A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques

Carlos A. Coello Coello*

Laboratorio Nacional de Informatica Avanzada
Rébsamen 80, AP 696, Xalapa, Veracruz 91090, México
ccoello@xalapa.lania.mx

Abstract. This paper presents a critical review of the most important evolutionary-based multiobjective optimization techniques developed over the years, emphasizing the importance of analyzing their Operations Research roots as a way to motivate the development of new approaches that exploit the search capabilities of evolutionary algorithms. Each technique is briefly described mentioning its advantages and disadvantages, their degree of applicability and some of their known applications. Finally, the future trends in this discipline and some of the open areas of research are also addressed.
Keywords: multiobjective optimization, multicriteria optimization, vector optimization, genetic algorithms, evolutionary algorithms, artificial intelligence.

1 Introduction

Since the pioneer work of Rosenberg in the late 60s regarding the possibility of using genetic-based search to deal with multiple objectives, this new area of research (now called evolutionary multiobjective optimization) has grown considerably as indicates the notable increment (mainly in the last 15 years) of technical papers in peer-reviewed journals, special sessions in international conferences and interest groups in the Internet[1].

Multiobjective optimization is with no doubt a very important research topic both for scientists and engineers, not only because of the multiobjective nature of most real-world problems, but also because there are still many open questions in this area. In fact, there is not even a universally accepted definition of "optimum" as in single-objective optimization, which makes it difficult to even compare results of one method to another, because normally the decision about what the "best" answer is corresponds to the so-called (human) *decision maker*.

* Most of this work was performed while the author was affiliated to the Plymouth Engineering Centre, in the United Kingdom.

[1] The author maintains a list on Evolutionary Multiobjective Optimization at: http://www.lania.mx/~ccoello/EMOO/EMOObib.html

Evolutionary algorithms seem particularly desirable to solve multiobjective optimization problems because they deal simultaneously with a set of possible solutions (the so-called population) which allows to find an entire set of Pareto optimal solutions 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 succeptible to the shape or continuity of the Pareto front, whereas these two issues are a real concern for mathematical programming techniques.

Surprisingly, despite the considerable volume of research in evolutionary multiobjective optimization in the last 15 years, there have been only two surveys of this area published in the technical literature[2]: Tamaki et al. [91], which is a very short and quick review of some of the main approaches, and Fonseca and Fleming [18, 21] which is a remarkable account of the issues that make this problem interesting from the evolutionary computing perspective. In both cases, however, little detail was provided on how each method worked, just a few applications of each technique were mentioned and their corresponding Operations Research roots were only scarcely mentioned. Furthermore, several other approaches have arisen since the publication of these 2 papers, and the intention of the present work is to provide researchers and students interested in this topic with an updated survey that can be used (to a certain extent) as a self-contained document for anyone interested in this area who has a previous (at least basic) knowledge of genetic algorithms in general. Those who may need additional information about genetic algorithms should refer to Goldberg [27], Holland [35], Michalewicz [54], and Mitchell [56] for more information.

2 Statement of the Problem

Multiobjective optimization (also called multicriteria optimization, multiperformance or vector optimization) can be defined as the problem of finding [65]:

> a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the designer.

Formally, we can state it as follows:

Find the vector $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which will satisfy the m inequality constraints:

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

[2] Right after the submission of this paper, David A. Van Veldhuizen and Gary B. Lamont made available a technical report [99] that contains another remarkable survey of the area that complements the material contained in this paper.

the p equality constraints

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

and optimizes the vector function

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T \tag{3}$$

where $\bar{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables.

In other words, we wish to determine from among the set $\mathcal{F}$ of all numbers which satisfy (1) and (2) the particular set $x_1^*, x_2^*, \ldots, x_k^*$ which yields the optimum values of all the objective functions.

2.1 Pareto Optimum

The concept of *Pareto optimum* was formulated by Vilfredo Pareto in the XIX century [66], and constitutes by itself the origin of research in multiobjective optimization. We say that a point $\bar{x}^* \in \mathcal{F}$ is *Pareto optimal* if for every $\bar{x} \in \mathcal{F}$ either,

$$\bigwedge_{i \in I} (f_i(\bar{x}) = f_i(\bar{x}^*)) \tag{4}$$

or, there is at least one $i \in I$ such that

$$f_i(\bar{x}) > f_i(\bar{x}^*) \tag{5}$$

In words, this definition says that $\bar{x}^*$ is Pareto optimal if there exists no feasible vector $\bar{x}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, the Pareto optimum almost always gives not a single solution, but rather a set of solutions called *non-inferior* or *non-dominated* solutions.

2.2 Pareto Front

The minima in the Pareto sense are going to be in the boundary of the design region, or in the locus of the tangent points of the objective functions. In Fig. 1, a bold line is used to mark this boundary for a biobjective problem. The region of points defined by this bold line is called the *Pareto Front*. In general, it is not easy to find an analytical expression of the line or surface that contains these points, and the normal procedure is to compute the points $\mathcal{F}^k$ and their corresponding $f(\mathcal{F}^k)$. When we have a sufficient amount of these, we may proceed to take the final decision.

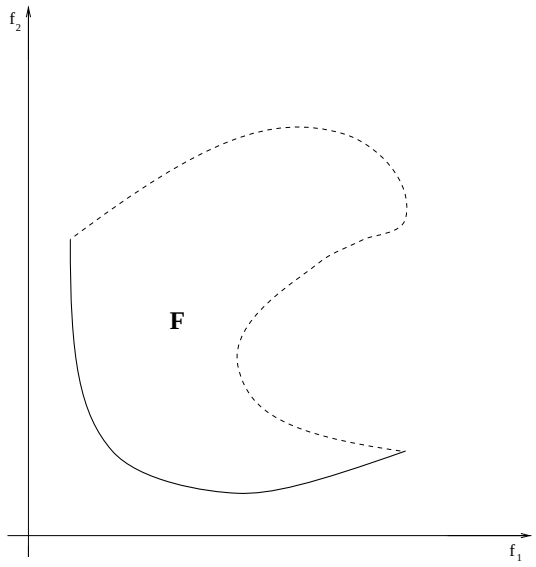

Fig. 1. An example of a problem with two objective functions. The Pareto front is marked with a bold line.

3 Approaches That Use Aggregating Functions

The notion of genetic search in a multicriteria problem dates back to the late 60s, in which Rosenberg's [80] study contained a suggestion that would have led to multicriteria optimization if he had carried it out as presented. His suggestion was to use multiple *properties* (nearness to some specified chemical composition) in his simulation of the genetics and chemistry of a population of single-celled organisms. Since his actual implementation contained only one single property, the multiobjective approach could not be shown in his work, but it nevertheless showed the possibility of using an evolutionary-based technique to handle multiple objective functions.

Knowing that a genetic algorithm needs scalar fitness information to work, it is almost natural to propose 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 devise. There are, however, obvious problems with this approach. The first is that we have to provide some accurate scalar information on the range of the objectives, to avoid having one of them to dominate the others. This implies that we should know, to a certain extent, the behavior of each of the objective functions, which is normally (at least in most real-world applications) a very expensive process (computationally speaking) that we can not afford in most cases. Obviously, if this combination of objectives is possible (and it is possible in some applications), this is not only the simplest approach, but also is one of the most efficient procedures, because no further interaction

with the decision maker is required, and if the GA succeeds at optimizing the resulting fitness function, then the results will be at least sub-optimum in most cases.

The approach of combining objectives into a single function is normally denominated *aggregating functions*, and it has been attempted several times in the literature with relative success in problems in which the behavior of the objective functions is more or less well-known. This section includes the most popular aggregating approaches.

3.1 Weighted Sum Approach

This method consists of adding all the objective functions together using different weighting coefficients for each one of them. This means that our multiobjective optimization problem is transformed into a scalar optimization problem of the form:

$$\min \sum_{i=1}^{k} w_i f_i(\bar{x}) \tag{6}$$

where $w_i \geq 0$ are the weighting coefficients representing the relative importance of the objectives. It is usually assumed that

$$\sum_{i=1}^{k} w_i = 1 \tag{7}$$

Since the results of solving an optimization model using (6) can vary significantly as the weighting coefficients change, and since very little is usually known about how to choose these coefficients, a necessary approach is to solve the same problem for many different values of w_i. But in this case, the designer is still, of course, confronted with the decision of having to choose the most appropriate solution based on his/her intuition.

Note that the weighting coefficients do not reflect proportionally the relative importance of the objectives, but are only factors which, when varied, locate points in the Pareto set. For the numerical methods that can be used to seek the minimum of (6), this location depends not only on the w_i values, but also on the units in which the functions are expressed.

If we want w_i to reflect closely the importance of the objectives, all functions should be expressed in units of approximately the same numerical values. Additionally, we can also transform (6) to the form:

$$\min \sum_{i=1}^{k} w_i f_i(\bar{x}) c_i \tag{8}$$

where c_i are constant multipliers that will scale properly the objectives.

The best results are usually obtained if $c_i = 1/f_i^0$. In this case, the vector function is normalized to the form $\bar{f}(\bar{x}) = [\bar{f}_1(\bar{x}), \bar{f}_2(\bar{x}), \ldots, \bar{f}_k(\bar{x})]^T$, where $\bar{f}_i(\bar{x}) = f_i(\bar{x})/f_i^0$.

Applications

- Syswerda and Palmucci [90] used weights in their fitness function to add or subtract values during the schedule evaluation of a resource scheduler, depending on the existence or absence of penalties (constraints violated).
- Jakob et al. [41] used a weighted sum of the several objectives involved in a task planning problem : to move the tool center point of an industrial robot to a given location as precisely and quickly as possible, avoiding certain obstacles and aiming to produce a path as smooth and short as possible.
- Jones et al. [42] used weights for their genetic operators in order to reflect their effectiveness when a GA was applied to generate hyperstructures from a set of chemical structures.
- Wilson & Macleod [103] used this approach as one of the methods incorporated into a GA to design multiplierless IIR filters in which the two conflicting objectives were to minimize the response error and the implementation cost of the filter.
- Liu et al. [51] used this technique to optimize the layout and actuator placement of a 45-bar plane truss in which the objectives were to minimize the linear regulator quadratic control cost, the robustness and the modal controllability of the controlled system subject to total weight, asymptotical stability and eigenvalues constraints.
- Yang and Gen [104] used a weighted sum approach to solve a bicriteria linear transportation problem. More recently, Gen et al. [25, 26] extended this approach to allow more than two objectives, and added fuzzy logic to handle the uncertainty involved in the decision making process. A weighted sum is still used in this approach, but it is combined with a fuzzy ranking technique that helps to identify Pareto solutions, since the coefficients of the objectives are represented with fuzzy numbers reflecting the existing uncertainty regarding their relative importance.

Strengths and Weaknesses

This method was the first technique developed for the generation of non-inferior solutions for multiobjective optimization. This is an obvious consequence of the fact that it was implied by Kuhn and Tucker in their seminal work on numerical optimization [45]. The main strength of this method is its efficiency (computationally speaking), and can be applied to generate a strongly non-dominated solution that can be used as an initial solution for other techniques. Its main weakness is the difficulty to determine the appropriate weights when we do not have enough information about the problem. In this case, any optimal point obtained will be a function of the coefficients used to combine the objectives. Most researchers prefer to use a simple linear combination of the objectives and then generate the *trade-off surface*[3] by varying the weights. This approach is very

[3] The term "trade-off" in this context refers to the fact that we are trading a value of one objective function for a value of another function or functions.

simple and easy to implement, but it has the disadvantage of missing concave portions of the trade-off curve (in other words, **the approach does not generate proper Pareto optimal solutions in the presence of non-convex search spaces**) [77], which is a serious drawback in most real-world applications.

3.2 Goal Programming

Charnes and Cooper [5] and Ijiri [39] are credited with the development of the goal programming method for a linear model, and have played a key role in applying it to industrial problems. In this method, the decision maker has to assign targets or goals that he/she wishes to achieve for each objective. These values are incorporated into the problem as additional constraints. The objective function will then try to minimize the absolute deviations from the targets to the objectives. The simplest form of this method may be formulated as follows [16]:

$$\min \sum_{i=1}^{k} |f_i(\bar{x}) - T_i|, \quad \text{subject to } \bar{x} \in \mathcal{F} \tag{9}$$

where T_i denotes the target or goal set by the decision maker for the ith objective function $f_i(\bar{x})$, and $\mathcal{F}$ represents the feasible region. The criterion, then, is to minimize the sum of the absolute values of the differences between target values and actually achieved values. A more general formulation of the goal programming objective function is a weighted sum of the pth power of the deviation $|f_i(\bar{x}) - T_i|$ [32]. Such a formulation has been called *generalized goal programming* [37, 38]. This technique has also been called "target vector optimization" by other authors [12].

Applications

- Wienke et al. [102] used this approach in combination with a genetic algorithm to optimize simultaneously the intensities of six atomic emission lines of trace elements in alumina powder as a function of spectroscopic excitation conditions.
- Eric Sandgren [82] also used goal programming coupled with a genetic algorithm to optimize plane trusses and the design of a planar mechanism.

Strengths and Weaknesses

This technique will yield a dominated solution if the goal point is chosen in the feasible domain [16]. Its main strength is its efficiency (computationally speaking) in case we know the desired goals that we wish to achieve, and if they are in the feasible region. However, its main weakness is that the decision maker is given the task of devising the appropriate weights or priorities for the objectives that will eliminate the non-commensurable characteristics of the problem, which in most cases is difficult unless there is prior knowledge about the shape of the

search space. Also, if the feasible region is difficult to approach, this method could become very inefficient. Nevertheless, this technique may be useful in cases in which a linear or piecewise-linear approximation of the objective functions can be made, because of the availability of excellent computer programs for that, and the possibility of eliminating dominated goal points easily. On the other hand, in non-linear cases, other approaches may be more efficient.

3.3 Goal Attainment

In this approach, a vector of weights $w_1, w_2, \ldots, w_k$ relating the relative under- or over-attainment of the desired goals must be elicited from the decision maker in addition to the goal vector $b_1, b_2, \ldots, b_k$ for the objective functions $f_1, f_2, \ldots, f_k$. To find the best-compromise solution x^*, we solve the following problem:

$$\text{Minimize} \quad \alpha \tag{10}$$

subject to:

$$\begin{aligned} g_j(\bar{x}) &\leq 0; \quad j = 1, 2, \ldots, m \\ b_i + \alpha \cdot w_i &\geq f_i(\bar{x}); \quad i = 1, 2, \ldots, k \end{aligned} \tag{11}$$

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

$$\sum_{i=1}^{k} |w_i| = 1 \tag{12}$$

If some $w_i = 0$ $(i = 1, 2, \ldots, k)$, it means that the maximum limit of objectives $f_i(\bar{x})$ is b_i.

It can be easily shown [6] that the set of non-dominated solutions for a problem can be generated by varying the weights, with $w_i \geq 0$ $(i = 1, 2, \ldots, k)$ even for nonconvex problems.

It should be pointed out that the optimum value of α will inform the decision maker of whether the goals are attainable or not. A negative value of α implies that the goal of the decision maker is attainable and an improved solution will be obtained. Otherwise, if $\alpha > 0$, then the decision maker goal is unattainable.

Applications

- Wilson & MacLeod [103] used this approach as another of the methods incorporated into their GA to design multiplierless IIR filters.

Strengths and Weaknesses

As Wilson and MacLoud [103] indicate, goal attainment has several weaknesses, from which probably the main one is the misleading selection pressure that it can generate in some cases. 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 GA none of them will be better than the other. Its main advantage is the simplicity of its implementation and its computational efficiency.

3.4 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 optima set. The method may be formulated as follows:

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

$$f_r(\bar{x}^*) = \min_{x \in \mathcal{F}} f_r(\bar{x}) \tag{13}$$

subject to additional constraints of the form

$$f_i(\bar{x}) \leq \varepsilon_i \quad for\ i = 1, 2, \ldots, k \ \text{ and } i \neq r \tag{14}$$

where ε_i are assumed values of the objective functions which we wish not 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 is stopped when the decision maker 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 GAs). For each objective function f_i $(i = 1, 2, \ldots, k)$, there is an optimal design vector $\bar{x}_i^*$ for which $f_i(\bar{x}_i^*)$ is a minimum. Let $f_i(\bar{x}_i^*)$ be the lower bound on ε_i, i.e.

$$\varepsilon_i \geq f_i(\bar{x}_i^*) \quad i = 1, 2, \ldots, r-1, r+1, \ldots, k \tag{15}$$

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

$$\varepsilon_i \leq f_i(\bar{x}_r^*) \quad i = 1, 2, \ldots, r-1, r+1, \ldots, k \tag{16}$$

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

This approach was suggested by Ritzel and Wayland [77] as a simple and naïve way of solving multiobjective optimization problems using a GA. The idea

was to code the GA in such a way that all the objectives, except for one, were kept constant (constrained to a single value), and the remaining objective would then become the fitness function for the GA. Thus, through a process of running the GA numerous times with different values of the constrained objectives, a trade-off surface can be developed.

Applications

- Quagliarella and Vicini [71] suggested the use of this technique coupled with a hybrid GA (a genetic algorithm that used gradient based optimization techniques to speed up the search in order to reduce the computational cost required in a real-world application) to solve multiobjective optimization problems.
- Ranjithan et al. [72] used this approach to solve groundwater pollution containment problems.
- Loughlin and Ranjithan [52] used a variation of this technique in which they incorporated target satisfaction levels (similar to those used in Goal Programming), and combined it with a neighboorhood selection procedure according to which only individuals within a certain radius were allowed to mate (individuals in the population were indexed and placed in a matrix format). Additional genetic operators such as elitism and dynamic scaling of the target satisfaction levels were also implemented. Loughlin and Ranjithan applied this technique to a real-world air quality management problem with two conflicting objectives: minimize the cost of controlling air pollutant emissions and maximize the amount of emissions reduction (this is a combinatorial problem that is suitable for integer programming techniques).

Strengths and Weaknesses

The most obvious weakness of this approach is that it is time-consuming, and the coding of the objective functions may be difficult or even impossible for certain problems, particularly if there are too many objectives. Furthermore, this method tends to find weakly non-dominated solutions, which may not be appropriate in some applications (e.g., structural optimization). Nevertheless, the relative simplicity of the technique (its main strength) has made it popular among some GA practitioners.

4 Other Approaches Not Based on The Notion of Pareto Optimum

To overcome the difficulties involved in the aggregating approaches, much work has been devoted to the development of alternative techniques based on population policies or special handling of the objectives [70]. Some of the most popular approaches that fall into this category will be examined in this section.

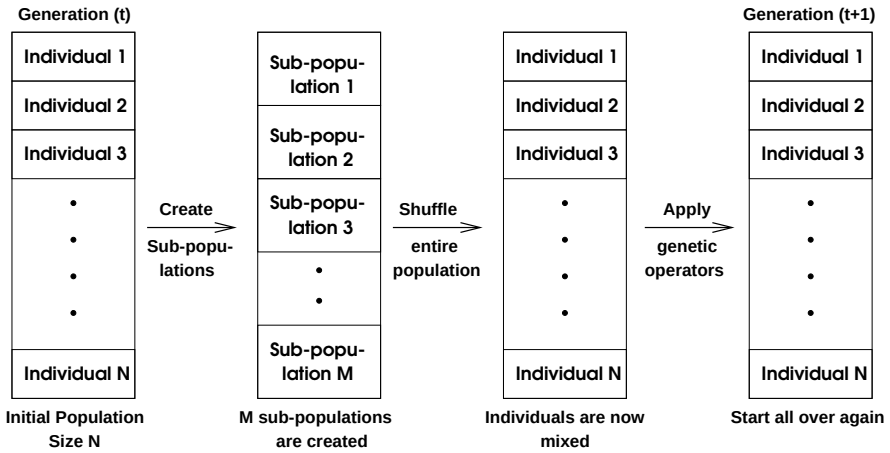

Fig. 2. Schematic of VEGA selection. It is assumed that the population size is N and that there are M objective functions.

4.1 VEGA

David Schaffer [83] extended Grefenstette's GENESIS program [31] to include multiple objective functions. Schaffer's approach was to use an extension of the Simple Genetic Algorithm (SGA) that he called the *Vector Evaluated Genetic Algorithm* (VEGA), and that differed of the first only in the way in which selection was performed. This operator was modified so that at each generation a number of sub-populations was 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 would 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 Fig. 2. Schaffer realized that the solutions generated by his system were non-dominated in a local sense, because their non-dominance was limited to the current population, and while a locally dominated individual is also globally dominated, the converse is not necessarily true [83]. An individual who 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 calls "middling" performance[4] in all dimensions, which could be very useful for compromise solutions, but that 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 some heuristics to deal with this problem. For example, to use a heuristic selection preference approach for non-dominated 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.

Applications

- Ritzel and Wayland [77] used a variation of VEGA in which they incorporated a parameter to control the selection ratio. In the case of the groundwater pollution containment problem that Ritzel and Wayland solved, there were only two objectives, and the selection ratio was defined as the ratio of the fraction of strings selected on the basis of the first objective (reliability) to the fraction selected via the second objective (cost).
- Surry et al. [89] proposed an interesting application of VEGA to model constraints in a single-objective optimization problem to avoid the need of a penalty function. Surry et al., however, modified the standard procedure of VEGA and introduced a form of ranking based on the number of constraints violated by a certain solution, and they reported that their approach worked well in the optimization of gas supply networks, since the tendency of VEGA to favor certain solutions can actually be an advantage when handling constraints, because in that case we want to favor precisely any solution that does not violate any constraint over those which do.
- Cvetković et al. [13] proposed several approaches to overcome VEGA's problems. For example, to wait for a certain amount of generations before shuffling together the population, or avoid shuffling the individuals, and instead copy or migrate a certain amount of individuals from one sub-population to another. They used these and other traditional multiobjective optimization approaches for preliminary airframe design.
- Tamaki et al. [92, 91] developed a technique in which at each generation, non-dominated individuals in the current population are kept for the following generation. This approach is really a mixture of Pareto selection (see next section) and VEGA, because if the number of non-dominated individuals is less that the population size, the remainder of the population in the following generation is filled applying VEGA to the dominated individuals. On the other hand, if the number of the non-dominated individuals exceeds the population size, individuals in the following generation are selected among

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

the non-dominated individuals using VEGA. In a later version of this algorithm, called Pareto Reservation strategy, Tamaki et al. [91] used also fitness sharing among the non-dominated individuals to maintain diversity in the population.

Strengths and Weaknesses

Although Schaffer reported some success, and the main strength of this approach is its simplicity, Richardson et al. [76] noted that the shuffling and merging of all the sub-populations corresponds to averaging the fitness components associated with each of the objectives. Since Schaffer used proportional fitness assignment [27], these fitness components were in turn proportional to the objectives themselves [18]. 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. [76]. The main consequence of this is that when we have a concave trade-off surface certain points in concave regions will not be found through this optimization procedure in which we are using just a linear combination of the objectives, and it has been proved that this is true regardless of the set of weights used [76]. Therefore, the main weakness of this technique is its inability to produce Pareto-optimal solutions in the presence of non-convex search spaces.

4.2 Lexicographic Ordering

In this method, the objectives are ranked in order of importance by the designer. The optimum solution $\bar{x}^*$ 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, $f_1(\bar{x})$ and $f_k(\bar{x})$ denote the most and least important objective functions, respectively. Then the first problem is formulated as

$$\text{Minimize} \quad f_1(\bar{x}) \tag{17}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{18}$$

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

$$\text{Minimize} \quad f_2(\bar{x}) \tag{19}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{20}$$

$$f_1(\bar{x}) = f_1^* \tag{21}$$

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

$$\text{Minimize} \quad f_i(\bar{x}) \tag{22}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{23}$$

$$f_l(\bar{x}) = f_l^*, \quad l = 1, 2, \ldots, i-1 \tag{24}$$

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

Applications

- Fourman [24] suggested a selection scheme based on lexicographic ordering. In a first version of his algorithm, objectives were assigned different priorities by the user and each pair of individuals were 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 was randomly selected at each run. Fourman used this approach to design compact symbolic layouts [24].
- Kursawe [47] formulated a multiobjective version of evolution strategies [84] (ESs) based on lexicographic ordering. Selection consisted of as many steps as objective functions had the problem. At each step, one of these objectives was selected randomly according to a probability vector, and used to delete a fraction of the current population. After selection, the survivors became the parents of the next generation. The map of the trade-off surface was produced from the points evaluated during the run. Since the environment was allowed to change over time, diploid individuals were necessary to keep recessive information stored.

Strengths and Weaknesses

Selecting randomly an objective 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 neggligible [18, 21]. 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 more 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 [12]. However, its simplicity (its main strength) makes it competitive with a weighted sum of objectives and VEGA.

4.3 Use of Game Theory

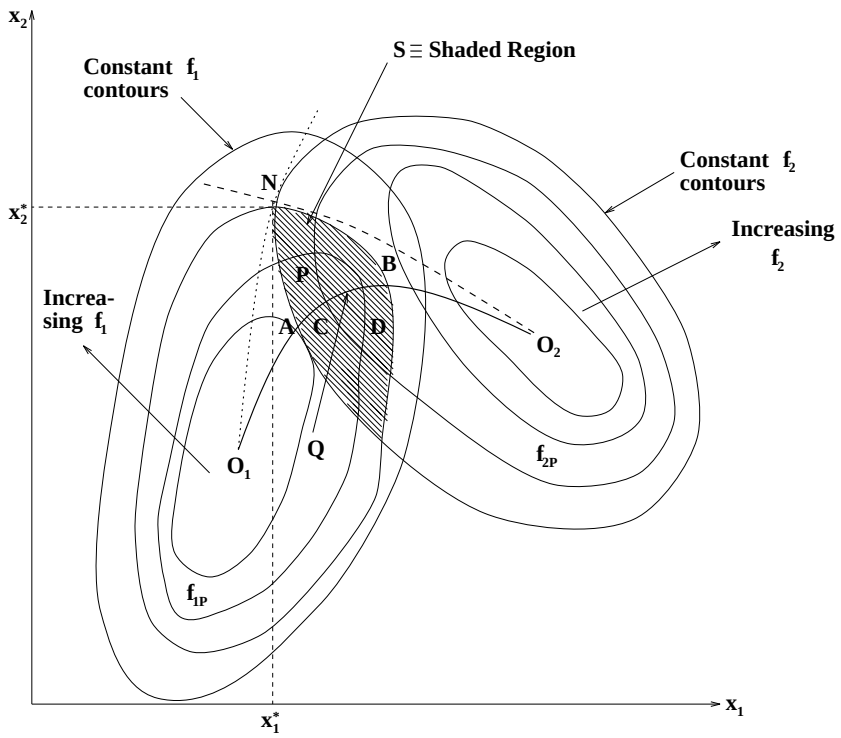

Fig. 3. Example of cooperative and non-cooperative game solutions.

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 Fig. 3. 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 design variables. It is assumed that one player is associated with each objective. The first player wants to select a design 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 Fig. 3. 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 Fig. 3, the point $N(x_1^*, x_2^*)$ represents such 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/her own criterion [57].

This point has the characteristic that

$$f_1(x_1^*, x_2^*) \leq f_1(x_1, x_2^*) \tag{25}$$

and

$$f_2(x_1^*, x_2^*) \leq f_2(x_1^*, x_2) \tag{26}$$

where x_1 may be to the left or right of x_1^* in (25) and x_2 may lie above or below x_2^* in (26).

Applications

- Périaux et al. [68] proposed a GA-based approach that uses the concept of Nash equilibrium to solve a biobjective optimization problem (the optimal distribution of active control elements which minimizes the backscatering of aerodynamic reflectors). The main idea of this work was to use 2 non-cooperative players represented by 2 independent sub-populations in a genetic algorithm, and then make them to interact in the following way:
  If f_1 and f_2 are the 2 objectives to be optimized, let P_1 and P_2 represent the 2 non-cooperative players. We start at generation zero with P_1 trying to optimize f_1 while f_2 remains fixed and P_2 trying to optimize f_2 while f_1 remains fixed. After one generation is over (i.e., when all the individuals in both populations have been evaluated and the genetic operators have been applied independently to each of those populations), we send (or migrate) the best individual from population 1 to population 2 and the best individual from population 2 to population 1. This process is repeated for as many generations as needed, until the Nash equilibrium is reached.

Strenghts and Weaknesses

The main strength of this approach is that it seems to be very efficient (computationally speaking), but in the state presented in the work by Périaux et al. [68] is

not possible to generate more than one non-dominated solution which, hopefully will be the best overall solution to the problem[5]. However, it is indeed 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 Pareto front can actually be found, although a *cooperative game* may be preferred in that case over a *non-cooperative* approach [75, 74].

4.4 Using Gender to Identify Objectives

Robin Allenson [2] used a population-based approach modelled after VEGA in which gender was used to distinguish between the two objective functions of a problem consisting of the planning of a route composed of a number of straight pipeline segments. With this approach, only male-female mating was allowed, and such gender was randomly assigned at birth. In the initial population, Allenson made sure that there was an equal number of males and females, but such balance was not kept after applying the genetic operators. At each generation, the worst individual (chosen according to one of the two genders) was eliminated and replaced by another (randomly picked) individual of the same gender. Allenson used evolution strategies to implement some form of sexual attractors that would modify the way in which mating was performed. The idea was to model the sexual attraction that some individuals have over others in nature, which determines a not so random mating.

Lis and Eiben [50] also incorporated gender in their GA, but in a more general sense. In this case, the number of genders (or sexes), was not limited to two, but it could be as many as objectives we had. Another distinction of this approach is that the crossover operator was modifed as to allow panmictic reproduction, in which several parents generate a single child (instead of having two parents generate two children as in the traditional genetic algorithm). The idea was to select one parent from each sex to contribute to the generation of a child. This child will have the sex of the parent that contributed with the largest amount of genes (if there is a tie, then the sex is randomly chosen from the parents that contributed the same amount of genes). If no crossover takes place, then one of the individuals in the old generation is copied exactly the same (including its sex) to the following generation. In this approach, individuals are evaluated using different fitness functions (according to their corresponding sex). The mutation operator is only slightly restricted to avoid changes in the sex of an individual. As generations progress, a list of non-dominated individuals is updated, removing from it any individual that is no longer non-dominated after the list is modified. At the end, this list will contain the Pareto optimal solutions.

Applications

- Lis and Eiben [50] tested successfully their approach with the two multi-objective optimization problems provided in the paper by Srinivas and Deb

[5] Périaux et al. did not succeed at that in the example presented in their paper.

[86], but no further applications of this technique seem to be available at the moment.

Strengths and Weaknesses

The use of genders is really another way of defining separate subpopulations for each objective. The difference of this approach with VEGA [83] lies on the fact that Lis and Eiben used panmictic crossover, which imposes certain mating restrictions, avoiding the random crossing among different subpopulations performed by Schaffer. However, the main weakness of this approach is that as we increase the number of objectives (or genders), we will have many subpopulations and panmictic crossover will become more inefficient (computationally speaking), because we will need to use more parents to generate a child. Additionally, the population size will have to be large enough as we increase the number of objectives, to keep a reasonably diverse spread of genders across the entire population.

4.5 Weighted Min-max Approach

The idea of stating the *min-max optimum* and applying it to multiobjective optimization problems, was taken from game theory, which deals with solving conflicting situations. The min-max approach to a linear model was proposed by Jutler [43] and Solich [85], and was further developed by Osyczka [59, 60, 64], Rao [73] and Tseng and Lu [95].

The min-max optimum compares relative deviations from the separately attainable minima. Consider the ith objective function for which the relative deviation can be calculated from

$$z_i^{'}(\bar{x}) = \frac{|f_i(\bar{x}) - f_i^0|}{|f_i^0|} \tag{27}$$

or from

$$z_i^{''}(\bar{x}) = \frac{|f_i(\bar{x}) - f_i^0)|}{|f_i(\bar{x})|} \tag{28}$$

It should be clear that for (27) and (28) we have to assume that for every $i \in I$ and for every $\bar{x} \in \mathcal{F}$, $f_i(\bar{x}) \neq 0$.

If all the objective functions are going to be minimized, then equation (27) defines function relative increments, whereas if all of them are going to be maximized, it defines relative decrements. Equation (28) works conversely.

Let $\bar{z}(\bar{x}) = [z_1(\bar{x}), \ldots, z_i(\bar{x}), \ldots, z_k(\bar{x})]^T$ be a vector of the relative increments which are defined in $\mathcal{R}^k$. The components of the vector $z(\bar{x})$ will be evaluated from the formula

$$\forall_{i \in I}(z_i(\bar{x})) = max\ \{z_i^{'}(\bar{x}), z_i^{''}(\bar{x})\} \tag{29}$$

Now we define the min-max optimum as follows [64]:

A point $\bar{x}^* \in \mathcal{F}$ is min-max optimal, if for every $\bar{x} \in \mathcal{F}$ the following recurrence formula is satisfied:

Step 1:

$$v_1(\bar{x}^*) = \underset{x \in \mathcal{F}}{min} \ \underset{i \in I}{max} \{z_i(\bar{x})\} \tag{30}$$

and then $I_i = \{i_1\}$, where i_1 is the index for which the value of $z_i(\bar{x})$ is maximal.

If there is a set of solutions $x_1 \subset \mathcal{F}$ which satisfies Step 1, then

Step 2:

$$v_2(\bar{x}^*) = \underset{x \in x_1}{min} \ \underset{i \in I,\, i \notin I_1}{max} \{z_i(\bar{x})\} \tag{31}$$

and then $I_2 = \{i_1, i_2\}$, where i_2 is the index for which the value of $z_i(x)$ in this step is maximal.

If there is a set of solutions $x_{r-1} \subset \mathcal{F}$ which satisfies step $r-1$ then

Step r:

$$v_r(\bar{x}^*) = \underset{x \in x_{r-1}}{min} \ \underset{i \in I,\, i \notin I_{r-1}}{max} \{z_i(\bar{x})\} \tag{32}$$

and then $I_r = \{I_{r-1}, i_r\}$, where i_r is the index for which the value of $z_i(\bar{x})$ in the rth step is maximal.

If there is a set of solutions $x_{k-1} \subset \mathcal{F}$ which satisfies Step $k-1$, then

Step k:

$$v_k(\bar{x}^*) = \underset{\bar{x} \in x_{k-1}}{min} z_i(\bar{x}) \ \underset{i \in I,\, i \notin I_{k-1}}{max} \quad \text{for } i \in I \text{ and } i \notin I_{k-1} \tag{33}$$

where $v_1(\bar{x}^*), \ldots, v_k(\bar{x})$ is the set of optimal values of fractional deviations ordered non-increasingly.

This optimum can be described in words as follows. Knowing the extremes of the objective functions which can be obtained by solving the optimization problems for each criterion separately, the desirable solution is the one which gives the smallest values of the relative increments of all the objective functions.

The point $\bar{x}^* \in \mathcal{F}$ which satisfies the equations of Steps 1 and 2 may be called the best compromise solution considering all the criteria simultaneously and on equal terms of importance.

Applications

- Hajela and Lin [33] included the weights of each objective in the chromosome, and promoted their diversity in the population through fitness sharing. Their goal was to be able to simultaneously generate a family of Pareto optimal designs corresponding to different weighting coefficients in a single run of the GA. Besides using sharing, Hajela and Lin used a vector evaluated approach

based on VEGA to achieve their goal. They proposed the use of a utility function of the form:

$$\bar{U} = \sum_{i=1}^{l} W_i \frac{F_i}{F_i^*} \tag{34}$$

where F_i^* are the scaling parameters for the objective criterion, l is the number of objective functions, and W_i are the weighting factors for each objective function F_i.

Hajela's approach also uses a sharing function of the form:

$$\phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{sh}}\right)^{\alpha}, & d_{ij} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \tag{35}$$

with $\alpha = 1$, and σ_{share} chosen between 0.01 and 0.1. The fitness of a design i is then modified as:

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{M} \phi(d_{ij})} \tag{36}$$

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

Under Hajela's representation, weight combinations are incorporated into the chromosomic string, and a single number represents not the weight itself, but a combination of weights. For example, the number 4 (assuming integer representation) could represent the vector $X_w = (0.4, 0.6)$ for a problem with two objective functions. Then, sharing is done on the weights.

Finally, a mating restriction mechanism was imposed, to avoid members within a radius σ_{mat} to cross. The value of $\sigma_{mat} = 0.15$ was suggested by Hajela and Lin in their paper [33].

Hajela and Lin [33] used their approach to optimize a 10-bar plane truss in which weight and displacement were to be minimized, and a wing-box structure in which they wanted to minimize its weight while maximizing its natural frequency.

Strengths and Weaknesses

The main weakness of this approach is that it can create a very high selection pressure if certain combinations of weights are produced at early stages of the search [12]. The use of sharing will avoid to a certain extent to have a premature convergence, but the use of a sharing factor (which is not easy to determine) increases the number of parameters required by the GA, and is therefore subject to further experimenting. Its main weakness is its simplicity and efficiency, because it does not require to check for non-dominance.

4.6 Two Variations of the Weighted Min-Max Strategy

Coello [12, 11] proposed two variations of the weighted min-max strategy used by Hajela and Lin. In his first approach, the decision maker has to provide a predefined set of weights that will be used to spawn several small subpopulations that will evolve separately (and concurrently), trying to converge to a single point of the Pareto front each. Mating restrictions were imposed to guarantee feasibility of all the solutions, and constraints were handled by not allowing the generation of any infeasible solutions through the evolution process (death penalty). This approach also requires the knowledge of the ideal vector, or some estimate of it that lies in the feasible region.

In a second approach, Coello [12] proposed the use of a local ideal vector that was computed at each generation, and the selection process was modified as to allow the incorporation of min-max dominance. That means that a certain individual would be considered the winner of a tournament if its maximum deviation from the ideal vector was the smallest from the set under competition. Also, mating restrictions were imposed to keep only feasible solutions at all generations. Finally, sharing had to be used to overcome the high selection pressure introduced by the use of min-max tournament selection.

Applications

- Coello and Christiansen applied these two approaches to the optimization of I-beams [8] and manufacturing problems [9], and to the design of a robot arm [10].

Strengths and Weaknesses

The main strength of this technique is its efficiency and relative simplicity. The use of weights is its main weakness because it is not always easy to find an appropriate set that can delineate correctly the part of the Pareto region that we wish to find. However, Coello [12] showed through several engineering design examples that it was actually possible to find a good approximation of the Pareto front with a relatively small amount of weights chosen systematically (using a deterministic technique). The use of mating restrictions and feasibility checks during the entire evolution process may be seen as an important weakness, since it has been shown that such constraint-handling approach will not work in concave search surfaces. However, this was an attempt to incorporate the handling of constraints into the search process in another way different from the traditional penalty approach, and it does not preclude the algorithm from handling constraints in a different manner.

The second approach, in which weights are not used, is much more efficient and produces good Pareto fronts [12]. However, its main weakness is its dependence on the value of σ_{share}, but the idea of using a utility function that is dynamically modified, as in this case, has also been exploited more recently by other researchers [96, 4, 30].

4.7 Use of the Contact Theorem to Detect Pareto Optimal Solutions

Osyczka and Kundu [62] proposed the use of an algorithm based on the contact theorem (one of the main theorems in multiobjective optimization [49]) to determine relative distances of a solution vector with respect to the Pareto set. In this paper [62], the contact theorem was used to determine the fitness of each individual in the population.

This approach is in a way, very similar to the Min-Max approach previously described, only that in this case no weights are required for each objective, nor a sharing function is needed to keep diversity in the population[6].

Applications

- Although the GA-based technique presented by Osyczka and Kundu [62] has not been applied to real-world problems (they used only two simple biobjective optimization functions in their paper), Osyczka's algorithm for detecting Pareto optimality has been applied before to several problems, mainly in machine design [63, 59, 60, 61, 64].

Strengths and Weaknesses

The main strengths of this approach are its efficiency and relative simplicity. Additionally, it does not require an explicit sharing function. However, its main weakness is that it is highly sensitive to the values of the penalty factor used to incorporate the constraints into each objective function, and its performance relies heavily on the so-called starting distance, which is some sort of scaling factor used to compare relative quality among the different solutions. If any of these 2 values is not chosen properly, too much selection pressure may be generated, or the GA may often jump back and forth between the feasible and infeasible regions at any given generation, producing too many dominated points in the process, and consequently losing portions of the Pareto front.

4.8 A Non-Generational Genetic Algorithm

Valenzuela-Rendón & Uresti-Charre [96] 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) [27], 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 Valenzuela-Rendón and Uresti-Charre 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

[6] The algorithm used by Osyczka to identify Pareto optimal solutions and the contact theorem are described in detail in his book [64].

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 taking a linear combination of these 2 objectives.

Applications

- Valenzuela-Rendón and Uresti-Charre [96] obtained better results than NPGA [36] (see below) in 3 biobjective optimization problems, both in terms of the number of points in the Pareto front at the final iteration, and in terms of the total number of function evaluations. However, no further comparisons with other methods or in problems with more objectives was provided.

Strengths and Weaknesses

This approach is really a more elaborate version of the weighted ranking techniques used by Bentley and Wakefield [4] (particularly the technique that they called weighted average ranking—WAR). The main strength 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 weakness 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 additional parameters 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).

4.9 Use of Randomly Generated Weights and Elitism

Ishibuchi and Murata [40] proposed an algorithm similar to Hajela's weighted min-max technique, but the weights were generated in a slightly different way in this case, and the set of non-dominated solutions produced at each generation was kept separately from the current population.

Applications

- Ishibuchi and Murata [40] used this technique to solve biobjective optimization flowshop scheduling problems in which the makespan and maximum tardiness were to be minimized.

Strengths and Weaknesses

This approach is very similar to the technique called Sum of Weighted Rations (SWR) by Bentley and Wakefield [4] and to the attribute value functions used by Greenwood et al. [30]. Bentley and Wakefield [4] claim that this approach

maintains enough diversity as to keep a wide spread of solutions through many generations. However, Coello [12] has shown (using a similar approach), that such spread may not be kept in problems in which there is an objective in the ideal vector that can be easily achieved by a wide set of solutions. In such case, it is necessary to use sharing techniques or a local search technique (as proposed by Ishibuchi and Murata [40]) to keep diversity, which constitutes the main weakness of this approach.

Bentley and Wakefield [4] showed also another variation of this algorithm called Sum of Weighted Global Ratios (SWGR) which visibly reduces the spread of solutions produced (i.e., the size of the Pareto set) by using the globally best and worst values instead of the current ones. The idea is nevertheless interesting and its main strength is that the implementation of this algorithm seems to be not only easy, but also quite efficient with respect to most of the Pareto-based approaches described next.

5 Pareto-Based Approaches

The idea of using Pareto-based fitness assignment was first proposed by Goldberg [27] to solve the problems of Schaffer's approach. He suggested the use of non-dominated 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 non-dominated 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. A niching mechanism such as sharing [29] would allow the GA to maintain individuals all along the non-dominated frontier.

Applications

- Hilliard et al. [34] used a Pareto optimality ranking method to handle the objectives of minimizing cost and minimizing delay in a scheduling problem. They tentatively concluded that the Pareto optimality ranking method outperformed the VEGA method.
- The Pareto method was found to be superior to a VEGA by Liepins et al. [48] when applied to a variety of set covering problems.
- Ritzel et al. [77] also used non-dominated ranking and selection combined with deterministic crowding [53] as the niching mechanism. They applied the GA to a groundwater pollution containment problem in which cost and reliability were the objectives. Though the actual Pareto front was unknown, Ritzel et al. used the best trade-off surface found by a domain-specific algorithm, called MICCP (Mixed Integer Chance Constrained Programming), to compare the performance of the GA. They found that selection according

to Pareto non-domination was superior to both VEGA and non-domination with deterministic crowding, at least for finding points near or on the front found by MICCP.
- Stanley and Mudge [88] implemented Goldberg's Pareto ranking technique to solve a microprocessor design problem in which the constraints were handled as additional objectives.

Strengths and Weaknesses

The main weakness of Pareto ranking in general is that there is no efficient algorithm to check for non-dominance in a set of feasible solutions [12]. Traditional algorithms have 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 σ_{share}, which is not easy, and the performance of the method relies a lot on such value. However, Pareto ranking is the most appropriate way to generate an entire Pareto front in a single run of the GA and its main strength 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 mathematical programming techniques.

5.1 Multiple Objective Genetic Algorithm

Fonseca and Fleming [17] have proposed a scheme 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 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 [17]:

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{37}$$

All non-dominated 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 [17]:

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 [27], 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 [28] 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 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 [87].

Applications

- Chen Tan and Li [93] reported success in the use of MOGA for the multi-objective optimization of ULTIC controllers that satisfy a number of time domain and frequency domain specifications. Also, Chipperfield and Fleming [7] reported success in using MOGA for the design of a multivariable control system for a gas turbine engine.
- Obayashi [58] used Pareto ranking with phenotypic sharing and *best-N* selection (the best N individuals are selected for the next generation among N parents and N children) for the aerodynamic design of compressor blade shapes.
- Rodríguez Vázquez et al. [78] extended MOGA to use it in genetic programming, introducing the so-called MOGP (Multiple Objective Genetic Programming). Genetic programming [44] replaces the traditional linear chromosomic representation by a hierarchical tree representation that, by definition, is more powerful, but also requires larger population sizes and specialized operators. MOGP was used for the identification of non-linear model structures, as an alternative that the authors reported to work better (in terms of representation power) than the use of the conventional linear representation of MOGA that they had attempted before [22].
- Aherne et al. [1] used MOGA to optimize the selection of parameters for an object recognition scheme called the Pairwise Geometric Histogram paradigm.
- Todd and Sen [94] used a variant of MOGA for the preplanning of containership layouts (a large scale combinatorial problem). In Todd and Sen's approach, a population of non-dominated individuals is kept and updated at each generation, removing individuals that become dominated and duplicates. The traditional genetic operators and sharing are applied only to this population. Niche sizes are computed automatically for each criterion by substracting the maximum value for that criterion from the minimum and dividing it by the population size. Crossover was restricted so that only individuals that were very similar could mate, and because of the permutations encoded, a repair algorithm had to be used afterwards. Finally, a heuristic mutation that basically defined rules to exchange bit positions had to be used to avoid premature convergence of the population.

Strenghts and Weaknesses

It has been cited in the literature [87, 14] that the main weakness 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 the parameter value space, and apparently this choice has been taken in some of the applications reported above.

The main strenghts of MOGA is that is efficient and relatively easy to implement [12]. 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 [17] have developed a good methodology to compute such value for their approach.

5.2 Non-Dominated Sorting Genetic Algorithm

The Non-dominated Sorting Genetic Algorithm (NSGA) was proposed by Srinivas and Deb [86], and is based on several layers of classifications of the individuals. Before selection is performed, the population is ranked on the basis of nondomination: all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, 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 used for this approach. 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 quick convergence of the population toward such regions. Sharing, by its part, helps to distribute it over this region. The efficiency of NSGA lies in the way in which multiple objectives are reduced to a dummy fitness function using a nondominated sorting procedure. With this approach, any number of objectives can be solved [87], and both maximimization and minimization problems can be handled.

Applications

- Périaux et al. [68] used the NSGA to find an optimal distribution of active control elements which minimizes the backscattering of aerodynamic reflectors.
- Vedarajan et al. [97] used the NSGA for investment portfolio optimization, but interestingly they used binary tournament selection instead of stochastic remainder selection as suggested by Srinivas and Deb [86]. The authors claim that this approach worked well in their examples, although they do not provide any argument for their choice of selection strategy. Tournament selection is expected to introduce a high selection pressure that may dilute the effect of sharing. However, since Vedarajan et al. used fairly large population sizes (above 1000 individuals), the counter-effect of tournament selection may had been absorbed by the extra individuals in the population.
- Michielssen and Weile [55] used also the NSGA to design an electromagnetic system.

Strengths and Weaknesses

The main strengths of this technique is that can handle any number of objectives, and that does sharing iin the parameter value space instead of the objective value space, which ensures a better distribution of individuals, and allows multiple equivalent solutions exist. Some researchers [12] have reported that its main weakness is that it is more inefficient (both computationally and in terms of quality of the Pareto fronts produced) than MOGA, and more sentitive to the value of the sharing factor σ_{share}. Other authors [106, 105] 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 their experiments no comparisons with MOGA were provided.

5.3 Niched Pareto Genetic Algorithm

Horn and Nafpliotis [36] proposed a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals, a number of other individuals in the population was used to help determine dominance (typically around 10). When both competitors were either dominated or non-dominated (i.e., there was a tie), the result of the tournament was decided through fitness sharing [29]. Population sizes considerably larger than usual with other approaches were used so that the noise of the selection method could be tolerated by the emerging niches in the population [18].

Horn and Nafpliotis [36] 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 *nested sharing*.

Applications

- Belegundu et al. [3] used the NPGA for the design of laminated ceramic composites.
- Poloni and Pediroda [69] used it for the design of a multipoint airfoil that has its minimum drag at two given lift values with a constraint in the maximum allowed pitching moment.
- A variation of the NPGA was proposed by Quagliarella and Vicini [71]. They introduced the dominance criteria of the problem in the selection mechanism (as in the NPGA), but then selected the individuals to be reproduced to generate the following population using a random walk operator. This obviously produces a locally dominating individual rather than a globally dominating one. Additionally, if more than one non dominated individual is found, then the first one encountered is selected (instead of doing sharing like in the NPGA). At the end of every new generation, the set of Pareto optimal solutions is updated and stored. They used this approach for airfoil design [71].

Strengths and Weaknesses

Since this approach does not apply Pareto selection to the entire population, but only to a segment of it at each run, its main strenghts are that is very fast and that it produces good non-dominated fronts that can be kept for a large number of generations [12]. However, its main weakness is that besides requiring a sharing factor, this approach also requires a good choice of the size of the tournament to perform well, complicating its appropriate use in practice.

6 Future Research Paths

Although a lot of work has been done in this area, most of it has concentrated on application of conventional or ad-hoc techniques to certain difficult problems. Therefore, there are several research issues that still remain to be solved, some of which will be briefly described next:

- Since the size of the Pareto set is normally considerably large, and in the particular case of using a genetic algorithm, depends on the size of the population, it may be desirable in some cases to devise ways of reducing the number of elements in such set, in order to facilitate the analysis for the decision maker. Kunha, Oliveira and Covas [46] proposed the incorporation of Roseman and Gero's algorithm [79] into the GA to cluster together points that are within a certain distance (defined by the user) of each other in the Pareto front.
- Probably one of the most difficult problems in multiobjective optimization is to determine how to measure the quality of a solution. So far, practically visual inspection is the only technique used, unless there is some previous knowledge of the points which lie in the Pareto front (in which case there is obviously no need for a multiobjective optimization technique). Fonseca and Fleming [23] proposed the definition of certain (arbitrary) goals that we wish the GA to attain; then we can perform multiple runs and apply standard non-parametric statistical procedures to evaluate the quality of the solutions (i.e., the non-dominated fronts) produced by the technique under study, and/or compare it against other similar techniques. However, these arbitrary goals are not easy to define either.
  Other (similar) metrics have been proposed in the literature. For example, Van Veldhuizen and Lamont [98] proposed the so-called *generational distance*, which is a measure of how close is our current Pareto front from the real Pareto front (assuming we know where it lies). Zitzler and Thiele [105] proposed two measures: the first concerns the size of the objective value space 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. Finally, Srinivas and Deb [87] proposed the use of an statistical measure (the chi-square distribution) to estimate the spread of the population on the Pareto front with respect to the sharing factor used.

All these metrics are interesting proposals but there are almost no comparative studies of techniques that substantiate their suitability in general test problems, which implies that more work in this area is required.

- In some cases it may be necessary to be able to assign more importance to certain objective or objectives. Interestingly, in such cases, an aggregating approach allows us to change the importance of the objectives very easily, as opposed to any ranking technique (i.e., Pareto-based approaches) which normally do not provide the means to do it directly. Fonseca and Fleming [17] proposed the use of a utility function combined with MOGA [18, 21] to produce a method for the progressive articulation of preferences. The idea that they proposed was to have a feedback loop between the decision maker and the GA, so that certain solutions (from the Pareto set) are given more preference than others. Ideally, such process could be done automatically by replacing the decision maker with an expert system [17]. The problem with Fonseca's approach is that it requires previous knowledge of the ranges of each objective function, which could be excessively expensive or even impossible to obtain in some cases.

  Bentley and Wakefield [4] proposed the use of weights to estimate the importance of solutions that are already identified as Pareto optimal, in an attempt to overcome the problems with Fonseca's approach. Also, in a more elaborate approach, Greenwood et al. [30] proposed a compromise between the aggregated approach (i.e., the use of weights) and ranking techniques in which the level of preference may be defined. Greenwood et al. [30] used an approach called *specified multi-attribute value theory* (ISMAUT) [101] which, combined with a GA, allows the definition of preferences by the GA itself, rather than asking the intervention of the decision maker. However, the decision maker still gets to decide what particular area of the trade-off surface wants to explore, so that the GA constrains the search to that specific area. Additionally, Greenwood et al. [30] defined a certain metrics that allows us to obtain a single value (or utility function) that will guide the search to the particular Pareto region that is of interest to the decision maker.

  Finally, Voget and Kolonko [100] proposed the use of a fuzzy controller that regulates the selection pressure automatically by using a set of predefined goals that define the 'desirable' behavior of the population. An interesting aspect of this work is that they actually combine Pareto ranking with VEGA during the same run of the GA, to allow the desired reduction of deviations from the goals specified by the authors [100].

  These 3 proposals are quite interesting, but still more work needs to be done in this area, preferrently with real-world problems (Fonseca's approach was an appropriate choice for the optimization of a gas turbine engine [17], and Greenwood et al. [30] showed that their approach performed well in two hardware/software codesign problems), so that more general guidelines can be derived from the different approaches proposed.
- Directly related to the problem of measuring the quality of the solutions found with a multiobjective optimization technique lies the need to have a set of benchmark problems that can be used to test existing and new ap-

proaches. This set should include both constrained and unconstrained problems[7], examples with few objectives (2 or 3) suitable for graphical inspection, problems with few and several variables, and problems in which is possible to achieve the ideal vector (these could be used to tune up any technique to be tried). In this direction, Deb [14] has recently proposed ways to create controllable test problems for evolutionary multiobjective optimization techniques using single-objective optimization problems as a basis. This is an interesting proposal that could allow to transform deceptive and massively multimodal problems into very difficult multiobjective optimization problems. However, in his technical report, Deb [14] only defines test problems with 2 objective functions and the scalability of these test functions to more objectives is not straightforward. Van Veldhuizen and Lamont [99] have also proposed some guidelines to design a test function suite for evolutionary multiobjective optimization techniques, and have included in their report some sample test problems.

Using benchmark problems such as those proposed by Deb [14] and Van Veldhuizen & Lamont [99], it should be possible to perform detailed studies of performance of different GAs (assuming certain quality measures).

Coello [12] and Zitzler & Thiele [106, 105] have conducted comparative studies of this type using engineering design problems and the 0/1 knapsack problem respectively, but it is necessary to have a database of test problems (as suggested by Van Veldhuizen and Lamont [99]) and to define better metrics to evaluate the performance of each technique. It is particularly important to introduce in this database problems that have constraints, since this aspect has been overlooked by most researchers in the last few years. This will improve our understanding of the strengths and weaknesses of each technique and should lead us to the development of new and more powerful approaches.

- As Deb indicates [14], it would be very useful to understand the dynamics of the population of a GA over different generations when applied to multiobjective optimization problems. If we knew how is the population behaving and what issues are making it difficult to keep nondominated solutions, we could devise techniques in which the progress towards the global Pareto front could be considerably faster than with the current approaches.
- It is also important to define stopping criteria for a GA-based multiobjective optimization technique, because it is not obvious to know when the population has reached a point from which no further improvement can be reached (i.e., how do we know that the global Pareto front has been found?). Currently, the main approaches used to stop this kind of GA haven been to either use a fixed number of generations, or to monitor the population at certain intervals and interpret visually the results to determine when to halt the evolution process.

[7] Most current papers that introduce new GA-based multiobjective optimization techniques, use 2 or 3 simple unconstrained biobjective functions, particularly those used originally by Schaffer [83].

- The use of sharing in these techniques introduces another problem, because the value of σ_{share} becomes another parameter with which the user has to experiment until a reasonable setting is found. Even when important work has been done in this area (see for example Deb and Goldberg [15] and Fonseca & Fleming [17]), most of that research is focused on single-objective optimization, or multimodal optimization.
- Some researchers have also found alternative applications of multiobjective optimization techniques that are quite interesting. The most remarkable is perhaps the attempt to use ranking techniques or similar approaches to handle constraints in a single objective optimization problem, as to avoid the use of penalty functions. Surry et al. [89] proposed the COMOGA (Constrained Optimization by Multi-Objective Genetic Algorithms) approach, which treats each constraint as a separate objective and therefore transforms a constrained single objective optimization problem into an unconstrained multiobjective optimization problem, which is solved using Fonseca's MOGA [17]. This approach was used by Surry et al. to optimize gas supply networks [89]. Fonseca and Fleming [19] also proposed to handle constraints as objectives, and applied their approach to the design of a gas turbine [20]. Parmee and Purchase [67] implemented a version of VEGA [83] to handle constraints relating to a gas turbine design problem as objectives to allow the GA to locate a feasible region within the highly constrained search space of this application. Having identified a feasible point region, specialized operators were introduced to create a variable-size hypercube around each feasible point in an attempt to define the feasible region [67]. Finally, Stanley and Mudge [88] used also Pareto ranking to handle constraints treated as objectives in a combinatorial optimization problem (microprocessor design).
  With no doubt, the number of applications of evolutionary multiobjective optimization techniques to real-world problems will increase over the years, and a probable trend in research could be to reformulate many problems that are currently considered as if they only had one objective. This will constitute a more realistic approach to the solution of problems that frequently arise in areas such as engineering, because they are normally reduced to a single objective and the remaining objectives are treated as constraints instead of handling all (conflicting) objectives simultaneously.
- Finally, a very important topic that has been only scarcely addressed by researchers in multiobjective optimization is the development of a theory that can explain issues such as the effect of the parameters used (i.e., population size, crossover and mutation rates, niche sizes, and elitism) and the way in which the selection technique adopted affects the performance of an algorithm. In this direction, Günter Rudolph [81] has recently showed that theoretical results of convergence derived from single-objective evolutionary optimization cannot be used in the presence of multiple objectives. In his study, Rudolph proposes a methodology to prove convergence to the Pareto front, but only shows results for a specific problem, indicating in the process the main difficulties that theoreticians have to face to derive a more general proof.

7 Conclusions

This paper has attempted to provide a comprehensive review of the most popular evolutionary-based approaches to multiobjective optimization, giving also some insights of their Operations Research roots, their advantages and disadvantages, and their possible range of applicability. Additionally, some representative real-world applications of each approach (when found) have also been included, together with a very rich bibliography that should be enough to guide a newcomer into this important and growing area of research.

In the final section of the paper, the most promising areas of future research (according to the author's opinion) were briefly described, and some of the work already done around them has also been briefly addressed.

8 Acknowledgments

The author would like to thank the two anonymous reviewers for their valuable comments that helped him improve this paper.

References

[1] F. J. Aherne, N. A. Thacker, and P. I. Rockett. Optimal Pairwise Geometric Histograms. In Adrian F. Clark, editor, *Electronic Proceedings of the Eighth British Machine Vision Conference, BMVC97*, page http://peipa.essex.ac.uk/bmva/bmvc97/papers/071/bmvc.html, University of Essex, United Kingdom, 1997.

[2] Robin Allenson. Genetic algorithms with gender for multi-function optimisation. Technical Report EPCC-SS92-01, Edinburgh Parallel Computing Centre, Edinburgh, Scotland, 1992.

[3] A. D. Belegundu, D. V. Murthy, R. R. Salagame, and E. W. Constants. Multiobjective Optimization of Laminated Ceramic Composites Using Genetic Algorithms. In *Fifth AIAA/USAF/NASA Symposium on Multidisciplinary Analysis and Optimization*, volume Paper 84-4363-CP, pages 1015–1022, Panama City, Florida, 1994. AIAA.

[4] P. J. Bentley and J. P. Wakefield. Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. In *Proceedings of the 2nd On-Line World Conference on Soft Computing in Engineering Design and Manufacturing (WSC2)*, page `http://users.aol.com/docbentley/dispaper.htm`, June 1997.

[5] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley, New York, 1961.

[6] Y. L. Chen and C. C. Liu. Multiobjective VAR planning using the goal-attainment method. *IEE Proceedings on Generation, Transmission and Distribution*, 141(3):227–232, may 1994.

[7] A. Chipperfield and P. Fleming. Gas Turbine Engine Controller Design using Multiobjective Genetic Algorithms. In A. M. S. Zalzala, editor, *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems : Innovations and Applications, GALESIA '95*, pages 214–219, Halifax Hall, University of Sheffield, UK, September 1995. IEEE.

[8] Carlos A. Coello Coello and Alan D. Christiansen. MOSES : A Multiobjective Optimization Tool for Engineering Design. *Engineering Optimization*, 1998. (Accepted for publication).

[9] Carlos A. Coello Coello and Alan D. Christiansen. Two new GA-based methods for multiobjective optimization. *Civil Engineering Systems*, 1998. (In Press).

[10] 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*, 1998. (In Press).

[11] Carlos A. Coello Coello, Filiberto Santos Hernández, and Francisco Alonso Farrera. Optimal design of reinforced concrete beams using genetic algorithms. *Expert Systems with Applications : An International Journal*, 12(1), January 1997.

[12] 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, apr 1996.

[13] Dragan Cvetković, Ian Parmee, and Eric Webb. Multi-objective Optimisation and Preliminary Airframe Design, April 1998.

[14] Kalyanmoy Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, 1998.

[15] 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, California, jun 1989. George Mason University, Morgan Kaufmann Publishers.

[16] L. Duckstein. Multiobjective Optimization in Structural Design: The Model Choice Problem. In E. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz, editors, *New Directions in Optimum Structural Design*, pages 459–481. John Wiley and Sons, 1984.

[17] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[18] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. Technical report, Department

of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U. K., 1994.

[19] Carlos M. Fonseca and Peter J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms I: A Unified Formulation. Technical Report 564, University of Sheffield, Sheffield, UK, January 1995.

[20] Carlos M. Fonseca and Peter J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms II: Application Example. Technical Report 565, University of Sheffield, Sheffield, UK, January 1995.

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

[22] Carlos M. Fonseca and Peter J. Fleming. Nonlinear System Identification with Multiobjective Genetic Algorithms. In *Proceedings of the 13th World Congress of IFAC*, pages 187–192, San Francisco, California, 1996.

[23] Carlos M. Fonseca and Peter J. Fleming. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 584–593. Springer-Verlag, Berlin, Germany, September 1996.

[24] M. 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. Lawrence Erlbaum, 1985.

[25] M. Gen, K. Ida, and Y. Li. Solving bicriteria solid transportation problem with fuzzy numbers by genetic algorithm. *International Journal of Computers and Industrial Engineering*, 29:537–543, 1995.

[26] Mitsuo Gen and Runwei Cheng. *Genetic Algorithms and Engineering Design*. John Wiley and Sons, Inc., New York, 1997.

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

[28] 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, California, 1991.

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

[30] Garrison W. Greenwood, Xiaobo Sharon Hu, and Joseph G. D'Ambrosio. Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings. In Richard K. Belew and Michael D.

Vose, editors, *Foundations of Genetic Algorithms 4*, pages 437–455. Morgan Kaufmann, San Mateo, California, 1997.

[31] J. J. Grefenstette. GENESIS: A system for using genetic search procedures. In *Proceedings of the 1984 Conference on Intelligent Systems and Machines*, pages 161–165, 1984.

[32] Y. Y. Haimes, W. Hall, and H. Freedman. *Multi-Objective Optimization in Water Resources Systems: The Surrogate Trade-Off Method.* Elsevier, Amsterdam, 1975.

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

[34] M. R. Hilliard, G. E. Liepins, M. Palmer, and G. Rangarajen. The computer as a partner in algorithmic design: Automated discovery of parameters for a multiobjective scheduling heuristic. In R. Sharda, B. L. Golden, E. Wasil, O. Balci, and W. Stewart, editors, *Impacts of Recent Computer Advances on Operations Research.* North-Holland Publishing Company, New York, 1989.

[35] John H. Holland. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* MIT Press, Cambridge, Massachusetts, second edition, 1992.

[36] J. Horn and N. Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.

[37] J. P. Ignizio. *Goal Programming and Extensions.* Heath, Lexington, Massachusetts, 1976.

[38] J. P. Ignizio. The determination of a subset of efficient solutions via goal programming. *Computing and Operations Research*, 3:9–16, 1981.

[39] Y. Ijiri. *Management Goals and Accounting for Control.* North-Holland, Amsterdan, 1965.

[40] Hisao Ishibuchi and Tadahiko Murata. Multi-Objective Genetic Local Search Algorithm. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation*, pages 119–124, Nagoya, Japan, 1996. IEEE.

[41] W. Jakob, M. Gorges-Schleuter, and C. Blume. Application of genetic algorithms to task planning and learning. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2nd Workshop*, Lecture Notes in Computer Science, pages 291–300, Amsterdam, 1992. North-Holland Publishing Company.

[42] Gareth Jones, Robert D. Brown, David E. Clark, Peter Willett, and Robert C. Glen. Searching Databases of Two-Dimensional and Three-Dimensional Chemical Structures using Genetic Algorithms. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 597–602, San Mateo, California, 1993. Morgan Kaufmann.

[43] H. Jutler. Liniejnaja modiel z nieskolkimi celevymi funkcjami (linear model with several objective functions). *Ekonomika i matematiceckije Metody*, 3:397–406, 1967. (In Polish).

[44] John R. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* The MIT Press, 1992.

[45] 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, California, 1951. University of California Press.

[46] A. Gaspar Kunha, Pedro Oliveira, and José A. Covas. Use of Genetic Algorithms in Multicriteria Optimization to Solve Industrial Problems. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 682–688, San Mateo, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.

[47] Frank Kursawe. A variant of evolution strategies for vector optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Berlin, Germany, oct 1991. Springer-Verlag.

[48] G. E. Liepins, M. R. Hilliard, J. Richardson, and M. Palmer. Genetic algorithms application to set covering and travelling salesman problems. In D. E. Brown and C. C. White, editors, *Operations research and Artificial Intelligence: The integration of problem-solving strategies*, pages 29–57. Kluwer Academic, Norwell, Massachusetts, 1990.

[49] J. G. Lin. Maximal vectors and multi-objective optimization. *Journal of Optimization Theory and Applications*, 18(1):41–64, jan 1976.

[50] Joanna Lis and A. E. Eiben. A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation*, pages 59–64, Nagoya, Japan, 1996. IEEE.

[51] 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.

[52] Daniel H. Loughlin and S. Ranjithan. The Neighborhood constraint method: A Genetic Algorithm-Based Multiobjective Optimization Technique. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 666–673, San Mateo, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.

[53] S. M. Mahfoud. Crowding and preselection revisited. In R. Männer and B. Manderick, editors, *Parallel problem Solving from Nature, 2nd Workshop*, Amsterdam, 1992. North-Holland Publishing Company.

[54] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, second edition, 1992.

[55] E. Michielssen and D. S. Weile. Electromagnetic System Design using Genetic Algorithms. In *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, pages 267–288. John Wiley and Sons, England, 1995.

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

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

[58] Shigeru Obayashi. Pareto Genetic Algorithm for Aerodynamic Design using the Navier-Stokes Equations. 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 12, pages 245–266. John Wiley and Sons, West Sussex, England, 1997.

[59] A. Osyczka. An approach to multicriterion optimization problems for engineering design. *Computer Methods in Applied Mechanics and Engineering*, 15:309–333, 1978.

[60] A. Osyczka. An approach to multicriterion optimization for structural design. In *Proceedings of International Symposium on Optimal Structural Design*. University of Arizona, 1981.

[61] A. Osyczka and J. Koski. Selected works related to multicriterion optimization methods for engineering design. In *Proceedings of Euromech Colloquium*, University of Siegen, 1982.

[62] A. Osyczka and S. Kundu. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural Optimization*, 10:94–99, 1995.

[63] Andrzej Osyczka. Optimization of the steady state parameters for machine tool gear trains. *International Journal of Machine Tool Design and Research*, 15:31–68, 1975.

[64] Andrzej Osyczka. *Multicriterion Optimization in Engineering with FORTRAN programs*. Ellis Horwood Limited, 1984.

[65] Andrzej Osyczka. Multicriteria optimization for engineering design. In John S. Gero, editor, *Design Optimization*, pages 193–227. Academic Press, 1985.

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

[67] I. C. Parmee and G. Purchase. The development of a directed genetic search technique for heavily constrained design spaces. In I. C. Parmee, editor, *Adaptive Computing in Engineering Design and Control- '94*, pages 97–102, Plymouth, UK, 1994. University of Plymouth, University of Plymouth.

[68] 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.

[69] 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.

[70] David Powell and Michael M. Skolnick. Using genetic algorithms in engineering design optimization with non-linear constraints. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 424–431, San Mateo, California, jul 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers.

[71] Domenico Quagliarella and Alessandro Vicini. Coupling Genetic Algorithms and Gradient Based Optimization Techniques. 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 14, pages 289–309. John Wiley and Sons, West Sussex, England, 1997.

[72] S. Ranjithan, J. W. Eheart, and J. C. Liebman. Incorporating fixed-cost component of pumping into stochastic groundwater management: A genetic algorithm-based optimization approach. *Eos Transactions AGU*, 73(14):125, 1992. Spring meeting supplement.

[73] S. Rao. Game theory approach for multiobjective structural optimization. *Computers and Structures*, 25(1):119–127, 1986.

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

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

[76] 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, George Mason University, 1989. Morgan Kaufmann Publishers.

[77] 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.

[78] Katya Rodríguez-Vázquez, Carlos M. Fonseca, and Peter J. Fleming. Multiobjective Genetic Programming : A Nonlinear System Identification Application. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 207–212, Stanford University, California, July 1997. Stanford Bookstore.

[79] M. A. Roseman and J. S. Gero. Reducing the Pareto optimal set in multicriteria optimization. *Engineering Optimization*, 8:189–206, 1985.

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

[81] Günter Rudolph. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 511–516, Piscataway, New Jersey, 1998. IEEE Press.

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

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

[84] H. P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley and sons, Great Britain, 1981.

[85] R. Solich. Zadanie programowania liniowego z wieloma funkcjami celu (linear programming problem with several objective functions). *Przeglad Statystyczny*, 16:24–30, 1969. (In Polish).

[86] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. Technical report, Department of Mechanical Engineering, Indian Institute of Technology, Kanput, India, 1993.

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

[88] Timothy J. Stanley and Trevor Mudge. A Parallel Genetic Algorithm for Multiobjective Microprocessor Design. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 597–604, San Mateo, California, July 1995. University of Pittsburgh, Morgan Kaufmann Publishers.

[89] 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*, Lecture Notes in Computer Science, pages 166–180. Springer-Verlag, Sheffield, U.K., 1995.

[90] Gilbert Syswerda and Jeff Palmucci. The Application of Genetic Algorithms to Resource Scheduling. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 502–508, San Mateo, California, 1991. Morgan Kaufmann.

[91] Hishashi 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*, pages 517–522, Nagoya, Japan, 1996. IEEE.

[92] Hishashi Tamaki, M. Mori, M. Araki, and H. Ogai. Multicriteria optimization by genetic algorithms : a case of scheduling in hot rolling process. In *Proceedings of the 3rd APORS*, pages 374–381, 1995.

[93] Kay Chen Tan and Yun Li. Multi-Objective Genetic Algorithm Based Time and Frequency Domain Design Unification of Linear Control Systems. Technical Report CSC-97007, Department of Electronics and Electrical Engineering, University of Glasglow, Glasglow, Scotland, 1997.

[94] David S. Todd and Pratyush Sen. A Multiple Criteria Genetic Algorithm for Containership Loading. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 674–681, San

Mateo, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.

[95] C. H. Tseng and T. W. Lu. Minimax multiobjective optimization in structural design. *International Journal for Numerical Methods in Engineering*, 30:1213–1228, 1990.

[96] 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, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.

[97] Ganesh Vedarajan, Louis Chi Chan, and David E. Goldberg. Investment Portfolio Optimization using Genetic Algorithms. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 255–263, Stanford University, California, July 1997. Stanford Bookstore.

[98] 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 University, California, July 1998. Stanford University Bookstore.

[99] 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, Ohio, 1998.

[100] S. Voget and M. Kolonko. Multidimensional Optimization with a Fuzzy Genetic Algorithm. *Journal of Heuristics*, 1998. (In Press).

[101] C. White, A. Sage, and S. Dozono. A model of multiattribute decision-making and tradeoff weight determination under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14:223–229, 1984.

[102] 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.

[103] 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.

[104] X. Yang and M. Gen. Evolution program for bicriteria transportation problem. In M. Gen and T. Kobayashi, editors, *Proceedings of the 16th International Conference on Computers and Industrial Engineering*, pages 451–454, Ashikaga, Japan, 1994.

[105] Eckart Zitzler and Lothar Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998.

[106] Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Par-*

allel Problem Solving from Nature V, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

Carlos A. Coello Coello was born in México in 1967. He received a Civil Engineering degree with honors from the Autonomous University of Chiapas in 1991. Then he pursued graduate studies in Computer Science at Tulane University, in New Orleans, Louisiana, USA, from where he received a Masters degree in 1993 and a PhD in 1996. He has been visiting professor of Computer Science at DePauw University, in Greencastle, Indiana, USA, and Senior Research Fellow at the Engineering Design Centre of the University of Plymouth, in the United Kingdom. He is currently a full-time researcher at LANIA (Laboratorio Nacional de Informática Avanzada), in Xalapa, Veracruz, México. His current interest is the use of evolutionary techniques for engineering optimization, manufacturing and design.