1

Multi-Objective Combinatorial Optimization: Problematic and Context

Clarisse Dhaenens[1,2], Laetitia Jourdan[1,2] and Carlos A. Coello Coello[3]

[1] Laboratoire d'Informatique Fondamentale de Lille (LIFL), UMR CNRS 8022, Université Lille 1, Bâtiment M3, 59655 Villeneuve d'Ascq cedex, France
[2] INRIA Lille-Nord Europe, Parc Scientifique de la Haute Borne, 40 avenue Halley, 59650 Villeneuve d'Ascq, France
[3] CINVESTAV-IPN, Departamento de Computación, Av. IPN No. 2508, Col. San Pedro Zacatenco, México, D.F. 07300, Mexico
clarisse.dhaenens@lifl.fr
laetitia.jourdan@inria.fr
ccoello@cs.cinvestav.mx

Combinatorial optimization problems are characterized for having discrete decision variables, but an objective function and constraints that can take any form (i.e., linear or nonlinear) [50]. Such problems have been studied for several years in mathematics and computer science where they have attracted a lot of attention, mainly because of their wide applicability (e.g., for planning and scheduling tasks, which are very common in everyday life). However, because of their complexity (combinatorial optimization problems tend to be NP-hard or NP-complete), the use of approximation algorithms (mainly metaheuristics[4]) to solve them has become relatively popular in the last few years [20].

On the other hand, many real-world problems have two or more objectives (often conflicting) which we aim to optimize at the same time (e.g., we aim to minimize the time to complete a task, but simultaneously, we wish the task to be as cheap as possible, which is an objective that normally opposes to the previous one). These problems are called "multi-objective", and their solution involves finding not a single solution, but several, that represent the best possible trade-offs among all the objectives that we aim to optimize. Numerous mathematical programming techniques exist to deal with multi-objective optimization problems [17, 49]. However, the use of metaheuristics in this field has become increasingly popular [8, 11].

[4] A **metaheuristic** is a high-level strategy for explosing the search space of a problem using a variety of methods [7]. Typically, a metaheuristic consists of both a diversification (i.e., mechanisms to explore the search space) and an intensification (i.e., mechanisms that exploit previously found solutions) procedure.

This chapter deals with the use of metaheuristics for solving multi-objective combinatorial optimization problems, and aims to provide a short introduction to the field, which can be used for the rest of the book. It is worth noting, however, that no effort is made to be comprehensive, since excellent surveys of this field already exist (see for example [19, 20]).

The remainder of this chapter is organized as follows. Section 1.1 provides some basic definitions required to understand the rest of the chapter. Then, a brief discussion on the incorporation of user's preferences is provided in Section 1.2. Some of the most commonly adopted multi-objective evolutionary algorithms used for multi-objective optimization are briefly discussed in Section 1.3. The most commonly adopted approaches to assess the performance of multi-objective metaheuristics and some possible methodologies to provide a statistical validation of their results are discussed in Section 1.4. Finally, our conclusions are provided in Section 1.5.

1.1 Basic Definitions

It is worth noting that some of the definitions provided next are used both in single- and multi-objective combinatorial optimization, but others are specific to the latter (in multi-objective optimization, the notion of optimality is different).

1.1.1 Multi-Objective Combinatorial Optimization Problem

A *Multi-Objective Combinatorial Optimization Problem* (MOCOP) is defined as:

$$(MOP) \begin{cases} \text{Optimize} & F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\ \text{with} & x \in \mathcal{D} \end{cases} \tag{1.1}$$

where n is the number of objectives ($n \geq 2$), $x = (x_1, x_2, \ldots, x_k)$ is the vector of decision variables, $\mathcal{D}$ is the set of feasible solutions and each objective function $f_i(x)$ has to be optimized (i.e. minimized or maximized). The bound of each decision variable x_i constitutes a decision variable space often denoted as $\mathcal{D}$. In multi-objective optimization, the objective functions F constitute a multi-dimensional space that is often called $\mathcal{Z}$. For each solution x in decision variable space (or decision space), there exists a point in objective space defined by $F(x)$. The mapping takes place between a k-dimensional solution vector and an n-dimensional objective vector (see Fig. 1.1).

Unlike single-objective optimization, the solution of a MOCOP is not unique, but is composed instead of a set of solutions representing the best possible trade-offs among the objectives. Such solutions are contained in the so-called **Pareto optimal set** (PO). When plotting the objective function values corresponding to the solutions stored in the Pareto optimal set, we obtain the **Pareto front** of the problem.

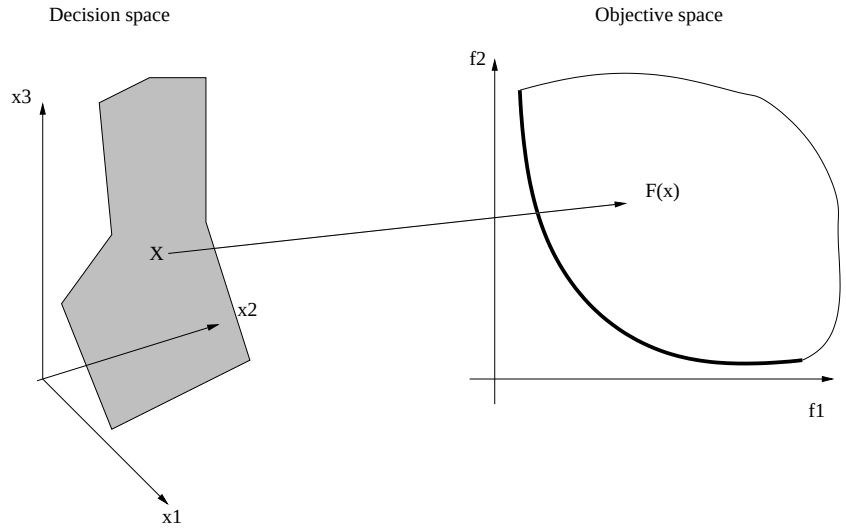

Fig. 1.1. Mapping from decision space to objective space.

1.1.2 Dominance and Pareto Optimality

In multi-objective optimization, there is a different notion of optimality than in single-objective optimization, since in this case, we are interested in finding good compromises (or trade-offs) among the objectives that we wish to optimize. The notion of optimality most commonly adopted is that originally proposed by Francis Ysidro Edgeworth in 1881 [14] and later (in 1896) generalized by Vilfredo Pareto [51]. Although some authors call this notion the **Edgeworth-Pareto optimality**, the most commonly accepted term is **Pareto optimality**.

A feasible solution $x^* \in \mathcal{D}$ is called **Pareto optimal** (also called efficient or nondominated) if and only if there is no solution $x \in \mathcal{D}$ such that x dominates x^*.

A solution $y = (y_1, y_2, \ldots, y_k)$ **dominates** a solution $z = (z_1, z_2, \ldots, z_k)$, in a minimization context, iff $\forall i \in [1 \ldots n]$, $f_i(y) \leq f_i(z)$ and $\exists i \in [1 \ldots n]$ such that $f_i(y) < f_i(z)$.

In this context, any solution of the Pareto optimal set may be considered as optimal, since no improvement may be found for an objective without degrading another objective value.

In the case of a bi-ojective minimization problem, the Pareto front of the efficient solutions obtained may be easily plotted (see the thick line in objective space from Fig. 1.1).

1.1.3 Ideal, Nadir and Utopian Points

In order to have reference points that may help to discuss the interest of the solutions found, some specific points have been defined in objective function space [49]. These points may represent either feasible or infeasible solutions.

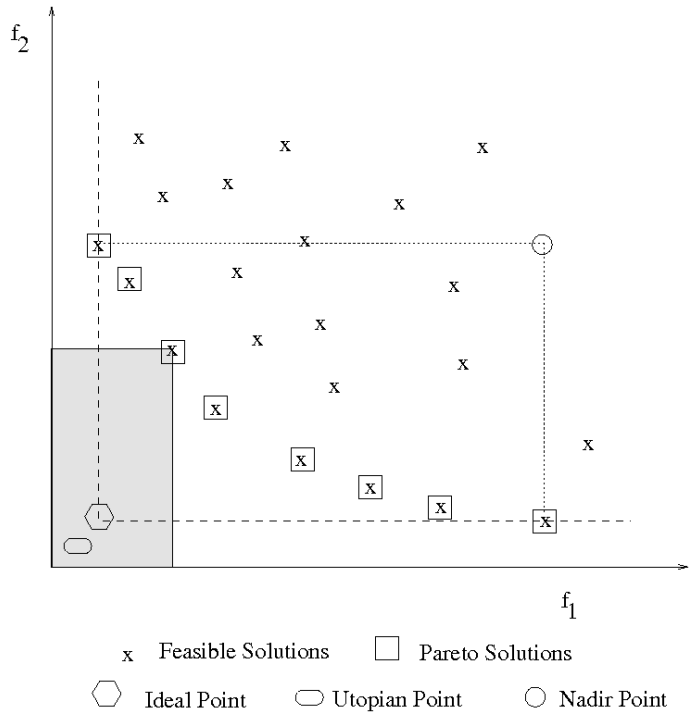

Fig. 1.2. Graphical representation of the ideal, nadir and utopian points.

First, the **ideal point** z^I represents the point that has, for each objective function, the optimal value (obtained by optimizing separately such objective). Formally, it is defined as:

$$z^I \text{such that } \forall i \in [1...n], f_i(z^I) = opt_{x \in \mathcal{D}} f_i(x) \tag{1.2}$$

This point does not correspond to a feasible solution; otherwise, this would indicate that the absence of conflict among the objectives. Should that be the case, optimizing the objectives separately would generate this single solution and no need for Pareto optimality would arise.

From this ideal point may be defined the **utopian point** z^U as follows:

$$z^U = z^I - \epsilon\, U \tag{1.3}$$

where $\epsilon > 0$ and U is the unitary vector ($U = (1, \ldots, 1) \in \Re^n$). This point is, by definition, not feasible.

Finally, the **Nadir point** may be defined in the bi-objective context as:

$$z^N \text{such that } \forall i \in [1...2], f_i(z^N) = opt_{x \in \mathcal{D}/f_j(x)=f_j(z^I)} f_i(x) \text{ with } j \neq i \quad (1.4)$$

This corresponds to affect to each objective of the nadir point the best possible value among solutions optimizing the other criterion.

These points are graphically displayed in Fig. 1.2.

1.2 Decision Making

Solving a multi-objective combinatorial problem leads to the determination of a set of Pareto optimal solutions, and it is often necessary to ask a (human) decision maker to express his/her preferences in order to obtain a single final solution.

Hence, before starting to solve a multi-objective combinatorial optimization problem, a decision must be made regarding the way in which such preferences from the decision maker are to be incorporated. There are three forms of incorporating user's preferences into the search:

- **A priori approaches**: The decision maker gives indications regarding the relative importance of the different criteria before the optimization process takes place. A single optimization is required to obtain the wanted solution. This method may be fast, but the time necessary for the modelling has to be considered. Moreover, the decision maker may not be satisfied with the solution found and may want to do another optimization expressing a different compromise among the objectives.
- **Progressive approaches**: The decision maker participates during the optimization process, expressing his/her preferences, which are then used to guide the search. This approach allows to take into account the preferences of the decision maker in a very accurate manner, but requires a lot of interaction throughout the search.
- **A posteriori approaches**: In this third category of methods, the decision maker is presented a set of (nondominated) solutions generated by an optimization method and he/she has to chooce from them those that he considers the most appropriate. In this case, the preferences do not need to modelled ahead of time (a task that can be very difficult to achieve) but the approaches within this category normally require a lot of computational effort.

Each of these types of approaches have advantages and disadvantages and the choice of one of them in particular depends on the decision maker and on the problem itself. It is worth emphasizing, however, that multi-objective metaheuristics have been traditionally adopted as a posteriori approaches [8].

1.3 Algorithms for Solving MOCOPs

Most MOCOPs are $\mathcal{NP}$-hard. Hence, exact methods can be used only for small-scale problems and approximation methods is the most common choice to deal with large instances. Within such approximation methods we can find both algorithms that produce an approximation with a guaranteed quality [50] and metaheuristics [18, 16, 20].

As far as exact methods are concerned, as indicated before, they are limited to small problems often having only two objectives. The two-phase method proposed by B. Ulungu et J. Teghem is an interesting example [59]. This method has been improved in [52] and in [42] for flowshop problems. Moreover, another method called PPM has been proposed for two [43] and more objectives [44].

Nevertheless, for medium and large size problems, it is necessary to use metaheuristics. Multi-objective metaheuristics can be classified into three classes:

- **Scalar approaches:** Transform the problem into one or several single-objective problem(s). Among them, we have aggregation methods [33], the ϵ-constraint method [32], weighted metrics, goal programming, achievement functions, goal attainment, etc. These methods require *a priori* knowledge of the problem in order to define preferences among objectives, and, most of the time, they produce a single solution per run.
- **Population-based approaches:** They exploit the population adopted by several metaheuristics (e.g., evolutionary algorithms) to combine several scalar search procedures in a single run. The typical example within this group is the *Vector Evaluated Genetic Algorithm* proposed by Schaffer [56], which consists of a simple genetic algorithm that uses as many sub-populations as objectives are in the problem. Each subpopulation selects the best individuals based on a single objective. Then, all the subpopulations are merged and shuffled, and crossover and mutation is applied as usual. The idea is that by recombining individuals that are good in one objective good trade-offs will be generated. Such sort of approach, however, contradicts the notion of Pareto optimality and is rarely adopted these days [8].
- **Pareto-based approaches:** In this case, the selection mechanism incorporates the concept of Pareto optimality. Typical approaches within this class adopt a ranking of solutions based on Pareto optimality originally proposed by David E. Goldberg [29]. Most multi-objective evolutionary algorithms are based on Pareto ranking, although several variations are available: dominance rank (MOGA [23]), dominance depth (NSGA-II [13]), and dominance count (SPEA [66] and SPEA2 [65]).
- **Indicator based approaches:** In this case, instead of using Pareto ranking, a performance assessment measure [67] is adopted to select solutions. Typical methods within this class are the Indicator-based Evolutionary Al-

gorithm (IBEA) [64, 2, 3] and the S metric selection EMOA (SMS-EMOA) [21, 5]).

1.3.1 Nature Inspired Metaheuristics for Solving MOCOPs

In recent years, an overwhelming number of multi-objective metaheuristics designed for solving multi-objective combinatorial optimization problems have been proposed [8]. Next, we will limit ourselves to discuss some of the multi-objective evolutionary algorithms most commonly used nowadays. However, the interested reader should refer to [8, 11] for more information on this topic.

Nondominated Sorting Genetic Algorithm II (NSGA-II)

The Nondominated Sorting Genetic Algorithm II was proposed by Deb et al. [12, 13] and is probably the most commonly adopted multi-objective evolutionary algorithm in the current literature. At each generation, solutions from the current population are ranked into several classes. Individuals mapping to vectors from the first front all belong to the best efficient set; individuals mapping to vectors from the second front all belong to the second best efficient set, and so on. Two values are then assigned to population members. The first one corresponds to the *rank* the corresponding solution belongs to, and represents the quality of the solution in terms of convergence. The second one, called *crowding distance*, consists of estimating the density of solutions surrounding a particular point of the objective space, and represents the quality of the solution in terms of diversity. A solution is said to be better than another one if it has a better rank value, or, in case of equality, if it has a better crowding distance. The selection strategy is a deterministic tournament between two randomly selected solutions. At the replacement step, only the best individuals survive with respect to a predefined population size.

Indicator-Based Evolutionary Algorithm (IBEA)

The Indicator-Based Evolutionary Algorithm was introduced in [64] and is a framework that allows any performance indicator to be incorporated into the selection mechanism of a multi-objective evolutionary algorithm. The main idea behind IBEA is to introduce a total order among solutions by means of a binary quality indicator. Its fitness assignment scheme is based on a pairwise comparison of solutions from the current population with regards to an arbitrary indicator I. To each individual x is assigned a fitness value $F(x)$ measuring the "loss in quality" if x was removed from the current population P, *i.e.* $F(x) = \sum_{x' \in P \setminus \{x\}} (-e^{-I(x',x)/\kappa})$, where $\kappa > 0$ is a user-defined scaling factor. Different indicators can be used for such a purpose, such as the binary additive ϵ-indicator ($I_{\epsilon+}$) as defined in [64] or the hypervolume [63]. Selection for reproduction consists of a binary tournament between randomly

chosen individuals. Selection for replacement consists of iteratively removing the worst solution from the current population until the required population size is reached; fitness information of the remaining individuals is updated each time there is a deletion.

Strength Pareto Evolutionary Algorithm 2 (SPEA2)

The Strength Pareto Evolutionary Algorithm 2 [65] was proposed as an extension of the Strength Pareto Evolutionary Algorithm (SPEA) [66], introduced by the same authors. The main improvements are related to the use of an improved fitness assignment strategy. SPEA2 intrinsically handles an internal archive of fixed size that is used during the selection step to create offspring solutions. Also, an external archive that stores the nondominated solutions generated during the search is adopted. At a given iteration of SPEA2, to each population and archive member x is assigned a strength value $S(x)$ representing the number of solutions it dominates. Then, the fitness value $F(x)$ of solution x is calculated by summing up the strength values of all individuals to which solution x currently dominates. Additionally, a diversity preservation strategy, based on a nearest neighbor technique, is incorporated. The selection step consists of a binary tournament with replacement applied on the internal archive only.

Simple Elitist Evolutionary Algorithm (SEEA)

If evaluating a solution in objective function space is not too much time consuming, computing fitness values and diversity information are generally the most computationally expensive steps of a multi-objective evolutionary algorithm. Based on this observation, Liefooghe et al. [45] proposed a simple search method for which none of these phases is required. The resulting evolutionary algorithm, called *Simple Elitist Evolutionary Algorithm* (SEEA for short), is detailed in Algorithm 1. In SEEA, an archive of potentially efficient solutions is updated at each generation, and the individuals contained in the main population are generated by applying variation operators to randomly chosen archive members. The replacement step is a generational one, *i.e.* the parent population is replaced by the offspring population. Note that the initial population can, for instance, be filled with random solutions. Then, as proposed in [66] (among other authors), the archive is not only used as an external storage, but it is integrated into the optimization process during the selection phase of the multi-objective evolutionary algorithm. The preservation of the nondominated solutions generated during the search is called **elitism**, and its use is of great importance in multi-objective optimization, since it is required in order to guarantee convergence from a theoretical point of view [54, 55, 41]. SEEA is somehow related to other elitist multi-objective evolutionary algorithms such as the Pareto Archived Evolution Strategy (PAES) [38], the Pareto envelope-based Selection Algorithm for multi-objective optimization

(PESA) [10] and the Simple Evolutionary Algorithm for Multi-Objective Optimization (SEAMO) [60]. But, contrary to other approaches, no strategy to preserve diversity or to manage the size of the archive is involved here, as solutions are selected randomly and the archive is unbounded[5]. The biggest advantage of this approach is that the population (or the population size, if solutions are randomly initialized) is the only problem-independent parameter. If non-dominated solutions are relatively close to each other in decision variable space and if the archive is not too small compared to the main population, SEEA may converge to a good approximation of the efficient set requiring a short CPU time. However, in some cases, this method may prematurely converge or may appear inefficient if promising solutions are far from each other.

Algorithm 1 Simple Elitist Evolutionary Algorithm (SEEA)

Input: P Initial population
Output: A Efficient set approximation

Step 1: **Initialization.** $A \leftarrow$ non-dominated individuals of P; $N \leftarrow |P|$; $P' \leftarrow \emptyset$.
Step 2: **Selection.** Repeat until $|P'| = N$: randomly select an individual from A and add it to the offspring population P'.
Step 3: **Variation.** Apply crossover and mutation operators to individuals of the offspring population P'.
Step 4: **Replacement.** $P \leftarrow P'$; $P' \leftarrow \emptyset$.
Step 5: **Elitism.** $A \leftarrow$ non-dominated individuals of $A \cup P$.
Step 6: **Termination.** If a stopping criteria is satisfied return A, else go to Step 2.

In general, it can be seen that modern multi-objective evolutionary algorithms require three main components: (1) a mechanism to select solutions that are nondominated in a Pareto sense, (2) a mechanism to maintain diversity in order to promote convergence to the entire Pareto front and (3) an elitist mechanism that ensures that the global nondominated solutions are preserved throughout the search.

Finally, it is worth noting the use of other metaheuristics (different from evolutionary algorithms) in multi-objective optimization (particularly when dealing with combinatorial optimization problems). Approaches such as simulated annealing [46], tabu search [26], scatter search [4], ant colony optimization [27], particle swarm optimization [53], differential evolution [48] and artificial immune systems [25] have all been used to solve multi-objective optimization problems. The use of hybrid approaches (e.g., multi-objective memetic

[5] Note that in the continuous case, due to limited computing and memory resources, the archive is usually bounded.

algorithms [36, 28]) has also become popular in the last few years, since the use of local search is quite evident when dealing with multi-objective combinatorial optimization problems. However, the discussion of these approaches is beyond the scope of this chapter. The interested reader should refer to [8, 20] for more information on this topic.

1.4 Performance Assessment

An important task when validating a multi-objective metaheuristic is to be able to assess its performance with respect to other approaches. In order to assess performance of two different methods, two sets of solutions have to be compared. This is not an easy task, since the two algorithms compared generate sets of solutions rather than a single value, but several performance indicators have been proposed in order to perform this task (for more details on performance assessment of multi-objective metaheuristics and some of their caveats, see [31, 35, 67]).

Performance indicators can be classified according to different features [58]:

- **Unary/Binary Indicators:** Binary indicators allow us to compare directly two approximations of the true Pareto front, whereas unary indicators assign to each approximation of the Pareto optimal set a scalar value.
- **Requirement of the True Pareto front:** Some performance indicators require that the user provides the true Pareto front of the problem, which, in many cases, is unknown.
- **Need of Extra Information:** Some quality indicators require the definition of several values that may be difficult to obtain in some cases (e.g., the ideal vector, the Nadir point, a reference solution set, etc.).

A lot of performance indicators are currently available, but their use is not only standardized, but it is debatable in several cases. Usually, more than one performance indicator is adopted in order to assess performance of a multi-objective metaheuristic, and different indicator exist for different goals:

- **Convergence-Based Indicators:** Provide the closeness of the obtained approximation with respect to the true Pareto front. Examples of this sort of indicator are: contribution [47], generational distance [61, 62], ϵ-indicator [67].
- **Diversity-Based Indicators:** Provide information about the uniformity of the distribution of the obtained solutions along the Pareto front[6]. Examples of this sort of indicator are: spacing [57, 62], spread [11, 13] and entropy [22].

[6] Note that diversity-based indicators can be applied both in decision variable space and objective variable space, but the latter is the most commonly adopted.

- **Hybrid Indicators:** They attempt to combine, in a single value, the performance on both, convergence and diversity. Examples of this class of indicators are: hypervolume [63], and the R-metrics [31, 35].

In the following sections, A and B will denote the approximation set found by a multi-objective metaheuristic, $Z_N^\star$ will denote the Pareto optimal set, assuming it is known or it was produced from the union of all the previously obtained approximations.

1.4.1 Convergence-Based Indicators

These indicators evaluate the closeness of the obtained approximation with respect to the true Pareto front.

Contribution

The contribution indicator [62, 47] is a cardinality-based measure. Let A and B be two Pareto approximation sets, $Z_N^\star$ be the set of Pareto optimal solutions from $A \cup B$, AB be the set of solutions in $A \cap B$, W be the number of solutions from A that dominate some solutions from B, and N be the number of incomparable solutions from A. The contribution is defined as follows:

$$C(A/B) = \frac{\frac{|AB|}{2} + |W| + |N|}{|Z_N^\star|} \tag{1.5}$$

Let us remark that $C(A/B) + C(B/A) = 1$.

Generational Distance

This performance measure computes the average distance from the approximation A (obtained by a metaheuristic) to $Z_N^\star$ (i.e., the true Pareto front of the problem). It is defined as follows:

$$GD(A, Z_N^\star) = \frac{(\sum_{i=1}^{|A|} d_i^p)^{1/p}}{|A|} \tag{1.6}$$

For $p = 2$, d is the euclidean distance in objective function space between solution $i \in A$ and the nearest member of $Z_N^\star$.

ϵ-Indicator

The unary additive ϵ-indicator ($\mathrm{I}_{\epsilon+}^1$) gives the minimum factor by which an approximation A has to be translated in objective function space in order to weakly dominate[7] the reference set $Z_N^\star$ where N is the number of objectives.

[7] A solution $y = (y_1, y_2, \ldots, y_k)$ **weakly dominates** a solution $z = (z_1, z_2, \ldots, z_k)$, in a minimization context, iff $\forall i \in [1 \ldots n]$, $f_i(y) < f_i(z)$.

$I^1_{\epsilon+}$ can be defined as follows:

$$I^1_{\epsilon+}(A) = I_{\epsilon+}(A, Z^\star_N) \ , \tag{1.7}$$

where

$$I_{\epsilon+}(A, B) = \min_{\epsilon}\{\forall z \in B, \exists z' \in A : z'_i - \epsilon \leq z_i, \forall 1 \leq i \leq n\} \ . \tag{1.8}$$

1.4.2 Diversity-Based Indicators

Spacing

Schott [57] proposed a performance measure that estimates the diversity of the Pareto front obtained by a metaheuristic by computing the relative distance measure between consecutive solutions as follows:

$$S = \sqrt{\frac{1}{|A|}\sum_{i=1}^{|A|}(d_i - \bar{d})^2} \tag{1.9}$$

where $d_i = min_{k \in A \neq i} \sum_{m=1}^{M} |f^i_m - f^k_m|$ and $\bar{d}$ is the mean value of the distance.

Spread (Δ) Indicator

Deb [11] proposed the spread indicator to estimate the extent of the spread of the obtained Pareto front. It is formally defined as follows:

$$\Delta = \frac{\sum_{m=1}^{M} d^e_m + \sum_{i=1}^{|A|} |d_i - \bar{d}|}{\sum_{m=1} M d^e_m + |Q|d} \tag{1.10}$$

where d_i is a neighboring distance measure, $\bar{d}$ is the mean value of this distance measure. The parameter d^e_m is the distance between the extreme solutions of A and $Z^\star_N$ corresponding to the m-th objective function.

1.4.3 Hybrid Indicators

As indicated before, these indicators evaluate both closeness to the true Pareto front and spread along it. Next, we will briefly discuss three of them.

Hypervolume

The unary Hypervolume (HV [63]) reflects the volume in objective function space covered by the members $p_i(i = 1, \ldots, N)$ of a nondominated set ND of solutions. It is defined relative to an "anti-optimal" reference point Z_{ref}, which can be the worst possible point in objective function space. This point is usually not known and has to be chosen carefully [35]. It is mostly approximated by the worst objective function values in each dimension from any of the calculated Pareto fronts during the execution of the algorithm. Then, the HV is the union of the hypercuboids (bounded by Z_{ref}) in the Lebesgue measure Λ which are weakly dominated by the vector p_i:

$$HV(ND, Z_{ref}) = \Lambda(\{\cup h(p_i) | p_i \in ND, i = 1, ..., N\}) \tag{1.11}$$

and

$$h(p_i) = [p_{i1}, Z_{ref_1}] \times [p_{i2}, Z_{ref_2}] \times \ldots \times [p_{iM}, Z_{ref_M}] \tag{1.12}$$

Thus, the larger the hypervolume is, the wider is the range of Pareto optimal solutions. Therefore, hypervolume has to be maximized. The choice of the reference point affects the ordering of the nondominated sets and often the point defined as $(1.05 \times z_1^{max}, 1.05 \times z_2^{max})$ is chosen to be the reference point.

The hypervolume difference indicator (I_H^-) is a binary indicator that computes the portion of the objective function space that is weakly dominated by $Z_N^\star$ and not by ND. The more this measure is close to 0, the better is the approximation ND.

R-Metrics

There are three R_R indicators that are based on a set of utility functions u [31]. The R_1 indicator [31] is based on calculating the probability that the approximation A is better than B over an entire set of utility functions. R_1^R is R_1 when it is used with a reference set—i.e., as a reference indicator. This performance measure does then induce a total ordering on the approximation set:

$$R_1(A, B, U, p) = \int_{u \in U} C(A, R, u) p(u) du \tag{1.13}$$

where U is a set of utility functions, and A and B are two approximations of the Pareto set.

$$C(A, B, u) = \begin{cases} 1 & \text{if } u*(A) > u*(B) \\ 1/2 & \text{if } u*(A) = u*(B) \\ 0 & \text{if } u*(A) < u*(B) \end{cases} \tag{1.14}$$

We can notice that $R1(A, B, u, p) = 1 - R1(B, A, u, p)$. If $R1(A, B, u, p) > 1/2$ then A is better than B.

R_2 and R_3 are defined as follows:

$$R_2(A, B) = \frac{\sum_{\lambda \in \Lambda} u(\lambda, A) - u(\lambda, B)}{|\Lambda|} \tag{1.15}$$

and

$$R_3(A, B) = \frac{\sum_{\lambda \in \Lambda} [u(\lambda, B) - u(\lambda, A)]/u(\lambda, B)}{|\Lambda|} \tag{1.16}$$

Attainment Surfaces

An attainment surface represent a boundary which divides the objective function space into two parts: one that was attained by the objective function value vectors returned by the algorithm and another that was not [24]. The computation of several of these surfaces through sets of objective function value vectors obtained by several runs of the considered algorithm allows to have an estimation of the quantiles of the attainment surfaces, i.e., the boundaries in the objective space which are likely to be attained with the same probability. A formal definition can be found in [30].

Fonseca and Fleming [24] performed an empirical estimation of these distributions by considering arbitrary straight lines intersecting the attainment surfaces. They proposed to test inequality of the attainment surface by performing non-parametric statistical tests. This method could then be used to assess performance.

Knowles and Corne [39, 34] extended this approach by adopting the one-sided Mann-Whitney test for comparing a pair of algorithms at the intersections of lines where the test indicated statistical differences. This allowed them to say which algorithm was performing better and to identify the differences to points or regions of the objective function space.

1.4.4 Statistical Validation

The use of statistical analyses to assess performance of metaheuristics has become more important in recent years [1]. Such statistical validation has also been adopted in multi-objective optimization (see for example [37]) and has become more widespread in the last few years.

In order to perform a statistical validation of our results, the first step is to collect descriptive statistics on each performance measure adopted, such as the mean, the variance and the median. A common way to present these descriptive statistics is to produce a box-plot (see Fig. 1.3) and to present side-by-side box-plots of the different algorithms to be compared. There is

software available to plot them directly from data (for example, R, Excel, and Matlab, among others).

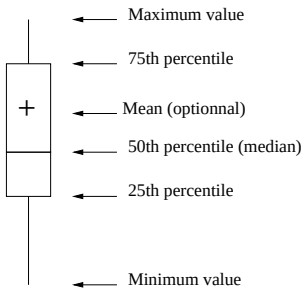

Fig. 1.3. An example of boxplot.

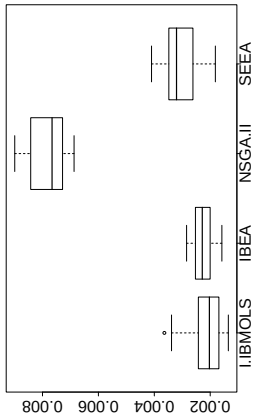

Fig. 1.4. An example of side-by-side boxplots.

Then, several statistical tests can be carried out, depending on our goal:

- **Comparison of Dominance Ranks:** For this, we can adopt the Mann-Whitney test [9] (this is a non-parametric test for comparing two groups of data) or the Kruskal-Wallis test [9] (this is a non-parametric test for comparing more than 2 groups of data).
- **Comparison of Indicators:** This can be done in two possible ways:
  - By using a single indicator: In this case, the Mann-Whitney rank sum test or Fisher's permutation test can be adopted [15]
  - By using a set of indicators: In this case, the Wilcoxon test (a non-parametric statistical hypothesis test) or Fisher's permutation test could be used.

1.4.5 Public-Domain Software for Validation

There is also some public-domain software available to validate the performance of multi-objective metaheuristics. Next, we briefly discuss two of them.

Guimoo

The *Graphical User Interface for Multi-Objective Optimization* (Guimoo) is a free software dedicated to the analysis of results in multi-objective optimization and it is available at: `http://guimoo.gforge.inria.fr/`.

Its main features include the following:

- The on-line visualization of approximate Pareto frontiers. Such information can be used by an expert to build more efficient metaheuristics. A Pareto frontier may be characterized by its (dis)continuity, (dis)convexity, (multi)modality, etc.
- Some performance measures for quantitative and qualitative performance evaluation (i.e., S-metric, R-metrics, contribution, entropy, generational distance, spacing, size of the dominated space, coverage of two sets and coverage difference).

PISA

The *Platform and Programming Language Independent Interface for Search Algorithms* (PISA) is available at `http://www.tik.ethz.ch/~sop/pisa/`.

PISA includes a set of statistical tools that allow to assess and compare the performance of several multi-objective evolutionary algorithms [6, 40]. It includes indicator modules using different quality indicators such as the ϵ-indicator, the R-metrics and hypervolume.

1.5 Conclusions

The aim of this chapter was to present the main definitions and concepts related to multi-objective combinatorial optimization using metaheuristics, such that this information can be used to understand the rest of the book.

The discussion has included multi-objective optimization algorithms, incorporation of user's preferences, performance measures and performance assessment, and the use of statistical tools (including public-domain software) to assess our obtained results.

References

1. Thomas Bartz-Beielstein. *Experimental Research in Evolutionary Computation. The New Experimentalism*. Springer, Berlin/Heidelberg, 2006. ISBN 978-3-540-32026-5.
2. Matthieu Basseur and Eckart Zitzler. Handling Uncertainty in Indicator-Based Multiobjective Optimization. *International Journal of Computational Intelligence Research*, 2(3):255–272, 2006.
3. Matthieu Basseur and Eckart Zitzler. A Preliminary Study on Handling Uncertainty in Indicator-Based Multiobjective Optimization. In Franz Rothlauf et al., editor, *Applications of Evolutionary Computing. EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC*, pages 727–739, Budapest, Hungary, April 2006. Springer, Lecture Notes in Computer Science Vol. 3907.

4. Ricardo P. Beausoleil. "MOSS" multiobjective scatter search applied to non-linear multiple criteria optimization. *European Journal of Operational Research*, 169(2):426–449, March 2006.
5. 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.
6. Stefan Bleuler, Marco Laumanns, Lothar Thiele, and Eckart Zitzler. PISA—A Platform and Programming Language Independent Interface for Search Algorithms. 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 494–508, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.
7. Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.
8. 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.
9. W.J. Conover. *Practical Nonparametric Statistics*. Wiley, USA, third edition, 1998. ISBN 978-0-471-16068-7.
10. David W. Corne, Joshua D. Knowles, and Martin J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
11. Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.
12. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
13. Kalyanmoy Deb, 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.
14. Francis Ysidro Edgeworth. *Mathematical Psychics*. P. Keagan, London, England, 1881.
15. Bradley Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994. ISBN 978-0-412-04231-7.
16. Matthias Ehrgott. Approximation algorithms for combinatorial multicriteria optimization problems. *International Transactions in Operational Research*, 7:5–31, 2000.
17. Matthias Ehrgott. *Multicriteria Optimization*. Springer, Berlin, second edition, 2005. ISBN 3-540-21398-8.
18. Matthias Ehrgott and Xavier Gandibleux. A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization. *OR Spektrum*, 22:425–460, 2000.

19. Matthias Ehrgott and Xavier Gandibleux. Approximative Solution Methods for Multiobjective Combinatorial Optimization. *Top*, 12(1):1–89, June 2004.
20. Matthias Ehrgott and Xavier Gandibleux. Hybrid Metaheuristics for Multiobjective Combinatorial Optimization. In Christian Blum, María J. Blesa Aguilera, Andrea Roli, and Michael Sampels, editors, *Hybrid Metaheuristics*, pages 221–259. Springer. Studies in Computational Intelligence Vol. 114, 2008.
21. Michael Emmerich, Nicola Beume, and Boris Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 62–76, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.
22. Ali Farhang-Mehr and Shapour Azarm. Diversity Assessment of Pareto Optimal Solution Sets: An Entropy Approach. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 723–728, Piscataway, New Jersey, May 2002. IEEE Service Center.
23. 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.
24. Carlos M. Fonseca and Peter J. Fleming. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. 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 584–593, Berlin, Germany, September 1996. Springer-Verlag.
25. Fabio Freschi, Carlos A. Coello Coello, and Maurizio Repetto. Multiobjective Optimization and Artificial Immune Systems: A Review. In Hongwei Mo, editor, *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies*, pages 1–21. Medical Information Science Reference, Hershey, New York, 2009. ISBN 978-1-60566-310-4.
26. Xavier Gandibleux and Arnaud Freville. Tabu Search Based Procedure for Solving the 0-1 Multi-Objective Knapsack Problem: The Two Objectives Case. *Journal of Heuristics*, 6(3):361–383, August 2000.
27. C. García-Martínez, O. Cordón, and F. Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bicriteria TSP. *European Journal of Operational Research*, 180(1):116–148, July 2007.
28. 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.
29. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
30. Viviane Grunert da Fonseca, Carlos M. Fonseca, and Andreia O. Hall. Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 213–225. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

31. Michael Pilegaard Hansen. *Metaheuristics for multiple objective combinatorial optimization*. PhD thesis, Institute of Mathematical Modelling, Technical University of Denmark, March 1998.
32. Alain Hertz, Brigitte Jaumard, C.C. Ribeiro, and W.P. Formosinho Filho. A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives. *RAIRO/Operations Research*, 28(3):303–328, 1994.
33. Hisao Ishibuchi and Tadahiko Murata. Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, 28(3):392–403, August 1998.
34. Joshua Knowles. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *Fifth International Conference on Intelligent Systems Design and Applications (ISDA'2005)*, pages 552–557. IEEE, 2005.
35. Joshua Knowles and David Corne. On Metrics for Comparing Nondominated Sets. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716, Piscataway, New Jersey, May 2002. IEEE Service Center.
36. Joshua Knowles and David Corne. Memetic Algorithms for Multiobjective Optimization: Issues, Methods and Prospects. In William E. Hart, N. Krasnogor, and J.E. Smith, editors, *Recent Advances in Memetic Algorithms*, pages 313–352. Springer. Studies in Fuzziness and Soft Computing, Vol. 166, 2005.
37. Joshua Knowles, Lothar Thiele, and Eckart Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, February 2006. revised version.
38. Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
39. Joshua D. Knowles, David W. Corne, and Martin J. Oates. On the Assessment of Multiobjective Approaches to the Adaptive Distributed Database Management Problem. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 869–878, Berlin, September 2000. Springer.
40. Simon Künzli, Stefan Bleuler, Lothar Thiele, and Eckart Zitzler. A Computer Engineering Benchmark Application for Multiobjective Optimizers. In Carlos A. Coello Coello and Gary B. Lamont, editors, *Applications of Multi-Objective Evolutionary Algorithms*, pages 269–294. World Scientific, Singapore, 2004.
41. Marco Laumanns, Eckart Zitzler, and Lothar Thiele. On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 181–196. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
42. J. Lemesre, C. Dhaenens, and E-G. Talbi. An exact parallel method for a bi-objective permutation flowshop problem. *European Journal of Operational Research*, 177(3):1641–1655, 2007.

43. J. Lemesre, C. Dhaenens, and E-G. Talbi. Parallel partitioning method (PPM): A new exact method to solve bi-objective problems. *Computers & Operations Research*, 34(8):2450–2462, 2007.
44. J. Lemesre, C. Dhaenens, and E-G. Talbi. K-PPM: A New Exact Method to solve Multi-Objective Combinatorial Optimization Problems. *European Journal of Operational Research*, 200(1):45–53, January 2010.
45. Arnaud Liefooghe, Laetitia Jourdan, Matthieu Basseur, El-Ghazali Talbi, and Edmund K. Burke. Metaheuristics for the Bi-objective Ring Star Problem. In Jano van Hemert and Carlos Cotta, editors, *Evolutionary Computation in Combinatorial Optimization, 8th European Conference, EvoCOP 2008*, pages 206–217, Naples, Italy, March 2008. Springer. Lecture Notes in Computer Science Vol. 4972.
46. Panta Lučić and Dušan Teodorović. Simulated annealing for the multi-objective aircrew rostering problem. *Transportation Research Part A*, 33:19–45, 1999.
47. Herve Meunier, El-Ghazali Talbi, and Philippe Reininger. A Multiobjective Genetic Algorithm for Radio Network Optimization. In *2000 Congress on Evolutionary Computation*, volume 1, pages 317–324, Piscataway, New Jersey, July 2000. IEEE Service Center.
48. 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.
49. Kaisa M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.
50. Christos Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization. Algorithms and Complexity*. Dover Publications, Inc., New York, USA, 1998. ISBN 0-486-40258-4.
51. Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.
52. A. Przybylski, X. Gandibleux, and M. Ehrgott. Seek and cut algorithm computing minimal and maximal complete efficient solution sets for the biobjective assignment problem. In *6th International Conference on Multi-Objective Programming and Goal Programming conf (MOPGP'04)*, Tunisia, April 2004.
53. 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.
54. Günter Rudolph. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 511–516, Piscataway, New Jersey, 1998. IEEE Press.
55. Günter Rudolph and Alexandru Agapie. Convergence Properties of Some Multi-Objective Evolutionary Algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, volume 2, pages 1010–1016, Piscataway, New Jersey, July 2000. IEEE Press.
56. 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.

57. Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.
58. El-Ghazali Talbi. *Metaheuristics. From Design to Implementation*. Wiley, USA, 2009. ISBN 978-0-470-27858-1.
59. E.L. Ulungu and J. Teghem. The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundation of Computing and Decision Sciences*, 20(2):149–165, 1995.
60. Christine L. Valenzuela. A Simple Evolutionary Algorithm for Multi-Objective Optimization (SEAMO). In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 717–722, Piscataway, New Jersey, May 2002. IEEE Service Center.
61. David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.
62. David A. Van Veldhuizen and Gary B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. In *2000 Congress on Evolutionary Computation*, volume 1, pages 204–211, Piscataway, New Jersey, July 2000. IEEE Service Center.
63. Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.
64. Eckart Zitzler and Simon Künzli. Indicator-based Selection in Multiobjective Search. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.
65. 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.
66. 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.
67. 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.