Chapter 1

EVOLUTIONARY ALGORITHMS AND MULTIPLE OBJECTIVE OPTIMIZATION

Carlos A. Coello Coello*
CINVESTAV-IPN
Departamento de Ingeniería Eléctrica
Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D.F. 07300, MEXICO
ccoello@cs.cinvestav.mx

Carlos E. Mariano Romero†
Instituto Mexicano de Tecnología del Agua
Paseo Cuauhnáhuac 8532
Col. Progreso
Jiutepec, Morelos, 62550, MEXICO
cmariano@tlaloc.imta.mx

Abstract This chapter presents a review of the most important evolutionary multiobjective optimization techniques developed to date. Using as a basis a simple taxonomy of approaches, we briefly describe and analyze the advantages and disadvantages of each of them, together with some of their applications reported in the literature. Other important issues such as diversity and some of the main techniques developed to preserve it, as well as the need of suitable test functions and metrics that can properly evaluate the performance of these multiobjective optimization

*The first author acknowledges support from the mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 34201-A.
†The second author acknowledges support from the mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 33000-A, and from the Instituto Mexicano de Tecnología del Agua.

techniques are also addressed. We conclude this chapter with a brief outline of some potential paths of future research in this area.

Keywords: evolutionary algorithms, evolutionary multiobjective optimization, genetic algorithms, multiobjective optimization, vector optimization

1. Introduction

The idea of using techniques based on the emulation of the mechanism of natural selection to solve problems can be traced as long back as the 1930s [12]. However, it was not until the 1960s that the three main techniques based on this notion were developed: genetic algorithms [75], evolution strategies [142] and evolutionary programming [50]. These approaches, which are now collectively denominated "evolutionary algorithms", have been very effective for single-objective optimization [58, 144, 51].

Evolutionary algorithms seem also particularly desirable for solving multiobjective optimization problems because they deal simultaneously with a set of possible solutions (the so-called population) which allows us to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous and concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques.

The potential of evolutionary algorithms in this field was indicated in the late 1960s by Rosenberg [132], but the first implementation was not produced until the mid-1980s [137, 138]. Since then, a considerable amount of research has been done in this area, now known as evolutionary multi-objective optimization (EMOO for short). The growing importance of this field is reflected by a significant increment (mainly during the last five years) of technical papers in international conferences and peer-reviewed journals, special sessions in international conferences and interest groups on the Internet[1].

The content of this chapter is organized as follows: first, we will define the terminology that we will adopt and we will describe the general multiobjective optimization problem. Then, we will give some basic notions

[1]The first author maintains an EMOO repository with over 850 bibliographical entries at: `http://delta.cs.cinvestav.mx/~ccoello/EMOO`, with mirrors at `http://www.lania.mx/~ccoello/EMOO/` and `http://www.jeo.org/emo/`

of evolutionary algorithms. After that, we will analyze the main evolutionary multiobjective optimization techniques that have been proposed in the specialized literature. Each technique will be briefly described and criticized. We will also provide some sample applications of each. Then, we will describe some of the main approaches proposed to maintain diversity, emphasizing the importance that this process has in multiobjective optimization. Test functions and metrics proposed for EMOO techniques are also discussed together with some representative applications reported in the literature. Finally, we will describe some of the potential research paths in this area.

2. Definitions

The emphasis of this chapter is the solution of multiobjective optimization problems (MOPs) of the form:

$$\text{minimize } [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \tag{1.1a}$$

subject to the m inequality constraints:

$$g_i(\vec{x}) \geq 0 \quad i = 1, 2, \ldots, m \tag{1.2b}$$

and the p equality constraints:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{1.3c}$$

where k is the number of objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. We call $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We wish to determine from among the set $\mathcal{F}$ of all vectors which satisfy (1.2a) and (1.3b) the particular set of values $x_1^*, x_2^*, \ldots, x_n^*$ which yield the optimum values of all the objective functions.

2.1 Pareto optimality

It is rarely the case that there is a single point that simultaneously optimizes all the objective functions. Therefore, we normally look for "trade-offs", rather than single solutions when dealing with multiobjective optimization problems. The notion of "optimality" is therefore, different. The most commonly adopted notion of optimality is that originally proposed by Francis Ysidro Edgeworth [44] and later generalized by Vilfredo Pareto [114]. Although some authors call this notion *Edgeworth-Pareto optimality* (see for example [152]), we will use the most commonly accepted term: *Pareto optimality*.

We say that a vector of decision variables $\vec{x}^* \in \mathcal{F}$ is *Pareto optimal* if there does not exist another $\vec{x} \in \mathcal{F}$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\vec{x}) < f_j(\vec{x}^*)$ for at least one j.

In words, this definition says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\vec{x} \in \mathcal{F}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions called the *Pareto optimal set*. The vectors $\vec{x}^*$ correspoding to the solutions included in the Pareto optimal set are called *nondominated*. The image of the Pareto optimal set under the objective functions is called *Pareto front*.

3. Notions of Evolutionary Algorithms

The term evolutionary computing or evolutionary algorithms is generically applied to a set of biologically-inspired techniques (inspired by the Neo-Darwinian theory of natural evolution[2]. Although three main paradigms are normally considered (evolutionary programming [50, 51], evolution strategies [143, 144], and genetic algorithms [76, 58]), nowadays it becomes increasingly difficult to distinguish the differences among them, and researchers tend to use the broader term "evolutionary algorithms" to refer to any technique that is based in the principle of natural selection (or survival of the fittest) originally defined by Charles Darwin [29].

In nature, individuals have to adapt to their environment in order to survive in a process called "evolution", in which those features that make an individual more suited to compete are preserved when it reproduces, and those features that make it weaker are eliminated. Such features are controlled by units called genes which form sets called chromosomes. Over subsequent generations not only the fittest individuals survive, but also their fittest genes which are transmitted to their descendants during the sexual recombination process which is called crossover.

In general terms, to simulate an evolutionary process in a computer, we need the following [105]:

- A representation for potential solutions to the problem.

[2]The Neo-Darwinian theory of natural evolution combines the original evolutionary theory of Charles Darwin (based on the survival of the fittest), the selectionism of August Weismann and Mendel's inheritance laws. It is called "Neo-Darwinian", because it improves the original proposal of Charles Darwin.

- A way to create an initial population of potential solutions (this is normally done randomly, but deterministic approaches can also be used).
- An evaluation function that plays the role of the environment, rating solutions in terms of their "fitness".
- Genetic operators that alter the composition of the offspring generated (normally, crossover and mutation).
- Values for various parameters that the evolutionary algorithm uses (population size, probabilities of applying genetic operators, etc.).

These elements are important both for single- and for multi-objective optimization. However, in multi-objective optimization, two more issues must be kept in mind: how to select individuals so that they correspond to elements of the Pareto optimal set, and how to keep diversity to avoid convergence of all the population to a single solution.

4. Classifying Techniques

A considerable amount of EMOO techniques have been developed in recent years [19, 162]. In an attempt to discuss the most important approaches proposed, we decided to classify these techniques using the following scheme:

- Non-Pareto Techniques
  - Aggregating approaches
  - VEGA
  - Lexicographic ordering
  - The ε-constraint method
  - Target-vector approaches
  - Game theory
- Pareto-based Techniques
  - Pure Pareto ranking
  - MOGA
  - NSGA
  - NPGA
  - Non-generational approaches

- Recent Approaches
  - PAES
  - SPEA
  - Micro-Genetic Algorithm

5. Non-Pareto Techniques

Under this category, we will consider approaches that do not incorporate directly the concept of Pareto optimality (or Pareto dominance). The approaches discussed in this section are all very efficient (computationally speaking), but most of them are incapable of producing certain portions of the Pareto front. Others could be appropriate to handle only a few objectives. However, their simplicity and efficiency has made them popular among a certain sector of researchers.

5.1 Aggregating approaches

Perhaps the most straightforward approach to handle multiple objectives with any technique is to use a combination of all the objectives into a single one using either an addition, multiplication or any other combination of arithmetical operations that we could think of. These techniques are normally known as "aggregating functions", because they combine (or "aggregate") all the objectives of the problem into a single one. In fact, aggregating approaches are the oldest mathematical programming methods for multiobjective optimization, since they can be derived from the Kuhn-Tucker conditions for nondominated solutions [89].

An example of this approach is a sum of weights of the form:

$$\min \sum_{i=1}^{k} w_i f_i(\vec{x}) \tag{1.4d}$$

where $w_i \geq 0$ are the weighting coefficients representing the relative importance of the k objective functions of our problem. It is usually assumed that

$$\sum_{i=1}^{k} w_i = 1 \tag{1.5e}$$

Aggregating functions have been used with evolutionary algorithms in a number of occasions, with relative success in problems in which the behavior of the objective functions is more or less well-known.

It is normal practice in aggregating approaches to vary the weighting coefficients used, so that different portions of the Pareto front can be generated. However, it is important to realize that the weighting coefficients do not reflect proportionally the relative importance of the objectives (unless a proper scaling of the objectives takes place), but are only factors which, when varied, locate elements from the Pareto optimal set.

5.1.1 Advantages and disadvantages. The main advantages of this method are its simplicity (it is easy to implement and use) and its efficiency (computationally speaking). Its main disadvantage is the difficulty to determine the appropriate weight coefficients to be used when we do not have enough information about the problem (this is an important concern, particularly in real-world applications). Also, a proper scaling of the objectives requires a considerable amount of extra knowledge about the problem. To obtain this information could be a very expensive process (computationally speaking). A more serious drawback of this approach is that it cannot generate certain portions of the Pareto front when its shape is concave, regardless of the weights combination used [30]. Nevertheless, aggregating functions could be very useful to get a preliminary sketch of the Pareto front of a certain problem, or to provide prior information to be exploited by another approach.

5.1.2 Some applications.

- Water quality control [15].
- Controller design [40].
- Design of optical filters for lamps [46].
- Improvement of wire-antenna geometries [166].

5.2 VEGA

This is the first actual implementation of an evolutionary multiobjective optimization technique, which was made by Schaffer [137, 138] in the mid-1980s. The approach was called the *Vector Evaluated Genetic Algorithm* (VEGA), and it basically consisted of a simple genetic algorithm (GA) with a modified selection mechanism. At each generation, a number of sub-populations were generated by performing proportional selection according to each objective function in turn. Thus, for a problem with k objectives, k sub-populations of size N/k each would be generated (assuming a total population size of N). These sub-populations

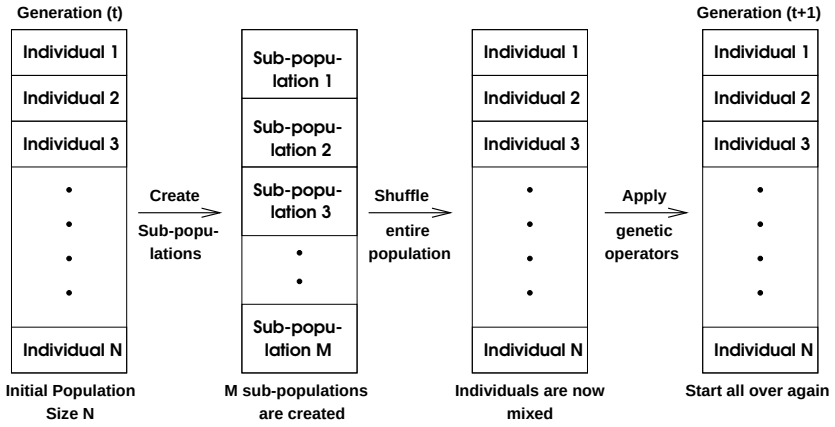

Figure 1.1a. Scheme of VEGA's selection mechanism. It is assumed that the population size is N and that there are M objective functions.

would then be shuffled together to obtain a new population of size N, on which the GA would apply the crossover and mutation operators in the usual way. This process is illustrated in Figure 1.1a. Schaffer realized that the solutions generated by his system were nondominated in a local sense, because their nondominance was limited to the current population, and while a locally dominated individual is also globally dominated, the converse is not necessarily true [138]. An individual which is not dominated in one generation may become dominated by an individual who emerges in a later generation. Also, he noted a problem that in genetics is known as "speciation" (i.e., we could have the evolution of "species" within the population which excel on different aspects of performance). This problem arises because this technique selects individuals who excel in one dimension of performance, without looking at the other dimensions. The potential danger doing that is that we could have individuals with what Schaffer called "middling" performance[3] in all dimensions, which could be very useful for compromise solutions, but which will not survive under this selection scheme, since they are not in the extreme for any dimension of performance (i.e., they do not produce the best value for any objective function, but only moderately good values for all of them). Speciation is undesirable because it is opposed to our goal of finding a compromise solution. Schaffer suggested

[3]By "middling", Schaffer meant an individual with acceptable performance, perhaps above average, but not outstanding for any of the objective functions.

some heuristics to deal with this problem. For example, to use a heuristic selection preference approach for nondominated individuals in each generation, to protect our "middling" chromosomes. Also, crossbreeding among the "species" could be encouraged by adding some mate selection heuristics instead of using the random mate selection of the traditional GA.

5.2.1 Advantages and disadvantages. VEGA is very simple and easy to implement, since only the selection mechanism of a traditional GA has to be modified. However, the shuffling and merging of all the sub-populations that VEGA does corresponds to averaging the fitness components associated with each of the objectives [60]. Since Schaffer used proportional fitness assignment [58], these fitness components were in turn proportional to the objectives themselves [53]. Therefore, the resulting expected fitness corresponded to a linear combination of the objectives where the weights depended on the distribution of the population at each generation as shown by Richardson et al. [128]. This means that VEGA has the same problems as the aggregating approaches (i.e., it is not able to generate concave portions of the Pareto front). Nevertheless, VEGA has been found useful in other domains such as constraint-handling, where its biased behavior can be of great help [154, 22].

5.2.2 Some applications.

- Groundwater pollution containment [129].
- Optimum placement of aerodynamic actuators for aircraft control [131, 130].
- Design of combinational circuits at the gate-level [22].
- Constraint-handling in evolutionary algorithms used for single-objective optimization [21, 154, 153].

5.3 Lexicographic ordering

In this method, the user is asked to rank the objectives in order of importance. The optimum solution $\vec{x}^*$ is then obtained by minimizing the objective functions, starting with the most important one and proceeding according to the assigned order of importance of the objectives.

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

$$\text{Minimize} \quad f_1(\vec{x}) \tag{1.6f}$$

subject to

$$g_j(\vec{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{1.7g}$$

and its solution $\vec{x}_1^*$ and $f_1^* = (\vec{x}_1^*)$ is obtained. Then the second problem is formulated as

$$\text{Minimize} \quad f_2(\vec{x}) \tag{1.8h}$$

subject to

$$g_j(\vec{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{1.9}$$

$$f_1(\vec{x}) = f_1^* \tag{1.10}$$

and the solution of this problem is obtained as x_2^* and $f_2^* = f_2(x_2^*)$. This procedure is repeated until all k objectives have been considered. The ith problem is given by

$$\text{Minimize} \quad f_i(\vec{x}) \tag{1.11i}$$

subject to

$$g_j(\vec{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{1.12}$$

$$f_l(\vec{x}) = f_l^*, \quad l = 1, 2, \ldots, i-1 \tag{1.13}$$

The solution obtained at the end, i.e., x_k^* is taken as the desired solution x^* of the problem.

Fourman [55] suggested a selection scheme based on lexicographic ordering. In a first version of his algorithm, objectives are assigned different priorities by the user and each pair of individuals are compared according to the objective with the highest priority. If this resulted in a tie, the objective with the second highest priority was used, and so on. In another version of this algorithm (that apparently worked quite well), an objective is randomly selected at each run.

5.3.1 Advantages and disadvantages. This technique explores objective space unequally, in the sense that priority is given to solutions performing well in one objective over another(s). Or, in other words, one objective is optimized at all costs. This approach appears most suitable only when the importance of each objective (in comparison to the others) is clearly known.

Selecting randomly an objective (as in the case of Fourman [55]) is equivalent to a weighted combination of objectives, in which each weight is defined in terms of the probability that each objective has of being selected. However, the use of tournament selection with this approach makes an important difference with respect to other approaches such as VEGA, because the pairwise comparisons of tournament selection will make scaling information negligible [53]. This means that this approach may be able to see as convex a concave trade-off surface, although that really depends on the distribution of the population and on the problem itself. Its main weakness is that this approach will tend to favor certain objectives when many are present in the problem, because of the randomness involved in the process, and this will have the undesirable consequence of making the population to converge to a particular part of the Pareto front rather than to delineate it completely [27]. The main advantage of this approach is its simplicity and computational efficiency. These two properties make it highly competitive with other non-Pareto approaches such as a weighted sum of objectives or VEGA.

5.3.2 Some applications.

- Symbolic layout compaction [55].
- Tuning of a fuzzy controller for the guidance of an autonomous vehicle in an elliptic road [56].

5.4 The ε-constraint method

This method is based on minimization of one (the most preferred or primary) objective function, and considering the other objectives as constraints bound by some allowable levels ε_i. Hence, a single objective minimization is carried out for the most relevant objective function f_1 subject to additional constraints on the other objective functions. The levels ε_i are then altered to generate the entire Pareto optimal set. The method may be formulated as follows:

1 Find the minimum of the rth objective function, i.e. find $\vec{x}^*$ such that

$$f_r(\vec{x}^*) = \min_{x \in \mathcal{F}} f_r(\vec{x}) \tag{1.14j}$$

subject to additional constraints of the form

$$f_i(\vec{x}) \leq \varepsilon_i \quad \text{for } i = 1, 2, \ldots, k \quad \text{and } i \neq r \tag{1.15k}$$

where ε_i are assumed values of the objective functions which we do not wish to exceed.

2 Repeat (1) for different values of ε_i. The information derived from a well chosen set of ε_i can be useful in making the decision. The search ends when the user finds a satisfactory solution.

It may be necessary to repeat the above procedure for different indices r.

To get adequate ε_i values, single-objective optimizations are normally carried out for each objective function in turn by using mathematical programming techniques (or independent EAs). For each objective function f_i $(i = 1, 2, \ldots, k)$, there is an optimal solution vector $\vec{x}_i^*$ for which $f_i(\vec{x}_i^*)$ is a minimum. Let $f_i(\vec{x}_i^*)$ be the lower bound on ε_i, i.e.

$$\varepsilon_i \geq f_i(\vec{x}_i^*) \quad i = 1, 2, \ldots, r-1, r+1, \ldots, k \tag{1.16l}$$

and $f_i(\vec{x}_r^*)$ be the upper bound on ε_i, i.e.

$$\varepsilon_i \leq f_i(\vec{x}_r^*) \quad i = 1, 2, \ldots, r-1, r+1, \ldots, k \tag{1.17m}$$

When the bounds ε_i are too low, there is no solution and at least one of these bounds must be relaxed.

This technique has been hybridized with EAs on several occasions. The idea is to use only one objective function at a time as the fitness function of the EA, and keep the others constant (constrained to a single value). Then, the EA is run several times varying the constrained values, so that the Pareto front of the problem can be generated.

5.4.1 Advantages and disadvantages. The main disadvantage of this approach is its (potentially high) computational cost. Also, the encoding of the objective functions may be extremely difficult or even impossible for certain applications, particularly if there are too many objectives. Nevertheless, the relative simplicity of the technique (its main advantage) has made it popular among some researchers (particularly in engineering).

5.4.2 Some applications.

- Preliminary design of a marine vehicle [94].
- Groundwater pollution containment problems [149].
- Fault tolerant system design [139].

5.5 Target-vector approaches

This category encompasses methods in which we have to define a set of goals (or targets) that we wish to achieve for each objective function under consideration. The EA in this case will then try to minimize the difference between the current solution generated and the vector of desirable goals (different metrics can be used for this purpose). Although target vector approaches can be considered as another aggregating approach, we decided to discuss them separately because these techniques can generate (under certain conditions) concave portions of the Pareto front, whereas approaches based on simple weighted sums cannot.

The most popular techniques included here are hybrids of EAs with: Goal Programming [32, 170, 135], Goal Attainment [171, 177], and the min-max algorithm [67, 23].

5.5.1 Advantages and disadvantages. The main advantage of these methods is their simplicity and their efficiency (computationally speaking) because they do not require a Pareto ranking procedure. However, their main disadvantage is the definition of the desired goals which requires some extra computational effort. Some target vector approaches have additional problems. For example, Wilson and MacLeod [171] found that goal attainment could generate, under certain circumstances, a misleading selection pressure. For example, if we have two candidate solutions which are the same in one objective function value but different in the other, they will still have the same goal-attainment value for their two objectives, which means that for the EA neither of them will be better than the other.

An additional problem with these techniques is that they will yield a nondominated solution only if the goals are chosen in the feasible domain, and such conditions may certainly limit their applicability.

5.5.2 Some applications.

- Design of multiplierless IIR filters [171].
- Structural optimization [135, 67].
- Optimization of the counterweight balancing of a robot arm [25].

5.6 Game theory

We can analyze this technique with reference to a simple optimization problem with two objectives and two design variables whose graphical representation is shown in Figure 1.1b. Let $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ represent two scalar objectives and x_1 and x_2 two scalar variables. It

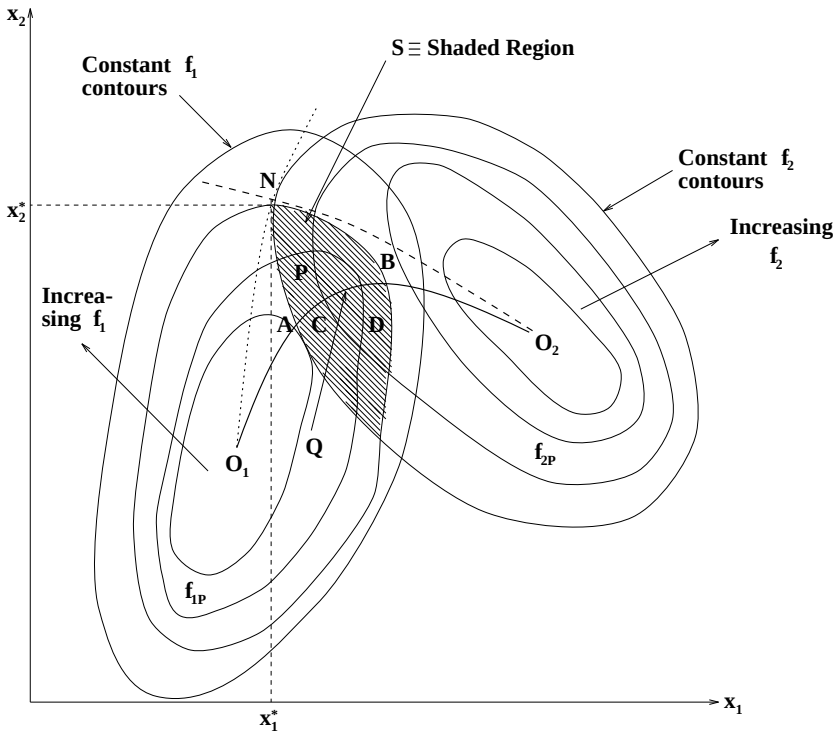

Figure 1.1b. Example of cooperative and non-cooperative game solutions.

is assumed that one player is associated with each objective. The first player wants to select a variable x_1 which will minimize his objective function f_1, and similarly the second player seeks a variable x_2 which will minimize his objective function f_2. If f_1 and f_2 are continuous, then the contours of constant values of f_1 and f_2 appear as shown in Figure 1.1b. The dotted lines passing through O_1 and O_2 represent the loci of rational (minimizing) choices for the first and second player for a fixed value of x_2 and x_1, respectively. The intersection of these two lines, if it exists, is a candidate for the two objective minimization problem, assuming that the players do not cooperate with each other (*non-cooperative game*). In Figure 1.1b, the point $N(x_1^*, x_2^*)$ represents such an intersection point. This point, known as a *Nash equilibrium* solution, represents a stable equilibrium condition in the sense that no player can deviate unilaterally from this point for further improvement of his own criterion [110].

This point has the characteristic that

$$f_1(x_1^*, x_2^*) \leq f_1(x_1, x_2^*) \tag{1.18n}$$

and

$$f_2(x_1^*, x_2^*) \leq f_2(x_1^*, x_2) \tag{1.19o}$$

where x_1 may be to the left or right of x_1^* in (1.18m) and x_2 may lie above or below x_2^* in (1.19n).

5.6.1 Advantages and disadvantages. The main advantage of this approach is that it is very efficient (computationally speaking). However, under certain circumstances, it could generate a single non-dominated vector instead of a set of them (as in [117]). Nevertheless, it is possible to extend this approach to k players (where k is the number of objectives of a problem), and to have several Nash equilibrium points, with which the entire Pareto front of a problem can actually be found, although a *cooperative game* may be preferred in that case over a *non-cooperative* approach [124, 123].

5.6.2 Some applications.

- Truss optimization [37, 125].
- Minimization of the backscattering of aerodynamic reflectors [116, 117].

6. Pareto-Based Techniques

The idea of using Pareto-based fitness assignment was first proposed by Goldberg [58] to solve the problems of Schaffer's approach [138]. He suggested the use of nondominated ranking and selection to move a population toward the Pareto front in a multiobjective optimization problem. The basic idea is to find the set of strings in the population that are Pareto nondominated by the rest of the population. These strings are then assigned the highest rank and eliminated from further contention. Another set of Pareto nondominated strings are determined from the remaining population and are assigned the next highest rank. This process continues until the population is suitably ranked. Goldberg also suggested the use of some kind of niching technique to keep the GA from converging to a single point on the front [34]. A niching mechanism such as sharing [60] would allow the EA to maintain individuals all along the nondominated frontier.

6.1 Pure Pareto ranking

Although several variations of Goldberg's proposal have been proposed in the literature (see the following subsections), several authors have used what we call "pure Pareto ranking". The idea in this case is to follow Goldberg's proposal as stated in his book [58].

6.1.1 Advantages and disadvantages. The main weakness of Pareto ranking in general is that there is no efficient algorithm to check for nondominance in a set of feasible solutions (the conventional process is $O(kM^2)$, where k is the number of objectives and M is the population size). Therefore, any traditional algorithm to check for Pareto dominance exhibits a serious degradation in performance as we increase the size of the population and the number of objectives. Also, the use of sharing requires to estimate the value of the sharing factor, which is not easy, and the performance of the method relies a lot on this value. However, Pareto ranking is the most appropriate way to generate an entire Pareto front in a single run of an EA and its main advantage is that the approach is less succeptible to the shape or continuity of the Pareto front, whereas these two issues are a serious concern for traditional mathematical programming techniques.

6.1.2 Applications.

- Optimal location of a network of groundwater monitoring wells [18].

- Pump scheduling [141, 136].
- Feasibility of full stern submarines [158].
- Optimal planning of an electrical power distribution system [121].

6.2 MOGA

Fonseca and Fleming [52] proposed a scheme called "Multi-Objective Genetic Algorithm" (MOGA), in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual[4] x_i at generation t, which is dominated by $p_i^{(t)}$ individuals in the current generation. Its current position in the individuals' rank can be given by [52]:

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{1.20p}$$

All nondominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface.

Fitness assignment is performed in the following way [52]:

1 Sort population according to rank.

2 Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $n \leq N$) in the way proposed by Goldberg [58], according to some function, usually linear, but not necessarily.

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

As Goldberg and Deb [59] point out, this type of blocked fitness assignment is likely to produce a large selection pressure that might produce premature convergence. To avoid that, Fonseca and Fleming [52] used a niche-formation method to distribute the population over the Pareto-optimal region, but instead of performing sharing on the parameter values, they have used sharing on the objective function values [150].

[4] An individual encodes the decision variables of the problem.

6.2.1 Advantages and disadvantages. It has been indicated in the literature [150, 31] that the main drawback of MOGA is that it performs sharing on the objective value space, which implies that two different vectors with the same objective function values can not exist simultaneously in the population under this scheme. This is apparently undesirable, because these are precisely the kind of solutions that the user normally wants. However, nothing in the algorithm precludes it from performing sharing in decision variable space, and apparently this choice has been taken in some of the applications reported below.

The main advantage of MOGA is that it is efficient and relatively easy to implement [27, 162]. Its main weakness is that, as all the other Pareto ranking techniques, its performance is highly dependent on an appropriate selection of the sharing factor. However, it is important to add that Fonseca and Fleming [52] have developed a good methodology to compute this value for their approach.

6.2.2 Some applications.

- Co-synthesis of hardware-software embedded systems [39].
- Design of active magnetic bearing controllers [140].
- Fault Diagnosis [100, 101, 99].
- Plane truss optimization [109, 3].

6.3 NSGA

Srinivas and Deb [150] proposed the "Nondominated Sorting Genetic Algorithm" (NSGA). This algorithm is based on several layers of classifications of the individuals as shown in Figure 1.1c. Before selection is performed, the population is ranked on the basis of nondomination: all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population are classified. A stochastic remainder proportionate selection was adopted by the authors. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows to search for nondominated regions, and results in convergence of the population toward such regions. Sharing, by its part, helps to

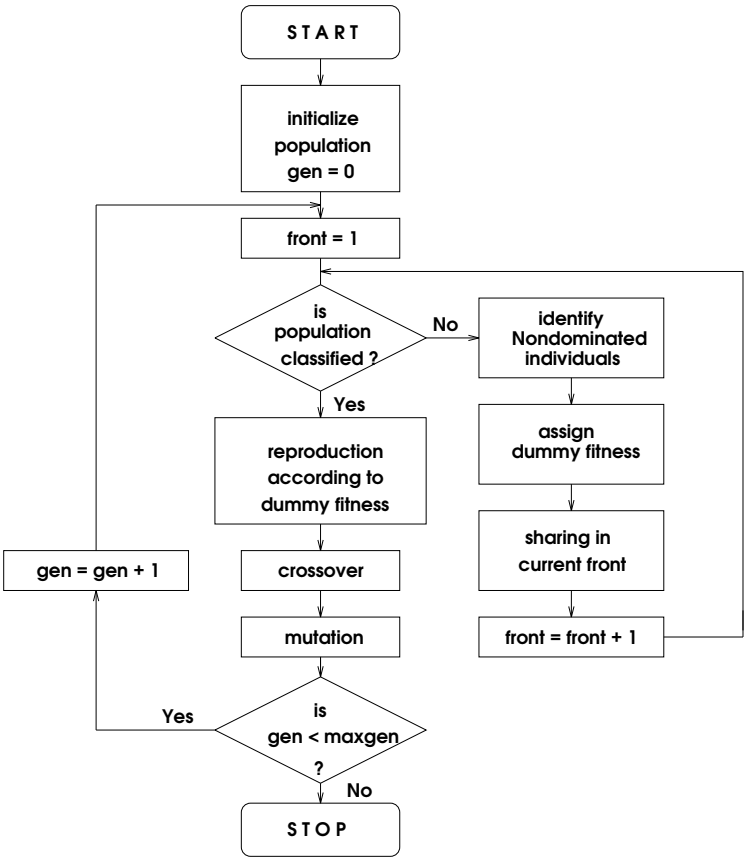

Figure 1.1c. Flowchart of the Nondominated Sorting Genetic Algorithm (NSGA).

distribute the population over this region (i.e., the Pareto front of the problem).

6.3.1 Advantages and disadvantages. Some researchers have reported that NSGA has a lower overall performance than MOGA, and it seems to be also more sensitive to the value of the sharing factor than MOGA [27, 162]. Other authors [180] report that the NSGA performed quite well in terms of "coverage" of the Pareto front (i.e., it spreads in a more uniform way the population over the Pareto front) when applied to the 0/1 knapsack problem, but in these experiments no comparisons with MOGA were provided.

In any case, Deb et al. [33] have recently proposed a new version of this algorithm, called NSGA-II, which is more efficient (computationally speaking), uses elitism and a crowded comparison operator that keeps diversity without specifying any additional parameters. The new approach has not been extensively tested yet, but it certainly looks promising.

6.3.2 Some applications.

- Computational fluid dynamics [98].
- Design of multilayer microwave absorbers [169], and thinned antenna arrays with digital phase shifters [168].
- Robust trajectory tracking problems [8].
- Design of optimal earth orbiting satellite constellations [103].

6.4 NPGA

Horn and Nafpliotis [77, 78] proposed a tournament selection scheme based on Pareto dominance. Two individuals randomly chosen are compared against a subset from the entire population (typically, around 10% of the population). When both competitors are either dominated or nondominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing [60].

The pseudocode for Pareto domination tournaments assuming that all of the objectives are to be maximized is presented below [77]. S is an array of the N individuals in the current population, *random_pop_index* is an array holding the N indices of S, in a random order, and t_{dom} is the size of the comparison set.

function selection /* Returns an individual from the current population S */
begin
  shuffle(random_pop_index); /* Re-randomize random index array */

```
candidate_1 = random_pop_index[1];
candidate_2 = random_pop_index[2];
candidate_1_dominated = false;
candidate_2_dominated = false;
for comparison_set_index = 3 to t_dom + 3 do
/* Select t_dom individuals randomly from S */
    begin
        comparison_individual = random_pop_index[comparison_set_index];
        if S[comparison_individual] dominates S[candidate_1]
            then candidate_1_dominated = true;
        if S[comparison_individual] dominates S[candidate_2]
            then candidate_2_dominated = true;
    end /* end for loop */
if ( candidate_1_dominated AND ¬ candidate_2_dominated )
    then return candidate_2;
else if ( ¬ candidate_1_dominated AND candidate_2_dominated )
    then return candidate_1;
else
    do sharing;

end
```

Horn and Nafpliotis [77, 78] also arrived at a form of fitness sharing in the objective domain, and suggested the use of a metric combining both the objective and the decision variable domains, leading to what they called *equivalent class sharing*.

6.4.1 Some applications.

- Fault tolerant system design [139].
- Planning of a traffic route [64].
- Analysis of experimental spectra and monochromatic images [62].
- Partitioning and allocation of objects in heterogeneous distributed environments [17].

6.4.2 Advantages and disadvantages. Since this approach does not apply Pareto selection to the entire population, but only to a segment of it at each run, its main advantage is that it is very fast and that it produces good nondominated fronts that can be kept for a large number of generations [27, 162]. However, its main disadvantage is that

besides requiring a sharing factor, this approach also requires a good choice of the size of the set against which the two reference individuals will be compared (i.e., the tournament size), in order to perform well. This adds an extra parameter to the EA, which is also subject to certain fine tuning. Also, the NPGA has normally been used with population sizes considerably larger than usual with other approaches so that the noise of the selection method can be tolerated by the emerging niches in the population [53].

6.5 Non-generational approaches

Valenzuela-Rendón and Uresti-Charre [161] proposed a GA that uses non-generational selection and in which the fitness of an individual is calculated incrementally. The idea comes from Learning Classifier Systems (LCS),[5] in which it has been shown that a simple replacement of the worst individual in the population followed by an update of fitnesses of the rest of the population works better than a traditional (generational) GA. In the context of multiobjective optimization, what the authors did was to transform the problem with N objectives into another one with only two objectives: the minimization of domination count (weighted average of the number of individuals that have dominated this individual so far) and the minimization of the moving niche count (weighted average of the number of individuals that lie close according to a certain sharing function). Then, this biobjective optimization problem is transformed into a single objective optimization problem by performing a linear combination of these two objectives.

More recently, Borges & Barbosa [9] proposed another non-generational GA that reduces all the objectives of the problem to two measures related to dominance and population distribution. Such measures, however, are different in this case. The domination measure expresses the state of domination of a certain individual with respect to the current population. The neighbor density measure represents the size of the niche in which a certain individual is in. Fitness is then computed using a combination of these two measures. This approach presents several differences with respect to the previous one. For example, the dominance and neighborhood measures in this case consider the entire population instead of using a sampling of the population (as in the previous approach). Also, the several parameters required by the previous approach become unnec-

[5] A classifier system is a machine learning system that learns syntactically simple string rules to guide its performance in an arbitrary environment [58].

essary. This approach also compared well with respect to other EMOO techniques in several test functions.

6.5.1 Advantages and disadvantages. The approach proposed by Valenzuela-Rendón and Uresti-Charre (1997) is really a more elaborate version of the weighted ranking techniques used by Bentley and Wakefield [6] (particularly the technique that they called weighted average ranking—WAR). The main advantage of this approach is that it seems to provide good distributions in an efficient manner using well-known techniques taken from LCS. However, its main disadvantage is that it does not seem feasible to incorporate in this approach preferences of the objectives defined by the decision maker, which may be a drawback in real-world applications. Also, it does not seem clear how to define the six additional parameters (two more are fixed by the authors) required by this algorithm, which apparently require an empirical fine tuning as the other normal parameters of the GA (e.g., crossover and mutation rates).

The approach proposed by Borges & Barbosa [9] eliminates most of the drawbacks of Valenzuela-Rendón and Uresti-Charre's technique. However, the use of this approach has not been too widespread and we are not aware of its performance with a larger amount of objectives and in constrained search spaces.

6.5.2 Some applications.

- Structural optimization [10].

7. Recent approaches

Recently, several new EMOO approaches have been developed. We consider important to discuss briefly at least two of them: PAES and SPEA. Also, we will discuss some of our recent work regarding the use of a micro-genetic algorithm for multiobjective optimization.

7.1 PAES

The *Pareto Archived Evolution Strategy* (PAES) was introduced by Knowles & Corne [85]. The idea of the approach is very simple. A (1+1) evolution strategy (i.e., a single parent that generates a single offspring) is used in combination with a historical archive that records all nondominated solutions previously found. This archive is used as a reference set against which each mutated individual will be compared. This is analogous to the tournament competitions held with the NPGA [78].

PAES also uses a novel approach to keep diversity, which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its "coordinates" or "geographical location"). A map of this grid is maintained, indicating the number of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space). Furthermore, the procedure has a lower computational complexity than traditional niching methods [85].

Since PAES is a very recent approach, only a few applications of it have been reported in the literature, all of them related to telecommunications problems [84, 85, 86].

7.2 SPEA

The *Strength Pareto Evolutionary Algorithm* (SPEA) was introduced by Zitzler & Thiele [181]. This approach was conceived as a way of integrating different EMOO techniques. SPEA uses an archive containing nondominated solutions previously found (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set. For each individual in this external set, a strength value is computed. This strength is similar to the ranking value of MOGA, since it is proportional to the number of solutions to which a certain individual dominates. The fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. Additionally, a clustering technique called "average linkage method" [107] is used to keep diversity.

SPEA has been used to explore trade-offs of software implementations for programmable digital signal processors (PDSP) [179] and to solve 0/1 knapsack problems [181].

7.3 A micro-GA for multiobjective optimization

Currently, we have been experimenting with a a micro-GA (a GA with small population and a reinitialization mechanism [88]) for multiobjective optimization [26]. This approach uses two memories: the population memory, which is used as the source of diversity of the approach, and the external memory, which is used to archive members of the Pareto optimal set. Population memory is divided in two parts: a replaceable and a non-replaceable portion (the percentages of each can be regulated by the user).

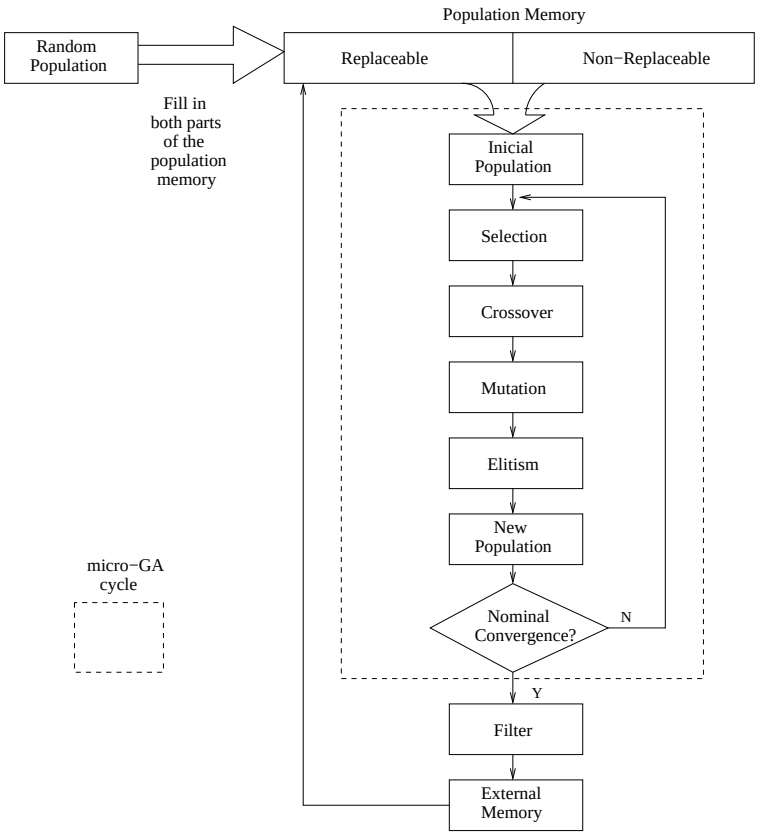

Figure 1.1d. Diagram that illustrates the way in which the micro-GA for multiobjetive optimization works.

The way in which this technique works is illustrated in Figure 1.1d. First, an initial random population is generated. This population feeds the population memory, which is divided in two parts as indicated before. The non-replaceable portion of the population memory will never change during the entire run and is meant to provide the diversity required by the algorithm. The initial population of the micro-GA at the beginning of each of its cycles is taken (with a certain probability) from both portions of the population memory as to allow a greater diversity.

During each cycle, the micro-GA undergoes conventional genetic operators: tournament selection, two-point crossover, uniform mutation, and elitism (regardless of the amount of nondominated vectors in the population only one is arbitrarily selected at each generation and copied intact to the following one).

This approach uses three types of elitism. The first is based on the notion that if we store the nondominated vectors produced from each cycle of the micro-GA, we will not lose any valuable information obtained from the evolutionary process. The second is based on the idea that if we replace the population memory by the nominal solutions (i.e., the best solutions found when nominal convergence is reached), we will gradually converge, since crossover and mutation will have a higher probability of reaching the true Pareto front of the problem over time. This notion was hinted at by Goldberg [58]. Nominal convergence, in this case, is defined in terms of a certain (low) number of generations (typically, two to five in our case). However, similarities among the strings (either at the phenotypical or genotypical level) could also be used as a criterion for convergence. The third type of elitism is applied at certain intervals (defined by a parameter called "replacement cycle"). We take a certain amount of points from all the regions of the Pareto front generated so far and we use them to fill in the replaceable memory. Depending on the size of the replaceable memory, we choose as many points from the Pareto front as necessary to guarantee a uniform distribution. This process intends to use the best solutions generated so far as the starting point for the micro-GA, so that we can improve them (either by getting closer to the true Pareto front or by getting a better distribution). This also avoids that the content of the replaceable memory becomes homogeneous.

To keep diversity in the Pareto front, the micro-GA uses an approach similar to the adaptive grid proposed by Knowles & Corne [85]. The idea is that once the archive that stores nondominated solutions has reached its limit, the search space that this archive covers is divided, assigning a set of coordinates to each solution. Then, each newly generated nondominated solution will be accepted only if the geographical location to

where the individual belongs is less populated than the most crowded location. Alternatively, the new nondominated solution could also be accepted if the individual belongs to a location outside the previously speficied boundaries. In other words, the less crowded regions are given preference so that the spread of the individuals on the Pareto front can be more uniform.

This approach allows the regulation of the amount of points from the Pareto front that the user wishes to find through the size of the external memory. Our preliminary results indicate that our micro-GA is able to generate the Pareto front of difficult test functions (i.e., disconnected and concave Pareto fronts) that have been previously adopted to evaluate EMOO techniques. Furthermore, the approach seems to exhibit a lower computational cost than the NSGA II and PAES while obtaining Pareto fronts of similar quality. However, it also requires certain additional parameters and the sensitivity of the approach to them is still subject of ongoing research [26].

8. Diversity

Due to stochastic errors associated with its genetic operators, evolutionary algorithms tend to converge to a single solution when used with a finite population [34]. As long as our goal is to find the global optimum (or at least a very good approximation of it), this behavior is acceptable. However, there are certain applications in which we are interested in finding not one, but several solutions. Multiobjective optimization is certainly one of those applications, because we want to find the entire Pareto front of a problem, and not only a single nondominated solution. The question is then how to keep the EA from converging to a single solution.

Early evolutionary computation researchers identified this convergence phenomenon of EAs, called *genetic drift* [36], and found that it happens in Nature as well. They correctly stated that the key to solve this problem is to find a way of preserving diversity in the population, and several proposals, modelled after natural systems were made. Holland [76] suggested the use of a "crowding" operator, which was intended to identify situations in which more and more individuals dominate an environmental niche, since in those cases the competition for limited resources increases rapidly, which will result in lower life expectancies and birth rate. DeJong [36] experimented with such a *crowding* operator, which was implemented by having a newly formed offspring to replace the existing individual more similar to itself. The similarity between two individuals was measured in the genotype by counting the number

of bits along each chromosome that were equal in the two individuals being compared. DeJong used two parameters in his model: generation gap (G) and crowding factor (CF) [34]. The first parameter indicates the percentage of the population that is allowed to reproduce. The second parameter specifies the number of individuals initially selected as candidates to be replaced by a particular offspring [36]. Therefore, CF=1 means that no crowding will take place, and as we increase the value of CF, it becomes more likely that similar individuals replace one another [36].

Goldberg and Richardson [60] used a different approach in which the population was divided in different subpopulations according to the similarity of the individuals in two possible solution spaces: the decoded parameter space (phenotype) and the gene space (genotype). They defined a sharing function $\phi(d_{ij})$ as follows [60]:

$$\phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{sh}}\right)^{\alpha}, & d_{ij} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \tag{1.21q}$$

where normally $\alpha = 1$, d_{ij} is a metric indicative of the distance between designs i and j, and σ_{share} is the sharing parameter which controls the extent of sharing allowed. The fitness of a design i is then modified as:

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{M} \phi(d_{ij})} \tag{1.22r}$$

where M is the number of designs located in vicinity of the i-th design.

Deb and Goldberg [34] proposed a way of estimating the parameter σ_{share} in both phenotypical and genotypical space. In phenotypical sharing, the distance between 2 individuals is measured in decoded parameter space, and can be calculated with a simple Euclidean distance in a p-dimensional space, where p refers to the number of variables encoded in the GA; the value of d_{ij} can then be calculated as:

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{k,i} - x_{k,j})^2} \tag{1.23s}$$

where $x_{1,i}, x_{2,i}, \ldots, x_{p,i}$ and $x_{1,j}, x_{2,j}, \ldots, x_{p,j}$ are the variables decoded from the EA.

To estimate the value of σ_{share}, Deb and Goldberg [34] proposed to use the expression:

$$\sigma_{share} = \frac{r}{\sqrt[p]{q}} = \frac{\sqrt{\sum_{k=1}^{p}(x_{k,max} - x_{k,min})^2}}{\sqrt[p]{2q}} \tag{1.24t}$$

where r is the volume of a p-dimensional hypersphere of radius σ_{share} and q is the number of peaks that we want the EA to find.

In genotypical sharing, d_{ij} is defined as the Hamming distance between the strings and σ_{share} is the maximum number of different bits allowed between the strings to form separate niches in the population. The experiments performed by Deb and Goldberg [34] showed sharing as a better way of keeping diversity than crowding, and indicated that phenotypic sharing was better than genotypic sharing.

It should be added that much further work has been done regarding keeping the diversity in the population. Deb and Goldberg [34] suggested the use of restrictive mating with respect to the phenotypic distance. The idea is to allow two individuals to reproduce only if they are very similar (i.e., if their phenotypic distance is less than a factor called σ_{share}). This is intended to produce distinct "species" (mating groups) in the population [106]. Other researchers such as Eshelman [47] and Eshelman & Schaffer [48] did exactly the opposite: they did not allow mating between individuals that were too similar (they said to be "preventing incest").

Smith et al. (1993) [148] proposed an approach, modelled after the immune system, that can maintain the diversity of the population without the use of an explicit sharing function. This approach has been actually used by Hajela et al. [66, 68] to handle constraints in structural optimization problems.

Poloni and Pediroda [119] proposed an interesting alternative to preserve diversity. They called their approach "local Pareto selection", and it basically consists of placing the population on a toroidal grid and choosing the members of the local tournament by means of a random walk in the neighborhoods of the given grid point.

Kita et al. [83] proposed the so called "Thermodynamical Genetic Algorithm" (TDGA) to maintain diversity when using a Pareto ranking technique for multiobjective optimization. The TDGA is inspired by the principle of minimal free energy used in simulated annealing [82]. The idea is to select the individuals for a new generation in such a way that the free energy F is minimized, and

$$F = \langle E \rangle - HT \tag{1.25u}$$

where $\langle E \rangle$ is the mean energy of the system, H is the entropy and T is the temperature. The diversity of the population is controlled by

adjusting T according to a certain schedule (as in simulated annealing). Presumably, T is less sensitive to the population size and to the size of the feasible region than traditional sharing functions [156].

Goldberg & Wang [61] proposed a coevolutionary adaptive niching scheme inspired on the economic model of monopolistic competition. The idea is to create two populations, one of businessmen and another one of customers. The population of customers is in fact the population of solutions to our problem (e.g., members of the Pareto optimal set) that will try to maximize a certain set of criteria, whereas the businessmen will try to locate themselves in such a way that their "profit" can be maximized. Customers will create niches according to their own criteria being optimized. Businessmen will then have to adapt to the current fitness landscape so that they can serve as many customers as possible. By enforcing a competition between these two populations, a uniform spread of the population of customers is expected to emerge.

Tan et al. [157] proposed the use of a dynamic sharing distance. The idea is to approximate the curvarture of the trade-off curve formed by the nondominated solutions in objective space. The procedure then attempts to perform a uniform distribution of points along the Pareto front without requiring any prior parameters (the information required to bias the search is obtained from the evolutionary process itself).

Deb et al. [33] proposed the use of a crowding distance measure which represents the amount of solutions that lie within a certain neighborhood (in objective space). This approach is more efficient (computationally speaking) than traditional fitness sharing and does not require an extra parameter (i.e., σ_{share}).

Several other proposals exist (see [96] for a more detailed review of approaches to keep diversity). In fact, some researchers tend to develop their own variation of a certain technique or (in a few cases) to design an entirely new approach.

9. Test Functions

A very important aspect of this research area that has been generally disregarded in the technical literature is the use of appropriate test functions. In the early days of evolutionary multiobjective optimization, many researchers tested their approaches only with the two classic test functions provided by Schaffer in his seminal work on EMOO [138]. These functions are not only very simple (they have only two objectives), but are also unconstrained and do not show any of the most important aspects that would be interesting to analyze using an EMOO

approach (e.g., ability of the algorithm to deal with concave or discontinuous Pareto fronts).

In recent years, several researchers have addressed the design of standard benchmarks against which any EMOO algorithm can be validated. Deb [31] has proposed ways to create controllable test problems for evolutionary multiobjective optimization techniques using single-objective optimization problems as a basis. Under this proposal, some problems that have been of great interest in evolutionary computation could be transformed into multiobjective optimization problems (e.g., deceptive and massively multimodal problems). Recently, this study has been extended to constrained multiobjective optimization problems [35].

Van Veldhuizen and Lamont [164, 165] have also proposed some guidelines to design a test function suite for evolutionary multiobjective optimization techniques (mainly combinatorial optimization problems). In more recent work, Van Veldhuizen [162] has also summarized most of the test functions that have been previously suggested in the specialized literature.

Nevertheless, a more complete test suite is still required. Such a suite should contain problems of different degrees of difficulty (both constrained and unconstrained) and some real-world applications. If possible, good approximations of the true Pareto front of each problem should also be included. Furthermore, the test suite should be easily accessible (i.e., through the Internet), so that anyone interested in using it could use it. Such a test suite would become an important benchmark to validate any new EMOO technique developed.

10. Metrics

Closely related to the previous issue is the importance of defining good metrics to assess the effectiveness of an EMOO technique. The definition of such metrics is not an easy task since it is difficult to compare two vectors. Three are normally the issues to take into consideration to design a good metric in this domain [178]:

1 Minimize the distance of the Pareto front produced by our algorithm with respect to the true Pareto front (assuming we know its location).

2 Maximize the spread of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible.

3 Maximize the amount of elements of the Pareto optimal set found.

There are several interesting proposals in the specialized literature that take into consideration at least one of these issues. The main proposals will briefly be described next:

1 **Enumeration**: Van Veldhuizen & Lamont [164, 162] have proposed the use of parallel processing techniques to enumerate the entire intrinsic search space explored by an EA. This obviously allows to obtain the Pareto front that is global with respect to the granularity used. Knowing the global Pareto front of the problem, we can compare results against it, and devise different metrics for estimating how well our EA is performing.

   This approach might work with relatively short binary strings (Van Veldhuizen & Lamont [164] report success with strings ≤ 26 bits), but might not be suitable when using alphabets of higher cardinality (e.g., real-coded GAs) or longer binary strings.

2 **Spread**: Srinivas and Deb [150] proposed to measure the "spread" of points along the Pareto front using a statistical metric such as the chi-square distribution. This metric also assumes knowledge of the true Pareto front, and emphasizes the good distribution of points (determined through a set of niches) rather than a direct comparison between our Pareto front and the true Pareto front.

3 **Attainment Surfaces**: Fonseca and Fleming [54] proposed to draw a boundary in objective space separating those points which are dominated (by a certain set of points) from those which are nondominated. Such boundary was called "attainment surface". This attainment surface could then be used to determine the quality and the distribution of the nondominated points found by an EMOO approach. Multiple runs would then have to be performed and standard non-parametric statistical procedures would have to be applied to evaluate the quality of the nondominated vectors found. Several EMOO approaches can then be compared using this approach, but it is unclear how we can really assess how much better a certain approach is with respect to others [178].

4 **Generational Distance**: Van Veldhuizen & Lamont [163] proposed the use of a metric that estimates how far our current Pareto front is from the true Pareto front of a problem. This metric uses the Euclidean distance (measured in objective space) between each vector and the nearest member of the true Pareto front. Similar metrics have also been proposed by Schott [139], Rudolph [133], and Zitzler et al. [178]. The problem with this metric is that only

distance to the true Pareto front is considered and not uniform spread along the Pareto front.

5 **Coverage**: Zitzler and Thiele [181] proposed two measures: the first concerns the size of the objective value space area which is covered by a set of nondominated solutions and the second compares directly two sets of nondominated solutions, using as a metric the fraction of the Pareto front covered by each of them. The first metric combines the three issues previously mentioned (distance, spread and amount of elements of the Pareto optimal set found) into a single value. Therefore, sets differing in more than one criterion could not be distinguished. The second metric is similar to the attainment surfaces of Fonseca & Fleming and it also has the same drawbacks.

In more recent work, Zitzler et al. [178] proposed several additional metrics for EMOO algorithms and also performed a detailed comparative study using such metrics. More work in this area is, however, still needed.

11. Applications

An analysis of the evolution of the EMOO literature reveals some interesting facts. From the first EMOO approach published in 1985 [138] up to the first survey of the area published in 1995 [53], the number of published papers related to EMOO is relatively low. However, from 1995 to our days, the increase of EMOO-related papers is exponential. Today, the EMOO repository registers over 850 papers, from which a vast majority are applications. Given the large number of EMOO papers that currently exist, we will not attempt to produce a detailed review of applications in this section. Instead, we will delineate the most popular application fields, indicating some of the specific areas within them in which researchers have focused their efforts.

Current EMOO applications can be roughly classified in three large groups: engineering, industrial and scientific. Some specific areas within each of these groups are indicated next.

We will start with the engineering applications, which are, by far, the most popular in the literature. This should not be too surprising, since engineering disciplines normally have problems with better understood mathematical models which facilitates the use of evolutionary algorithms. A representative sample of engineering applications is the following (aeronautical engineering seems to be the most popular subdiscipline within this group):

- Electrical engineering [159, 108, 122]
- Hydraulic engineering [141, 136, 174]
- Structural engineering [95, 24, 173]
- Aeronautical engineering [72, 111, 167]
- Robotics [41, 57, 112]
- Control [40, 97, 42]
- Telecommunications [104, 86, 175]
- Civil engineering [49, 5, 81]
- Transport engineering [120, 2, 93]

Industrial applications occupy the second place in popularity in the EMOO literature. Within this group, scheduling is the most popular subdiscipline. A representative sample of industrial applications is the following:

- Design and manufacture [63, 127, 113]
- Scheduling [155, 4, 14]
- Management [11, 87, 43]

Finally, we have a variety of scientific applications, from which the most popular are (for obvious reasons) those related to computer science:

- Chemistry [170, 74, 90]
- Physics [115, 117, 62]
- Medicine [176, 145, 92]
- Computer science [147, 13, 7]

The above distribution of applications indicates a strong interest for developing real-world applications of EMOO algorithms (something not surprising considering that most problems are of a multiobjective nature). Furthermore, the previous sample of EMOO applications should give a general idea of the application areas that have not been explored yet, some of which are mentioned in the following section.

12. Future Research Paths

Despite the noticeable increment in the amount of EMOO research in the last few years, there are still several open research areas. Some of them will be described next.

- New Approaches: Several new techniques have been proposed in the last few years. However, only a fistful of them have been adopted by a significant portion of the scientific community. In fact, some of these techniques widely used are already undergoing updates. MOGA [52], for example, has been recently hybridized with neural networks to improve its efficiency [42]. The NSGA [150] has undergone significant changes in its algorithmic structure and its diversity preservation approach, in order to make it more efficient [33]. But this may be only the beginning. We believe that the next few years will witness the development of many other new approaches (and updates of those currently in use). However, the focus of these developments will be different. Right now, for example, efficiency is the main issue. Researchers try to defeat the inherent inefficiency associated with Pareto ranking and with traditional niching in order to produce new approaches whose computational cost is lower and therefore more suitable to be scaled to larger (real-world) problems. The use of local search with archival memories [85, 79, 26] and parallel selection strategies [104, 99, 73] are two of the alternatives currently explored, but several others are also possible. For example, little attention has been paid to the data structures used to store nondominated vectors in the current EMOO literature. In contrast, operational researchers have used efficient data structures for discrete multiobjective optimization (e.g., domination-free quad trees where a nondominated vector can be retrieved from the tree very efficiently). Checking if a new vector is dominated by the vectors in one of these trees can also be done very efficiently [65].

  We also believe that multiobjective extensions of other heuristics will become popular in the next few years [102, 71, 28, 146, 126, 160, 16], as well as the hybridization of EAs with other heuristics (particularly to deal with multiobjective combinatorial optimization problems) [91, 38].

- New Applications: Despite the large amount of applications reported in the literature, many other domains remain practically unexplored. For example, the coordination of distributed agents is a problem that frequently involves globally conflicting solutions to

multiple (local) objectives and it therefore lends itself naturally to a multiobjective optimization approach [118]. Other domain areas such as shape design [151] and constraint-handling [20] seem also very appropriate for testing new EMOO techniques. Additionally, EMOO researchers have not paid enough attention to multiobjective combinatorial optimization problems, which are not only challenging, but have also been studied in great depth [45]. Few EMOO researchers have actually used well-studied combinatorial optimization problems such as the 0/1 knapsack problem to validate EMOO approaches [181, 79, 80]. Finally, more real-world applications of EMOO techniques are also lacking in the current literature.

- Theory: There is a noticeable lack of research in theoretical issues related to EMOO. Most of the current studies available deal with convergence issues of EMOO algorithms [133, 134, 69, 70, 163], or with ways to compute niche sizes [52, 78]. However, many other important areas have not been studied. It would be very interesting to study, for example, the structure of fitness landscapes in MOPs [172, 1]. Such study could provide some insights regarding the sort of problems that are particularly difficult for EAs and could also provide clues regarding the design of more powerful EMOO techniques. Furthermore, there is a need for detailed studies of the different aspects involved in the parallelization of EMOO techniques (e.g., load balancing, impact on Pareto convergence, performance issues, etc.), including new algorithms that are more suitable for parallelization than those currently in use.

- Benchmarks: We have mentioned some of the current work regarding the design of test functions that can be properly used to validate EMOO approaches. Despite these recent efforts, more work in this area is still necessary. Other domains such as constraint-handling in the context of single-objective optimization could be used to validate in a more quantitative way the performance of EMOO approaches [20, 153]. A more systematic way of designing test functions is also required, focusing on the aspects that are more important to evaluate from an EMOO algorithm (e.g., its ability to deal with concave, discontinuous and highly-constrained search spaces). Closely related to this issue is the notorious lack of comparative studies in the current literature. Also, it is necessary to have more in-depth studies of metrics appropriate to evaluate the performance of EMOO techniques. Some of the efforts in that

direction have also been discussed in this chapter, but more work is still required.

13. Summary

This chapter has reviewed some of the most important research done in evolutionary multiobjective optimization. We have discussed the main EMOO techniques currently in use, together with their advantages and disadvantages and some of their applications. Also, we have discussed the importance of diversity in the context of multiobjective optimization, reviewing some of the most important proposals found in the literature. Then, we have included a brief discussion of test functions and metrics used to validate EMOO techniques, addressing their importance to estimate (in a quantitative way) how good a certain technique is with respect to others. Finally, we have provided a representative sample of the types of applications of EMOO algorithms reported in the literature.

In the last section of this chapter, we have discussed some potential research areas that would be interesting to explore in more depth in the next few years. Some of them are already being studied, but others have not been addressed by any EMOO researchers. We expect that the general view of this relatively new field presented in this chapter can be of some use to the newcomers who want to become familiar with the research in this area in order to identify some possible research topic. Additionally, we also expect mature researchers and practitioners interested in evolutionary multiobjective optimization to find enough pointers as to allow them to initiate work in this area. As we mentioned before, this research discipline still has several open areas and possible application domains for those who may be interested.

References

[1] Lee Altenberg. NK fitness landscapes. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter B2.7.2, pages B2.7:5–B2.7:10. Oxford University Press, New York, NY, 1997.

[2] Jessica M. Anderson, Tessa M. Sayers, and M. G. H. Bell. Optimization of a fuzzy logic traffic signal controller by a multiobjective genetic algorithm. In *Proceedings of the Ninth International Conference on Road Transport Information and Control*, pages 186–190, London, April 1998. IEE.

[3] Shapour Azarm, Brian J. Reynolds, and Sanjay Narayanan. Comparison of two multiobjective optimization techniques with and within genetic algorithms. In *CD-ROM Proceedings of the 25th ASME Design Automation Conference*, volume Paper No. DETC99/DAC-8584, Las Vegas, NV, September 1999.

[4] Tapan P. Bagchi. *Multiobjective Scheduling by Genetic Algorithms*. Kluwer Academic Publishers, Boston, MA, 1999.

[5] Richard Balling and Scott Wilson. The maximim fitness function for multi-objective evolutionary computation: Application to city planning. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 1079–1084, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[6] P. J. Bentley and J. P. Wakefield. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. In P. K. Chawdhry, R. Roy, and R. K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, Part 5, pages 231–240, London, June 1997. Springer Verlag London Limited. (Pre-

sented at the 2nd On-line World Conference on Soft Computing in Design and Manufacturing (WSC2)).

[7] Ester Bernadó i Mansilla and Josep M. Garrell i Guiu. MOLeCS: Using multiobjective evolutionary algorithms for learning. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 696–710. Springer-Verlag. Lecture Notes in Computer Science No. 1993, Berlin, Germany, 2001.

[8] Anna L. Blumel, Evan J. Hughes, and Brian A. White. Fuzzy autopilot design using a multiobjective evolutionary algorithm. In *2000 Congress on Evolutionary Computation*, volume 1, pages 54–61, Piscataway, NJ, July 2000. IEEE Service Center.

[9] Carlos C.H. Borges and Helio J.C. Barbosa. A non-generational genetic algorithm for multiobjective optimization. In *2000 Congress on Evolutionary Computation*, volume 1, pages 172–179, Piscataway, NJ, July 2000. IEEE Service Center.

[10] Carlos Cristiano Hasenclever Borges. *Algoritmos Genéticos para Otimização em Dinâmica de Estruturas.* PhD thesis, Coordinação dos Programas de Põs-Graduação da Universidade Federal do Rio de Janeiro, Engenharia Civil, 1999. (In Portuguese).

[11] Rob A. C. M. Broekmeulen. Facility management of distribution centers for vegetables and fruits. In J. Biethahn and Volker Nissen, editors, *Evolutionary Algorithms in Management Applications*, pages 199–210. Springer-Verlag, Berlin, Germany, 1995.

[12] W. D. Cannon. *The Wisdom of the Body.* Norton and Company, New York, NY, 1932.

[13] Alain Cardon, Thierry Galinho, and Jean-Philippe Vacher. Genetic algorithms using multi-objectives in a multi-agent system. *Robotics and Autonomous Systems*, 33(2–3):179–190, November 2000.

[14] W. Matthew Carlyle, Bosun Kim, John W. Fowler, and Esma S. Gel. Comparison of multiple objective genetic algorithms for parallel machine scheduling problems. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 472–485. Springer-Verlag. Lecture Notes in Computer Science No. 1993, Berlin, Germany, 2001.

[15] H. W. Chen and N.-B. Chang. Water pollution control in the river basin by fuzzy genetic algorithm-based multiobjective programming modeling. *Water Science and Technology*, 37(8):55–63, 1998.

[16] A.J. Chipperfield, J.F. Whidborne, and P.J. Fleming. Evolutionary algorithms and simulated annealing for MCDM. In T. Gal, T.J. Stewart, and T. Hanne, editors, *Multicriteria Decicion Making—Advances in MCDM Models, Algorithms, Theory, and Applications*, pages 16.1–16.32. Kluwer Academic Publishing, Boston, MA, 1999.

[17] Seunghoon Choi and Chisu Wu. Partitioning and allocation of objects in heterogeneous distributed environments using the niched pareto genetic-algorithm. In *Proceedings of 1998 Asia Pacific Software Engineering Conference (APSEC 98)*, Taipei, Taiwan, December 1998.

[18] Scott E. Cieniawski, J. W. Eheart, and S. Ranjithan. Using genetic algorithms to solve a multiobjective groundwater monitoring problem. *Water Resources Research*, 31(2):399–409, February 1995.

[19] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, August 1999.

[20] Carlos A. Coello Coello. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering Systems*, 17:319–346, 2000.

[21] Carlos A. Coello Coello. Treating constraints as objectives for single-objective evolutionary optimization. *Engineering Optimization*, 32(3):275–308, 2000.

[22] Carlos A. Coello Coello, Arturo Hernández Aguirre, and Bill P. Buckles. Evolutionary multiobjective design of combinational logic circuits. In Jason Lohn, Adrian Stoica, Didier Keymeulen, and Silvano Colombano, editors, *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pages 161–170, Los Alamitos, CA, July 2000. IEEE Computer Society.

[23] Carlos A. Coello Coello and Alan D. Christiansen. Two new GA-based methods for multiobjective optimization. *Civil Engineering Systems*, 15(3):207–243, 1998.

[24] Carlos A. Coello Coello and Alan D. Christiansen. Multiobjective optimization of trusses using genetic algorithms. *Computers and Structures*, 75(6):647–660, May 2000.

[25] Carlos A. Coello Coello, Alan D. Christiansen, and Arturo Hernández Aguirre. Using a new GA-based multiobjective optimization technique for the design of robot arms. *Robotica*, 16(4):401–414, July–August 1998.

[26] Carlos A. Coello Coello and Gregorio Toscano. A micro-genetic algorithm for multiobjective optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 127–141. Springer-Verlag. Lecture Notes in Computer Science No. 1993, Berlin, Germany, 2001.

[27] Carlos Artemio Coello Coello. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design.* PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, April 1996.

[28] P. Czyzak and A. Jaszkiewicz. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.

[29] Charles Darwin. *The Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life.* Random House, New York, NY, 1929. Originally published in 1859.

[30] Indraneel Das and John Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, 1997.

[31] Kalyanmoy Deb. Evolutionary algorithms for multi-criterion optimization in engineering design. In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, chapter 8, pages 135–161. John Wiley & Sons, Chichester, UK, 1999.

[32] Kalyanmoy Deb. Solving goal programming problems using multi-objective genetic algorithms. In *1999 Congress on Evolutionary*

Computation, pages 77–84, Piscataway, NJ, July 1999. IEEE Service Center.

[33] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[34] Kalyanmoy Deb and David E. Goldberg. An investigation of niche and species formation in genetic function optimization. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, San Mateo, CA, June 1989. Morgan Kaufmann Publishers.

[35] Kalyanmoy Deb, Amrit Pratap, and T. Meyarivan. Constrained test problems for multi-objective evolutionary 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 284–298. Springer-Verlag. Lecture Notes in Computer Science No. 1993, Berlin, Germany, 2001.

[36] A. K. DeJong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, Ann Arbor, MI, 1975.

[37] A. K. Dhingra and B. H. Lee. A genetic algorithm approach to single and multiobjective structural optimization with discrete-continuous variables. *International Journal for Numerical Methods in Engineering*, 37:4059–4080, 1994.

[38] Robert P. Dick and Niraj K. Jha. CORDS: hardware-software co-synthesis of reconfigurable real-time distributed embedded systems. In *Proceedings of the International Conference on Computer-Aided Design*, pages 62–68, November 1998.

[39] Robert P. Dick and Niraj K. Jha. MOGAC: A multiobjective genetic algorithm for hardware-software co-synthesis of hierarchical heterogeneous distributed embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(10):920–935, October 1998.

[40] D. C. Donha, D. S. Desanj, and M. R. Katebi. Genetic algorithm for weight selection in h_∞ control design. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 599–606, San Mateo, CA, July 1997. Morgan Kaufmann Publishers.

[41] Gerry V. Dozier, Shaun McCullough, Abdollah Homaifar, and Loretta Moore. Multiobjective evolutionary path planning via fuzzy tournament selection. In *IEEE International Conference on Evolutionary Computation (ICEC'98)*, pages 684–689, Piscataway, NJ, May 1998. IEEE Press.

[42] N.M. Duarte, A. E. Ruano, C.M. Fonseca, and P.J. Fleming. Accelerating multi-objective control system design using a neuro-genetic approach. In *2000 Congress on Evolutionary Computation*, volume 1, pages 392–397, Piscataway, NJ, July 2000. IEEE Service Center.

[43] Els I. Ducheyne, Robert R. De Wulf, and Bernard De Baets. Bi-objective genetic algorithm for forest management: a comparative study. In *Proceedings of the 2001 Genetic and Evolutionary Computation Conference. Late-Breaking Papers*, pages 63–66, San Francisco, CA, July 2001.

[44] F. Y. Edgeworth. *Mathematical Physics.* P. Keagan, London, England, 1881.

[45] Matthias Ehrgott and Xavier Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22:425–460, 2000.

[46] Neil H. Eklund and Mark J. Embrechts. GA-based multi-objective optimization of visible spectra for lamp design. In Cihan H. Dagli, Anna L. Buczak, Joydeep Ghosh, Mark J. Embrechts, and Okan Ersoy, editors, *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Complex Systems*, pages 451–456, New York, NY, November 1999. ASME Press.

[47] Larry J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In Gregory E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 265–283. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[48] Larry J. Eshelman and J. David Schaffer. Preventing premature convergence in genetic algorithms by preventing incest. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 115–122, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[49] Chung-Wei Feng, Liang Liu, and Scott A. Burns. Using genetic algorithms to solve construction time-cost trade-off problems. *Journal of Computing in Civil Engineering*, 10(3):184–189, 1999.

[50] Lawrence J. Fogel. *Artificial Intelligence through Simulated Evolution.* John Wiley, New York, NY, 1966.

[51] Lawrence J. Fogel. *Artificial Intelligence through Simulated Evolution. Forty Years of Evolutionary Programming.* John Wiley & Sons, New York, NY, 1999.

[52] 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, CA, 1993. Morgan Kaufmann Publishers.

[53] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995.

[54] 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*, pages 584–593. Springer-Verlag. Lecture Notes in Computer Science No. 1141, Berlin, Germany, September 1996.

[55] Michael P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153, Hillsdale, NJ, 1985. Lawrence Erlbaum.

[56] L. Gacôgne. Research of pareto set by genetic algorithm, application to multicriteria optimization of fuzzy controller. In *5th European Congress on Intelligent Techniques and Soft Computing EUFIT'97*, pages 837–845, Aachen, Germany, September 1997.

[57] Louis Gacôgne. Multiple objective optimization of fuzzy rules for obstacles avoiding by an evolution algorithm with adaptative operators. In Pavel Ošmera, editor, *Proceedings of the Fifth International Mendel Conference on Soft Computing (Mendel'99)*, pages 236–242, Brno, Czech Republic, June 1999. Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automation and Computer Science.

[58] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, Reading, MA, 1989.

[59] David E. Goldberg and Kalyanmoy Deb. A comparison of selection schemes used in genetic algorithms. In G.J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, San Mateo, CA, 1991.

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

[61] David E. Goldberg and Liwei Wang. Adaptive niching via coevolutionary sharing. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, chapter 11, pages 22–38. John Wiley and Sons, West Sussex, England, 1998.

[62] Igor E. Golovkin, Roberto C. Mancini, Sushil J. Louis, Richard W. Lee, and Lewis Klein. Multi-criteria search and optimization: an application to x-ray plasma spectroscopy. In *2000 Congress on Evolutionary Computation*, volume 2, pages 1521–1527, Piscataway, NJ, July 2000. IEEE Service Center.

[63] R. Groppetti and R. Muscia. On a genetic multiobjective approach for the integration and optimization of assembly product design and process planning. In P. Chedmail, J. C. Bocquet, and D. Dornfeld, editors, *Integrated Design and Manufacturing in Mechanical Engineering*, pages 61–70. Kluwer Academic Publishers, The Netherlands, 1997.

[64] P. Haastrup and A. Guimaraes-Pereira. Exploring the use of multi-objective genetic algorithms for reducing traffic generated

urban air and noise pollution. In *Proceedings of the 5th European Congress on Intelligent and Soft Computing*, pages 819–825, Aachen, Germany, September 1997.

[65] W. Habenicht. Quad trees: A data structure for discrete vector optimization problems. In *Lecture Notes in Economics and Mathematical Systems*, volume 209, pages 136–145, 1982.

[66] P. Hajela and J. Lee. Constrained genetic search via scheme adaptation: An immune network solution. *Structural Optimization*, 12(1):11–15, 1996.

[67] P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.

[68] P. Hajela, J. Yoo, and J. Lee. GA based simulation of immune networks—applications in structural optimization. *Journal of Engineering Optimization*, 29:131–149, 1997.

[69] T. Hanne. On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, 117(3):553–564, September 2000.

[70] Thomas Hanne. Global multiobjective optimization using evolutionary algorithms. *Journal of Heuristics*, 6(3):347–360, August 2000.

[71] Michael Pilegaard Hansen. *Metaheuristics for multiple objective combinatorial optimization.* PhD thesis, Institute of Mathematical Modelling, Technical University of Denmark, March 1998.

[72] John W. Hartmann. Low-thrust trajectory optimization using stochastic optimization methods. Master's thesis, Department of Aeronautical and Astronautical Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, January 1999.

[73] Alberto Herreros López. *Diseño de Controladores Robustos Multiobjetivo por Medio de Algoritmos Genéticos.* PhD thesis, Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid, Valladolid, Spain, Septiembre 2000. (In Spanish).

[74] Mark Hinchliffe, Mark Willis, and Ming Tham. Chemical process systems modelling using multi-objective genetic programming. In John R. Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick L. Riolo, editors, *Proceedings of*

the Third Annual Conference on Genetic Programming, pages 134–139, San Mateo, CA, July 1998. Morgan Kaufmann Publishers.

[75] John H. Holland. Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery*, 9:297–314, 1962.

[76] John H. Holland. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* University of Michigan Press, Ann Arbor, MI, 1975.

[77] Jeffrey Horn and Nicholas Nafpliotis. Multiobjective optimization using the niched pareto genetic algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, IL, 1993.

[78] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, NJ, June 1994. IEEE Service Center.

[79] Andrzej Jaszkiewicz. On the performance of multiple objective genetic local search on the 0/1 knapsack problem. a comparative experiment. Technical Report RA-002/2000, Institute of Computing Science, Poznan University of Technology, Poznań, Poland, July 2000.

[80] Kosuke Kato, Masatoshi Sakawa, and Toshinori Ikegame. Interactive decision making for multiobjective block angular 0-1 programming problems with fuzzy parameters through genetic algorithms. *Japanese Journal of Fuzzy Theory and Systems*, 9(1):49–59, 1997.

[81] S. Khajehpour. *Optimal Conceptual Design of High-Rise Office Buldings.* PhD thesis, Civil Engineering Department, University of Waterloo, Ontario, Canada, 2001.

[82] S. Kirkpatrick, C.D. Gellatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[83] Hajime Kita, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-objective optimization by means of the thermodynamical genetic algorithm. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel*

Problem Solving from Nature—PPSN IV, pages 504–512. Springer-Verlag. Lecture Notes in Computer Science No. 1141, Berlin, Germany, September 1996.

[84] Joshua D. Knowles and David W. Corne. The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation. In *1999 Congress on Evolutionary Computation*, pages 98–105, Piscataway, NJ, July 1999. IEEE Service Center.

[85] Joshua D. Knowles and David W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[86] Joshua D. Knowles, Martin J. Oates, and David W. Corne. Multiobjective evolutionary algorithms applied to two problems in telecommunications. *BT Technology Journal*, 18(4):51–64, October 2000.

[87] M. Krause and V. Nissen. On using penalty functions and multicriteria optimisation techniques in facility layout. In J. Biethahn and V. Nissen, editors, *Evolutionary Algorithms in Management Applications*. Springer-Verlag, Berlin, Germany, 1995.

[88] Kalmanje Krishnakumar. Micro-genetic algorithms for stationary and non-stationary function optimization. *SPIE Proceedings: Intelligent Control and Adaptive Systems*, 1196:289–296, 1989.

[89] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, CA, 1951. University of California Press.

[90] A. Gaspar Kunha, Pedro Oliveira, and José A. Covas. Genetic algorithms in multiobjective optimization problems: An application to polymer extrusion. In Annie S. Wu, editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, pages 129–130, Orlando, FL, July 1999.

[91] Setsuya Kurahashi and Takao Terano. A genetic algorithm with tabu search for multimodal and multiobjective function optimization. In Darrell Whitley, David Goldberg, Erick Cantú-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pages 291–298, San Francisco, CA, 2000. Morgan Kaufmann.

[92] Michael Lahanas, Natasa Milickovic, Dimos Baltas, and Nikolaos Zamboglou. Application of multiobjective evolutionary algorithms for dose optimization problems in brachytherapy. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 574–587. Springer-Verlag. Lecture Notes in Computer Science No. 1993, Berlin, Germany, 2001.

[93] Nando Laumanns, Marco Laumanns, and Dirk Neunzig. Multiobjective design space exploration of road trains with evolutionary algorithms. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 612–623. Springer-Verlag. Lecture Notes in Computer Science No. 1993, Berlin, Germany, 2001.

[94] Dongkon Lee. Multiobjective design of a marine vehicle with aid of design knowledge. *International Journal for Numerical Methods in Engineering*, 40:2665–2677, 1997.

[95] Xiaojian Liu, D. W. Begg, and R. J. Fishwick. Genetic approach to optimal topology/controller design of adaptive structures. *International Journal for Numerical Methods in Engineering*, 41:815–830, 1998.

[96] Samir W. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Department of General Engineering, Urbana, IL, May 1995.

[97] M. Mahfouf, M. F. Abbod, and D. A. Linkens. Multi-objective genetic optimization of the performance index of self-organizing fuzzy logic control algorithm using a fuzzy ranking approach. In H. J. Zimmerman, editor, *Proceedings of the Sixth European Congress on Intelligent Techniques and Soft Computing*, pages 1799–1808, Aachen, Germany, 1998. Verlag Mainz.

[98] N. Marco, S. Lanteri, J.-A. Desideri, and J. Périaux. A parallel genetic algorithm for multi-objective optimization in computational fluid dynamics. In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Périaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, chapter 22, pages 445–456. John Wiley & Sons, Chichester, UK, 1999.

[99] T. Marcu, L. Ferariu, and P. M. Frank. Genetic evolving of dynamic neural networks with application to process fault diagnosis.

In *Procedings of the EUCA/IFAC/IEEE European Control Conference ECC'99*, Karlsruhe, Germany, 1999. CD-ROM, F-1046,1.

[100] Teodor Marcu. A multiobjective evolutionary approach to pattern recognition for robust diagnosis of process faults. In R. J. Patton and J. Chen, editors, *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes: SAFEPROCESS'97*, pages 1183–1188, Kington Upon Hull, United Kingdom, August 1997.

[101] Teodor Marcu and Paul M. Frank. Parallel evolutionary approach to system identification for process fault diagnosis. In Prasad S. Dhurjati and Sylvie Cauvin, editors, *Procedings of the IFAC Workshop on 'On-line Fault Detection and Supervision in the Chemical Process Industries'*, pages 113–118, Solaize (Lyon), France, 1998.

[102] Carlos Eduardo Mariano Romero and Eduardo Morales Manzanares. MOAQ an ant-Q algorithm for multiple objective optimization problems. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Genetic and Evolutionary Computing Conference (GECCO 99)*, volume 1, pages 894–901, San Francisco, CA, July 1999. Morgan Kaufmann Publishers.

[103] W. Mason, V. Coverstone-Carroll, and J. Hartmann. Optimal earth orbiting satellite constellations via a pareto genetic algorithm. In *1998 AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pages 169–177, Boston, MA, August 1998. Paper No. AIAA 98-4381.

[104] 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, NJ, July 2000. IEEE Service Center.

[105] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, Berlin, Germany, third edition, 1996.

[106] Melanie Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA, 1996.

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

[108] Dongkyung Nam, Yun Deuk Seo, Lae-Jeong Park, Cheol Hoon Park, and Bumsup Kim. Parameter optimization of a voltage reference circuit using EP. In David B. Fogel, editor, *Proceedings of the 1998 International Conference on Evolutionary Computation*, pages 245–266, Piscataway, NJ, 1998. IEEE.

[109] S. Narayanan and S. Azarm. On improving multiobjective genetic algorithms for design optimization. *Structural Optimization*, 18:146–155, 1999.

[110] J. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[111] Shigeru Obayashi, Takanori Tsukahara, and Takashi Nakamura. Multiobjective evolutionary computation for supersonic wing-shape optimization. *IEEE Transactions on Evolutionary Computation*, 4(2):182–187, July 2000.

[112] Matthias Ortmann and Wolfgang Weber. Multi-criterion optimization of robot trajectories with evolutionary strategies. In *Proceedings of the 2001 Genetic and Evolutionary Computation Conference. Late-Breaking Papers*, pages 310–316, San Francisco, CA, July 2001.

[113] Keith A. Osman, Anthony M. Higginson, and John Moore. Improving the efficiency of vehicle water-pump designs using genetic algorithms. In C. Dagli, M. Akay, A. Buczak, O. Ersoy, and B. Fernandez, editors, *Smart Engineering Systems: Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE '98)*, volume 8, pages 291–296, New York, NY, 1998. ASME, ASME Press.

[114] Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.

[115] Geoffrey T. Parks. Multiobjective pressurised water reactor reload core design using a genetic algorithm. In George D. Smith, Nigel C. Steele, and Rudolf F. Albrecht, editors, *Artificial Neural Nets and Genetic Algorithms*, pages 53–57, Norwich, UK, 1997. Springer-Verlag.

[116] Jacques Périaux, Mourad Sefrioui, and Bertrand Mantel. RCS multi-objective optimization of scattered waves by active control elements using GAs. In *Proceedings of the Fourth International Conference on Control, Automation, Robotics and Vision (ICARCV'96)*, Singapore, 1996.

[117] Jacques Périaux, Mourad Sefrioui, and Bertrand Mantel. Ga multiple objective optimization strategies for electromagnetic backscattering. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, chapter 11, pages 225–243. John Wiley and Sons, West Sussex, England, 1997.

[118] Charles J. Petrie, Teresa A. Webster, and Mark R. Cutkosky. Using pareto optimality to coordinate distributed agents. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 9:269–281, 1995.

[119] Carlo Poloni and Valentino Pediroda. Ga coupled with computationally expensive simulations: tools to improve efficiency. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, chapter 13, pages 267–288. John Wiley and Sons, West Sussex, England, 1997.

[120] Min Qiu. Prioritizing and scheduling road projects by genetic algorithm. *Mathematics and Computers in Simulation*, 43:569–574, 1997.

[121] I. J. Ramírez Rosado, J. L. Bernal Agustín, L. M. Barbosa Proença, and Vladimiro Miranda. Multiobjective planning of power distribution systems using evolutionary algorithms. In M. H. Hamza, editor, *8th IASTED International Conference on Modelling, Identification and Control—MIC'99*, pages 185–188, Innsbruck, Austria, February 1999.

[122] Ignacio J. Ramírez Rosado and José L. Bernal Agustín. Reliability and cost optimization for distribution networks expansion using an evolutionary algorithm. *IEEE Transactions on Power Systems*, 16(1):111–118, February 2001.

[123] S. S. Rao. Multiobjective optimization in structural design with uncertain parameters and stochastic processes. *AIAA Journal*, 22(11):1670–1678, November 1984.

[124] S. S. Rao. Game theory approach for multiobjective structural optimization. *Computers and Structures*, 25(1):119–127, 1987.

[125] S. S. Rao. Genetic algorithmic approach for multiobjective optimization of structures. In *ASME Annual Winter Meeting, Structures and Controls Optimization*, volume AD-Vol. 38, pages 29–38, New Orleans, LA, November 1993. ASME.

[126] T. Ray, R.P. Gokarn, and O.P. Sha. A global optimization model for ship design. *Computers in Industry*, 26:175–192, 1995.

[127] Brahim Rekiek. *Assembly Line Design (multiple objective grouping genetic algorithm and the balancing of mixed-model hybrid assembly line)*. PhD thesis, Free University of Brussels, CAD/CAM Department, Brussels, Belgium, December 2000.

[128] Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some guidelines for genetic algorithms with penalty functions. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197, San Mateo, CA, 1989. Morgan Kaufmann Publishers.

[129] Brian J. Ritzel, J. Wayland Eheart, and S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Research*, 30(5):1589–1603, may 1994.

[130] James L. Rogers. Optimum actuator placement with a genetic algorithm for aircraft control. In Cihan H. Dagli, Anna L. Buczak, Joydeep Ghosh, Mark J. Embrechts, and Okan Ersoy, editors, *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems (ANNIE'99)*, pages 355–360, New York, NY, November 1999. ASME Press.

[131] James L. Rogers. A parallel approach to optimum actuator selection with a genetic algorithm. In *AIAA Paper No. 2000-4484, AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August 14–17 2000.

[132] R. S. Rosenberg. *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Arbor, MI, 1967.

[133] 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, NJ, 1998. IEEE Press.

[134] 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, NJ, July 2000. IEEE Press.

[135] Eric Sandgren. Multicriteria design optimization by goal programming. In Hojjat Adeli, editor, *Advances in Design Optimization*, chapter 23, pages 225–265. Chapman & Hall, London, 1994.

[136] Dragan A. Savic, Godfrey A. Walters, and Martin Schwab. Multiobjective genetic algorithms for pump scheduling in water supply. In *AISB International Workshop on Evolutionary Computing*, pages 227–236, Berlin, Germany, April 1997. Springer-Verlag. Lecture Notes in Computer Science No. 1305.

[137] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Nashville, TN, 1984.

[138] J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, NJ, 1985. Lawrence Erlbaum.

[139] 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, MA, May 1995.

[140] P. Schroder, A. J. Chipperfield, P. J. Fleming, and N. Grum. Multiobjective optimization of distributed active magnetic bearing controllers. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 13–18. IEE, September 1997.

[141] M. Schwab, D. A. Savic, and G. A. Walters. Multi-objective genetic algorithm for pump scheduling in water supply systems. Technical Report 96/02, Centre For Systems And Control Engineering, School of Engineering, University of Exeter, Exeter, United Kingdom, 1996.

[142] Hans-Paul Schwefel. Kybernetische evolution als strategie der experimentellen forschung inder strömungstechnik. Dipl.-Ing. thesis, 1965. (in German).

[143] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Birkhäuser, Basel, Switzerland, 1977. (In German).

[144] Hans-Paul Schwefel. *Numerical Optimization of Computer Models.* Wiley, Chichester, UK, 1981.

[145] E. Science, N. Marvin, M. Bower, and R. J. Rowe. An evolutionary approach to constructing prognostic models. *Artificial Intelligence in Medicine*, 15(2):155–165, February 1999.

[146] Paolo Serafini. Simulated annealing for multiple objective optimization problems. In G.H. Tzeng, H.F. Wang, U.P. Wen, and P.L. Yu, editors, *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application*, volume 1, pages 283–292, Berlin, Germany, 1994. Springer-Verlag.

[147] Mitsuhiro Shibuya, Hajime Kita, and Shigenobu Kobayashi. Integration of multi-objective and interactive genetic algorithms and its application to animation design. In *Proceedings of IEEE Systems, Man, and Cybernetics*, volume III, pages 646–651, 1999.

[148] Robert E. Smith, Stephanie Forrest, and Alan S. Perelson. Population diversity in an immune system model: Implications for genetic search. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 153–165. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[149] Kishan Chetan Srigiriraju. Noninferior surface tracing evolutionary algorithm (NSTEA) for multi objective optimization. Master's thesis, North Carolina State University, Raleigh, NC, August 2000.

[150] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[151] W. Stadler. Natural structural shapes (the static case). *The Quarterly Journal of Mechanics and Applied Mathematics*, XXXI, Pt. 2:169–217, 1978.

[152] W. Stadler. Fundamentals of multicriteria optimization. In W. Stadler, editor, *Multicriteria Optimization in Engineering and the Sciences*, pages 1–25. Plenum Press, New York, NY, 1988.

[153] Patrick D. Surry and Nicholas J. Radcliffe. The COMOGA method: Constrained optimisation by multiobjective genetic algorithms. *Control and Cybernetics*, 26(3):391–412, 1997.

[154] Patrick D. Surry, Nicholas J. Radcliffe, and Ian D. Boyd. A multi-objective approach to constrained optimisation of gas supply networks : The COMOGA method. In Terence C. Fogarty, editor, *Evolutionary Computing. AISB Workshop. Selected Papers*, pages 166–180, Sheffield, U.K., 1995. Springer-Verlag. Lecture Notes in Computer Science No. 993.

[155] T. Tagami and T. Kawabe. Genetic algorithm based on a pareto neighborhood search for multiobjective optimization. In *Proceedings of the 1999 International Symposium of Nonlinear Theory and its Applications (NOLTA'99)*, pages 331–334, Hawaii, 1999.

[156] Hisashi Tamaki, Hajime Kita, and Shigenobu Kobayashi. Multi-objective optimization by genetic algorithms : A review. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation (ICEC'96)*, pages 517–522, Piscataway, NJ, 1996. IEEE.

[157] K. C. Tan, T. H. Lee, and E. F. Khor. Evolutionary algorithms with goal and priority information for multi-objective optimization. In *1999 Congress on Evolutionary Computation*, pages 106–113, Piscataway, NJ, July 1999. IEEE Service Center.

[158] Mark W. Thomas. *A Pareto Frontier for Full Stern Submarines via Genetic Algorithm.* PhD thesis, Ocean Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, june 1998.

[159] Effie Tsoi, Kit Po Wong, and Chun Che Fung. Hybrid GA/SA algorithms for evaluating trade-off between economic cost and environmental impact in generation dispatch. In David B. Fogel, editor, *Proceedings of the Second IEEE Conference on Evolutionary Computation (ICEC'95)*, pages 132–137, Piscataway, NJ, 1995. IEEE Press.

[160] E.L. Ulungu, J. Teghem, Ph. Fortemps, and D. Tuyttens. MOSA method: A tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 8(4):221–236, 1999.

[161] Manuel Valenzuela-Rendón and Eduardo Uresti-Charre. A non-generational genetic algorithm for multiobjective optimization. In

Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 658–665, San Mateo, CA, July 1997. Morgan Kaufmann Publishers.

[162] 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, OH, May 1999.

[163] David A. Van Veldhuizen and Gary B. Lamont. Evolutionary computation and convergence to a pareto front. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, Stanford, CA, July 1998. Stanford University Bookstore.

[164] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1998.

[165] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithm test suites. In Janice Carroll, Hisham Haddad, Dave Oppenheim, Barrett Bryant, and Gary B. Lamont, editors, *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351–357, San Antonio, TX, 1999. ACM.

[166] David A. Van Veldhuizen, Brian S. Sandlin, Robert M. Marmelstein, and Gary B. Lamont. Finding improved wire-antenna geometries with genetic algorithms. In David B. Fogel, editor, *Proceedings of the 1998 International Conference on Evolutionary Computation*, pages 102–107, Piscataway, NJ, 1998. IEEE.

[167] Jiang Feng Wang and Jacques Périaux. Multi-point optimization using GAs and Nash/Stackelberg games for high lift multi-airfoil design in aerodynamics. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 1, pages 552–559, Piscataway, NJ, May 2001. IEEE Service Center.

[168] D. S. Weile and E. Michielssen. Integer coded pareto genetic algorithm design of constrained antenna arrays. *Electronics Letters*, 32(19):1744–1745, September 1996.

[169] D. S. Weile, E. Michielssen, and D. E. Goldberg. Genetic algorithm design of pareto optimal broadband microwave absorbers.

IEEE Transactions on Electromagnetic Compatibility, 38(3):518–525, August 1996.

[170] P. B. Wienke, C. Lucasius, and G. Kateman. Multicriteria target optimization of analytical procedures using a genetic algorithm. *Analytical Chimica Acta*, 265(2):211–225, 1992.

[171] P. B. Wilson and M. D. Macleod. Low implementation cost IIR digital filter design using genetic algorithms. In *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, pages 4/1–4/8, Chelmsford, U.K., 1993.

[172] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In D.F. Jones, editor, *Proceedings of the Sixth International Conference on Genetics*, volume 1, pages 356–366, 1932.

[173] Jin Wu and Shapour Azarm. On a new constraint handling technique for multi-objective genetic algorithms. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 741–748, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[174] P. O. Yapo, H. V. Gupta, and S. Sorooshian. Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, 204:83–97, 1998.

[175] Habib Youssef, Sadiq M. Sait, and Salman A. Khan. Fuzzy evolutionary hybrid metaheuristic for network topology design. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 400–415. Springer-Verlag. Lecture Notes in Computer Science No. 1993, Berlin, Germany, 2001.

[176] Yan Yu. Multi-objective decision theory for computational optimization in radiation therapy. *Medical Physics*, 24:1445–1454, 1997.

[177] R. S. Zebulum, M. A. Pacheco, and M. Vellasco. A multi-objective optimisation methodology applied to the synthesis of low-power operational amplifiers. In Ivan Jorge Cheuri and Carlos Alberto

dos Reis Filho, editors, *Proceedings of the XIII International Conference in Microelectronics and Packaging*, volume 1, pages 264–271, Curitiba, Brazil, August 1998.

[178] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[179] Eckart Zitzler, Jürgen Teich, and Shuvra S. Bhattacharyya. Multidimensional exploration of software implementations for DSP algorithms. *Journal of VLSI Signal Processing*, 24(1):83–98, February 2000.

[180] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, The Netherlands, September 1998. Springer-Verlag. Lecture Notes in Computer Science No. 1498.

[181] 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.