Chapter 10
On Multi-Objective Evolutionary Algorithms

Dalila B.M.M. Fontes and António Gaspar-Cunha

Abstract In this chapter Multi-Objective Evolutionary Algorithms (MOEAs) are introduced and some details discussed. A presentation of some of the concepts in which this type of algorithms are based on is given. Then, a summary of the main algorithms behind these approaches and their applications is provided, together with a brief discussion including their advantages and disadvantages, degree of applicability, and some known applications. Finally, future trends in this area and some possible paths for future research are pointed out.

10.1 Introduction

Most real-world optimization problems are multi-objective since they require the simultaneous satisfaction of several objectives. The most usual approach to deal with the multi-objective nature of these problems consists on congregating the various individual objectives into a unique function in order to form a single-objective optimization problem. In this case, it is necessary to define a priori a compromise between the objectives considered. If the relative importance of the criteria is changed a new optimization run needs to be carried out.

Dalila B.M.M. Fontes
LIAAD - INESC Porto L.A. and Faculdade de Economia, Universidade do Porto
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal.
e-mail: fontes@fep.up.pt

António Gaspar-Cunha
IPC/I3N - Institute of Polymers and Composites
Department of Polymer Engineering, University of Minho
Campus de Azurém, 4800-058 Guimarães, Portugal.
e-mail: agc@dep.uminho.pt

C. Zopounidis and P.M. Pardalos (eds.), *Handbook of Multicriteria Analysis*, Applied Optimization 103, DOI 10.1007/978-3-540-92828-7_10,

Other possible approach takes advantages of the fact that Evolutionary Algorithms (EAs) work with a population of points processed in each iteration, yielding a set of non-dominated vectors, designated as Pareto optimal solutions. In this case, all the objectives are optimized simultaneously.

EAs mimic the process of natural evolution where an analogy between the mechanisms of natural selection and a learning (or optimization) process is made through the application of certain heuristic techniques [38]. These techniques can be classified into four main categories, Genetic Algorithms [42], Evolution Strategies [64], Evolutionary Programming [30] and Genetic Programming [52]. However, this classification is due to historical developments rather than to major functioning differences, since the basis of these techniques is essentially the same.

After using evolutionary techniques for single-objective optimization during more than two decades, the incorporation of more than one objective has finally become a popular area of research. As a consequence, many new evolutionary-based approaches and variations of existing techniques have recently been published in the literature [14]. The large number of applications [4, 8] and the continuously growing interest in this field are due to several advantages of EAs:

1. In-depth mathematical understanding of the problems to which they are applied to is not required;
2. Some of the solutions obtained by the EAs were previously out of range of the solutions obtained by other methods;
3. EAs can be applied to problems that cannot be solved by analytical mathematical techniques or that involve so many variables that other methods would take too long to solve them.
4. EAs can be applied to a high range of problems since they are robust;
5. EAs are relatively cheap and simple to implement;
6. It is easy to combine EAs with other techniques, such as local search and other heuristics (hybridization);
7. EAs are extremely adaptable due to the fact that the evolutionary mechanism is separate from the problem representation. Therefore, they can be transferred from problem to problem, that is, they are modular and portable;
8. EAs allow for the use of arbitrary constraints, simultaneous multiple objectives and the mixing of continuous and discrete parameters;
9. In addition, EAS are intrinsically parallel, i.e., they can be easily adapted to parallel computation.

Regarding multi-objective optimization problems they also have the advantage of working with a population of solutions rather than with a single solution. The ability to simultaneously search different regions of a solution space not only makes it possible to find a diverse set of solutions but also to address problems with non-convex, discontinuous, and multi-modal solutions spaces. These features enable the creation of Pareto fronts representing the trade-off between the criteria.

Multi-Objective Optimization (MOO) is undoubtedly a very important research topic both for scientists and practitioners, not only because of the multi-objective nature of most real-world problems but also because there are still many open questions in this area. The conflict of objectives entailed in MOO places the issue of compromise in a central position. Edgeworth [26] and Pareto [61] captured this notion mathematically in the criterion widely known as Pareto optimality [14]. Solutions belonging to the Pareto optimal set of a particular MOO problem perform better in one or more objectives and worst in the other objectives. In other words, solutions in the Pareto optimal set display different trade-offs.

Since, usually, no single solution optimizes simultaneously all objectives, decision making based on subjective human preference is an inherent aspect in solving MOO problems. Only a single solution out of the Pareto optimal set is required. Preference is the basis of tie-breaking between solutions in the Pareto optimal set. In the areas of Multi-criteria Decision Making (MCDM) and multi-objective Decision Aid (MCDA) a variety of frameworks capturing the decision maker(s) preferences have been proposed. Multi-attribute utility theory (MAUT) [49], Analytic Hierarchy Process [66] and outranking synthesis [78] are some of the most popular preference specification schemes. The multiplicity of preference articulation schemes highlights the complexity of human preference. Many approaches to this type of problems have been suggested, going all the way from naively combining objectives into one, to the use of game theory to coordinate the relative importance of each objective.

This chapter emphasizes the importance of looking at previous work in operations research, not only to get a good understanding of some of the existing techniques, but also to motivate the development of new EA-based approaches. Finally, some real applications are also described to provide the reader with a more complete idea of the range of applicability and the underlying motivation of this technique.

10.2 Multi-Objective Optimization

10.2.1 Definitions and Concepts

As soon as there are many (possibly conflicting) objectives to be optimized simultaneously, there is no longer a single optimal solution but rather a whole set of possible solutions of equivalent quality.

The general Multi-Objective Optimization Problem (MOOP) may be stated as finding the value for a set of n decision variables which must satisfy some constraints (J inequalities and K equalities) such that the M objective functions are optimized and can be modeled as follows:

$$
\begin{array}{llll}
(\mathscr{P}) & \text{Optimize} & f_m(x_i) & \text{for all } m = 1, 2, \dots, M \\
 & \text{subject to} & & \\
 & & g_j(x_i) \geq 0 & \text{for all } j = 1, 2, \dots, J \\
 & & h_k(x_i) = 0 & \text{for all } k = 1, 2, \dots, K
\end{array}
$$

where $x_i = \{x_1, x_2, \dots, x_n\}$ is the vector of decision variables.

In general the objectives are non-commensurable and in conflict with each other. Therefore, optimizing means finding a solution having values, for all objective functions, which satisfy the decision maker.

Generally speaking, in MOOPs two different solutions are related to each other in two possible ways: either one dominates the other or none of them is dominated. The set of Pareto solutions consists of good solutions, where none can be said to be better than the others, that is, the set of nondominated solutions. This concept is illustrated in Figure 10.1, where solutions 1, 2, 3 and 4 are non-dominated and constitute the Pareto front.

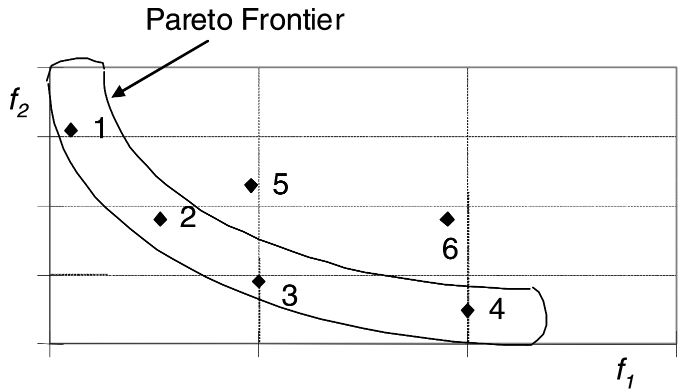

Fig. 10.1 Concept of non-dominance.

The optimal solution of a MOOP is not a single solution but a set of solutions composed by all the potential solutions such that the components of the objectives vector cannot be simultaneously improved. These solutions are known as Pareto optimal solutions, i.e., the set of non-dominated solutions. A solution is optimal when it is non-dominated by all other feasible solutions. In practice, it is generally impossible to know the actual optimal set and the corresponding Pareto optimal front, but, instead the optimization algorithms find an approximation to this set. The above mentioned concepts can be formally defined as follows:

Pareto Dominance:

Given the vector of objective functions $f_m = (f_1, \ldots, f_M)$ is said that candidate x^1 dominates x^2 (for minimizing), written as $x^1 \preceq x^2$, if

$$\begin{aligned} &f_m(x^1) \leq f_m(x^2), \ \ \forall m \in \{1, \ldots, M\} \quad \text{and} \\ &\exists m \in \{1, \ldots, M\} : f_m(x^1) < f_m(x^2). \end{aligned} \tag{10.1}$$

Pareto Optimality:
For a MOP, a given solution x^* is Pareto optimal if and only if there is no vector $x \in \mathscr{F}$ ($\mathscr{F}$ is the set of feasible candidate solutions), so that

$$\begin{aligned} &f_m(x) \leq f_m(x^*), \ \ \forall m \in \{1, \ldots, M\} \quad \text{and} \\ &f_m(x) < f_m(x^*) \ \ \text{for at least one objective function.} \end{aligned} \tag{10.2}$$

Pareto Optimal Set:
For a MOP, the Pareto Optimal Set ($\mathscr{P}^*$) is defined as

$$\mathscr{P}^* := \{x \in \Omega | \neg \exists x' \in F(x\prime) \preceq f(x)\}. \tag{10.3}$$

Pareto Front:
For a MOP and Pareto Optimal Set ($\mathscr{P}^*$), the Pareto Front ($\mathscr{P}\mathscr{F}^*$) is defined as

$$\mathscr{P}\mathscr{F}^* := \{f_m(x) = (f_1(x), f_2(x), \cdots, f_M(x) | x \in \mathscr{P}\}. \tag{10.4}$$

In extending the ideas of single-objective EAs to multi-objective cases, three major problems must be addressed (Figure 10.2).

1. How to accomplish **fitness assignment** and selection in order to guide the search towards the Pareto optimal set;
2. How to maintain a **diverse population** in order to prevent premature convergence and achieve a well distributed, wide spread trade-off front;
3. How to prevent, during the successive generations, that some good solutions are lost.

It should be noticed that the objective function itself no longer qualifies as fitness function since it is a vector of values and not a scalar value. Different approaches to relate the fitness function to the objective functions are discussed in the following section, further details can be found, for example, in [14, 11].

To maintain good diversity of the population it is necessary to have a density estimation operator, such as, for example, niching [16]. It consists basically in counting the number of neighborhoods around each solution in order to deteriorate the fitness of the different individuals, i.e., the fitness decreases for the individuals with more neighbors.

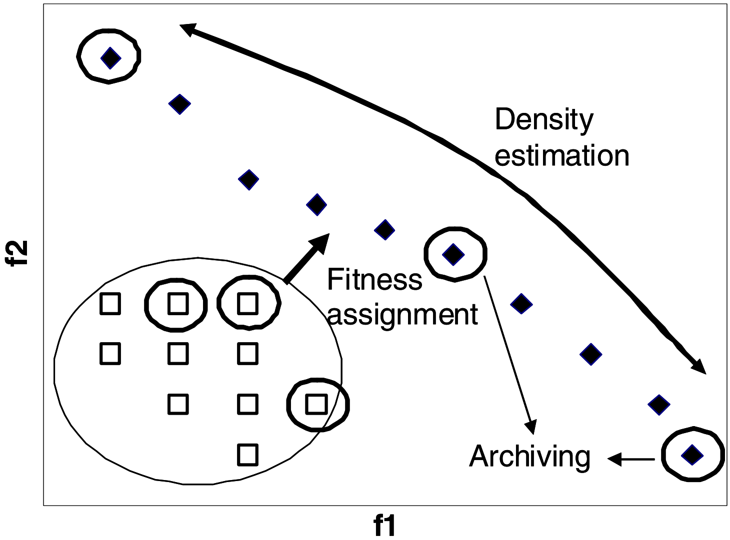

Fig. 10.2 Basic functions of a MOEA.

The third problem is usually solved by keeping the best solutions found so far in an archive in order to ensure that good individuals do not get lost, by mutation or recombination.

10.2.2 Addressing Multi-Objectives

Since there is no accepted definition of optimum as in single-objective optimization problems, it is difficult to compare reported results, as normally, the decision about the best answer corresponds to the so-called (human) decision maker. The decision maker preference for a particular solution is vague, base on perceptive information, and highly dependent on the application context. The vagueness and context-dependence of the decision makers preference structure have lead to the development of various mathematical models and techniques.

The approaches can be divided into two main categories. One that solves a single-objective problem, achieved by combining the objectives into a single-objective function, and another that searches for the Pareto optimal solutions set. In the former case, determination of a single-objective is possible with methods such as utility theory, weighted sum, etc., but the problem lies in the proper selection of the weights or utility functions to characterize the decision maker's preferences. Following the classification of Veldhuizen [77], Multi-objective Evolutionary Algorithms (MOEAs) may be a priori, interactive, or a posteriori algorithms based on the treatment of preference. A priori MOEAs involve preference specification prior to the optimization stage, and are traditionally implemented by aggregating objectives into a single fitness function with parameters reflecting the preference of the decision maker(s). Interactive MOEAs allow decision maker(s) to alter parameters during the search, effectively influencing the direction of the search. A posteriori MOEAs find the set of Pareto optimal solutions and relegate decision making based on human preference to a later stage.

In **a priori** algorithms, the decision maker states the preferences, which are then incorporated into the objective function through aggregation, prior to the optimization. This new formulation is then incorporated in the fitness function computation and cannot be changed throughout the optimization process. Aggregation of the objectives can be made in lexicographic order, that is the objectives are optimized in their order of the importance, or by linear/nonlinear combination of the objectives. In the latter case a single-objective function, reflecting the decision maker preferences, is obtained. This is the simpler approach to MOO, therefore a good choice if the preferences can be captured by the mathematical model and no practical computational difficulties arise. However, this is rarely the case, since often the non-commensurability of objectives makes it very hard, if not impossible to model them in a priori preference specification. In addition, this type of approach requires deep knowledge of the problem in hands, which usually is not possible. Furthermore, practical computational difficulties may also arise due to the non-convex nature of the Pareto front introduced through the combination of the objectives. The main drawback of this type of approach is that scaling amongst objectives is needed and small perturbations in the weights can lead to quite different solutions. In addition, the optimization method devised would return a single solution rather than a set of solutions that can be examined for trade-offs. See Figure 10.3 for an illustration of this approach. For a recent review on preference incorporation in multi-objective

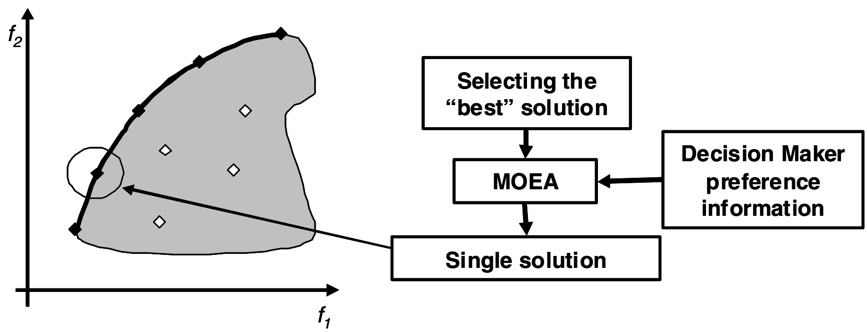

Fig. 10.3 Decision making in MOOP addressing preferences a priori.

evolutionary algorithms the reader is referred to the work by Abass and Sarker [63].

Interactive algorithms tend to be the most adequate approaches to MOO problems since they involve the decision maker in the optimization process. For this type of approach the decision maker only needs to specify a few parameters a priori. As the search progresses and more information on the problem becomes available, the decision maker can make better judgments and therefore parameter specifications. However, interactive approaches require intensive collaboration with the decision maker and quite often become a challenge due to the existence of more than one decision maker. Fonseca and Fleming [31] proposed the incorporation of an expert system to "replace" the decision maker. Nevertheless, the construction of an expert system still requires extensive problem knowledge and, as it is widely recognized, its success is highly dependent on the application context.

A posteriori algorithms tend to be the most popular since its application allows for independent optimization and decision making processes. The optimization and decision maker issues are separated by leaving the latter ones to a post-optimization stage. An illustration of such approach is provided in Figure 10.4. In these ap-

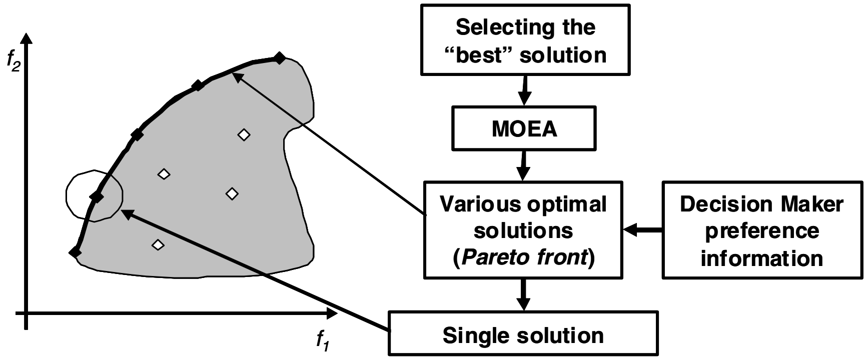

Fig. 10.4 Decision making in MOOP addressing preferences a posteriori.

proaches, the aim of the optimization process is, therefore, to find a set of well-distributed solutions along the Pareto front. Some challenging computational issues are avoided by only looking for such a set of solutions and leaving the choice of the preferred solution to the decision maker. However, ensuring that these solutions represent a wide range of trade-offs may be computationally expensive. Population-wide Pareto ranking, archiving strategy, and diversity preservation measures are features commonly found in the MOEAs which are computationally expensive. Efforts have been introduced to cut down on the computational burden, see ,e.g., [46]. In this case, the algorithm seeks to find the entire Pareto optimal solution set or, at least a representative subset of it. Pareto optimal solution sets are often preferred to single solutions because they can be practical when considering real-life problems since the final solution of the decision maker is always a trade-off.

Over the years several different alternative techniques have been proposed, some of which will be discussed below. However, identifying the entire Pareto optimal set, for many multi-objective problems, is practically impossible due to its size and, depending on the techniques, also due to its shape and properties. In addition, for many problems, especially for combinatorial optimization problems, proof of solution optimality is computationally infeasible. Thus, the approach commonly used is to search for a set of solutions that represent the Pareto optimal set. Therefore, such an approach should find a set of Pareto solutions such that [81]:

1. it is as close as possible to the true Pareto front,
2. the solutions are uniformly distributed and diverse, to cover a wide range of trade-offs,
3. it includes solutions at the extreme ends of the objective function space.

However, these characteristics are conflicting since (for a given computational time limit) the first one requires deep search on a particular region of the search space, the second leads to a distributed search effort, while the third requires the search to be directed to the extremes.

10.3 Multi-Objective Evolutionary Algorithms

After the seminal work of Schaffer [67, 68], a substantial number of different Multi-Objective Evolutionary Algorithms (MOEAs) have been proposed. Good reviews about this subject have been prepared by [14, 11]. Usually, these algorithms can be divided into three classes.

The first class is based on non-Pareto approaches, including techniques such as aggregating functions [14] and VEGA (Vector Evaluated Genetic Algorithm) [67]. In these cases, the decision maker's preferences are stated before the search (a priori), and the solution obtained is a single point. These techniques do not incorporate directly the concept of Pareto optimum, are unable to find some portions of the Pareto front, and are only capable of handling a small number of objectives. However, they are easy to implement.

The second class emerged after Goldbergs suggestion [38] that selection should be made using a non-dominated ranking scheme and that diversity should be maintained with the use of a sharing function, being based on the concept of Pareto Optimality. Some examples of such approach are referred to next. The algorithm proposed in [31] (MOGA) uses a ranking scheme where the rank of each individual corresponds to the number of individuals in the current population by which it is dominated. Fitness sharing is used in order to maintain diversity, together with a mating restriction scheme that avoids crossover between very distant individuals in the search space. Latter, Srinivas and Deb [74] implemented a Pareto based ranking scheme in the Non-dominated Sorting Genetic Algorithm (NSGA). They sort the population in various fronts. The non-dominated individuals belonging to the first front are more fit, hence they are removed from the population and the process is repeated until the entire population is classified. Then, Horn et al. [44] proposed a different algorithm (NPGA) that uses a tournament selection scheme based on the concept of Pareto dominance.

Lately, a third class of MOEAs based on the use of an elite-preserving operator, that suppresses the deterioration of the population fitness along the successive generations, has been proposed. These algorithms perform sequentially the three basic tasks of fitness assignment, density estimation and archiving. Deb and co-authors ([15, 17]) suggested an elitist non-dominated sorting GA (known as NSGA-II). The method uses simultaneously an elite preservation strategy and an explicit diversity preserving mechanism. First, an offspring population is created using the parent population, both of size N. These populations are combined together to form a single population of size $2N$. Then, a classification of the population using a non-dominated sorting is performed. Finally, the new population is filled with the individuals of the best fronts, until its size becomes equal to N. If the population becomes larger than N, a niching strategy is used to select the individuals of the last front. The algorithm proposed by Zitzler and Thiele [83], called Strength Pareto EA (SPEA), introduces elitism by maintaining an external population. Initially, a random population of size N and an empty external population of size N_e are created. In each generation the solutions belonging to the best front are copied to the external population. Then, the dominated solutions found in this modified population are deleted. When the number of solutions in the external population exceeds N_e, a clustering algorithm is used to eliminate the more crowded solutions. This algorithm was modified recently, in order to incorporate a fine-grained fitness assignment strategy, a density estimation technique and an enhanced archive truncation method - the SPEA2 algorithm [82]. Corne et al. [12] proposed PESA (Pareto Envelope-based Selection Algorithm), which uses a small internal population and a larger external population. Initially, an internal population and an empty external population are created. Then, the non-dominated points of the internal population are incorporated in the external population. When a stop criterion is reached, the result will be the non-dominated individuals of the external population. Otherwise, the individuals of the internal population are deleted and new ones are created by crossover and mutation, using as parents the individuals of the external population. Finally, Knowles and Corne [51] introduced an algorithm based on the use of an (1+1) evolution strategy and of an external archive of all the non-dominated solutions. Diversity is maintained though the use of an adaptive grid technique, which is based on a new crowding procedure where the objective space is divided recursively. According to the authors this technique has lower computational cost and the setting of the niche-size parameter is carried out in an adaptive mode [51].

10.3.1 Reduced Pareto Set Genetic Algorithm (RPSGA)

As stated before, in MOEAs only the selection phase of the traditional EA must be changed in order to be possible to deal with the multiple objectives (Figure 10.5).

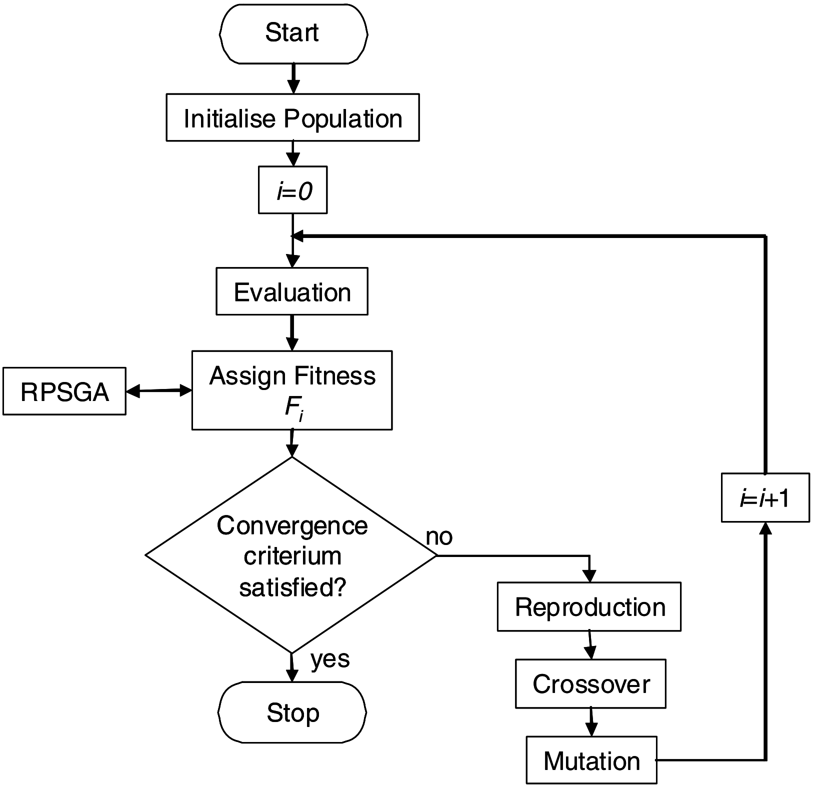

Fig. 10.5 Flowchart of a MOEA.

In this work a MOEA developed previously by one of the authors will be adopted to optimize the processes described in the next section. This algorithm is named Reduced Pareto Set Genetic Algorithm (RPSGA), involving the use of a clustering technique to reduce the number of solutions on the Pareto front [34, 36]. The main steps of this algorithm are illustrated below (Algorithm 1).

Algorithm 1

1. Generate a random initial population (internal).
2. Create an empty external population.
3. **While** not Stop-Condition **Do**
   a) Evaluate internal population.
   b) Calculate the Fitness of the individuals using clustering.
   c) Copy the best individuals to the external population.

d) **If** the external population becomes full **then**
Apply the clustering to this population.
Copy the best individuals to the internal population.
End If

e) Select the individuals for reproduction.

f) Apply crossover.

g) Apply mutation.

End While

Initially, an internal population of size N is randomly generated (step 1) and an empty external population created (step 2). At each generation, i.e., while a stop condition is not found (step 3) the following operations are performed in turn: i) The internal population is evaluated using the modeling routine (step 3a); ii) A clustering technique is applied to reduce the number of solutions on the efficient front and to calculate the fitness of the individuals of the internal population (step 3b) [36]; iii) A fixed number of best individuals are copied to an external population (step 3c); iv) If the external population is not totally full, the genetic operators of selection (step 3e), crossover (step 3f) and mutation (step 3g) are applied to the internal population to generate a new better population; v) When the external population becomes full (step d) the clustering technique is applied to sort the individuals of the external population, and a pre-defined number of the best individuals are incorporated into the internal population by replacing lowest fitness individuals. Detailed information about this algorithm can be found elsewhere [34, 36]. The influence of some important algorithm parameters, such as the size of the internal and of the external populations, the number of individuals copied to the external population and from the external population (to the internal population) in each generation, and the limits of the indifference of the clustering technique have been studied, see [36] for further details.

10.3.2 Recent Developments

Often, solving real optimization problems is very complex since the system performance and characteristics are influenced by more than one field of knowledge and, in addition, requires the use of powerful computational tools. Good examples of such type of problems are the optimization and design of, amongst others, aeroplanes, automobiles and building structures. The traditional way of tackling this type of problems consists of using approximation and decomposition techniques to split a problem into simpler blocks, which are individually solved. A global solution, to the original problem, is then obtained by integrating the solutions to the simpler blocks. This type of approach does not satisfy the actual needs as far as the increasing cost of the design life cycle is concerned. Simultaneously, the high efficiency of the numerical methods available for analyzing specific engineering problems (e.g., computational fluid dynamics and structural mechanics) and the existence of high performance computers enabled the possibility of numerically solving such problems. However, these advantages must go together with the development of more efficient and advanced approaches. Nevertheless, some difficulties persists. First, the result of a MOEA is a set of solutions, but in real problems only a single solution can be used. As a consequence, it will be necessary to provide additional information regarding the relative importance of every objective on the system. This is usually, accomplished by introducing, in the optimization system, the preferences of a Decision Maker [14]. Although some recent work enabled the development of an efficient decision making methodology based on the preferences of the decision maker [28, 29], additional developments are still needed.

In addition, robustness of the solutions should be seek [35, 27] since in real applications small changes on the design variables or on the environmental parameters may happen or may be imposed. Such changes should not affect, or at least affect only slightly, the quality of the proposed solution. Changes in a problem under consideration may arise due to several reasons:

1. parameter values may change due to, for example, data noise (originated by sources such as sensors) or environmental changes,
2. change in the design variables magnitude or on themselves (some may become parameters and new ones may appear),
3. uncertainty on or approximation of some values (parameters, assessment function, etc.),
4. dynamic nature of the problem or evaluation criteria.

From the above, it is clear that robustness is an important aspect to consider during optimization, nevertheless it is rarely included in traditional algorithms. Recently, problems of the second category have been addressed by the authors [35, 27].

One of the major difficulties in applying MOEAs to real problems is the large number of evaluations of the objective functions needed to obtain an acceptable solution - typically of the order of several thousands. Often these are time-consuming evaluations obtained by solving numerical codes with expensive methods like finite-differences or finite-elements. Therefore, reducing the number of evaluations needed, to reach an acceptable solution, is thus of major importance. Finding good approximate methods is even harder for multi-objective problems due to the number of objectives and to the possible interactions between them. Two different efficient methodologies were recently proposed, an Inverse Artificial Neural Network (IANN) approach [37] and a hybrid algorithm based on the use of a filter method as local search procedure [57].

Finally, another important issue that needs to be addressed is dimension of multi-objective problems since as the number of objectives grows, the number of incomparable solutions also grows. Therefore, the problem becomes much more difficult to solve, regarding the point of view of the EAs, since a large number of solutions move from one generation to the next, reducing the selection pressure. Moreover, with more than two objectives, the visualization of the compromises between different solutions becomes extremely complex. In order to tackle real problems with several objectives, it is necessary to investigate ways of reducing the number of objectives. Several approaches based on statistical techniques are reported in the literature [13].

Clearly, the research in multidisciplinary design optimization methodologies and its application to multi-objective engineering problems requires expertise in optimization (e.g., optimization methods, decision support, solution robustness, reduction of the computational requirements, and reduction of the number of objectives) and engineering tools (e.g., computational fluid dynamics, aerodynamics, structural mechanics, aesthetics, etc.). Further research in MOEAs will, certainly, encompass methodologies for dealing and linking these different tools.

10.4 Applications

The very large number of papers published in the last few years, either on international journals or conferences, dealing with applications of MOEAs can be considered a measure of its importance both for the practitioner and scientific communities. A good example is the book by Coello and Lamont [8], where applications on the areas of engineering, industry, economics and management, science and others have been presented.

10.4.1 Engineering

The use of MOEAs in engineering has been quite extensive. Therefore, engineering applications include many different problems, such as: design of welded beams, bulk carriers, airfoil, industrial magnetic devices, optimization of ground water monitoring networks, combinatorial logic circuits, autonomous vehicles navigation, control systems, polymer extrusion problems, truss optimization, city and regional planning, covering tour problem, routing and supersonic wings. Several of the above mentioned applications are described in [8].

Recent works still report new problems or improved methodologies for problems previously addressed. Examples of such works are, for example, the works by Gaspar-Cunha and Covas [36], Gong et al. [39] and Herrero et al. [41], to name just a few.

In [36] an automatic optimization methodology of the polymer extrusion process, using a Multiobjective Optimization Genetic Algorithms approach is proposed. The Reduce Pareto Set Genetic Algorithm with Elitism (RPSGAe), see [34], was applied to the optimization of the operating conditions and to the screw design of a polymer extrusion process, to automatically optimize, in terms of prescribed attributes, the values of important parameters, such as operating conditions and screw geometry. The results obtained for specific case studies have physical meaning and correspond to a successful optimization of the process.

In [39] it is proposed a differential evolution algorithm that adopts the orthogonal design method with quantization technique to generate the initial archive and evolutionary population. In order to keep the nondominated solutions found the authors use a secondary population, which is updated by a new relaxed form of Pareto dominance, at each generation. The authors found their method to be capable of yielding a good coverage and convergence to the true Pareto-optimal fronts for four engineering design problems, namely: Two-bar truss design, Welded beam design, Speed reducer design, and Disc brake design.

Herrero et al. [41] address the problem of designing nonlinear tracking filters under up to several hundreds performance specifications. The suitability of different evolutionary computation techniques for solving such a problem has been analyzed. The authors have found that the application of different MOEA techniques can lead to very different performances. Based on the results obtained and after trying several combination strategies to build the fitness function, the authors propose to build a fitness function based on an operator that selects worst cases of multiple specifications in different situations. Results are obtained for the design of an air traffic control (ATC) tracking filter that should accomplish a specific normative with 264 specifications. The results show good performance, both in terms of effectiveness and computational load.

Let us now illustrate the use of a MOEA to solve a feature extraction problem. The aim here is not to present all the details, but rather show how the RPSGAe algorithm can be used to solve such a problem. The problem to be addressed is a classification problem that can be solved using different methods such as logistics, support vector machines and artificial neural networks. The objective is to find the minimum number of features needed to obtain the maximum accuracy of the companies evaluation. Examples of the features considered are the number of employees, the fixed assets, the current ratio, the liquidity ratio, and the stock turnover days. For that purpose we consider a database with 30 features or attributes characterizing 1200 different companies, for a given year. Regarding the company evaluation we measure whether the company has survived or gone into bankruptcy. In the present study, and for illustration purposes, we have used the logistics and support vector machines methods with a gradient descent and holdout validation, having learning rate and training fraction of 0.01 and 0.6, respectively. The runs performed used the following RPSGA parameters (see [36] for more details): the main and the elitist populations had 100 and 200 individuals, respectively; a roulette wheel selection strategy was adopted; the crossover and mutation probabilities were, respectively, set to 80% and 5%, the number of ranks and the limits of indifference for the clustering technique were chosen to be 30 and 0.01, respectively. The results obtained are reported in Figure 10.6. As it can be seen, 100 generations of evolution lead to a considerable gain in accuracy while decreasing significantly the number of features needed. On the final population only 4 non-dominated solutions exist having, respectively 2, 3, 5 and 6 features, which are identified in Figure 10.7.

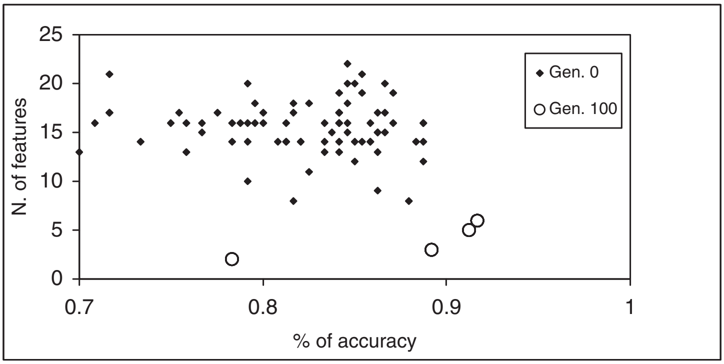

Fig. 10.6 MOEA results: initial population and non-dominated solutions after 100 generations.

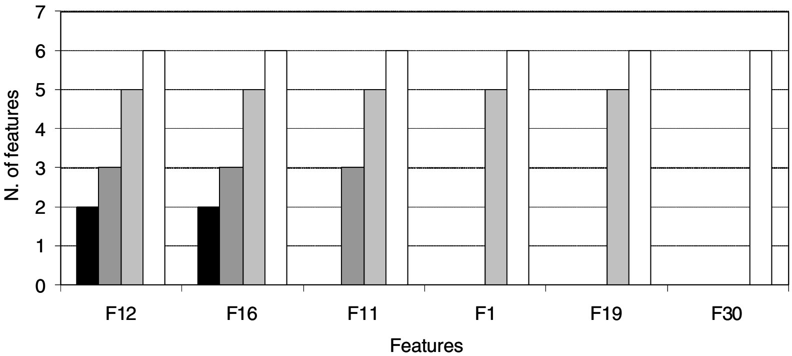

Fig. 10.7 Features used in the 4 non-dominated solutions after 100 generations.

10.4.2 Industrial

Industrial applications of EAs have been the subject of research for many years. Both the single-objective and multi-objective versions of several problem types have been addressed in the literature. In here, we look at recent work on multi-objective versions of cellular manufacturing systems, balancing assembly lines, and scheduling problems.

Cellular manufacturing attempts to bring the benefits of mass production to job-shop production systems. Recent surveys, see e.g., [79] indicate that the practical implementation of cellular manufacturing systems involves many conflicting objectives, however most of the literature is on the single-objective version of the problem. Even when the existence of multiple objectives is acknowledge the proposed solution methodologies, typically, aggregate objectives into a single one, see for example the reviews by Mansouri [56] and Dimopoulos [19]. A recent exception, although others exist, is that of Dimopoulos [20]. In this work the author proposes multi-objective genetic programming single-linkage cluster analysis (GP-SLCA) for the solution of the multi-objective cell-formation problem. The Multi-objective GP-SLCA combines an existing algorithm for the solution of single-objective cell formation problems [21] with NSGA-II, an elitist evolutionary multi-objective optimization technique, in order to automatically generate a set of non-dominated solutions for a given multi-objective cell-formation problem. Although the results presented in the article indicate that multi-objective GP-SLCA is a promising approach to the solution of the multi-objective cell-formation problem, the experimental basis that exists for this problem is small, and consequently there are no extensive comparative results.

The assembly line balancing problem (ALBP) is a decision problem arising when an assembly line has to be (re)-configured, and consists of determining the optimal partitioning of the assembly work among the workstations in accordance with some objectives [71]. These objectives usually take one of two forms: i) either minimising the number of workstations given the cycle time of the line, or ii) minimising the cycle time given the number of workstations. The multi-objective ALBP has attracted a considerable research attention in the last decade. There is a lack in the literature regarding the use of EAs for solving the multi-criteria ALBP of type ii above. Furthermore, as point out by Scholl and Becker [72], the computational testing of most EAs has been performed ignoring existing ALBP test bed. In a recent work, Nearchou [60] presents a differential evolution based approach , inspired on that of Murata [58], for solving the bi-criteria ALBP. The main objective was to minimize the cycle time of the line and secondary objectives to minimize balance delay time and workload smoothness index. The proposed method formulates the cost function of each individual solution as a weighted-sum of multiple objectives functions with self-adapted weights. It maintains and updates a separate population with diverse Pareto-optimal solutions, injects the actual evolving population with some Pareto-optimal solutions to preserve non-dominated solutions found over generations. The encoding scheme used maps real-valued vectors to integer strings corresponding to feasible solutions. The computational results reported are for benchmark problems taken from the open literature and are compared to that of two other previously proposed methods, namely, a weighted sum Pareto GA [58], and a Pareto-niched GA [50].

Production scheduling problems have been researched for many years, however the literature on multi-objective scheduling is notably sparser than on single-objective scheduling. The interest in multi-objective production scheduling, especially in the multi-objective deterministic problem has been sparked by some recent surveys. Particularly after the survey by Nagar et al. [59], which provides a good review of the research on this type of problems up to the early 1990s. They discuss the single machine bi-criteria problem, the single machine multi-criteria problem, the multiple machine bi-criteria problem, and the multiple machine multi-criteria problem. Later, Tkindt et al. [76] present a discussion on the one-machine job shop, the parallel-machine job shop, and the flow shop. For these problems more than 100 published papers have been listed. In a recent work, Hoogeveen [43] looks closer to the earliness-tardiness scheduling and the scheduling with the controllable processing time. More recently Lei [54] looks into multi-objective scheduling after 1995 and an extensively list of papers is provided. The author classifies the scheduling problems based on the nature of the problem, shop configuration, and the description method of uncertainty. Then, the main characteristics of the previous research are summarized, and finally the new trends in scheduling with multiple objectives are pointed out. The author also reviews some less researched problems.

10.4.3 Economics and Management

Economics and management are very promising research areas to apply Evolutionary Algorithms (EAs) for two kinds of reasons. On the one hand, problems within these areas are quite difficult, see for example Schlottmann and Seese [70] for proof of NP-completeness of some financial problems. On the other hand, although the use of EAs in these areas is not an emerging research area [7], the use of multi-objective EAs is still scarce and not many different problems have yet been addressed.

Regarding the use of MOEAs several problems have been addressed, such as time series forecasting, stock ranking, economic exploitation levels, risk management, forest management, space allocation. Recently, very good surveys have been written on applications of MOEAs to problems in economics and management, see [69, 24, 73, 75]. Here we only mention a few new approaches to the portfolio optimization problem, since it has been the most popular, just to show how active is the research on finding solutions to problems in economics and management using MOEAs.

The portfolio optimization problem is a well-known difficult problem occurring in the financial world. In this problem a collection of assets is chosen to be held by an institution or a private individual. The choice is done such that the expected return (mean profit) is maximized, while at the same time the risk is to be minimized. Since the optimal solution depends on the users risk aversion, various trade-offs are usually seek. In [6] the authors propose an approach that integrates an active set algorithm optimized for portfolio selection into a multi-objective evolutionary algorithm (MOEA). The MOEA provides some convex subsets of the set of all feasible portfolios and then a critical line algorithm is solved for each subset. Finally, the partial solutions are merged to obtain the solution to the original non-convex problem. The authors were able to show that their envelope-based MOEA significantly outperforms existing MOEAs, when solving some benchmark problems.

In another recent work Li proposes a multi-objective genetic programming system [55]. This system improves on the previous one, in two different ways, by taking advantage of the MOEAs. One the one hand, it improves on efficiency since a set of Pareto front solutions is obtained in one single execution. On the other hand, from the users perspective, it is simpler as it eliminates a number of user-supplied parameters previously required.

Other problems such as forest management have also been addressed by evolutionary algorithms, see for example [25] and the references therein.

10.4.4 Other Applications

In this section we include several application types that we designate by others. Such applications include for example spectroscopic data analysis, medical image processing, computer-aided diagnosis, treatment planning, machine learning, selection of attributes in data mining, regression and prediction, series forecasting, and biometric applications.

Since here we refer to several applications areas where vast research has been occurring we refer the reader to some recent surveys. In [62] the authors provide an overview of the application of evolutionary computation in the medical domains. First, six types of evolutionary algorithms are outlined (genetic algorithms, genetic programming, evolution strategies, evolutionary programming, classifier systems, and hybrid systems). Then their application to obtain solutions to medical problems, including diagnosis, prognosis, imaging, signal processing, planning, and scheduling, is discussed. Finally, the authors provide an extensive bibliography, classified both according to the medical task addressed and according to the evolutionary technique used. Another review is provided by Handl et al. [40]. In this work, the application of multi-objective optimization in the fields of bioinformatics and computational biology is reviewed. The literature reviewed in the survey, over 140 papers, is arranged by biological problem domain, namely classification problems (unsupervised, supervised, and semisupervised), inverse modeling problems (i.e. problems were it is intended to infer the original system from the observed data), sequence and structure alignment (which involves assessment of similarity and the identification of related sequences or structures), structure prediction and design, and system optimization and experimental design (investigate the degree of optimality of naturally occurring biochemical systems or to design optimal biochemical processes). Another important issue studied in this survey is the identification of five distinct problem contexts, giving rise to multiple objective problems.

A very interesting work, regarding the use of MOEAs, is that of Hruschka et al. [45]. This work provides an up-to-date overview of the use of evolutionary algorithms for clustering problems in different domains, such as image processing, computer security, and bioinformatics. It also provides a taxonomy that highlights some very important aspects in the context of evolutionary data clustering, namely, fixed or variable number of clusters, cluster-oriented or nonoriented operators, context-sensitive or context-insensitive operators, guided or unguided operators, binary, integer, or real encodings, centroid-based, medoid-based, label-based, tree-based, or graph-based representations, among others. The paper ends by addressing some important issues and open questions that can be subject of future research.

Two very successful scientific disciplines using evolutionary algorithms are data mining and machine learning. Data mining has emerged as a major research domain in the recent decades to extract implicit and useful knowledge. Initially, this knowledge extraction was computed and evaluated manually using statistical techniques. Subsequently, semi-automated data mining techniques emerged because of the advancement in technology. Such advancement was also in the form of storage which increased the demands for analysis. In such case, semi-automated techniques have become inefficient. Therefore, automated data mining techniques were introduced to synthesis knowledge efficiently. A critical literature review, highlighting the strengths and limitations of the automated techniques is given by Asghar and Iqbal [2]. Machine learning is concerned with the design and development of algorithms that allow computers to learn based on data. Therefore a research major focus is to automatically learn to recognize complex patterns and make intelligent decisions based on data. Machine learning is inherently multi-objective, since many of the applications where it is used are multi-objective and involve solving hard problems. However, until recently either only one of the objectives was adopted as the cost function or the multiple objectives were aggregated to a scalar cost function. Recently, this has been changing, mainly due to the great success of MOEAs [53]. MOEAs are now being used within machine learning techniques in order to find a number of non-dominated rule sets with respect to the several objectives, see for example the reviews in [48, 32]. For a more detailed account of the existing research on multi-objective learning, the reader is referred to [47].

10.5 Conclusions

This paper has presented a brief review of algorithms in the rapid growing area of Multi-objective Evolutionary Algorithms (MOEAs), as well as, some of their applications. Regarding the algorithms, their self robustness seems to be one of the main issues, since the conditions under which the solutions have been obtained are unlikely to be exactly the ones to be found during the implementation and usage of the method. This may happen due to several reasons, namely: data noise, changes in the design variables, environmental changes, quality measures or other changes with time, etc. Therefore, it is clear that it is an important aspect to be considered during optimization. However, it is rarely included in traditional algorithms.

Another important issue, that is still of major concern, is algorithmic efficiency. Recent research is looking at parallel implementation as a possible solution. Therefore, more in depth and detailed studies of the different aspects involved in parallelization need to be performed.

A recent trend, that most likely will continue to grow further is the use of nature inspired techniques, such as particle swarm optimization [10, 9, 65], differential evolution [1, 3, 80], ant colony systems [22, 23, 33], electromagnetism [5, 18], amongst others.

Reformulating some real problems, which are currently addressed as if they only have a single objective is likely to be one of the probable future trends in MOEAs. Thus, the research efforts should not only be put into the development of new algorithms but also on the adaptation of existing algorithms to new applications.

The main idea we would like to leave the reader with is that Evolutionary Algorithms are a viable alternative to solve difficult real-world problems in a reasonable amount of time. Sometimes, they might even be the only alternative providing good results. Given their heuristic nature there are no guarantees on the solution quality.

However, there is an overwhelming evidence showing their effectiveness to address complex real-world problems when compared to other heuristics, regardless of being deterministic or stochastic.

References

1. H.A. Abbass and R. Sarker. The Pareto differential evolution algorithm. *International Journal on Artificial Intelligence Tools*, 11(4):531–552, 2002.
2. S. Asghar and K. Iqbal. Automated data mining techniques:a critical literature review. In *Proceedings of the International Conference on Information Management and Engineering*, pages 75–76, 2009.
3. B.V. Babu and M.M.L. Jehan. Differential evolution for multi-objective optimization. In *The 2003 Congress on Evolutionary Computation*, volume 4, 2003.
4. D. Beasley. Possible applications of evolutionary computation. In T. Bäck, D.B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages 4–19. IOP Publishing Ltd. and Oxford University Press, 1997.
5. S.I. Birbil and S.C. Fang. An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 25:263–282, 2003.
6. J. Branke, B. Scheckenbach, M. Stein, K. Deb, and H. Schmeck. Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. *European Journal of Operational Research*, 199(3):684–693, 2009.
7. S.H. Chen. *Evolutionary Computation in Economics and Finance*. Physica-Verlag, 2002.
8. C.A.C. Coello and G.B. Lamont. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, 2004.
9. C.A.C. Coello and M.S. Lechuga. MOPSO: a proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation, 2002*, volume 2, pages 1051–1056, 2002.
10. C.A.C. Coello, G.T. Pulido, and M.S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, 2004.
11. C.A.C. Coello, D.A. Van Veldhuizen, and G.B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, Dordrecht, 2002.
12. D.W. Corne, J.D. Knowles, and M.J. Oates. The Pareto-Envelope based Selection Algorithm for Multiobjective Optimisation. In *Proceedings of the Parallel Problem Solving from Nature VI (PPSNVI)*, 2000.
13. L. Costa and P. Oliveira. Dimension reduction in multiobjective optimization. *PAMM*, 7(1):2060047–2060048, 2007.
14. K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, 2001.
15. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGAII. In *Proceedings of the Parallel Problem Solving from Nature VI (PPSNVI)*, pages 849–858, 2000.
16. K. Deb and D.E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proc. Third Int. Conf. on Genetic Algorithms*, 1989.
17. K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGAII. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
18. D. Debels and M. Vanhoucke. The electromagnetism meta-heuristic applied to the resource-constrained project scheduling problem. *Lecture Notes in Computer Science*, 3871:259–270, 2006.
19. C. Dimopoulos. A review of evolutionary multiobjective optimization applications in the area of production research. In *Congress on Evolutionary Computation (CEC 04)*, pages 1487–1494, 2004.

20. C. Dimopoulos. Explicit consideration of multiple objectives in cellular manufacturing. *Engineering Optimization*, 39(5):551–565, 2007.
21. C. Dimopoulos and N. Mort. Evolving knowledge for the solution of clustering problems in cellular manufacturing. *International Journal of Production Research*, 42(19):4119–4133, 2004.
22. K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*, 131(1):79–99, 2004.
23. K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer. Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, 171(3):830–841, 2006.
24. E.I. Ducheyne, B. De Baets, and R.R. De Wulf. *Applications of Multi-Objective Evolutionary Algorithms*, volume 1 of *Advances in Natural Computation*, chapter Even Flow Scheduling Problems in Forest Management, pages 701–726. World Scientific Pub Co Inc, 2004.
25. E.I. Ducheyne, R.R. De Wulf, and B. De Baets. A spatial approach to forest-management optimization: linking GIS and multiple objective genetic algorithms. *International Journal of Geographical Information Science*, 20(8):917–928, 2006.
26. F.Y. Edgeworth. *Mathematical Physics*. P. Keagan, London, England, 1881.
27. J.C. Ferreira, C.M. Fonseca, J.A. Covas, and A. Gaspar-Cunha. *Advances in Evolutionary Algorithms*, chapter Evolutionary Multi-Objective Robust Optimization, pages 261–278. I-Tech Education and Publishing, 2008.
28. J.C. Ferreira, C.M. Fonseca, and A. Gaspar-Cunha. Methodology to select solutions from the Pareto-optimal set: A comparative study. In *GECCO 2007, Genetic and Evolutionary Computation Conference*, 2007.
29. J.C. Ferreira, C.M. Fonseca, and A Gaspar-Cunha. Selection of solutions in a multi-objective environment: Polymer extrusion a case study. In *Evolutionary Methods For Design, Optimization and Control*, 2008.
30. L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley, New York, 1996.
31. C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *Multiple Criteria Problem Solving*, pages 416–423. Morgan Kauffman Publishers, San Mateo, California, 1993.
32. L. Galway, D. Charles, and M. Black. Machine learning in digital games: A survey. *Artificial Intelligence Review*, 29(2):123–161, 2009.
33. C. Garcia-Martinez, O. Cordon, and F. Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*, 180(1):116–148, 2007.
34. A. Gaspar-Cunha. *Modeling and Optimisation of Single Screw Extrusion*. PhD thesis, University of Minho, Guimares, Portugal, 2000.
35. A. Gaspar-Cunha and J.A. Covas. Robustness in multi-objective optimization using evolutionary algorithms. *Computational Optimization and Applications*, 39(1):75–96, 2008.
36. A. Gaspar-Cunha and Covas J.A. *Metaheuristics for Multiobjective Optimisation*, chapter RPSGAe - A Multiobjective Genetic Algorithm with Elitism: Application to Polymer Extrusion, pages 221–249. Lecture Notes in Economics and Mathematical Systems. Springer, 2004.
37. A. Gaspar-Cunha and A.S. Vieira. A hybrid multi-objective evolutionary algorithm using an inverse neural network. In *Hybrid Metaheuristics (HM 2004) Workshop at ECAI 2004*, 2004.
38. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
39. W.Y. Gong, Z.H. Cai, and L. Zhu. An efficient multiobjective differential evolution algorithm for engineering design. *Structural And Multidisciplinary Optimization*, 38(2):137–157, 2009.
40. J. Handl, D.B. Kell, and J. Knowles. Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions On Computational Biology And Bioinformatics*, 4(2):279–292, 2007.

41. J.G. Herrero, A. Berlanga, and J.M.M. Lopez. Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters. *IEEE Transactions on Evolutionary Computation*, 13(1):151–168, 2009.
42. J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
43. H. Hoogeveen. Multicriteria scheduling. *European Journal of Operational Research*, 167:592–623, 2005.
44. J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, 1994.
45. E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, and A.C.P.L.F. de Carvalho. A survey of evolutionary algorithms for clustering. *IEEE Transactions On Systems, Man, and Cybernetics Part C*, 39(2):133–155, 2009.
46. M.T. Jensen. Reducing the run-time complexity of multi-objective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5):503–515, 2003.
47. Y. Jin, editor. *Multi-Objective Machine Learning*. Springer-Verlag, New York, 2006.
48. Y. Jin and B. Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions On Systems, Man, and Cybernetics Part C*, 38(3):397–415, 2008.
49. R.L. Keeney and H. Raiffa. *Decision with Multiple Objectives. Preferences and Value Trade-offs*. Cambridge University Press, Cambridge, UK, 1993.
50. Y.K. Kim, Y.-J. Kim, and Y. Kim. Genetic algorithms for assembly line balancing with various objectives. *Computers and Industrial Engineering*, 30(3):397–409, 1996.
51. J.D. Knowles and D.W. Corne. Approximating the non-dominated front using the Pareto archived evolutionary strategy. *Evolutionary Computation Journal*, 8(2):149–172, 2000.
52. J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
53. R. Kumar. *Applications of Multi-Objective Evolutionary Algorithms*, chapter On machine learning with multiobjective genetic optimization, pages 393–425. World Scientific, 2004.
54. D. Lei. Multi-objective production scheduling: A survey. *International Journal on Advanced Manufacturing and Technology*, 43:926–938, 2009.
55. J. Li. Enhancing financial decision making using multi-objective financial genetic programming. In *In IEEE Congress on Evolutionary Computation*, 2006.
56. S.A. Mansouri, S.M.M. Husseini, and Newman. S.T. A review of the modern approaches to multi-criteria cell design. *International Journal of Production Research*, 38(5):1201–1218, 2000.
57. F. Mendes, V. Sousa, M.F.P. Costa, and A. Gaspar-Cunha. Multi-objective memetic algorithm using pattern search filter methods. In *EU/Meeting 2009*, 2009.
58. T. Murata, H. Ishibuchi, and H. Tanaka. Multi-objective genetic algorithms and its application to flow-shop scheduling. *Computers and Industrial Engineering*, 30(4):957–968, 1996.
59. A. Nagar, J. Haddock, and S. Heragu. Multiple and bicriteria scheduling: A literature survey. *European Journal of Operational Research*, 81:88–104, 1995.
60. A.C. Nearchou. Multi-objective balancing of assembly lines by population heuristics. *International Journal of Production Research*, 46(8):2275–2279, 2008.
61. V. Pareto. *Cours dEconomie Politique*. vol. I and II, F. Rouge, Lausanne, 1896.
62. C.A. Peña Reyes and M. Sipper. Evolutionary computation in medicine: An overview. *Artificial Intelligence in Medicine*, 19:1–23, 2000.
63. L. Rachmawati and D. Srinivasan. Preference incorporation in multi-objective evolutionary algorithms: A survey. In *IEEE Congress on Evolutionary Computation, 2006*, pages 962–968, 2006.
64. I. Rechenberg. *Evolutionstrategie: Optimierung Technisher Systeme Nach Prinzipien der Biologischen Evolution*. Frommann-Hoolzboog Verlag, 1973.
65. M. Reyes-Sierra and C.A.C. 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.

66. T.L. Saaty. *Multicriteria Decision Making: The Analytic Hierarchy Process - Planning, Priority Setting, Resource Allocation*. RWS Publications, Pittsburgh, PA, 1990.
67. J.D. Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Nashville, Tennessee, 1984.
68. J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications*, pages 93–100. L. Erlbaum Associates Inc. Hillsdale, NJ, USA, 1985.
69. F. Schlottmann and D. Seese. *Applications of Multi-Objective Evolutionary Algorithms*, volume 1 of *Advances in Natural Computation*, chapter Financial Applications of Multi-Objective Evolutionary Algorithms: Recent Developments and Future Research Directions, pages 627–652. World Scientific Pub Co Inc, 2004.
70. F. Schlottmann and D. Seese. *Handbook of Computational and Numerical Methods in Finance*, chapter Modern Heuristics for Finance Problems:A Survey of Selected Methods and Applications, pages 331–360. Birkhäuser, 2004.
71. A. Scholl, editor. *Balancing and Sequencing of Assembly Lines*. Physica-Verlag, 1999.
72. A. Scholl and C. Becker. State of the art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168:666–693, 2006.
73. L.J.D. Silva and E.K. Burke. *Applications of Multi-Objective Evolutionary Algorithms*, volume 1 of *Advances in Natural Computation*, chapter Using Diversity to Guide the Search in Multi-objective Optimization, pages 727–747. World Scientific Pub Co Inc, 2004.
74. N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2:221–248, 1995.
75. M.G.C. Tapia and C.A.C. Coello. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *IEEE Congress on Evolutionary Computation, CEC*, pages 532–539, 2007.
76. V. TKindt, J. Billaut, and C. Proust. Multicriteria scheduling problems: A survey. *RAIRO Operations Research*, 35:143–163, 2001.
77. D.A. Van Veldhuizen and G.B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary computation*, 8(2):125–148, 2000.
78. P. Vincke. Analysis of MCDA in Europe. *European Journal of Operational Research*, 25:160–168, 1995.
79. U. Wemmerlov and D.J. Johnson. Empirical findings in manufacturing cell design. *International Journal of Production Research*, 38:481–507, 2000.
80. F. Xue, A.C. Sanderson, and R.J. Graves. Pareto-based multi-objective differential evolution. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 2, 2003.
81. E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
82. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK report no. 103, Swiss Federal Institute of Technology, Zurich, Switzerland*, 2001.
83. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.