. The basic idea of an evolutionary algorithm is that the fittest individuals of a population are given a higher probability of reproducing, with the aim that their offspring can improve their fitness values. Over time, the population is expected to converge to a set of individual with high fitness values. Evolutionary algorithms also incorporate a mutation operator, which allows to explore unknown regions of the search space, and aims to avoid getting trapped in local optima. MOEAs incorporate two main mechanisms that are not present in single-objective evolutionary algorithms: (1) a selection mechanism that normally, but not necessarily, incorporates the definition of Pareto optimality and (2) a density estimator, which blocks the selection mechanism and allows the generation of as many different solutions as possible. A MOEA follows the basic scheme shown in Algorithm 1.

In the first step, a population of random solutions is created, and then the MOEA generational loop is entered. At each generation, the individual's fitness Φ_i is obtained which is further used to assign a probability of selection p_i. Individuals with higher fitness values will receive higher selection probabilities, and will be able more times to participate in the creation of offspring (descendants) for the next generation. From the selected parents, an offspring population is created by applying the Evolutionary Operators (EVOPs), namely crossover and/or mutation. In all MOEAs, descendants

Algorithm 1 Basic MOEA

1: $t \leftarrow 0$
2: Generate an initial population $P(t = 0)$
3: **while** Stopping criterion not met **do**
4: Evaluate the fitness Φ_i for each individual in $P(t)$, based on Pareto dominance, crowding distance, or any other metric.
5: Compute the probability selection p_i for each individual in $P(t)$ based on its fitness value Φ_i.
6: Select the fittest $P'(t)$ as parents from $P(t)$
7: Apply the Evolutionary Operators (EVOPs) of crossover and mutation to create an offspring population $P(t + 1)$.
8: From the populations $P(t)$ and $P(t+1)$, select the best individuals to pass to the next generation.
9: $t \leftarrow t + 1$
10: **end while**

can be imperfect clones of the parents with small variations (this corresponds to the naturally occurring mutations in nature), or the descendants are a combination of multiple parents (this corresponds to the sexual reproduction in nature) or both. Finally, from the populations of parents and offspring, a selection process is performed, allowing the best individuals, measured in terms of Pareto dominance or other related metric, to pass to the next generation.

MOEAs can be classified in several ways [54]. However, for the purposes of this chapter, we present a simple high-level classification that considers only three types of MOEAs: (a) Non-Pareto-based, (b) Pareto-based, and (c) Indicator-based. The first group contains MOEAs that do not adopt the concept of Pareto optimality in their selection mechanism, whereas the second comprises MOEAs that adopt Pareto optimality in their selection mechanism. Finally, the third group considers more recently developed MOEAs which adopt a performance measure in their selection process.

4.1 Non-Pareto based algorithms

Some of the most popular non-Pareto-based MOEAs are the following:

- **Lexicographic method**: The user ranks the objectives of the problem in a decreasing order and the optimization proceeds from higher to lower order objectives, one at a time. Once an objective is optimized, the aim is to improve as much as

possible the following objective(s) without decreasing the quality of the previous one(s) [54].

- **Aggregating functions**: All the objectives are added up into a single (scalar) value which constitutes the objective to be optimized. i.e. $\bar{f} = \sum_{i=1}^{k} \omega_i f_i(\mathbf{x})$, where $\omega_i \geq 0$, $i = 1, 2, \ldots, k$ are the weighting factors representing the decision maker's opinion for each objective (i.e., objective importance). Since objectives tend to be defined in very different ranges, a normalization is normally required. Also, weights tend to be assigned to each objective in order to define preferences from the user [54]. Varying the weights during the run allows, in general, the generation of different nondominated solutions in one run [55, 56]. Figure 9 illustrates the aggregating function approach.

- **Population-based methods**: A number of sub-populations (usually as many as the number of objective functions of the problem) are generated from a main population of an EA. Each sub-population optimizes a single objective function and then all the sub-populations are merged and mixed. The aim is that, when performing crossover, individuals that are good in one objective will recombine with individuals that are good in another one [57]. This sort of approach typically misses good compromises among the objectives because of the way in which individuals are selected in each population [54].

4.2 Pareto-based approaches

Among the Pareto-based methods, there are two sub-classes: the non-elitist MOEAs and the elitist MOEAs. Non-elitist MOEAs do not retain the nondominated solutions that they generate and could, therefore, lose them after applying the evolutionary operators. Elitist MOEAs retain these solutions either in an external archive or in the main population.

The most representative non-elitist MOEAs are the following:

- **Nondominated Sorting Genetic Algorithm** (NSGA): It was proposed by Srini-

vas and Deb [58]. It 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 individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, in order to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values (i.e., fitness sharing [59] is adopted). 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. Figure 10 illustrates the Pareto ranking and sharing mechanisms used in NSGA.

- **Niched-Pareto Genetic Algorithm** (NPGA): Proposed by Horn et al. [60]. It uses a tournament selection scheme based on Pareto dominance. The basic idea of the algorithm is the following: Two individuals are randomly chosen and compared against a subset from the entire population (typically, around 10% of the population). If one of them is dominated (by the individuals randomly chosen from the population) and the other is not, then the nondominated individual wins. When both competitors are either dominated or nondominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing [59].

- **Multi-Objective Genetic Algorithm** (MOGA): This algorithm was proposed by Fonseca and Fleming [61]. For this approach, the rank of a certain individual corresponds to the number of individuals in the current population by which it is dominated plus one. All nondominated individuals are assigned the lowest possible rank (i.e., one), while dominated ones receive as their rank the number of individuals that dominate them plus one, i.e. $rank(i) = 1 + q_i$ where q_i is the number of individuals that dominate individual i in objective space. Fitness sharing is applied in objective space in order to obtain a good distribution of solutions along the Pareto front. Figure 11 illustrates the ranking mechanism used in MOGA.

Among the most popular Pareto-based elitist MOEAs, we have the following:

- **Strength Pareto Evolutionary Algorithm** (SPEA): Zitzler and Thiele introduced this MOEA in [62]. It uses an archive containing nondominated solutions previously found (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set, removing the dominated solutions. For each individual in this external set, a *strength* value is computed. This strength is similar to the ranking value of MOGA [61], since it is proportional to the number of solutions to which a certain individual dominates. The fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. The fitness assignment process of SPEA considers both closeness to the true Pareto front and even distribution of solutions at the same time. The effectiveness of this approach relies on the size of the external nondominated set, since such a set participates in the selection process of SPEA. In fact, since the external nondominated set participates in the selection process of SPEA, if its size grows too large, it might reduce the selection pressure, thus slowing down the search. Because of this, the authors decided to adopt a technique that prunes the contents of the external nondominated set so that its size remains below a certain threshold. The approach adopted for this sake was a clustering technique called "average linkage method" [63].

- **Strength Pareto Evolutionary Algorithm 2** (SPEA2): It was proposed by Zitzler et al. [64] and has three main differences with respect to its predecessor [62]: (1) it incorporates a fine-grained fitness assignment strategy which, for each individual, takes into account both the number of individuals to which it dominates and the number of individuals that dominate it; (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.

- **Pareto Archived Evolution Strategy** (PAES): This algorithm was introduced by Knowles and Corne [65]. 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 the nondominated solutions previously found. This archive is used as a reference set against which each mutated individual is compared. Such a historical archive is the elitist mechanism adopted in PAES. However, an interesting aspect of this algorithm is the method used to maintain diversity which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location (see Figure 12) based on the values of its objectives (which are used as its "coordinates" or "geographical location"). The archive is fixed in size, and once its upper bound is reached, a new generated solution is inserted and the archive is pruned, by deleting individuals in the most crowded grid cells.

- **Nondominated Sorting Genetic Algorithm II** (NSGA-II): This approach was proposed by Deb et al. [66] as an improved version of the NSGA. In the NSGA-II, solutions are ranked using a nondominated sorting scheme, and the density of solutions surrounding a particular solution in the population is estimated by computing the average distance of two points on either side of this solution along each of the objectives of the problem. This value is the so-called *crowding distance* (see Figure 13). During selection, the NSGA-II uses a crowded-comparison operator which takes into consideration both the nondomination rank of an individual in the population and its crowding distance (i.e., nondominated solutions are preferred over dominated solutions, but between two solutions with the same nondomination rank, the one that resides in the less crowded region is preferred). The elitist mechanism of the NSGA-II consists of combining the best parents with the best offspring obtained (i.e., a $(\mu + \lambda)$-selection).

In general, MOEAs have two main goals: (i) to produce approximations that minimize the distance to the true Pareto-optimal set, and (ii) to maximize diversity along the Pareto front (i.e., to produce solutions that are spread along the Pareto front).

In the most popular MOEAs, these two goals are tackled using a selection mechanism based on Pareto optimality and a density estimator, which is responsible for maintaining different solutions in the population of a MOEA. The most popular density estimators

include the use of fitness sharing [59] (which penalizes solutions that are too close either in decision variable space or in objective function space), clustering [62], crowding [66], entropy [67] and adaptive grids [65].

4.3 Indicator-based algorithms

A relatively recent trend regarding the design of MOEAs has been the use of a selection mechanism based on a performance indicator. These MOEAs are referred to as Indicator-based MOEAs. Next, we describe two algorithms (IBEA and SMS-EMOA) that belong to this class:

- **Indicator-Based Evolutionary Algorithm** (IBEA): It was proposed by Zitzler and Künzli [68]. The main idea of this algorithm is to first define the optimization goal in terms of a binary performance indicator, and then to directly use the value of this indicator in the selection process. This MOEA can be considered as a general indicator-based approach, since any binary performance indicator can be used in the fitness assignment function for each of the solutions in the current population. The fitness function definition is:

$$Fitness(\mathbf{x}) = \sum_{\mathbf{y} \in P \setminus \{\mathbf{x}\}} -e^{-I(\{\mathbf{x}\},\{\mathbf{y}\})/\kappa} \tag{28}$$

  In equation (28), P is the actual population and κ is a scaling factor which needs to be defined by the user and depends on the problem being solved. Also, this fitness function definition requires that the binary quality indicator $I(\{\mathbf{x}\},\{\mathbf{y}\})$ be dominance preserving[3]. Next, we present the formal definition for a dominance preserving binary quality indicator.

  Definition 3. A binary quality indicator $I(\{\mathbf{x}\},\{\mathbf{y}\})$ is denoted as dominance preserving if (i) $\mathbf{x} \prec \mathbf{y} \Rightarrow I(\{\mathbf{x}\},\{\mathbf{y}\}) < I(\{\mathbf{y}\},\{\mathbf{x}\})$, and (ii) $\mathbf{x} \prec \mathbf{y} \Rightarrow I(\{\mathbf{z}\},\{\mathbf{x}\}) \geq I(\{\mathbf{z}\},\{\mathbf{y}\})$ for all $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z} \in \mathcal{X}$.

[3]Some binary quality indicators with this property can be found in [69].

With this condition, the fitness assignment scheme is also Pareto-compliant [69]. The fitness assignment mechanism tries to rank the population members according to their usefulness, regarding the reformulated optimization goal, i.e., to maximize/minimize the performance indicator. In summary, the proposed fitness function measures the *"loss in quality"* in the binary quality indicator if a solution is removed from the actual population.

In its basic form, IBEA performs binary tournaments for mating selection, and implements environmental selection by iteratively removing the worst individual from the population, in terms of the binary quality indicator measure, and updating the fitness values of the remaining individuals in the population. In their proposed approach, its authors make use of the binary additive ϵ-indicator $I_{\epsilon+}$ and the Hypervolume inidicator I_{HV}. One particular aspect of IBEA is that for both, mating selection and environmental selection processes, comparisons are made in a pairwise sense, reducing in consequence the computational overhead in computing the binary indicator values.

- **S-Metric Selection - Evolutionary Multi-Objective Algorithm** (SMS-EMOA): It was proposed by Beume et al. [70]. For this algorithm the hypervolume (or S-Metric) contribution is used in the environmental selection process. SMS-EMOA is a steady state algorithm in which only one solution is created at a time and inserted into the actual population for performing the environmental selection. Then, for each solution in the extended population, its contribution to the hypervolume measure is computed as the difference of the hypervolume measure with and without it. This difference is assigned as a fitness value to each solution in the population:

$$Fitness(\mathbf{x}) = I_{Hv}(P) - I_{Hv}(P \backslash \{\mathbf{x}\}) \tag{29}$$

In equation (29), P corresponds to the extended population, i.e., including the newly generated solution. Since the maximization of the hypervolume measure attains both, the goals of convergence towards the Pareto-optimal solutions and

maximizes the spread of solutions along the Pareto-front approximation [71, 72], the solution with the lowest contribution to the Hv measure is then discarded. In Figure 14, the basic ranking mechanism of solutions used in SMS-EMOA is illustrated. In this figure the number close to each solution corresponds to its rank, based on the hypervolume contribution, which is depicted as the shaded area to the right of each solution. Also, in this figure it can be observed that the Pareto extreme solutions receive the first k higher ranks (k is the number of objectives in the MOP) in order to avoid losing them.

At the beginning of the evolutionary process, many solutions in the current population will normally be dominated and, therefore, they do not contribute to the hypervolume measure of the Pareto-front approximation. For these cases, the SMS-EMOA algorithm relies on the Pareto ranking approach used in the NSGA-II. The hypervolume measure contribution is computed for each rank layer of solutions. In consequence, discarded solutions will be selected as the less contributing in the hypervolume measure but in the highest rank layer.

4.4 Use of Decomposition

Another relatively recent trend regarding the design of MOEAs has been the use of scalarization methods. The most representative approach within this class is the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [73]. This MOEA is based on the use of a decomposition process that transforms a MOP into a number of scalar optimization problems (SOPs). Each SOP (or subproblem), is a (linearly or nonlinearly) weighted aggregation of the individual objectives. Neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors. Thus, subproblem i is a neighbor of subproblem j if the weight vector of subproblem i is close to that of subproblem j. Each subproblem is optimized in MOEA/D by using information mainly from its neighboring subproblems. Figure 15 illustrates the MOEA/D approach.

In a simple version of MOEA/D, each individual subproblem keeps one solution in

its memory, which could be the best solution found so far for the subproblem. It then generates a new solution by applying evolutionary operators on several solutions from its neighboring subproblems, and updates its memory if the new solution is better than the old one for the subproblem. A subproblem also passes on its newly generated solution to some (or all) of its neighboring subproblems, which will update their current solutions if the received solution is better. A major advantage of MOEA/D is that scalar objective local search can be used in each subproblem in a natural way since its task is optimizing a scalar objective subproblem.

4.5 Other metaheuristics

There are several other multi-objective metaheuristics available. Two of them are briefly discussed next:

- **Particle Swarm Optimization** (PSO): This metaheuristic is inspired on the choreography of a bird flock which aim to find food [74]. It can be seen as a distributed behavioral algorithm that performs (in its more general version) a multidimensional search. The implementation of the algorithm adopts a population of particles, whose behavior is affected by either the best local (i.e., within a certain neighborhood) or the best global individual. PSO has been successfully used for both continuous nonlinear and discrete binary optimization [75]. For extending PSO to deal with MOPs, the main issues are: (1) how to select particles (to be used as leaders) in order to give preference to nondominated solutions over those that are dominated?, (2) how to retain the nondominated solutions found during the search process in order to report solutions that are nondominated with respect to all the past populations and not only with respect to the current one?, and 3) how to maintain diversity in the swarm in order to avoid convergence to a single solution? Normally, mechanisms very similar to those adopted with MOEAs (namely, Pareto-based selection and external archives) have been adopted in multi-objective particle swarm optimizers (MOPSOs). However, the addition of other mechanisms (e.g., a mutation operator) is also relatively common in MOPSOs. An important

number of multi-objective versions of PSO currently exist (see for example [76]), and this remains as a very active research area.

- **Differential Evolution** (DE): This metaheuristic was proposed by Kenneth Price and Rainer Storn [77, 78] to optimize problems over continuous domains. The core idea of this approach is to use vector differences for perturbing a vector population, and it aims to estimate the gradient in a region (rather than in a point). DE performs mutation based on the distribution of the solutions in the current population. In this way, search directions and possible step sizes depend on the location of the individuals selected to calculate the mutation values. Several DE variants are possible, and they differ in the way in which the parents are selected and in the form in which recombination and mutation takes place (see [78] for more information on DE). The high success of DE in single-objective optimization has made it an interesting candidate for solving MOPs. The main issues for extending DE to multi-objective optimization are very similar to those of PSO (i.e., how to select parents, how to store nondominated solutions and how to maintain diversity in the population). As with MOPSOs, very similar mechanisms to those adopted by MOEAs have been used with multi-objective differential evolution (MODE). A variety of MODE approaches currently exist (see for example [79]), and this also remains as a very active research area.

Although many other MOEAs exist (see for example [80, 81]), it is not the intention of this chapter to be comprehensive. The interested reader may refer to [54, 82, 83] for more information on this topic. The main advantages of MOEAs are their generality, ease of use and the fact that they require little or no specific domain information to operate. Also, they are less susceptible to the specific features of the problem (e.g., shape or continuity of the Pareto front) than traditional mathematical programming techniques [54].

5 MOO for Space Mission Design Problems

In the context of *Space Exploration*, recent initiatives involve developing a robust and flexible capability to visit potential destinations. A Lunar Gateway Station near L_1, a point of balance between the Earth and the Moon has been proposed as a 21st century hub for science and a jumping off point for deep space missions, eventually to land humans on Mars. Space missions to reach such a destination are complex and challenging to design, requiring new and unusual kinds of orbits to meet their goals, orbits that cannot be found by classical approaches to the problem [84, 85, 86]. Classical approaches to spacecraft trajectory design have been quite successful in the near past, but these missions were costly in terms of fuel, e.g., large burns for orbit entry are required. The minimization of fuel (i.e., energy) requirements for a spacecraft's trajectory is important for the feasibility of any deep space mission. An unreasonably high fuel requirement can render a mission infeasible.

In recent years, pressure to reduce the cost of interplanetary missions has derived in designing missions with shorter flight times, smaller launch vehicles, and simpler flight systems. This situation has increased interest in designing spacecraft systems with continuous low propulsion systems. Additionally the need to optimize the continuous thrust profile required in these systems, offers new challenges to trajectory designers. Traditional techniques used in interplanetary mission trajectory design needed only a model with a series of discrete events: launch and planetary flyby times (including characteristics of the flyby trajectory), plus any deep space maneuvers which may be required. Even this discrete case is not a trivial one when complicated multiple gravity assist trajectories are considered, but the additional requirement of optimizing a continuous thrust profile severely constrains, and in many cases exceeds the capability of traditional techniques.

The goal of trajectory design is to find a transfer trajectory, such as the one shown in Figure 16, which takes the spacecraft from a prescribed initial orbit (Earth's orbit in the example) to a prescribed final orbit (Mars' orbit in the example) using thrust controls during the flight and a different times. To effect this transfer, propulsion systems are used. In the low thrust propulsion system case, a small continuous control can operate at

any time. In the Figure 16 several transfer orbits are shown, each one requiring different energy budgets ($\sum_{i=1}^{n} \Delta V_i$), as well as different transfer times. It can be anticipated that the shortest trajectory, will require the highest energy budget, applied in a lesser number of high velocity impulses ΔV_i; while the longest trajectory will require the lowest energy budget, applied in a higher number of low velocity impulses ΔV_i. Both these conditions give rise to a Multiobjective Optimization Problem, which have been recently solved using several techniques. In this section, we briefly summarize some research works for which MOO techniques have been used to solve space mission design problems.

- Hartman et al. [85] and Coverstone-Carroll et al. [86] presented the application of a multi-objective evolutionary algorithm (MOEA) to the design of low-thrust spacecraft trajectories. The authors considered two study cases: a) Earth-Mars rendevouz [85, 86], and b) Earth-Mercury rendevouz [86]. The authors adopted the Nondominated Sorting Genetic Algorithm (NSGA) [58] and considered three objectives: i) maximize spacecraft mass delivery at rendevouz, ii) minimize the spacecraft mission flight time, and iii) maximize the spacecraft heliocentric revolutions. Three constraints were also imposed on the MOP, from which two were related to the minimum and maximum values for the heliocentric revolutions (i.e., they constrain the range value that the third objective can attain). The third constraint was the convergence error that results from solving a two-point boundary value problem (TPBVP), which includes two sets of seven nonlinear and coupled differential equations each. Since for this case there is no closed form solution, a numerical approximation, based on the calculus of variations is used. In fact, this latter process corresponds to an optimization process by itself, since it involves computing the optimal spacecraft thrust schedule as well as the thrust orientation, along with the optimal orbit that maximizes the delivered weight at the rendevouz point, with its specific constraints at launch/rendevouz points as well as along the transfer orbit. This last optimization process corresponds to the objective function evaluation, which is computationally intensive, since many of the solutions generated by the MOEA might not be feasible. The NSGA was hybridized with a local

search procedure [4] based on a gradient method implemented in NASA's JPL SEPTOP (Solar Electric Propulsion Trajectory Optimization Program) software. So, the MOEA (NSGA in this case) is used for the global search, and the parameters obtained for each individual in the population, are used as input parameters for the SEPTOP software. It is interesting to note that, as reported by Hartmann [87], after applying the local search, the individuals are not updated in their parameters, but only in their fitness values (i.e., the authors adopt a Baldwinian learning strategy). Thus, the authors argue that diversity is preserved in the population. The authors adopt a penalty function to handle the constraints of the problem. The authors were able to find several families of optimal trajectories for the two spacecraft missions analyzed, including some novel trajectories.

- Lee et al. [88] addressed a low-thrust orbit transfer from a geostationary orbit to a retrograde Molnya-type orbit. The challenge in this problem is that it requires to modify five out of six orbital parameters, which is performed with low-thrust applied during long periods of time. The authors considered two objectives: i) minimize the required propellant mass, and ii) minimize the toal flight time. The authors relied on the Q-law (a Lyapunov feedback control law) theory, which requires the tuning of 13 control parameters defining the decision vector. Three different MOEAs were adopted: 1) NSGA [58], 2) The Pareto-based Ranking Genetic Algorithm[5] (PRGA), and 3) the Strength Pareto Genetic Algorithm[6] (SPGA). The results obtained by these three MOEAs are compared based on two performance measures: the size of the dominated space, and the coverage of two sets. For each candidate solution in the MOEA's population, an optimal orbital transfer was estimated, using the Q-law, such that it satisfied the orbital's initial and final boundary conditions, while minimizing the total flight time. Once the schedule and orientation of the thrust along the orbit are obtained, the required propellant mass, and the flight

[4] In Hartmann [87] the approach is called **NSMA** which stands for Non-dominated Sorting Memetic Algorithm.

[5] The description of this algorithm provided by the authors corresponds to that of the Multi-Objective Genetic Algorithm (MOGA) [61].

[6] This is really the Strength Pareto Evolutionary Algorithm (SPEA) [62].

time, allow to evaluate the two objective functions previously indicated. From their comparative study, the authors concluded that both NSGA and SPGA had a similar performance with respect to the measures adopted. These two MOEAs outperformed PRGA. It is worth noting, however, that the authors performed only three runs with each algorithm, because of the high computational cost involved in the evaluation of the objective functions of this problem.

- Luo et al. [89] solved the problem of rendez-vous trajectory parameter optimization. In this case, three objective functions were considered: (i) the time of flight for the spacecrafts to accomplish the rendez-vouz, (ii) the total velocity characteristic which is a function of multiple impulses performed by the chaser spacecraft, and (iii) the trajectory safety performance index, which is a measure of the distance the chaser spacecraft attains in "free path" with respect to the target spacecraft, in case the thrust control ceases. A simplified model (linearized) was adopted for solving the trajectory of the rendez-vouz problem. The problem consisted of a decision vector that could vary in size due to the number of impulses considered in the optimization problem. In the application problems presented, the authors used either three or four impulses, originating decision vectors of seven or eight variables, respectively. Constraints were imposed on the times of applying the impulse and the interval time between two consecutive impulses. The authors adopted the Nondominated Sorting Genetic Algorithm-II (NSGA-II) [66] with real-numbers encoding, arithmetical crossover and nonuniform mutation. Binary tournament selection was adopted, making the solution with the lower front number the winner of the tournaments. If solutions were from the same front, then the solution with the highest crowding distance was selected as the winner. The original constraint-handling mechanism incorporated into the NSGA-II was adopted without any changes. The evaluation of the objective functions was obtained by an iterative method, i.e., a set of differential equations, governing the spacecraft motion. The example problems presented by the authors were for three and four impulses rendez-vouz trajectory optimization. In each case, 10 runs were performed and a "global" Pareto front was

constructed considering the Pareto fronts obtained in each execution. The authors did not report the number of nondominated solutions obtained in any case.

- In a similar work, Luo et al. [90] extended their application for the multiple-impulse rendez-vouz trajectory optimization problem, but in this case using a more sophisticated model (non-linear) for evaluating the objective functions. Additionally, constraints on the path were included to solve a problem with more realistic operational conditions. As before, the NSGA-II was adopted [66] with the modifications previously indicated. The problems that were solved corresponded to a three and four impulses rendez-vouz trajectory optimization. In both cases, trade-offs were obtained among the time of flight, the propellant cost, and the trajectory safety for rendez-vouz missions, with and without path constraints. This information was useful for identifying multiple solutions from which to select one that met specific needs.

- Vavrina and Howell [91], presented the design of three different spacecraft missions: a) Earth-Mars rendevouz, b) Earth-Jupiter rendevouz, and c) Earth-Earth-Jupiter rendevouz. In all cases, two objectives were considered: i) maximize spacecraft mass delivery at rendevouz, and ii) minimize the spacecraft mission flight time. The authors adopted the NSGA-II (as described in [92]) hybridized with a gradient-based local search approach, for obtaining optimal trajectories for each individual in the population of the NSGA-II. The local search engine adopted was a calculus-based method named GALLOP (Gravity-Assist Low-Thrust Local Optimization Program). The application of the local search operator allowed to apply both Lamarckian and Baldwinian learning strategies, but in the examples presented by the authors, a Lamarckian learning strategy was applied only to the feasible solutions obtained by GALLOP. The authors adopted a penalty function to handle the constraints of the problem. The authors reported finding promising and novel optimal trajectories, and discussed the benefits of having the trade-offs between the two objectives previously indicated.

- Croisard et al [93], and Vasile and Croisard [94] addressed the robust preliminary and multidisciplinary design for an interplanetary spacecraft mission, namely, the *BepiColombo* mission. The robust design considered uncertainties in several design parameters, and aims at reducing the impact of these on the optimal value for the design criteria. Unlike other approaches presented above, which make use of the Taguchi method as the robust design framework, in this case, the authors made use of Evidence theory [95, 96]. This allows to model both, stochastic and epistemic uncertainties (i.e., the authors assume a poor or incomplete knowledge of the design parameters) [97]. The latter situation is commonly present in the preliminary design phase of the spacecraft mission considered. The authors considered two objectives in this case: i) maximize the Cumulative Belief Function (CBF) (i.e, a measure of the maximum confidence that a design is better than a certain threshold, in the cost function), and ii) minimize a given cost function, which in the examples presented, corresponds to minimizing the wet mass (related to the mass of propellant required to perform the low-thrust transfer) of the spacecraft being designed. The MOEA used by the authors was the NSGA-II [66]. In the solution of robust design problems, design candidates are not evaluated at fixed values of the design parameters, but considering uncertainties in them. In this case, three uncertain parameters were considered with four threshold intervals and a corresponding BPA (Basic Probability Assignment) each. Thus, for evaluating the CBF, a total of 64 *Focal Elements* (intersection threshold regions for all the uncertain parameters with different BPAs each), had to be searched for. In each of these threshold regions, a local optimizer was used to estimate the maximum of the system's function. Thus, if the whole evolutionary process is considered, it is evident that this is a computationally expensive application. Furthermore, the authors reported the use of a Kriging model for approximating the relation between the spacecraft maximum thrust and the power to be generated by the solar arrays, with the Delta budget (ΔV), which is an important value for the objective function evaluation. The authors compared the use of the NSGA-II to a reference (nearly optimal) solution, and

concluded that their hybrid approach was very useful for estimating the optimum and for narrowing down the search in the presence of uncertainties.

In similar research work, Vasile and Zuiani [98] and Zuiani and Vasile [99], have recently proposed an interesting approach based on the collaboration of multiple agents (MACS). The first approach [98] blends a number of metaheuristics, including particle swarm optimization and differential evolution. This approach, has been succesfully applied to the design of multi-impulse trajectories [98], to the robust design optimization of low-thrust transfers, and aerocapture manoeuvres [97]. In the second approach [99], MACS is extended by adding to the action set of each agent; a first mechanism to perform a local search based on a decomposition method using the Tchebycheff approach [73]; and a second mechanism for Monotonic Basin Hopping (MBH) steps. These combined mechanisms proved to be very effective and efficient for solving the multi-gravity assist transfer to Saturn for the Cassini mission [98], attaining better performance both for convergence and spread of solutions, as compared to the base MACS approach.

- Schütze et al. [100] addressed the multiobjective design of low-thrust gravity-assisted trajectories, used for interplanetary misions, as well as for orbital transfers. The MOP comprised two objectives: i) minimize the propellant mass fraction, and ii) minimize the flight time of a given trajectory. It is important to remark that the design of low-thrust space trajectories requires the solution of an optimal control problem that does not have, in general, a closed form solution. For this latter problem, the optimization process needs to be initialized with a suitable first-guess solution. Contraints in the problem are considered both for velocity and trajectory positions. This situation renders the design space with many infeasible regions. For tackling this situation, Schütze et al. [100] have proposed a novel technique based on the pruning of the design space, and the application of a multiobjective optimization technique in each subregion. The application problems presented corresponded to both Earth-Venus-Mercury, and Earth-Venus-Earth-Jupiter orbital transfers.

- Dellnitz et al. in [101] proposed a novel three-step approach for solving low thrust space trajectories, specifically for solving flight formation of spacecrafts. In their approach, the optimization comprised two objectives: i) minimize the time of flight, and ii) minimize the propulsion effort. The particular problem that they solved consisted of a formation of four spacecrafts to be positioned along the $L_{1,2}$ halo orbits between the sun and the Earth. The authors' proposed approach for solving the Circular Restricted Three Body Problem (CRTBP) in this case, starts by solving, in the first phase, the global multiobjective problem, i.e., by finding feasible trajectories but with a relaxed thrust control law, in order to reduce the computational effort and to find the best possible trade-offs among the objectives. The algorithm used in this phase corresponded to a space subdivision technique similar to the one used in [100]. Then, in the second phase, Pareto-optimal solutions obtained in the previous phase were improved by the use of a local search operator and with a more accurate thrust control law. Finally, in the third phase, a compromise solution is selected in order to solve the fomation flight. A major conclusion of this study is that the pre-selection of trajectories by the global method, even when a simple control law is used, substantially contributes to improve the quality of the solutions obtained in the local search phases.

- Coffee et al. [102] adressed a similar problem to the one described by [101] and [103]. In the first research work, the aim was to generate Pareto optimal solutions for obtaininig $L_{1,2}$ halo orbits between the sun and the Earth. In this approach, two main phases were employed. The first consisted of a global multiobjective optimization process, but in this case, the authors solved for near-efficient Pareto solutions, extracting solutions from dynamical channels formed by invariants methods of the CRTBP problem. Different to the work of [101], this gloabl phase is solved by the use of Numerical Continuation Methods. From these near-optimal solutions, a local search operator is applied to improve the solutions by using an optimal control method.

- Schütze et al. [104] and Schütze et al. [105] proposed a multiobjective evolutionary

approach for two different space mission designs. In the first case, a two impulse orbital transfer from Earth to the Apophis Asteroid is solved, while in the second case, a space trajectory for the sequence Earth-Venus-Mercury is solved. In both examples, two objectives are considered and minimized: i) the propellant mass fraction, and the trajectory time of flight. In both examples, constraints are considered for velocities and position along the orbital trajectories. In this research work, authors proposed a novel approach for finding not only Pareto-optimal solutions, but also ϵ-efficient solutions. The evolutionary approach is based on the ϵ-NSGA-II, and includes an archiving technique for storing the ϵ-efficient solutions. The authors clearly show in these applications, how each part of the Pareto front can belong to a different launch opportunity, and most important, how the reliability of the mission design can be increased, i.e., by having a wider launch window for each mission design example.

- Nakamiya et al. [106], in a recent paper, studied the multiobjective trade-off from the Earth-to-Moon orbital trajectory for the DESTINY (DEmonstration and Space Technology for INterplanetary voYage) space mission. One particular aspect of this study is that a three objective MOP was defined, comprising the following objectives: i) minimize the transit time in the Van Allen Belt (VAB), ii) minimize the use of the ion engine, and iii) minimize the TOF to the Moon. Tha aim of the first objective was to reduce the degradation of the solar array which the electric propulsion depends on for having a succesful mission. This significantly increased the trade-off for the objectives and the design parameters, since design constraints have to account for eclipses, and solar incidences with respect to the solar array. The MOEA adopted to solve this problem was the NSGA-II, and, of particular interest is the MODE (Multi-Objective Design Exploration) approach adopted by the authors [107], which is based on correlation of the design variables with respect to the design objectives. With this approach, the authors were able to extract some basic knowledge from the problem that they solved.

6 Future Research Trends

As we have seen in this chapter, multi-objective space mission design problems are mainly defined by two objective functions or, in some cases, using three objectives. In this regard, there are several MOEAs that are designed to perform well with this low number of objective functions (e.g., NSGA-II). There are, however, certain difficulties that may arise in these problems. Here, it is important to realize that MOPs comprise a coupled and two-level optimization problem in which the design space might contain a very high number of infeasible solutions. For the two-level optimization problem, one level (the upper one) deals with obtaining the best possible trade-offs among the objectives, whereas the second level (the lower one) aims to solve an optimal control problem for finding a control law for a given system of differential equations, describing the paths of the control variables, such that a certain optimality criterion and constraints are achieved. This latter process is used to evaluate the objective functions for each design candidate, and its computational cost is very high, since it is done by iterative methods.

It is worth noticing that several multi-objective techniques have been used in the solution of multi-objective space mission design problems, ranging from MOEAs [85, 87, 86, 88, 89, 90, 92, 104, 105, 91, 93, 97, 94, 106], Set Oriented Numerics techniques based on the pruning of the design space [100, 101], Numerical Continuation Methods [102], to hyperheuristics [98]. Also, it is important to emphasize that most of the applications discussed in this chapter involve the use of a coupled global-local search optimization scheme. This is to say that a MOEA is used to find a set of good solutions, at a coarse granularity, which are further improved using a local search engine (gradient-based techniques are normally used for this sake). Although memetic MOEAs have existed for several years in the specialized literature [108], the development of specific MOEA-based approaches that properly combine a global and a local search scheme in an efficient and effective way when dealing with mission space design applications, is still an open research area. Issues such as how to couple the global search engine with the local search engine, how to handle the constraints (particularly when dealing with large scale applications having many nonlinear constraints), how to handle mixed problems that combine, for

example, integer and real-numbers decision variables (which could be handled separately or at different granularities by the global and the local search engines), how to make the search less expensive (computationally speaking) are some of the possible paths for future research in this area.

Another interesting issue that arises in the problems discussed in this chapter is the size of the feasible region, which can be very small with respect to the entire search space. In most of the applications reviewed in this chapter, constraints are dealt with, by using a simple external penalty function. However, many other constraint-handling approaches exist, which could be very useful in multi-objective optimization, since they can explore the boundary between the feasible and the infeasible region in a more efficient way than traditional penalty functions (see for example [109, 110, 111]).It would also be interesting to design approaches that can efficiently deal with problems having many nonlinear constraints. In this sense, a promising research area is the development of efficient and *ad-hoc* constraint-handling techniques for MOO in space mission design problems.

Most of the MOEAs that have been used so far for space mission design problems rely on the use of genetic algortithms (NSGA, NSGA-II, MOGA), and the use of alternative metaheuristics (e.g., evolution strategies, ant colony optimization, particle swarm optimization, differential evolution, etc.) is still scarce. Therefore, this is another interesting research path that could be explored, since some of these metaheuristics are very effective in certain domains (e.g., differential evolution is known to be very effective in continuous search spaces). Additionally, the use of more recent MOEAs that rely on the use of indicator-based selection mechanisms [70], decomposition [73] and relaxed forms of Pareto dominance [112] is also very scarce and could be another interesting venue for future research in this area.

The evaluation of the objective functions in space mission design problems is very costly (computationally speaking). Since MOEAs normally require a relatively high number of objective function evaluations, it seems natural to think about the use of parallelism, and, in fact, parallel MOEAs are relatively common in other aeronautical

and aerospace engineering applications [113]. However, in the applications reviewed in this chapter, no parallel MOEAs were adopted. This is, clearly, a very promising path for future research, since a wide variety of parallelization techniques are currently available for MOEAs, including those based on coevolution [114], cellular computing [115], GPU-based computing [116] and asynchronous techniques [117].

Another approach that is commonly used for reducing the computational cost involved in a multi-objective optimization task, is the adoption of approximation models (or surrogates). In this regard, it is important to remark the use of this type of techniques in the research work of Croisard et al. [93], and Vasile and Croisard [94], which make use of a Kriging model for approximating the relation between the spacecraft maximum thrust and the power to be generated by the solar arrays, with the Delta budget (ΔV), which is an important value for the objective function evaluation. Additionally, and with the aim of reducing the associated computational cost, Dellnitz et al. [101] applied in a first phase of the MOP, a relaxed thrust control law for finding Pareto feasible trajectories, which were further improved by the use of a local search operator and with a more accurate thrust control law. This procedure can be regarded as the use of an approximation model. In this sense, and for reducing the computational cost in MOO space mission design problems, a possible research path is the adoption of (probably local) surrogate/approximation models for the selection of promising solutions, specially for distinguishing between feasible and infeasible solutions. Another possible option for improving efficiency is to adopt knowledge extraction techniques and then reuse this information during the evolutionary search. Although such techniques have been normally used in an *a posteriori* manner (adopting self-organizing maps and ANOVA, as in [106]), it is also possible to use them as *a priori* techniques. For example, Gräning et al. [118] successfully applied this type of approach. The extension of this type of approach to MOO in space mission designs is, indeed, a very promising research path.

Acknowledgements

The first author acknowledges support from IPN-ESIME through the EDI and SIBE programs. The second author acknowledges support from CONACyT project no. 103570. The third author acknowledges support from CONACyT project no. 128554.

References

[1] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[2] M. Ehrgott and M. M. Wiecek. Multiobjective programming. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 667–722. Springer, 2005.

[3] V. Pareto. *Manual of Political Economy*. The MacMillan Press, 1971 (original edition in French in 1927).

[4] H. Kuhn and A. Tucker. Nonlinear programming. In J. Neumann, editor, *Proceeding of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, 1951.

[5] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[6] W. E. Karush. *Minima of functions of several variables with inequalities as side conditions*. PhD thesis, University of Chicago, 1939.

[7] G. Strang. *Calculus*. Cambridge University Press, 2010.

[8] C. Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.

[9] M. Köppen and K. Yoshida. Many-objective particle swarm optimization by gradual leader selection. In *Proceedings of the 8th international conference on Adaptive*

and Natural Computing Algorithms (ICANNGA 2007), pages 323–331, Berlin, Heidelberg, 2007. Springer-Verlag.

[10] O. Schütze, A. Lara, and C. A. Coello Coello. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Transactions on Evolutionary Computation*, 15(4):444–455, 2011.

[11] J. D. Schaffer. *Multiple objective optimization with vector evaluated genetic algorithms*. PhD thesis, Vanderbilt University, 1984.

[12] I. Das and J. Dennis. *Normal-boundary intersection*: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization*, 8:631–657, 1998.

[13] J. Fliege and B. Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[14] K. Klamroth, J. Tind, and M. Wiecek. Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research*, 56:413–437, 2002.

[15] J. Fliege. Gap-free computation of Pareto-points by quadratic scalarizations. *Mathematical Methods of Operations Research*, 59:69–89, 2004.

[16] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer, Berlin Heidelberg, 2008. ISBN 978-3-540-79157-7.

[17] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2:39–45, 1955.

[18] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization*, 14(1):63–69, 1997.

[19] V. J. Bowman Jr. On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Multiple Criteria Decision Making*, pages 76–85, 1976.

[20] K. Daechert, J. Gorski, and K. Klamroth. An augmented weighted Tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. Computers & Operations Research, to appear), 2012.

[21] 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*, 1:296–297, 1971.

[22] J. Branke, T. Kraušler, and H. Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32(6):499–507, 2001.

[23] K. Deb, J. Sundar, U. B. Rao N., and S. Chauduri. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal on Computational Intelligence Research*, 2(3):273–286, 2006.

[24] A. Mohammadi, M. Nabi Omidvar, and X. Li. Reference point based multi-objective optimization through decomposition. In *IEEE Congress on Evolutionary Computation*, pages 1150–1157, 2012.

[25] B. Roy. Problems and methods with multiple objective functions. *Mathematical Programming*, 1:239–266, 1971.

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

[27] P. Korhonen and J. Wallenius. A Pareto race. *Naval Research Logistics*, 35:615–623, 1988.

[28] A. Pascoletti and P. Serafini. Scalarizing vector optimization problems. *Journal of Optimization Theory and Applications*, 42:499–524, 1984.

[29] B. Schandl, K. Klamroth, and M.M. Wiecek. Introducing oblique norms into multiple criteria programming. *Journal of Global Optimization*, 23:925–942, 2002.

[30] O. Schütze, A. Lara, and C. A. Coello Coello. The directed search method for unconstrained multi-objective optimization problems. In *EVOLVE – A Bridge Between Probability, Set Oriented Numerics and Evolutionary Computation*, pages 1–4, 2011.

[31] S. Schäffler, R. Schultz, and K. Weinzierl. A stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications*, 114(1):209–222, 2002.

[32] P. A. N. Bosman and E. D. de Jong. Exploiting Gradient Information in Numerical Multi-Objective Evolutionary Optimization. In Hans-Georg Beyer et al., editor, *2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 755–762, New York, USA, June 2005. ACM Press.

[33] A. Lara Lopez, C. A. Coello Coello, and O. Schütze. A painless gradient-assisted multi-objective memetic mechanism for solving continuous bi-objective optimization problems. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

[34] A. Lara Lopez, S. Alvarado, G. Avigad, C. A. Coello Coello, and O. Schütze. The gradient free directed search method as local search within multi-objective evolutionary algorithms. In *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, pages 153–168, 2012.

[35] P. A. N. Bosman. On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. IEEE Transactions on Evolutionary Computation, to appear), 2012.

[36] E. L. Allgower and K. Georg. *Numerical Continuation Methods.* Springer, 1990.

[37] O. Schütze, A. Dell'Aere, and M. Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Ralph E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings. In-

ternationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005. <http://drops.dagstuhl.de/opus/volltexte/2005/349>.

[38] E. Miglierina, E. Molho, and M. C. Recchioni. Box-constrained multi-objective optimization: A gradient-like method without "a priori" scalarization. *European Journal of Operational Research*, 188(3):662–682, 2008.

[39] M. Ringkamp, S. Ober-Blöbaum, M. Dellnitz, and O. Schütze. Handling high dimensional problems with multi-objective continuation methods via successive approximation of the tangent space. *Engineering Optimization*, 44:1117–1146, 2012.

[40] Honggang Wang. Zigzag search for continuous multiobjective optimization. *INFORMS Journal on Computing*, 25(4), 2013.

[41] M. Dellnitz and A. Hohmann. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numerische Mathematik*, 75:293–317, 1997.

[42] O. Schütze. *Set Oriented Methods for Global Optimization*. PhD thesis, University of Paderborn, 2004. <http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/>.

[43] M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124:113–155, 2005.

[44] J. Jahn. Multiobjective search algorithm with subdivision technique. *Computational Optimization and Applications*, 35(2):161–175, 2006.

[45] O. Schütze, M. Vasile, O. Junge, M. Dellnitz, and D. Izzo. Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41(2):155–181, 2009.

[46] O. Schütze, K. Witting, S. Ober-Blöbaum, and M. Dellnitz. Set oriented methods for the numerical treatment of multiobjective optimization problems. In E. Tantar

et al., editor, *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, pages 187–219. Springer, 2013.

[47] K. Ichida and Y. Fujii. Multicriterion optimization using interval analysis. *Computing*, 44:47–57, 1990.

[48] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. SIAM, 2009.

[49] G. L. Soares, R. O. Parreiras, L. Jaulin, and J. A. Vasconcelos. Interval robust multi-objective algorithm. *Nonlinear Analysis*, 71:e1818–e1825, 2009.

[50] L. Majumder and S.S. Rao. Interval-based optimization of aircraft wings under landing loads. *Computers & Structures*, 87:225–235, 2009.

[51] C.S. Hsu. *Cell-to-cell mapping: a method of global analysis for nonlinear systems*. Applied mathematical sciences. Springer-Verlag, 1987.

[52] Carlos Hernández, Yousef Naranjani, Yousef Sardahi, Wei Liang, Oliver Schütze, and Jian-Qiao Sun. Simple cell mapping method for multi-objective optimal feedback control design. *International Journal of Dynamics and Control*, 1(3):231–238, 2013.

[53] Carlos Hernández, Jian-Qiao Sun, and Oliver Schütze. Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In M. Emmerich et al., editor, *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV*, pages 171–188. Springer, 2013.

[54] 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.

[55] Yaochu Jin, Tatsuya Okabe, and Bernhard Sendhoff. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How?

In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 1042–1049, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[56] Martina Hasenjäger and Bernhard Sendhoff. Crawling Along the Pareto Front: Tales From the Practice. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 174–181, Edinburgh, Scotland, September 2005. IEEE Service Center.

[57] 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. Lawrence Erlbaum, 1985.

[58] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[59] David E. Goldberg and Jon Richardson. Genetic algorithm with sharing for multimodal function optimization. In John J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, New Jersey, 1987. Lawrence Erlbaum.

[60] 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.

[61] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages

416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[62] 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.

[63] J.N. Morse. Reducing the size of the nondominated set: Pruning by clustering. *Computers and Operations Research*, 7(1–2):55–66, 1980.

[64] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.

[65] Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[66] 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.

[67] Hajime Kita, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 504–512, Berlin, Germany, September 1996. Springer-Verlag.

[68] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.

[69] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.

[70] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.

[71] M. Fleischer. The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 519–533, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[72] Joshua D. Knowles, David W. Corne, and Mark Fleischer. Bounded Archiving using the Lebesgue Measure. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2490–2497, Canberra, Australia, December 2003. IEEE Press.

[73] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[74] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.

[75] Andries P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2005.

[76] 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.

[77] Rainer Storn and Kenneth Price. Differential Evolution - A Fast and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[78] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Springer, Berlin, 2005. ISBN 3-540-20950-6.

[79] Efrén Mezura-Montes, Margarita Reyes-Sierra, and Carlos A. Coello Coello. Multi-Objective Optimization using Differential Evolution: A Survey of the State-of-the-Art. In Uday K. Chakraborty, editor, *Advances in Differential Evolution*, pages 173–196. Springer, Berlin, 2008. ISBN 978-3-540-68827-3.

[80] Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[81] Sanyou Y. Zeng, Lishan S. Kang, and Lixin X. Ding. An Orthogonal Multi-objective Evolutionary Algorithm for Multi-objective Optimization Problems with Constraints. *Evolutionary Computation*, 12(1):77–98, Spring 2004.

[82] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[83] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 2011.

[84] Wang Sang Koon, Martin W Lo, Jerrold E Marsden, and Shane D Ross. Dynamical systems, the three-body problem and space mission design. In *International Conference on Differential Equations. Control and Dynamical Systems.*, 2000.

[85] John W. Hartmann, Victoria L. Coverstone-Carroll, and Steven N. Williams. Optimal Interplanetary Spacecraft Trajectories via a Pareto Genetic Algorithm. *The Journal of the Astronautical Sciences*, 46(3):267–282, July–September 1998.

[86] V. Coverstone-Caroll, J.W. Hartmann, and W.M. Mason. Optimal Multi-Objective Low-Thrust Spacecraft Trajectories. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):387–402, 2000.

[87] John William Hartmann. Stochastic Optimization Algorithms for the Optimization of Low-Thrust Interplanetary Spacecraft Trajectories. Master's thesis, Department of Aeronautical and Astronautical Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, January 1999.

[88] Seungwon Lee, Paul von Allmen, Wolfgang Fink, A.E. Petropoulos, and R.J. Terrile. Multi-Objective Evolutionary Algorithms for Low-Thrust Orbit Transfer Optimization. In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*. ACM Press, June 2005.

[89] Ya-Zhong Luo, Guo-Jin Tang, and Yong-Jun Lei. Optimal multi-objective linearized impulsive rendezvous. *AIAA Journal of Guidance, Control, and Dynamics*, 30(2):383–389, March-April 2007. DOI: 10.2514/1.21433.

[90] Ya-Zhong Luo, Yong-Jun Lei, and Guo-Jin Tang. Optimal multi-objective nonlinear impulsive rendezvous. *AIAA Journal of Guidance, Control, and Dynamics*, 30(4):994–1002, July–August 2007. DOI: 10.2514/1.27910.

[91] M. Vavrina and K. Howell. Multiobjective Optimization of Low-Thrust Trajectories using a Genetic Algorithm Hybrid. In *AAS/AIAA Space Flight Mechanics Meeting*, Savannah, Georgia, USA, February 2009.

[92] Matthew A. Vavrina. A Hybrid Genetic Algorithm Approach to Global Low-Thrust Trajectory Optimization. Master's thesis, Department of Aerospace Engineering, Purdue University, West Lafayette, Indiana, USA, 2008.

[93] Nicolas Croisard, Massimiliano Vasile, Stephen Kemble, and Gianmarco Radice. Preliminary space mission design under uncertainty. *Acta Astronautica*, 66(5???6):654 – 664, 2010.

[94] Massimiliano Vasile and Nicolas Croisard. Robust Preliminary Space Mission Design Under Uncertainty. In Yoel Tenne and Chi-Keong Goh, editors, *Computational Intelligence in Expensive Optimization Problems*, pages 543–570. Springer, Berlin, Germany, 2010. ISBN 978-3-642-10700-9.

[95] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[96] Arthur Dempster. New methods for reasoning towards posterior distributions based on sample data. In Ronald R. Yager and Liping Liu, editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 35–56. Springer Berlin / Heidelberg, 2008.

[97] Massimiliano Vasile. Robust Mission Design Through Evidence Theory and Multi-agent Collaborative Search. In E. Belbruno, editor, *New Trends in Astrodynamics and Applications*, volume 1065 of *Annals of the New York Academy of Sciences*, pages 152–173. New York Academy of Sciences, New York, USA, December 2005.

[98] M. Vasile and F. Zuiani. Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(11):1211–1227, 2011.

[99] Federico Zuiani and Massimiliano Vasile. Multi-agent collaborative search with tchebycheff decomposition and monotonic basin hopping steps. *Bioinspired Optimization Methods and their Applications, BIOMA 2012*, 2012.

[100] Oliver Schütze, Massimiliano Vasile, Oliver Junge, Michael Dellnitz, and Dario Izzo. Designing optimal low-thrust gravity-assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41(2):155–181, February 2009.

[101] M. Dellnitz, S. Ober-Blöbaum, M. Post, O. Schütze, and B. Thiere. A multi-objective approach to the design of low thrust space trajectories using optimal control. *Celestial Mechanics and Dynamical Astronomy*, 105(1):33–59, 2009.

[102] T.M. Coffee, R.L. Anderson, and M.W. Lo. Multiobjective optimization of low-energy trajectories using optimal control on dynamical channels. 2011.

[103] Ashley Moore, Sina Ober-Blöbaum, and Jerrold E. Marsden. Trajectory design combining invariant manifolds with discrete mechanics and optimal control. *Journal of Guidance, Control, and Dynamics*, 35(5):1507 – 1525, 2012.

[104] Oliver Schütze, Massimiliano Vasile, and Carlos A. Coello Coello. Approximate Solutions in Space Mission Design. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature–PPSN X*, pages 805–814. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008.

[105] O. Schütze, M. Vasile, and C.A. Coello Coello. Computing the set of epsilon-efficient solutions in multiobjective space mission design. *Journal of Aerospace Computing, Information and Communication*, 8(3):53–70, 2011.

[106] M. Nakamiya, A. Oyama, M. Bando, C. Hirose, S. Campangnola, and Y. Kawakatsu.

[107] Shinkyu Jeong, Kazuhisa Chiba, and Shigeru Obayashi. Data mining for aerodynamic design space. In *AIAA Paper 2005–5079, 23rd AIAA Applied Aerodynamic Conference*, Toronto, Ontario Canada, June 6–9 2005.

[108] Chi-Keong Goh, Yew-Soon Ong, and Kay Chen Tan, editors. *Multi-Objective Memetic Algorithms*. Springer, Berlin, Germany, 2009. ISBN 978-3-540-88050-9.

[109] Efrén Mezura-Montes and Carlos A. Coello Coello. Constraint-Handling in Nature-Inspired Numerical Optimization: Past, Present and Future. *Swarm and Evolutionary Computation*, 1(4):173–194, December 2011.

[110] Efrén Mezura-Montes, editor. *Constraint-Handling in Evolutionary Optimization*, volume 198 of *Studies in Computational Intelligence*. Springer-Verlag, 2009.

[111] Dhish Kumar Saxena, Tapabrata Ray, Kalyanmoy Deb, and Ashutosh Tiwari. Constrained Many-Objective Optimization: A Way Forward. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 545–552, Trondheim, Norway, May 2009. IEEE Press.

[112] M. Farina and P. Amato. A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics Part A—Systems and Humans*, 34(3):315–326, May 2004.

[113] A. Arias-Montano, C.A.C. Coello, and E. Mezura-Montes. Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *Evolutionary Computation, IEEE Transactions on*, 16(5):662 –694, oct. 2012.

[114] K.C. Tan, Y.J. Yang, and C.K. Goh. A Distributed Cooperative Coevolutionary Algorithm for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 10(5):527–549, October 2006.

[115] Enrique Alba and Bernabé Dorronsoro. *Cellular Genetic Algorithms*. Springer, New York, 2008. ISBN 978-0-387-77609-5.

[116] Man-Leung Wong, Tien-Tsin Wong, and Ka-Ling Fok. Parallel evolutionary algorithms on graphics processing unit. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 3, pages 2286–2293, Edinburgh, Scotland, September 2005. IEEE Service Center.

[117] Varvara G. Asouti and Kyriakos C. Giannakoglou. Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes. *Engineering Optimization*, 41(3):241–257, March 2009.

[118] Lars Gräning, Stefan Menzel, Martina Hasenjäger, Thomas Bihrer, Markus Olhofer, and Bernhard Sendhoff. Knowledge extraction from aerodynamic design data

and its application to 3d turbine blade geometries. *J. Math. Model. Algorithms*, 7(4):329–350, 2008.

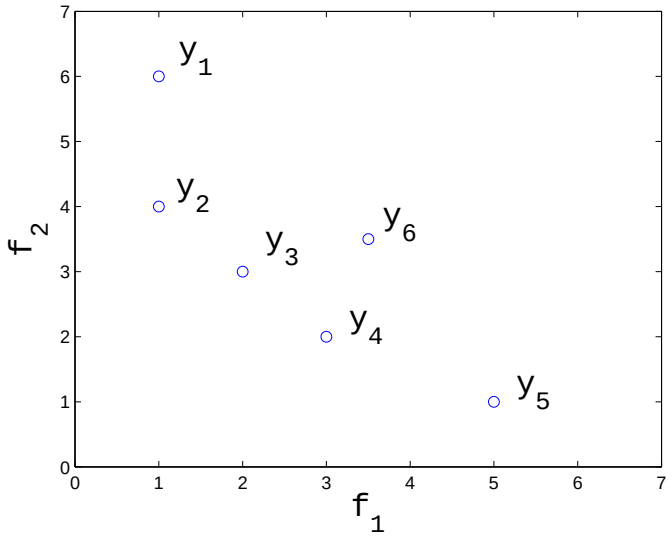
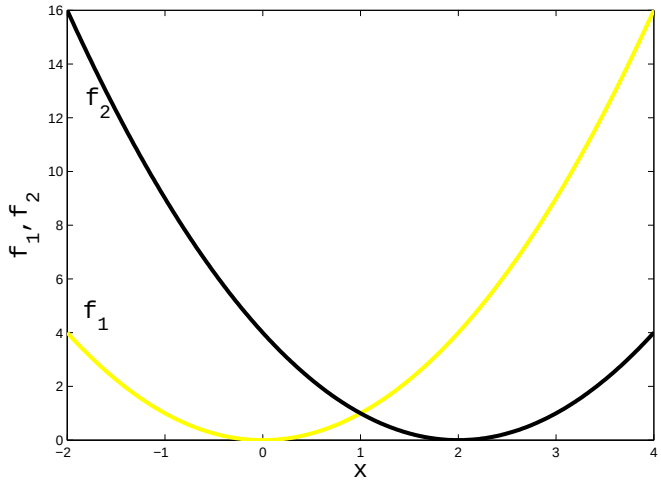
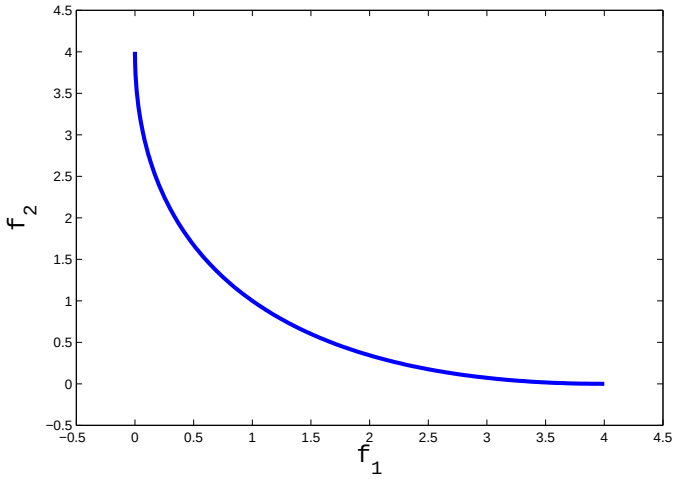


Figure 1: Hypothetical setting for a bi-objective problem



(a) Objective functions



(b) Pareto front

Figure 2: Objective functions and Pareto front of MOP (7). The Pareto set for this problem is $\mathcal{P} = [0, 2]$.

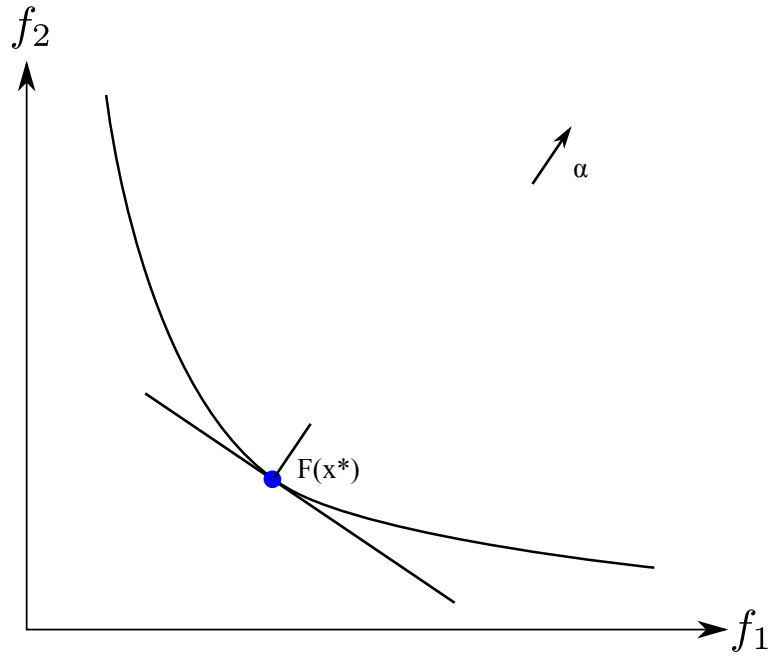


Figure 3: Scheme of the weighted sum method.

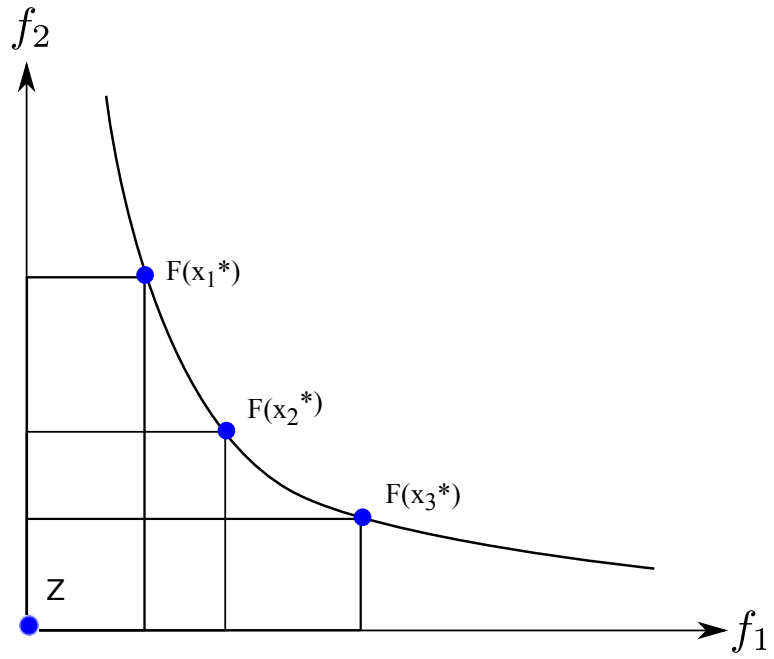


Figure 4: Example of the Tchebycheff method for $\alpha^1 = (0.75, 0.25)$, $\alpha^2 = (0.5, 0.5)$, and $\alpha^3 = (0.25, 0.75)$ and the respective solutions $F(x_i^*)$, $i = 1, 2, 3$.

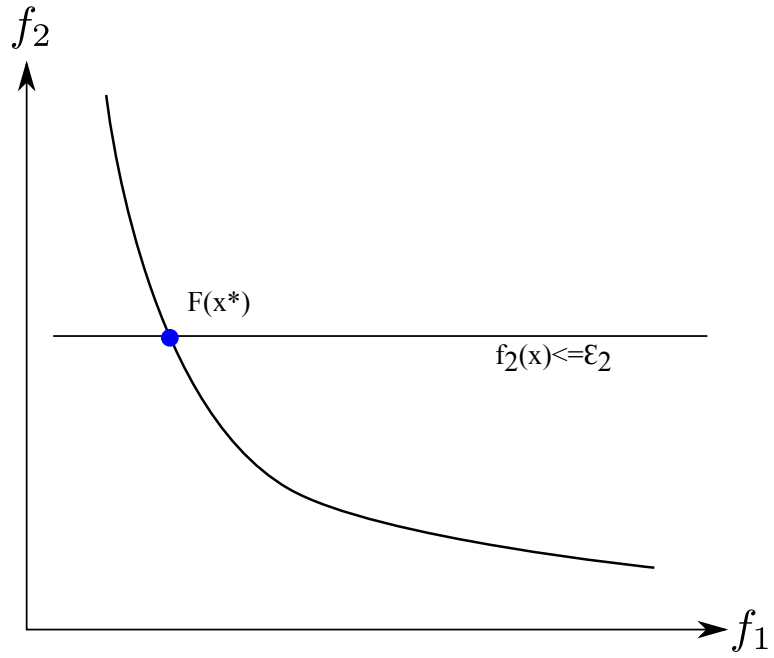


Figure 5: Example of an ϵ-constrained problem.

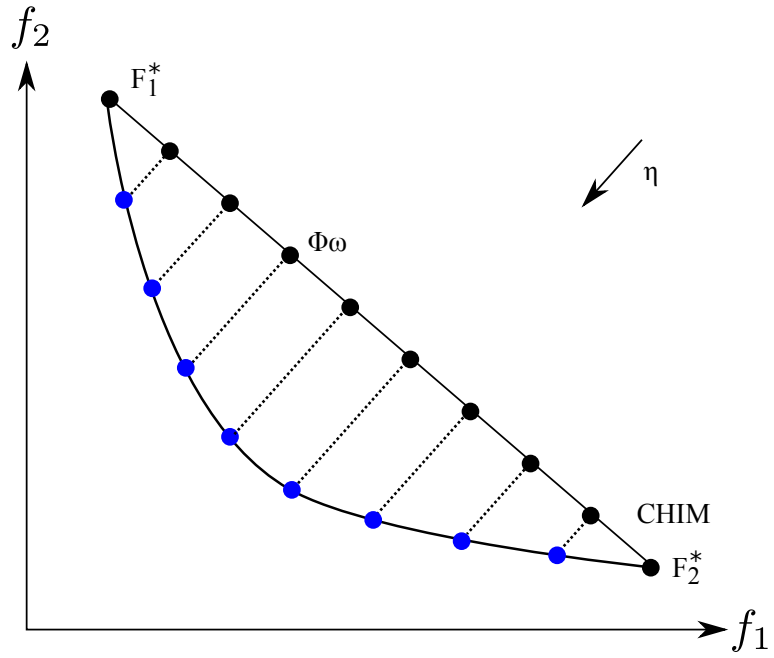


Figure 6: Scheme of the NBI method.

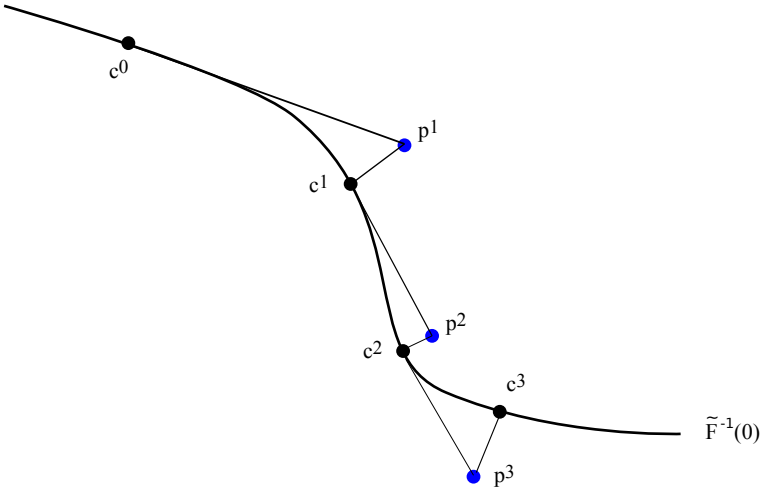
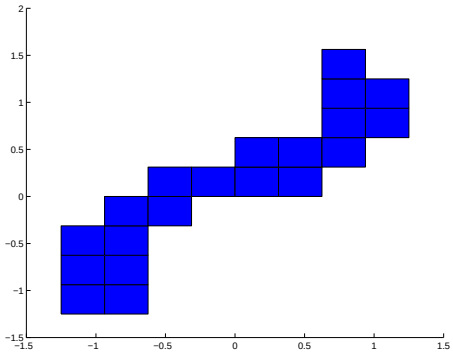
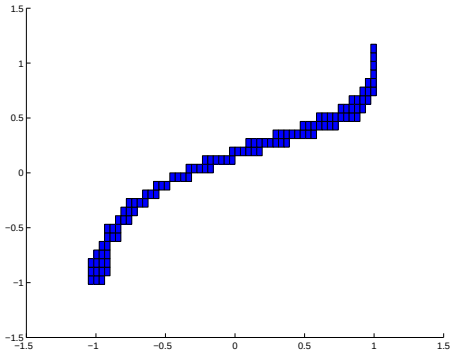


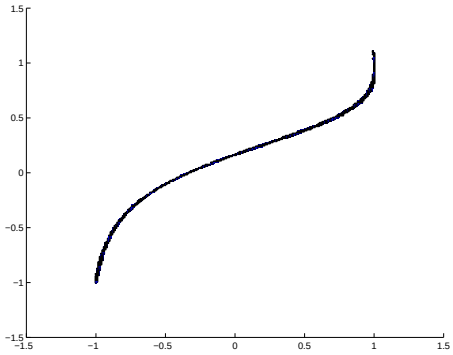
Figure 7: Scheme of continuation methods.



(a) 10 iteration steps



(b) 15 iteration steps



(c) 20 iteration steps

Figure 8: Box collections for different iteration steps obtained by the algorithm DS-Subdivision applied on MOP (27) for different iteration steps (taken from [46]).

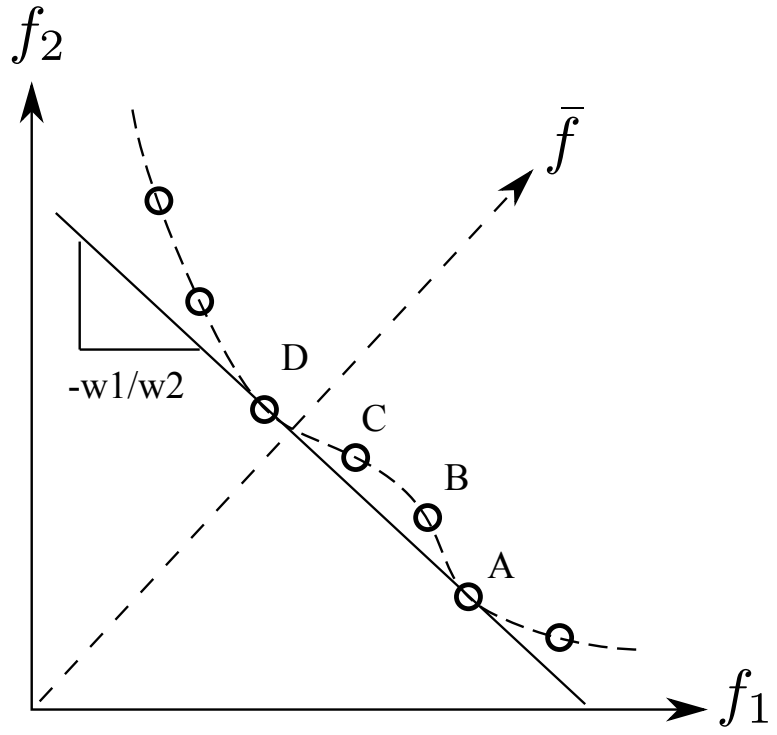


Figure 9: Illustration of the aggregating function approach for solving MOPs.

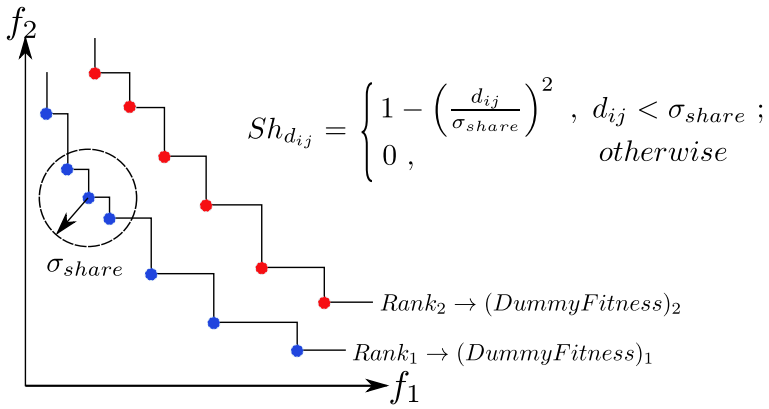


Figure 10: Illustration of the Pareto Ranking and Sharing mechanisms used in NSGA

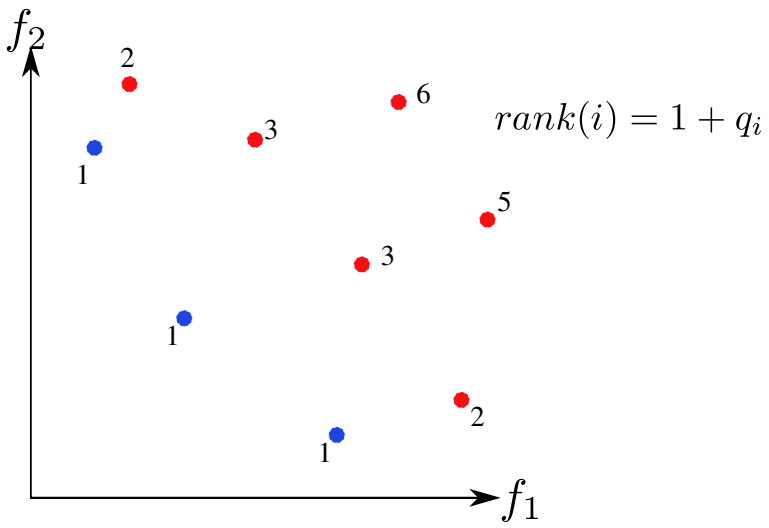


Figure 11: Illustration of the ranking mechanism used in MOGA

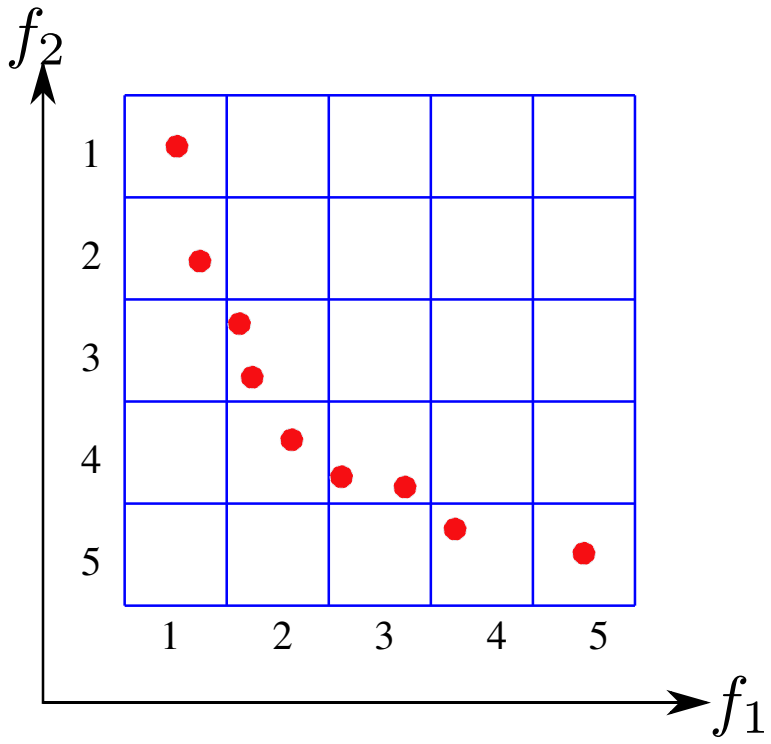


Figure 12: Illustration of the grid archive used in PAES

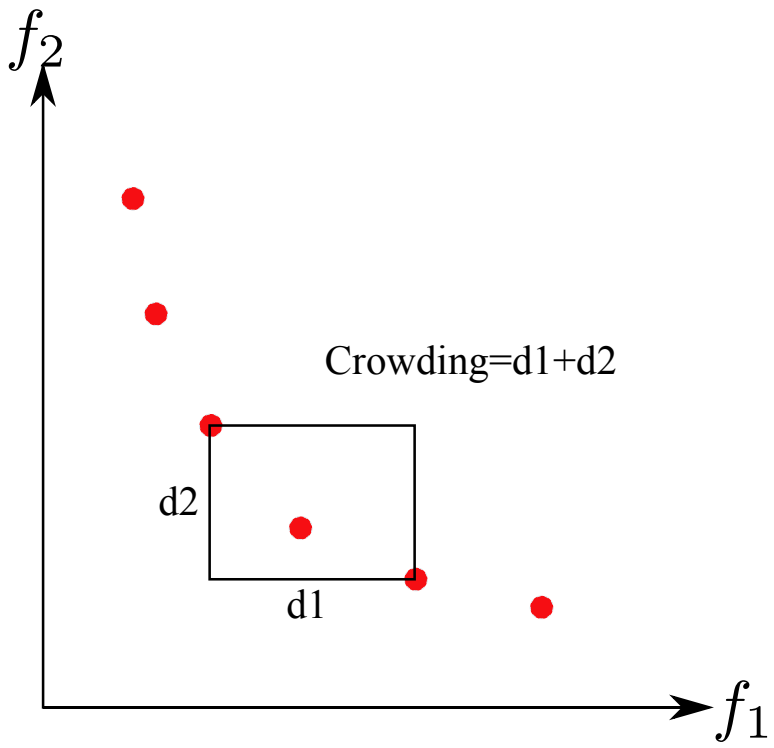


Figure 13: Illustration of the crowding mechanism used in NSGA-II

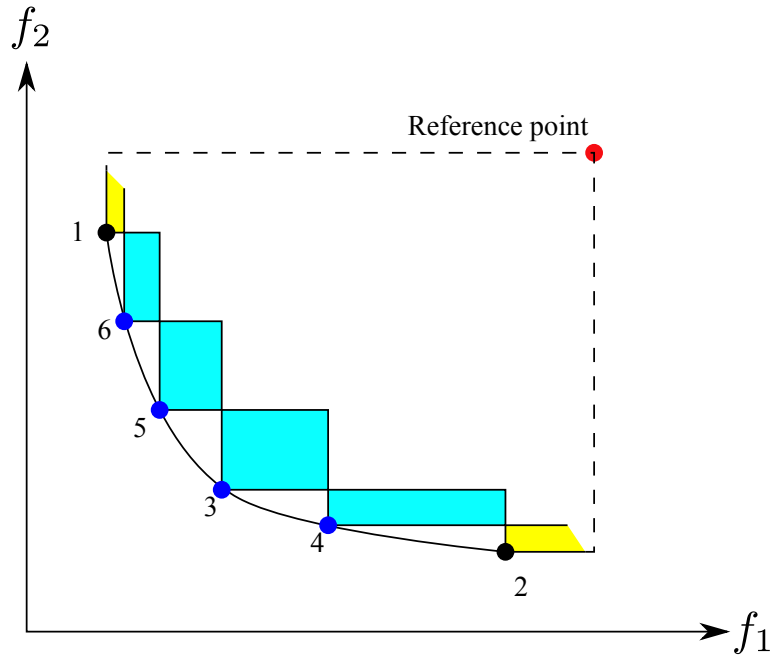


Figure 14: Illustration of the ranking mechanism of solutions based on the hypervolume contribution used in SMS-EMOA.

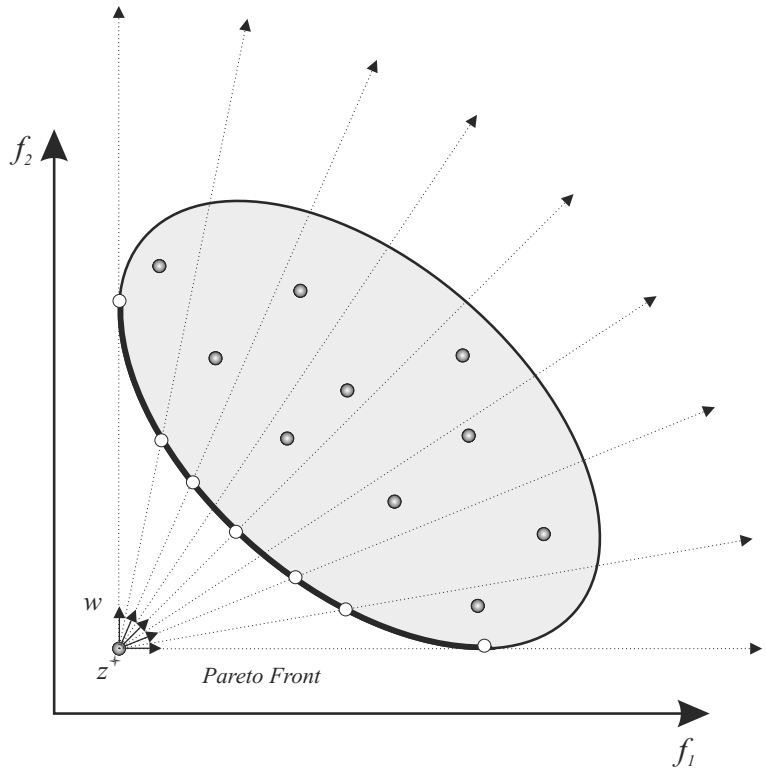


Figure 15: Illustration of decomposition approach used by MOEA/D for solving MOPs.

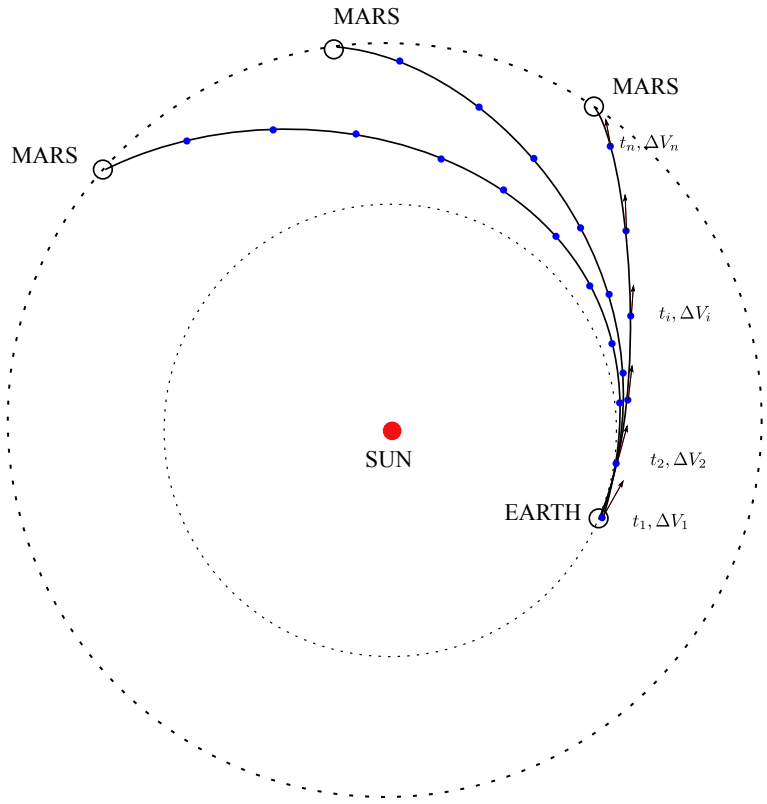


Figure 16: Illustration of different orbital transfer trajectories for the Earth-Mars case.