An Updated Survey of GA-Based Multiobjective Optimization Techniques

CARLOS A. COELLO COELLO

Laboratorio Nacional de Informática Avanzada

After using evolutionary techniques for single-objective optimization during more than two decades, the incorporation of more than one objective in the fitness function has finally become a popular area of research. As a consequence, many new evolutionary-based approaches and variations of existing techniques have recently been published in the technical literature. The purpose of this paper is to summarize and organize the information on these current approaches, emphasizing the importance of analyzing the operations research techniques in which most of them are based, in an attempt to motivate researchers to look into these mathematical programming approaches for new ways of exploiting the search capabilities of evolutionary algorithms. Furthermore, a summary of the main algorithms behind these approaches is provided, together with a brief criticism that includes their advantages and disadvantages, degree of applicability, and some known applications. Finally, future trends in this area and some possible paths for further research are also addressed.

Categories and Subject Descriptors: I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms: Algorithms

Additional Key Words and Phrases: Artificial intelligence, genetic algorithms, multicriteria optimization, multiobjective optimization, vector optimization

1. INTRODUCTION

Multiobjective optimization is without a 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 operations research, more than 20 techniques have been developed over the years to try to deal with functions that have multiple objectives, and many approaches have been suggested, going all the way from naively combining objectives into one, to the use of game theory to coordinate the relative importance of each objective. The fuzziness of this area lies in the fact that there is no accepted definition of "optimum" as in single-objective optimization. Hence it is difficult to even compare the results of one method to another's because, normally, the decision about

This work was performed while the author was affiliated to the Plymouth Engineering Design Centre, in the United Kingdom.

Author's address: Laboratorio Nacional de Informática Avanzada, Rébsamen No. 80, A.P. 696, Xalapa, Veracruz 91090, Mexico; email: ccoello@xalapa.lania.mx.

CONTENTS

the "best" answer corresponds to the so-called (human) *decision maker*.

There have been other surveys of multiobjective optimization techniques in the mathematical programming literature, from which the papers by Cohon and Marks [1975]; Hwang et al. [1980]; Stadler [1984]; Lieberman [1991]; Evans [1984]; Fishburn [1978]; and Boychuk and Ovchinnikov [1973] are probably the most comprehensive. The most remarkable survey of multiobjective optimization in the evolutionary computing literature is the one by Fonseca and Fleming [1994; 1995c]. However, little detail is provided in this work on how each method works, only a few applications of each technique are given, and their corresponding operations research roots are scarcely mentioned. Furthermore, several other approaches have arisen since the publication of Fonseca's paper, and the intention of the present work is to provide researchers and students with an updated survey that can be used (to a certain extent) as a self-contained document for anyone with a previous (at least basic) general knowledge of genetic algorithms (GAs). Those who need additional information about genetic algorithms should refer to Goldberg [1989]; Holland [1992]; Michalewicz [1992]; and Mitchell [1996].

This paper emphasizes the importance of looking at previous work in operations research, not only to get a full understanding of some of the existing techniques, but also to motivate the development of new GA-based approaches. Finally, applications of each method are also given to provide the reader with a more complete idea of the range of applicability and the underlying motivation of each of these techniques. A brief criticism appears after the description of each technique listing its advantages, possible drawbacks and limitations, and (in some cases) possible ways to exploit its characteristics or even improve performance.

2. STATEMENT OF THE PROBLEM

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

> *"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."* [Osyczka 1985]

Formally, we can state this as follows: Find the vector $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ that will satisfy the m inequality constraints:

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

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 F of all numbers that satisfy (1) and (2) the particular set $x_1^*, x_2^*, \ldots, x_k^*$ that yields the optimum values of all the objective functions.

The constraints given by (1) and (2) define the *feasible region* F and any point $\bar{x} \in F$ defines a *feasible solution*. The vector function $\bar{f}(\bar{x})$ is a function that maps the set F in the set X which represents all possible values of the objective functions. The k components of the vector $\bar{f}(\bar{x})$ represent the *noncommensurable* criteria[1] that must be considered. The constraints $g_i(\bar{x})$ and $h_i(\bar{x})$ represent the restriction imposed on the decision variables. The vector $\bar{x}^*$ is reserved to denote the optimal solutions (normally there is more than one).

The problem is that the meaning of *optimum* is not well defined in this context, since we rarely have an $\bar{x}^*$ such that for all $i = 1, 2, \ldots, k$

$$\bigwedge_{x \in F} (f_i(\bar{x}^*) \leq f_i(\bar{x})) \tag{4}$$

If this were the case, then $\bar{x}^*$ would be a desirable solution, but normally we never have a situation in which all the $f_i(\bar{x})$ have a minimum in F at a common point x^*. An example of the ideal situation is shown in Figure 1. However, since this situation is rare in real-world problems, we have to establish a criterion to determine what an "optimal" solution is.

[1]*Noncommensurable* means that the values of the objective functions are expressed in different units.

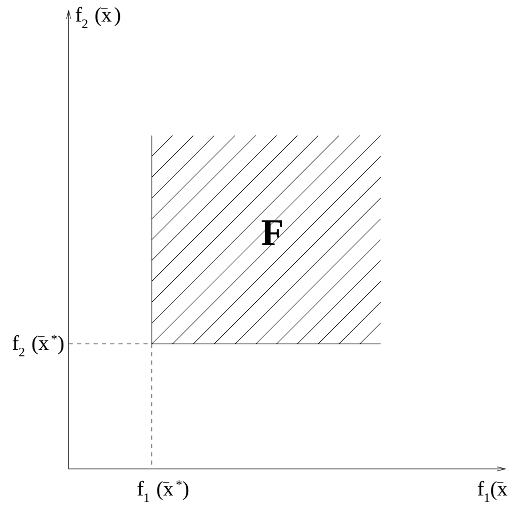

Figure 1. Ideal solution in which all our objective functions have their minimum at a common point.

2.1 Ideal Vector

Assume that we find the minimum (or maximum) of each of the objective functions $f_i(\bar{x})$ separately. Assuming that they can be found, let

$$\bar{x}^{0(i)} = [x_1^{0(i)}, x_2^{0(i)}, \ldots, x_n^{0(i)}]^T \tag{5}$$

be a vector of variables that optimizes (or minimizes or maximizes) the ith objective function $f_i(x)$. In other words, the vector $\bar{x}^{0(i)} \in X$ is such that

$$f_i(\bar{x}^{0(i)}) = opt_{x \in F} f_i(\bar{x}) \tag{6}$$

In general, there is a unified criterion with respect to "opt." Most authors prefer to treat it as a minimum, in which case $f_i(\bar{x}^{0(i)})$, or simply f_i^0 (more convenient notation) denotes the minimum value of the ith function. Hence, the vector $\bar{f}^0 = [f_1^0, f_2^0, \ldots, f_k^0]^T$ is ideal for a multiobjective optimization problem. The point in R^n that determines this vector is the ideal (utopian) solution, and consequently is called the *ideal vector*.

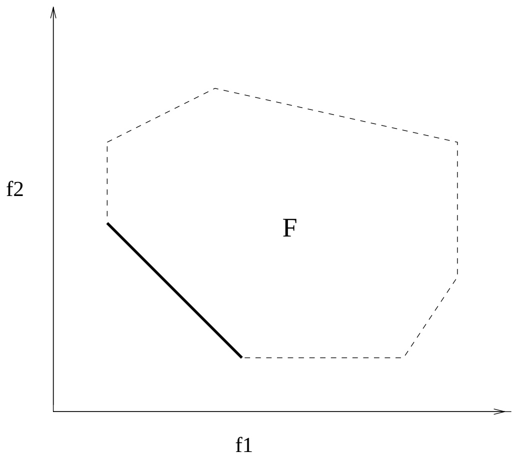

Figure 2. An example of a problem with two objective functions. The Pareto front is marked with a bold line (the two criteria are to be minimized).

2.2 Pareto Optimum

The concept of a *Pareto optimum* was formulated by Vilfredo Pareto in the nineteenth century [Pareto 1896], and by itself constitutes the origin of research in multiobjective optimization. We say that a point $\bar{x}^* \in F$ is *Pareto optimal* if for every $\bar{x} \in F$, either

$$\bigwedge_{i \in I} (f_i(\bar{x}) = f_i(\bar{x}^*)), \tag{7}$$

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

$$f_i(\bar{x}) > f_i(\bar{x}^*) \tag{8}$$

In words, this definition says that $\bar{x}^*$ is Pareto optimal if there exists no feasible vector $\bar{x}$ that decreases 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 *noninferior* or *nondominated* solutions.

2.3 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 Figure 2, a bold line is used to mark this boundary for a bi-objective 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 F^k and their corresponding $f(F^k)$. When we have a sufficient number of these, we may proceed to take the final decision.

A point $\bar{x}^* \in F$ is a *weakly nondominated solution* if there is no $\bar{x} \in F$ such that $f_i(\bar{x}) < f_i(\bar{x}^*)$ for $i = 1, \ldots, n$. A point $\bar{x}^* \in F$ is a *strongly nondominated solution* if there is no $\bar{x} \in F$ such that $f_i(\bar{x}) \leq f_i(\bar{x}^*)$, for $i = 1, \ldots, n$ and for at least one value of i, $f(\bar{x}) < f(\bar{x}^*)$.

Thus, if $\bar{x}^*$ is *strongly nondominated*, it is also *weakly nondominated*; but the converse is not necessarily true. Nondominated solutions for the biobjective case can readily be represented graphically by passing into the objective function space $\{f_1(\bar{x}), f_2(\bar{x})\}$. The so-called *minimal curve* corresponds to the locus of strongly nondominated points, and the *weakly minimal curve* to the locus of weakly nondominated points [Baier 1977]. These two curves are sketched in Figure 3 for a simple biobjective problem.

3. THE NEED TO PRESERVE DIVERSITY

Due to stochastic errors associated with its genetic operators, the genetic algorithm (GA) tends to converge to a single solution when used with a finite population [Deb and Goldberg 1989]. As long as our goal is to find the global optimum (or at least a very good approximation of it), this behavior is acceptable. However, there are certain applications in which we are interested in finding not one but several solutions. Multiobjective optimization is certainly one of those applications because we want to find

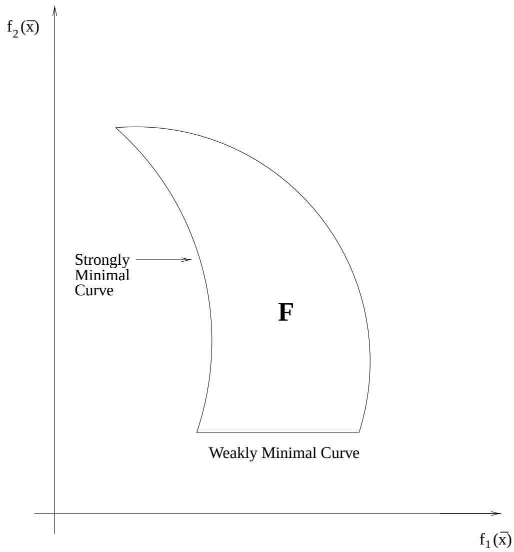

Figure 3. Weakly and strongly nondominated curves on the biobjective case.

the entire Pareto front of a problem, and not only a single nondominated solution. The question then is how to keep the GA from converging to a single solution.

Early researchers in genetic algorithms identified the GA convergence phenomenon, called *genetic drift* [DeJong 1975], and found that it occurs in nature as well. They stated correctly that the key to solving this problem is to find a way to preserve diversity in the population—this observation resulted in several proposals modeled after natural systems: Holland [1975] suggested the use of a "crowding" operator intended to identify situations in which more and more individuals dominate an environmental niche, since in such cases the competition for limited resources increases rapidly and results in lower life expectancies and birth rates. DeJong [1975] experimented with such a *crowding* operator, implemented by having a newly formed offspring replace an existing individual with one most similar to itself. The similarity between two individuals was measured in the genotype by counting the number of bits that match along each chromosome of each individual. DeJong used two parameters in his model: generation gap (G) and crowding factor (CF) [Deb and Goldberg 1989]. The first parameter indicates the percentage of the population that is allowed to reproduce. The second parameter specifies the number of individuals initially selected as candidates to be replaced by a particular offspring [DeJong 1975]. Therefore, CF=1 means that no crowding will take place, and as we increase the value of CF, it becomes more likely that similar individuals will replace one another [DeJong 1975].

Goldberg and Richardson [1987] used a different approach, in which the population is divided into different subpopulations according to the similarity of individuals in two possible solution spaces: the decoded parameter space (phenotype) and the gene space (genotype). They defined a sharing function $\phi(d_{ij})$ as follows [Goldberg and Richardson 1987]:

$$\phi(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\sigma_{sh}}\right)^{\alpha}, & d_{ij} < \sigma_{sh} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

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

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{M} \phi(d_{ij})} \tag{10}$$

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

Deb and Goldberg [1989] proposed a way of estimating the parameter σ_{share} in both phenotypical and genotypical space. In phenotypical sharing, the distance between two individuals is measured in decoded parameter space, and can be calculated with a simple Euclidean distance in a p-dimensional space, where p refers to the number of variables encoded in the GA; the value of d_{ij} can then be calculated as

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{k,i} - x_{k,j})^2} \qquad (11)$$

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

To estimate the value of σ_{share}, Deb and Goldberg [1989] proposed using the expression:

$$\sigma_{share} = \frac{r}{\sqrt[p]{q}} = \frac{\sqrt{\sum_{k=1}^{p} (x_{k,max} - x_{k,min})^2}}{\sqrt[p]{2q}} \qquad (12)$$

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

In genotypical sharing, d_{ij} is defined as the Hamming distance between the strings, and σ_{share} is the maximum number of different bits allowed between the strings in order to form separate niches in the population. Experiments by Deb and Goldberg [1989] show that sharing is a better way of maintaining diversity than crowding, and indicate that phenotypic sharing is better than genotypic sharing.

Much more work has been done on maintaining diversity in a population. Deb and Goldberg [1989] suggested the use of restrictive mating with respect to the phenotypic distance. The idea is to allow two individuals to reproduce only if they are very similar (i.e., if their phenotypic distance is less than a factor called σ_{share}). This is intended to produce distinct "species" (mating groups) in the population [Mitchell 1996]. Some researchers, Eshelman [1991] and Schaffer [1991], did exactly the opposite: they did not allow mating between individuals who were too similar (to "prevent incest").

Smith et al. [1993] proposed an approach, modeled after the immune system, that can maintain the diversity of the population without the use of an explicit sharing function. This approach has actually been used by Hajela [1996; 1997] to handle constraints in structural optimization problems.

To preserve diversity, Poloni and Pediroda [1997] proposed an interesting alternative approach, called "local Pareto selection," which basically consists of placing the population on a toroidal grid and choosing the members of the local tournament by means of a random walk in the neighborhoods of the given grid point.

Kita et al. [1996] proposed the so-called "thermodynamical genetic algorithm" (TDGA) to maintain diversity when using a Pareto ranking technique for multiobjective optimization. The TDGA is inspired by the principle of minimal free energy used in simulated annealing. The idea is to select the individuals for a new generation in such a way that the free energy F is minimized and

$$F = \langle E \rangle - HT \qquad (13)$$

where $\langle E \rangle$ is the mean energy of the system, H is the entropy, and T is the temperature. The diversity of the population is controlled by adjusting T according to a certain schedule (as in simulated annealing). T is presumably less sensitive to population size and the size of the feasible region than traditional sharing functions [Tamaki et al. 1996].

4. NAIVE APPROACHES TO MULTIOBJECTIVE OPTIMIZATION

The notion of genetic search in a multicriteria problem dates back to the late 1960s, when Rosenberg [1967] suggested using multiple *properties* (nearness to some specified chemical composition) in his simulation of the genetics and chemistry of a population of single-celled organisms. This would have led to multicriteria optimization if it had been carried out. Since his actual implementation contained only one single property, the multiobjective approach could not be shown.

Knowing that a genetic algorithm needs scalar fitness information to work, the simplest idea that we could devise is to combine all the objectives into a single one using addition, multiplication, or any other combination of arithmetical operations that we could think of. There are obvious problems with this approach, though. The first is that we have to provide some accurate scalar information on the range of the objectives, to avoid having one of them dominate the others. This implies that we should know, to a certain extent, the behavior of each of the objective functions. This is normally (at least in most real-world applications) a very expensive process (computationally speaking), which we could not afford in most cases. If this combination of objectives is possible (and it is in some applications), this is not only the simplest approach, but also one of the most efficient, since no further interaction with the decision maker is required. If the GA succeeds at optimizing the resulting fitness function, then the results will be at least suboptimum in most cases.

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

4.1 Weighted Sum Approach

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

$$\min \sum_{i=1}^{k} w_i f_i(\bar{x}) \tag{14}$$

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

$$\sum_{i=1}^{k} w_i = 1 \tag{15}$$

Since the results of solving an optimization model using (14) can vary significantly as the weighting coefficients change, and since very little is usually known about how to choose these coefficients, it is necessary to solve the same problem for many different values of w_i. But in this case the designer is still confronted with choosing the most appropriate solution based on intuition.

Note that the weighting coefficients do not proportionally reflect 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 (14), this location depends not only on w_i values, but also on the units in which the functions are expressed.

If we want w_i to closely reflect the importance of the objectives, all functions should be expressed in units of approximately the same numerical value. Additionally, we can also transform (14) to the form

$$\min \sum_{i=1}^{k} w_i f_i(\bar{x}) c_i \tag{16}$$

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

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

Applications

—Syswerda and Palmucci [1991] 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. [1992] 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. [1993] used weights for their genetic operators in order to reflect their effectiveness when a GA is applied to generate hyperstructures from a set of chemical structures.

—Wilson and Macleod [1993] used this approach as one of the methods in 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. [1998] 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, robustness, and modal controllability of the controlled system subject to total weight, asymptotical stability, and eigenvalues constraints.

—Yang and Gen [1994] used a weighted sum approach to solve a bicriteria linear transportation problem. More recently, Gen et al. [1995; 1997] extended this approach to allow more than two objectives and added fuzzy logic to handle the uncertainty 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 identify Pareto solutions, since the coefficients of the objectives are represented with fuzzy numbers reflecting the existing uncertainty of their relative importance.

Criticism. This was the first method developed for generating noninferior solutions for multiobjective optimization, and is an obvious consequence of the seminal work of Kuhn and Tucker on numerical optimization [Kuhn and Tucker 1951]. This method is very efficient computationally speaking, and can be applied to generate a strongly nondominated solution which can provide initial answers to other problems. The difficulty with this approach is determining 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 use a simple linear combination of the objectives and then generate the *trade-off surface*[2] by varying the weights. This approach is very simple and easy to implement, but it has the disadvantage of missing concave portions of the trade-off curve[3] [Ritzel et al. 1994], which is a serious drawback in most real-world applications.

4.2 Goal Programming

Charnes and Cooper [1961] and Ijiri [1965] 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, decision makers have to assign targets or goals that they wish 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

[2]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.

[3]In other words, it does not work properly with nonconvex Pareto fronts.

form of this method may be formulated as follows [Duckstein 1984]:

$$\min \sum_{i=1}^{k} |f_i(\bar{x}) - T_i|, \text{ subject to } \bar{x} \in F \quad (17)$$

where T_i denotes the target or goal set by the decision maker for the ith objective function $f_i(\bar{x})$, and 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 $|f_i(\bar{x}) - T_i|$ [Haimes et al. 1975]. Such a formulation has been called *generalized goal programming* [Ignizio 1976; 1981]. This technique has also been called "target vector optimization" by other authors [Coello 1996].

Applications. Wienke et al. [1992] used this approach in combination with a genetic algorithm to simultaneously optimize the intensities of six atomic emission lines of trace elements in alumina powder as a function of spectroscopic excitation conditions. Eric Sandgren [1994] also used goal programming coupled with a genetic algorithm to optimize plane trusses and the design of a planar mechanism.

Criticism. This technique will yield a dominated solution if the goal point is chosen in the feasible domain [Duckstein 1984]. It may be a very efficient approach (computationally speaking) if we know the desired goals and they are in the feasible region. However, the decision maker must devise the appropriate weights or priorities for the objectives that will eliminate the noncommensurable characteristics of the problem. This is difficult in most cases, 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, due to the availability of excellent computer programs and the possibility of easily eliminating dominated goal points, this technique may be useful in cases where a linear or piecewise-linear approximation of the objective functions can be made. On the other hand, in nonlinear cases, other approaches may be more efficient.

4.3 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:

$$\text{Minimize } \alpha \quad (18)$$

subject to

$$\begin{aligned} &g_j(\bar{x} \leq 0); && j = 1, 2, \ldots, m \\ &b_i + \alpha \cdot w_i \geq f_i(\bar{x}); && i = 1, 2, \ldots, k \end{aligned} \quad (19)$$

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

$$\sum_{i=1}^{k} |w_i| = 1 \quad (20)$$

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

It can easily be shown [Chen and Liu 1994] that the set of nondominated solutions can be generated by varying the weights, with $w_i \geq 0$ $(i = 1, 2, \ldots, k)$, even for nonconvex problems. The mechanism by which this method operates is illustrated in Figure 4. The vector $\bar{b}$ is represented by the decision goal of the decision maker, who also decides the direction of $\bar{w}$. Given vectors $\bar{w}$ and $\bar{b}$, the direction of the vector $\bar{b} + \alpha \cdot \bar{w}$ can be determined and the problem

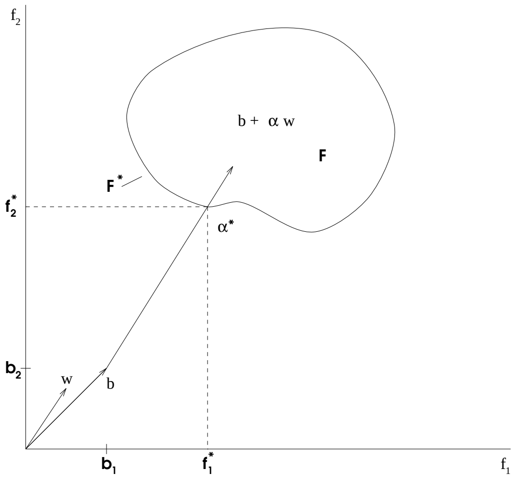

Figure 4. Goal-attainment method with two objective functions.

stated by Equation (18) is equivalent to finding a feasible point on this vector in objective space that is closest to the origin. It is obvious that the optimal solution of Equation (18) will be the first point at which $\bar{b} + \alpha \cdot \bar{w}$ intersects the feasible region F in the objective space. Should this point of intersection exist, it would clearly be a noninferior (or nondominated) solution.

Note that the optimum value of α will inform the decision maker 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's goal is unattainable.

Applications. Wilson and MacLeod [1993] used this approach as another method incorporated into their GA to design multiplierless IIR filters.

Criticism. As Wilson and MacLoud [1993] indicate, goal attainment has several problems: the main one is the misleading selection pressure that it can generate in some cases. For example, if we have two candidate solutions that are the same in one objective function value but different in another, they will still have the same goal-attainment value for their two objectives, which means that for the GA neither is better than the other.

4.4 The ε-Constraint Method

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

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

$$f_r(\bar{x}^*) = \min_{x \in F} f_r(\bar{x}) \tag{21}$$

subject to additional constraints of the form

$$f_i(\bar{x}) \leq \varepsilon_i \text{ for } i = 1, 2, \ldots, k \text{ and } i \neq r \tag{22}$$

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

(2) Repeat (1) for different values of ε_i. The information derived from a well-chosen set of ε_i can be useful in making the decision. The search 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

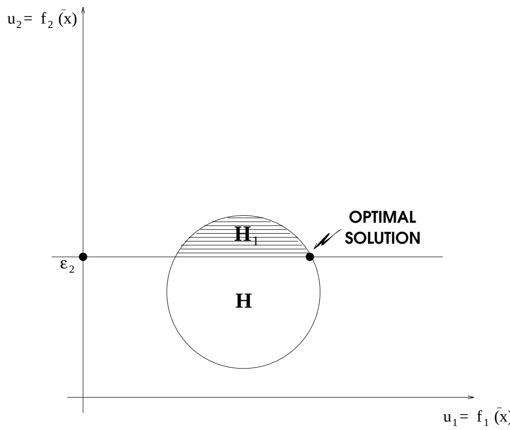

Figure 5. The ε-constraint method for a maximizing problem.

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.,

$$\varepsilon_i \geq f_i(\bar{x}_i^*)$$

$$i = 1, 2, \ldots, r-1, r+1, \ldots, k \quad (23)$$

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

$$\varepsilon_i \leq f_i(\bar{x}_r^*)$$

$$i = 1, 2, \ldots, r-1, r+1, \ldots, k \quad (24)$$

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

Figure 5 illustrates the ε-constraint method for a maximizing problem where H is the payoff set of the original problem, restricted to the shadowed area H_1 by the further constraint $f_2(\bar{x}) \geq \varepsilon_2$ (we are maximizing), and the objective function f_1 is maximized subject to the assumption that $\bar{x}$ belongs to H_1. Thus, the most important objective (in this case f_1) has been optimized, and the others, as mentioned before, are treated as additional constraints.

Szidarovszky and Duckstein [1982] showed that the ε-constraint method usually leads to *weakly nondominated solutions*; however, if the optimal solution is unique, then such solutions become *strongly nondominated*.

This approach was suggested by Ritzel and Wayland [1994] as a simple and naive way of solving multiobjective optimization problems using a GA. The idea is to code the GA in such a way that all the objectives, except one, are kept constant (constrained to a single value), and the remaining objective then becomes the fitness function for the GA. Thus, through the process of running the GA numerous times with different values of the constrained objectives, a trade-off surface can be developed.

In the mathematical programming literature, this approach is also known as the *trade-off method*, due to its main concept of trading a value of one objective function for a value of another function.

Applications. Quagliarella and Vicini [1997] suggested the use of this technique, coupled with a hybrid GA (a genetic algorithm that uses gradient-based optimization techniques to speed up the search, in order to reduce the computational cost in a real-world application) to solve multiobjective optimization problems. Ranjithan et al. [1992] used this approach to solve groundwater pollution containment problems.

Loughlin and Ranjithan [1997] 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

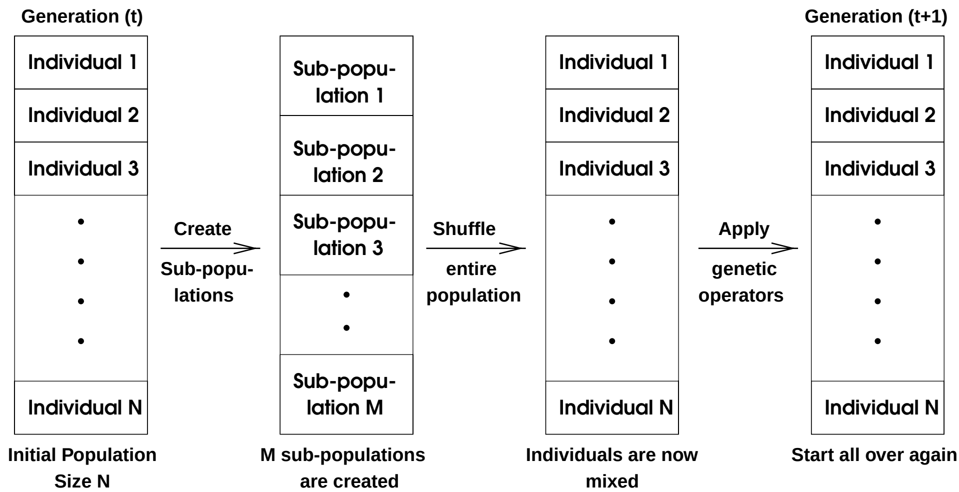

Figure 6. Schematic of VEGA selection. It is assumed that the population size is N and that there are M objective functions.

reduction (this is a combinatorial problem suitable for integer programming techniques).

Criticism. The obvious drawback 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, finding weakly nondominated solutions may not be appropriate in some applications (e.g., structural optimization). Nevertheless, the relative simplicity of the technique has made it popular among some GA practitioners.

5. NONAGGREGATING APPROACHES THAT ARE NOT PARETO-BASED

To overcome the difficulties 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 [Powell and Skolnick 1993]. Some of the most popular approaches that fall into this category are examined in this section.

5.1 VEGA

Schaffer [1985] extended Grefenstette's GENESIS program [Grefenstette 1984] to include multiple objective functions. Schaffer's approach uses an extension of the simple genetic algorithm (SGA), called the *vector evaluated genetic algorithm* (VEGA), which only differs from the first (SGA) algorithm in the way selection is performed. The operator was modified so that, at each generation, a number of subpopulations are generated by proportional selection, in turn, according to each objective function. Thus, for a problem with k objectives, k subpopulations of size N/k each are generated (assuming N total population size). These subpopulations are shuffled together to obtain a new population of size N, on which the GA applies the crossover and mutation operators in the usual way. This process is illustrated in Figure 6. Schaffer realized that the solutions generated by his system were nondominated in a local sense, since their nondominance was limited to the current population. And while a locally dominated individual is also globally dominated, the converse is not necessarily true [Schaffer 1985]. An individual who is not dominated in one generation may become dominated by an individual who emerges in a later generation. Schaffer also noted a problem which, in genetics, is known as "speciation" (i.e., the evolution of a "species" within a population that excels in some respect). This problem arises because this technique selects individuals who excel in one dimension, without looking at other dimensions. The potential danger is that we could evolve individuals with what Schaffer calls "middling" performance[4] in all dimensions, which could be very useful for compromise solutions, but will not survive under this selection scheme, since they do not excel in any dimension (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 suggests some heuristics to deal with this problem (e.g., a heuristic selection preference approach for nondominated individuals in each generation, so as to protect "middling" chromosomes). Crossbreeding among "species" could also be encouraged by adding some mate selection heuristics, instead of the traditional random GA mate selection.

Applications. Ritzel and Wayland [1994] used a variant of VEGA in which they incorporate a parameter to control the selection ratio. In the case of the groundwater pollution containment problem (which Ritzel and Wayland solved), there were only two objectives. 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. [1995] proposed an interesting application of VEGA to model constraints in a single-objective optimization problem so as to avoid the need for a penalty function. Surry et al., however, modified the standard VEGA procedure and introduced a form of ranking based on the number of constraints violated by a certain solution. They reported that their approach worked well in optimizing gas supply networks, since the tendency of VEGA to favor certain solutions can actually be an advantage when handling constraints. In such a situation we want to favor precisely the solution that does not violate any constraint over those that do.

Ćvetković et al. [1998] proposed several approaches to overcome VEGA's problems. For example, to wait for a certain number of generations before shuffling the population together, or not do it at all, but instead copy or migrate a certain number of individuals from one subpopulation to another. They used these and other traditional multiobjective optimization approaches for preliminary airframe design.

Tamaki et al. [1995; 1996] developed a technique in which, at each generation, nondominated 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. If the number of nondominated individuals is fewer than the population size, the remainder of the population in the following generation is filled by applying VEGA to the dominated individuals. On the other hand, if the number of the nondominated individuals exceeds the population size, by using VEGA, individuals in the following generation are selected among the nondominated individuals. In a later version of this algorithm, called the Pareto reservation strategy, Tamaki et al. [1996] also used fitness sharing among the nondominated individuals to maintain diversity in the population.

[4]By "middling," Schaffer means an individual with acceptable performance, perhaps above average in all objectives, but not outstanding when measured by any particular function.

Criticism. Although Schaffer reported some success, and this approach is easy enough to be tempting to try and implement, Richardson et al. [1989] noted that shuffling and merging all the subpopulations corresponds to averaging the fitness components associated with each of the objectives. Since Schaffer uses proportional fitness assignment [Goldberg 1989], these are in turn proportional to the objectives themselves [Fonseca and Fleming 1994]. Hence, the resulting expected fitness corresponds to a linear combination of the objectives where the weights depend on the distribution of the population at each generation, as shown by Richardson et al. [1989]. The main consequence is that when we have a concave trade-off surface, certain points in concave regions will not be found by this optimization procedure (in which we just use a linear combination of the objectives). It has been proved that this is true regardless of the set of weights used [Richardson et al. 1989].

5.2 Lexicographic Ordering

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

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

$$\text{Minimize } f_1(\bar{x}) \tag{25}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{26}$$

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

$$\text{Minimize } f_2(\bar{x}) \tag{27}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{28}$$

$$f_1(\bar{x}) = f_1^* \tag{29}$$

and the solution of this problem is 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

$$\text{Minimize } f_i(\bar{x}) \tag{30}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{31}$$

$$f_l(\bar{x}) = f_l^*, \quad l = 1, 2, \ldots, i-1 \tag{32}$$

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

Applications. Fourman [1985] suggested a selection scheme based on lexicographic ordering. In a first version of his algorithm, objectives are assigned different priorities by the user, and each pair of individuals compared according to the objective with the highest priority. If this results in a tie, the objective with the second highest priority is used, and so on. In another version of this algorithm (that apparently works quite well), an objective is selected randomly at each run. [Fourman 1985] used this approach to design compact symbolic layouts.

Kursawe [1991] formulated a multiobjective version of evolution strategies (ESs) [Schwefel 1981] based on lexicographic ordering. Selection consists of as many steps as objective functions with the problem. At each step, one of these objectives is selected randomly according to a probability vector and used to delete a fraction of the current population. After selection, the survivors

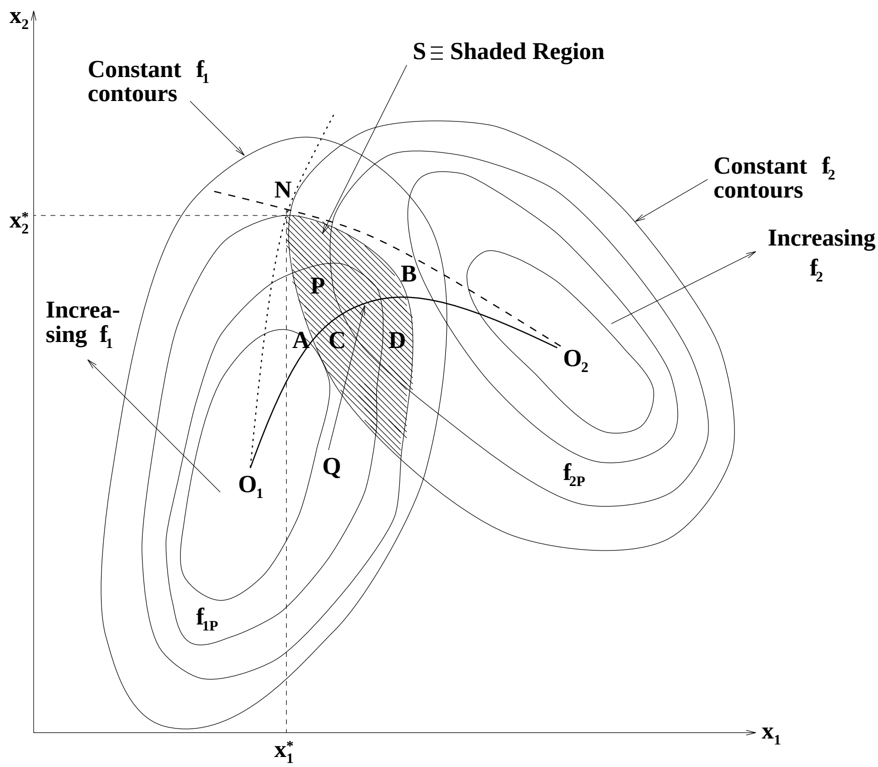

Figure 7. Example of cooperative and noncooperative game solutions.

became the parents of the next generation. The map of the trade-off surface is produced from the points evaluated during the run. Since the environment is allowed to change over time, diploid individuals are necessary to keep recessive information stored.

Criticism. Randomly selecting 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, since pairwise comparisons of tournament selection make scaling information negligible [Fonseca and Fleming 1994; 1995c]. Thus this approach may be able to see a concave trade-off surface as convex, although this really depends on the distribution of the population and on the problem itself. The main drawback is that, due to the randomness in the process, this approach tends to favor certain objectives over others when many are present. This has the undesirable consequence of making the population converge to a particular part of the Pareto front, rather than to delineate it completely [Coello 1996].

5.3 Game Theory

We can analyze this technique with reference to a simple optimization problem with two objectives and two design variables whose graphical representation is shown in Figure 7. 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 that will minimize his objective function f_1, and similarly the second player seeks a variable x_2 that will minimize his objective function f_2. If f_1 and f_2 are continuous, then the contours of constant values of f_1 and f_2 appear as shown in Figure 7. 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 (*noncooperative game*). In Figure 7, the point $N(x_1^*, x_2^*)$ represents such a intersection point. This point, known as the *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 [Nash 1950].

This point has the characteristic that

$$f_1(x_1^*, x_2^*) \leq f_1(x_1, x_2^*) \tag{33}$$

and

$$f_2(x_1^*, x_2^*) \leq f_2(x_1^*, x_2) \tag{34}$$

where x_1 may be to the left or right of x_1^* in Equation (33) and x_2 may lie above or below x_2^* in Equation (34).

Applications. Périaux et al. [1997] 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 that minimizes the backscattering of aerodynamic reflectors). The main idea is to use two noncooperative players represented by two independent subpopulations in a genetic algorithm, and then make them interact in the following way:

If f_1 and f_2 are the two objectives to be optimized, let P_1 and P_2 represent the two noncooperative 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 applied independently to each of the 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.

Criticism. This approach seems to be computationally very efficient, but in the state presented by Périaux et al. [1997] it is not possible to generate more than one nondominated solution (which would 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 in a problem). It is also possible to have several Nash equilibrium points with which the Pareto front can actually be found—although in that case a *cooperative game* may be preferred over a *noncooperative* one. For more information on cooperative games, see Rao [1987; 1984] and Coello [1996].

5.4 Using Gender to Identify Objectives

Allenson [1992] used a population-based approach modeled on VEGA, in which gender is used to distinguish between the two objective functions of a problem consisting of planning a route composed of a number of straight pipeline segments. With this approach, only male-female mating is allowed and gender is randomly assigned at birth. In the initial population, Allenson made sure that there was an equal number of males and females, but such a balance was not kept after applying the genetic operators. At each generation, the worst

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

individual (chosen from one of the two genders) is eliminated and replaced by another (randomly picked) individual of the same gender. Allenson used evolution strategies to implement sexual attractors that modify the way in which mating is performed. The idea is to model sexual attraction that in nature some individuals have over others, which determines a not-so-random mating.

Lis and Eiben [1996] also incorporated gender in their GA, but in a more general sense. In this case the number of genders (or sexes) is not limited to two, but to as many objectives as there are. Another distinction of this approach is that the crossover operator is modifed 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 is to select one parent from each sex to generate a child. This child will have the sex of the parent that contributed the largest number of genes (if there is a tie, then the sex is chosen randomly from the parents that contributed the same number of genes). If no crossover takes place, then one of the individuals in the old generation is copied exactly (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 restricted only slightly, to avoid changes in the sex of an individual. As generations progress, a list of nondominated individuals is updated, removing any individual who is no longer nondominated after the list is modified. At the end, this list will contain the Pareto optimal solutions.

Applications. Lis and Eiben [1996] successfully tested their approach with the two multiobjective optimization problems given by Srinivas and Deb [1993], but no further applications of this technique seem to be available.

Criticism. The use of genders is really another way of defining separate subpopulations for each objective. This approach differs from VEGA [Schaffer 1985] in that Lis and Eiben use panmictic crossover, which imposes certain mating restrictions and avoids the random crossing among different subpopulations that occur in Schaffer's approach. However, as we increase the number of objectives (or genders), many subpopulations and panmictic crossovers become more inefficient (computationally speaking) because we need to use more parents to generate a child. Population size will also have to increase as we increase the number of objectives, to keep a reasonably diverse spread of genders across the entire population.

5.5 Weighted Min-Max Approach

Stating the *min-max optimum* and applying it to multiobjective optimization problems is taken from game theory, which deals with solving conflicting situations. The min-max approach to a linear model was proposed by Jutler [1967] and Solich [1969], and further developed by Osyczka [1978; 1981; 1984]; Rao [1986]; and Tseng and Lu [1990]. The definitions shown below are taken from Osyczka [1978; 1981; 1984]. Notice that these definitions refer to nonlinear models because the procedure is simpler (there is no need to follow the steps mentioned below).

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

$$z'_i(\bar{x}) = \frac{|f_i(\bar{x}) - f_i^0|}{|f_i^0|} \tag{35}$$

or from

$$z''_i(\bar{x}) = \frac{|f_i(\bar{x}) - f_i^0|}{|f_i(\bar{x})|} \tag{36}$$

It should be clear that for (35) and (36) we have to assume that for every $i \in I$ and for every $\bar{x} \in F$, $f_i(\bar{x}) \neq 0$.

If all the objective functions are going to be minimized, then Equation (35) defines functions relative to increments, whereas if all of them are going to be maximized, it defines relative decrements. Equation (36) 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 R^k. The components of the vector $z(\bar{x})$ are evaluated from the formula

$$\forall_{i \in I}(z_i(\bar{x})) = \max\{z'_i(\bar{x}), z''_i(\bar{x})\} \quad (37)$$

We now define the min-max optimum as follows [Osyczka 1984]:

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

Step 1.

$$v_1(\bar{x}^*) = \min_{x \in F} \max_{i \in I} \{z_i(\bar{x})\} \quad (38)$$

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 $x_1 \subset F$ that satisfies Step 1, then

Step 2.

$$v_2(\bar{x}^*) = \min_{x \in x_1} \max_{i \in I,\, i \notin I_1} \{z_i(\bar{x})\} \quad (39)$$

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 $x_{r-1} \subset F$ that satisfies Step $r - 1$, then

Step r.

$$v_r(\bar{x}^*) = \min_{x \in x_{r-1}} \max_{i \in I,\, i \notin I_{r-1}} \{z_i(\bar{x})\} \quad (40)$$

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 $x_{k-1} \subset F$ that satisfies Step $k - 1$, then

Step k.

$$v_k(\bar{x}^*) = \min_{\bar{x} \in x_{k-1}} \{z_i(\bar{x})\} \max_{i \in I,\, i \notin I_{k-1}}$$

$$\text{for } i \in I \text{ and } i \notin I_{k-1} \quad (41)$$

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

This optimum can be described 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 that gives the smallest values of the relative increments of all the objective functions.

The point $\bar{x}^* \in F$ that satisfies the equations of all the previous steps may be called the best compromise solution, considering all the criteria simultaneously and on equal terms of importance.

Applications. Hajela and Lin [1992] included the weights of each objective in the chromosome and promoted their diversity in the population through fitness sharing. Their goal is 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 use a vector-evaluated approach based on VEGA to achieve their goal. They proposed the use of a utility function of the form:

$$\bar{U} = \sum_{i=1}^{l} W_i \frac{F_i}{F_i^*} \quad (42)$$

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 in Equation (9), with $\alpha = 1$, and σ_{share} chosen between 0.01 and 0.1. Under Hajela's representation, weight combinations are incorporated into the chromosomic string, and a single number does not represent the weight itself, but a combination of weights. For example, the number 4 (under floating point 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 is imposed to avoid members within a radius σ_{mat} crossing. The value of $\sigma_{mat} = 0.15$ was suggested by Hajela and Lin [1992].

Hajela and Lin [1992] used their approach to optimize a 10-bar plane truss in which weight and displacement were minimized. They also used it to minimize the weight of a wing-box structure in which its natural frequency was maximal.

Criticism. This approach can create a very high selection pressure if certain combinations of weights are produced at early stages of the search [Coello 1996]. To a certain extent, sharing avoids 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 experimentation.

5.6 Two Variations of the Weighted Min-Max Strategy

Coello [1996; 1997] 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), each trying to converge to a single point of the Pareto front. Mating restrictions are imposed to guarantee the feasibility of all the solutions, and constraints are handled through the evolution process by not allowing the generation of any infeasible solutions. This approach also requires knowledge of the ideal vector, or some estimate that lies in the feasible region.

In a second approach, Coello [1996] proposed the use of a local ideal vector computed at each generation. The selection process was modified to allow incorporation of min-max dominance. Thus an individual is considered the winner of a tournament if its maximum deviation from the ideal vector is the smallest in the set in ompetition. Mating restrictions are imposed to keep the feasible solutions at all generations only. Finally, sharing has to be used to overcome the high selection pressure introduced by the use of min-max tournament selection.

Applications. Coello and Christiansen [1999] applied these two approaches to the optimization of I-beams, to manufacturing problems [Coello and Christiansen 1998], and to the design of a robot arm [Coello et al. 1998].

Criticism. The use of weights is obviously a problem, since it is not always easy to find an appropriate set that can correctly delineate the part of the Pareto region that we wish to find. However, Coello [1996] showed through several engineering design examples that it is actually possible to find a good approximation of the Pareto front with a relatively small number 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 drawback, since it was shown that such a constraint-handling approach does not work in concave search surfaces. However, this was an attempt to incorporate constraint-handling into the search process in a different way from the traditional penalty approach, and 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 [Coello 1996]. However, its main drawback 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 [Valenzuela-Rendón and Uresti-Charre 1997; Bentley and Wakefield 1997; Greenwood et al. 1997].

5.7 The Contact Theorem to Detect Pareto Optimal Solutions

Osyczka and Kundu [1995] proposed an algorithm based on the contact theorem (one of the main theorems in multiobjective optimization [Lin 1976]) to determine relative distances of a solution vector with respect to the Pareto set (in fact, this approach has been called "the distance method" due to this characteristic [Kundu and Osyczka 1996]).

A solution is initially generated at random, and is considered Pareto optimal. Its fitness is d_1, which is an arbitrarily chosen value called the *starting* distance [Osyczka and Kundu 1995]. Then more solutions are generated and a "distance" value is computed for each of them, using the formula

$$z_l(\bar{x}) = \sqrt{\sum_{i=1}^{k}\left(\frac{f_{il}^{p} - \phi(\bar{x})}{f_{il}^{p}}\right)}, \quad \text{for } l = 1, 2, \ldots l_p \tag{43}$$

where k is the number of objectives and l_p is the number of Pareto optimal solutions found so far.

In the following step, the minimum value of the set $\{z_l(\bar{x})\}$ and its corresponding index l^* are found. This value is called $z_{l^*}(\bar{x})$. The procedure identifies the Pareto solution closest to the newly generated solution. We then have to verify if the newly generated solution is Pareto optimal; if it is, then its fitness is computed using

$$Fitness = d_{l^*} + z_{l^*}(\bar{x}) \tag{44}$$

where d_{l^*} is an arbitrary value at the beginning of the process (as indicated before). After the first generation, d_l is defined using the maximum value of the distances from all existing Pareto solutions.

If the newly generated solution is not a Pareto solution, then its fitness is computed using

$$Fitness = d_{l^*} - z_{l^*}(\bar{x}) \tag{45}$$

and $Fitness = 0$ in case a negative value results from this expression.

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

Applications. The method has been applied to control [Kundu et al. 1996], and to structural engineering [Kundu 1996] problems by its authors.

Criticism. Although this approach does not require an explicit sharing function, it is highly sensitive to the values of the penalty factor used to incorporate the constraints into each objective function. Its performance relies heavily on *starting* distance, which is a scaling factor used to compare the relative qualities of various solutions. If either of the two values is not chosen properly, too much selection pressure may be generated, or the GA may frequently 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.

5.8 A Nongenerational Genetic Algorithm

Valenzuela-Rendón and Uresti-Charre [1997] proposed a GA that uses nongenerational selection, and in which the fitness of an individual is calculated incrementally. The idea comes from learning classifier systems (LCS) [Goldberg 1989], where it was shown that a simple replacement of the worst individual in the population followed by an update on the fitness of the rest of the population works better than a traditional (generational) GA. In the context of multiobjective optimization, Valenzuela-Rendón and Uresti-Charre [1997] transformed the problem with N objectives into another one with only two objectives: the minimization of domination count (weighted average of the number of individuals that have dominated this individual so far) and the minimization of the moving niche count (weighted average of the number of individuals that lie close according to a sharing function). This biobjective optimization problem is then transformed into a single objective optimization problem by taking a linear combination of these two objectives. The basic algorithm follows:

(1) During the initialization of the population, each individual is compared to P randomly selected individuals (P can be seen as the tournament size used in tournament selection [Goldberg and Deb 1991]). After these comparisons take place, the domination count is set to the number of individuals who dominated other individuals in the group. Similarly, the moving niche count is updated using a measurement of closeness (normally some distance of its fitness value) among individuals.

(2) Loop an arbitrary number of times L and perform a comparison at each step of the loop, while the following is done:
   —Update fitness of each individual i using

$$fitness_i = c_d d_i + c_w w_i \tag{46}$$

   where d_i is the domination count, w_i is the moving niche count, and c_d and c_w are constants (arbitrarily chosen) that express the compromise between the two final objectives.
   —Update the domination count using

$$d(t+1) = d(t) - k_d d(t) + D(t) \tag{47}$$

   where k_d is set to zero in Valenzuela's experiments, and $D(t)$ is set to 1 if the individual was dominated in comparison t (t may be seen as the iteration number or the generation number in a generational GA) or to zero otherwise.
   —Update the moving niche count using

$$w(t+1) = w(t) - k_w w(t) + sharing(d) \tag{48}$$

   where k_w is set to $1/P$ in Valenzuela's experiments and $sharing(d)$ refers to the sharing expression based on the distance d allowed among individuals to make them part of a different niche. The sharing function used by Valenzuela-Rendón and Uresti-Charre is the same as the one used by Hajela and Lin [1992], explained before.
   —Perform proportional selection according to the maximum fitness in the population.
   —Apply crossover and mutation and produce a single new individual who will replace the worst individual in the current population (i.e., the individual with lowest fitness).

Applications. Valenzuela-Rendón and Uresti-Charre [1997] obtained better results than NPGA [Horn and Nafpliotis 1993] (see below) in three biobjective optimization problems, both in terms of the number of points in the Pareto front at the final iteration and in the total number of function evaluations. However, no further comparisons to other methods or to problems with more objectives was provided.

Criticism. This approach is really a more elaborate version of the weighted ranking techniques used by Bentley and Wakefield [1997], particularly the technique that they call weighted average ranking (WAR). Even when this approach seems to provide good distributions, it does not seem feasible to incorporate preferences of objectives defined by the decision maker, which may be a drawback in real-world applications. Also it is not clear how to define the additional parameters required by this algorithm, which seem to be subject to empirical fine tuning as are other normal parameters of the GA (e.g., crossover and mutation rates).

5.9 Randomly Generated Weights and Elitism

Ishibuchi and Murata [1996] proposed an algorithm that is similar to Hajela's weighted min-max technique, but the weights are generated in a slightly different way and the set of nondominated solutions produced at each generation is kept separately from the current population. The complete algorithm follows:

(1) Generate the initial population randomly.

(2) Compute the values of the p objectives for each individual in the population. Then determine the nondominated solutions and store them in a separate population, called NOND to distinguish it from the current population, which we denominate CURRENT.

(3) If N represents the number of nondominated solutions in NOND and M is the size of CURRENT, then we select $(M - N)$ pairs of parents using the following procedure:

—Let $r_1, r_2, \ldots, r_k$ be k random numbers in the interval [0,1]. The fitness function for each individual is

$$f(\bar{x}) = \sum_{i=1}^{p} w_i f_i(\bar{x}) \tag{49}$$

where p is the number of objectives and

$$w_i = \frac{r_i}{(r_1 + r_2 + \ldots + r_p)} \tag{50}$$

for $i = 1, 2, \ldots, p$. This ensures that all $w_i \geq 0$ (for $i = 1, 2, \ldots, p$) and that

$$\sum_{i}^{p} w_i = 1 \tag{51}$$

—Select a parent with probability:

$$P(\bar{x}) = \tag{52}$$

$$\frac{f(\bar{x}) - f_{min}(CURRENT)}{\sum_{x \in CURRENT} \{f(\bar{x}) - f_{min}(CURRENT)\}}$$

where f_{min} is the minimum fitness in the current population.

(4) Apply crossover to the selected $(M - N)$ pairs of parents. Apply the mutation to the newly generated solutions.

(5) Randomly select E solutions from NOND. Then add the selected E solutions to the $(M - N)$ solutions generated in the previous step to construct a population of size M.

(6) Since the goal of this work is to apply the GA to combinatorial optimization problems, the authors proposed the use of a local search procedure in which, for each individual, a set of solutions within a certain neighborhood are examined. If any one of them is better than the current individual, then the current individual replaces it. Local search is

applied to the M individuals in CURRENT.

(7) Finish if a prespecified stopping criterion is satisfied (e.g., the predefined maximum number of generations has been reached); otherwise, return to step 2.

Applications. Ishibuchi and Murata [1996] used this technique to solve biobjective optimization flowshop scheduling problems in which the makespan and maximum tardiness are to be minimized.

Criticism. This approach is very similar to the technique called sum of weighted rations (SWR) of Bentley and Wakefield [1997] and to the attribute value functions used by Greenwood et al. [1997]. Bentley and Wakefield [1997] claim that this approach maintains enough diversity to keep a wide spread of solutions through many generations. However, Coello [1996] has shown (using a similar approach), that such a spread may not be kept where there is an objective in the ideal vector that can be met easily by a large set of solutions. In such a case, it is necessary to use sharing techniques or a local search technique (as proposed by Ishibuchi and Murata [1996]) to keep diversity. Bentley and Wakefield [1997] also showed another variation of this algorithm, called the sum of weighted global ratios (SWGR), which visibly reduces the spread of solutions (i.e., the size of the Pareto set) by using the globally best and worst values, instead of the current ones. Nevertheless, the idea is interesting. The implementation of this algorithm seems easy and quite efficient with respect to most of the Pareto-based approaches described next.

6. PARETO-BASED APPROACHES

The idea of using Pareto-based fitness assignment was first proposed by Goldberg [1989] to solve the problems in Schaffer's approach. He suggested the use of nondomination ranking and selection to move a population toward the Pareto front in a multiobjective optimization problem. The basic idea is to find the set of strings in the population that are Pareto nondominated by the rest of the population. These strings are then assigned the highest rank and eliminated from further contention. Another set of Pareto nondominated strings are determined from the remaining population and are assigned the next highest rank. This process continues until the population is suitably ranked. Goldberg also suggested the use of some kind of niching technique to keep the GA from converging to a single point on the front. A niching mechanism such as sharing [Goldberg and Richardson 1987] allows the GA to maintain individuals all along the nondominated frontier.

Applications. Hilliard et al. [1989] used a Pareto optimality ranking method to handle the objectives of minimizing cost and delay in a scheduling problem. They tentatively concluded that the Pareto optimality ranking method outperforms the VEGA method. The Pareto method was found to be superior to a VEGA by Liepins et al. [1990] when applied to a variety of set covering problems. Ritzel et al. [1994] also used nondominated ranking and selection combined with deterministic crowding [Mahfoud 1992] 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 to. They found that selection according to Pareto nondomination is superior to both VEGA and to nondomination with deterministic crowding, at least for finding points near or on the front found by MICCP. Stanley and Mudge [1995] implemented

Goldberg's Pareto ranking technique to solve a microprocessor design problem in which the constraints are treated as additional objectives.

Criticism. The main problem with Pareto ranking is that there is no efficient algorithm to check for nondominance in a set of feasible solutions [Coello 1996]. Traditional algorithms have serious performance degradation as the size of the population and the number of objectives increases. Sharing also makes it necessary to estimate the value of σ_{share}, which is not easy. Nevertheless, this method's performance relies greatly on this value.

6.1 Multiple Objective Genetic Algorithm

Fonseca and Fleming [1993] proposed a scheme in which the rank of an individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual x_i of generation t dominated by $p_i^{(t)}$ individuals in the current generation. Its current position in the population can be given by Fonseca and Fleming [1993]:

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{53}$$

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

Fitness assignment is performed in the following way [Fonseca and Fleming 1993]:

(1) Sort population according to rank.

(2) Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $n \leq N$) as proposed by Goldberg [1989], according to some function, usually linear but not necessarily so.

(3) Average the fitnesses of individuals with the same rank, so that all of them are 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 [1991] point out, this type of blocked fitness assignment is likely to produce a large selection pressure that might produce premature convergence. To avoid this, Fonseca and Fleming use a niche-formation method to distribute the population over the Pareto-optimal region, but instead of performing sharing on the parameter values, they use sharing on the objective function values [Srinivas and Deb 1994].

With this approach it is possible to evolve only a certain region of the trade-off surface by combining Pareto dominance with partial preference information in the form of a goal vector. If the basic ranking scheme remains unaltered as we perform a Pareto comparison of the individuals, the objectives that have already been achieved will not be selected. If we specify fully unattainable goals, then objectives will never be excluded from comparison. Changing the goal values during the search alters the fitness landscape accordingly, and allows the decision maker to magnify a particular region of the trade-off surface [Fonseca and Fleming 1993].

Applications. MOGA has been used by several researchers in the past. For example, Tan and Li [1997] reported success using MOGA for the multiobjective optimization of ULTIC controllers that satisfy a number of time domain and frequency domain specifications. Chipperfield and Fleming [1995] reported success in using MOGA for the design of a multivariable control system for a gas turbine engine. Obayashi [1997] 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. [1997] extended MOGA to use it in genetic programming, introducing the so-called MOGP (multiple objective genetic programming). Genetic programming [Koza 1992] 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 nonlinear model structures as an alternative that the authors reported worked better (in terms of representation power) than the conventional linear representation of MOGA that they had attempted before [Fonseca and Fleming 1996a]. Aherne et al. [1997] used MOGA to optimize the selection of parameters for an object recognition scheme called the pairwise geometric histogram paradigm. Todd and Sen [1997] used a variant of MOGA to preplan containership layouts (a large-scale combinatorial problem). In Todd and Sen's approach, a population of nondominated individuals is kept and updated at each generation, removing individuals that become dominated and duplicated. 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 is restricted so that only individuals that are very similar could mate, and because of the permutations encoded, a repair algorithm had to be used afterwards. Finally, a heuristic mutation that defines rules to exchange bit positions must be used to avoid premature convergence of the population.

Criticism. In MOGA, sharing is done on the objective value space, which means that two different vectors with the same objective function values can not exist simultaneously in the population. This is precisely the kind of solution that the user normally wants, although the problem is theoretical and does not often occur in practice (but it should be said that in practice the method works quite well [Coello 1996]).

MOGA is a good approach, efficient and relatively easy to implement, but like all the other Pareto ranking techniques, its performance is highly dependent on an appropriate selection of σ_{share}. However, it is important to add that Fonseca and Fleming [1993] have developed a good methodology to compute such value.

6.2 Nondominated Sorting Genetic Algorithm

The nondominated sorting genetic algorithm (NSGA) was proposed by Srinivas and Deb [1993], and is based on several layers of classifications of individuals. Before selection, 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 population size to provide equal reproductive potential for these individuals). To maintain diversity in the population, classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is processed. The process continues until all individuals in the population are classified. A stochastic remainder proportionate selection is used in 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 the search for nondominated regions and results in quick convergence of the population toward such regions. Sharing, for its part, helps to distribute the population over this region. NSGA efficiency 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 [Srinivas and Deb 1994], and both maximization and minimization problems

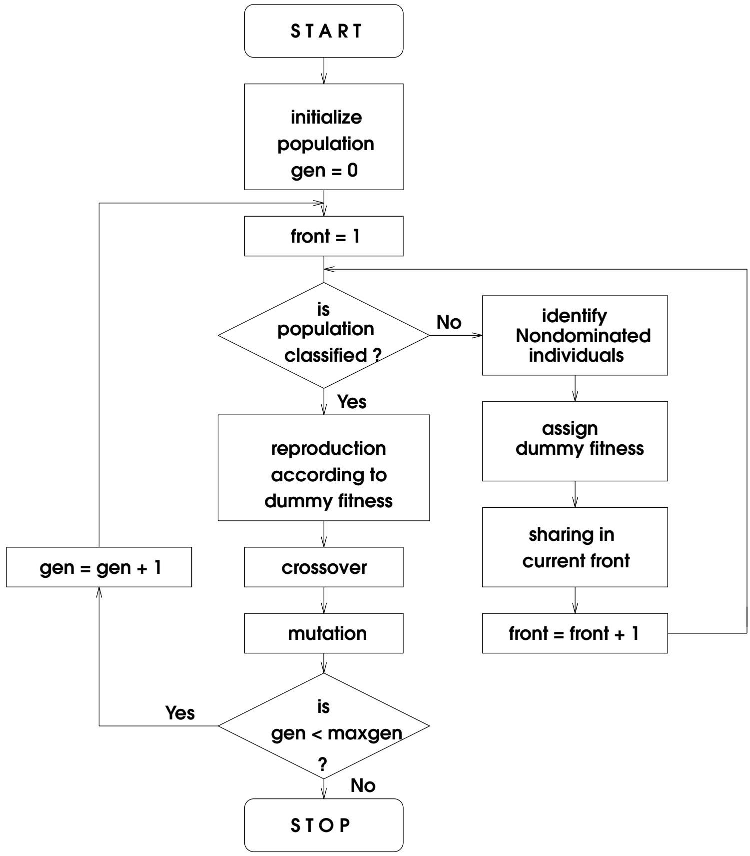

Figure 8. Flowchart of the nondominated sorting genetic algorithm (NSGA).

can be handled. Figure 8 shows the general flow chart of this approach.

Applications. Périaux et al. [1997] used NSGA to find an optimal distribution of active control elements, which minimized backscattering aerodynamic reflectors. Vedarajan et al. [1997] used NSGA to optimize portfolio investment, but interestingly they used binary tournament selection instead of stochastic remainder selection as suggested by Srinivas and Deb [1993]. 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 have been absorbed by the extra individuals in the population. Michielssen and Weile [1995] also used NSGA to design an electromagnetic system.

Criticism. In this case, sharing is done in the parameter values instead of the objective values, to ensure a better distribution of individuals, and to let multiple equivalent solutions exist. However, this technique is more inefficient (both computationally and in the quality of Pareto fronts produced) than MOGA, and more sensitive to the value of the sharing factor σ_{share}.[Coello 1996].

6.3 Niched Pareto GA

Horn and Nafpliotis [1993] proposed a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals, a number of individuals is used to help determine dominance (typically around 10). When both competitors are either dominated or nondominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing [Goldberg and Richardson 1987]. Population sizes considerably larger than usual with other approaches are used, so that the noise of the selection method is tolerated by the emerging niches in the population [Fonseca and Fleming 1994].

The pseudocode for Pareto domination tournaments, assuming that all of the objectives are maximized, is presented below [Horn and Nafpliotis 1993]. S is an array of the N individuals in the current population, *random_pop_index* is an array holding the N indices of S in a random order, and t_{dom} is the size of the comparison set.

```
function selection
/* Returns an individual from the current population S */
begin
  shuffle(random_pop_index);
  /* Rerandomize random index array */
  candidate_1 = random_pop_index[1];
  candidate_2 = random_pop_index[2];
  candidate_1_dominated = false;
  candidate_2_dominated = false;
  for comparison_set_index = 3 to t_dom + 3 do
  /* Select t_dom individuals randomly from S */
    begin
      comparison_individual = random_pop_index[comparison_set_index];
      if S [comparison_individual] dominates S [candidate_1]
        then candidate_1_dominated = true;
      if S [comparison_individual] dominates S [candidate_2]
        then candidate_2_dominated = true;
    end /* end for loop */
    if (candidate_1_dominated AND ¬ candidate_2_dominated)
    then return candidate_2;
    else if (¬candidate_1_dominated AND candidate_2_dominated)
      then return candidate_1;
    else
    do sharing;
end
```

The values of t_{dom} and σ_{share} should be provided by the user. Equivalence class sharing [Horn and Nafpliotis 1993] is done on the attribute values (i.e., on the vector of objective function values), and it should be implemented according to the following algorithm [Horn and Nafpliotis 1993]:

```
function selection
begin
    ⋮
  else if nichecount[candidate_1] > nichecount[candidate_2]
    then return candidate_2;
    else return candidate_1;
end
```

The value of *nichecount* is generated by the equivalence class sharing algorithm. The idea is that the best individual will be the one that has the least number of individuals in its niche, and thus the smallest niche count.

Horn and Nafpliotis [1993] 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 call *nested sharing*.

Applications. Belegundu et al. [1994] used the NPGA for the design of laminated ceramic composites. Poloni and Pediroda [1997] 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 [1997]. They introduced the dominance criteria in the selection mechanism (as in the NPGA), but then selected the individuals to be reproduced to generate the next population using a random walk operator. This obviously produces a locally dominating individual rather than a globally dominating one. Additionally, if more than one nondominated individual is found, then the first one encountered is selected (instead of sharing as in NPGA). At the end of every new generation, the set of Pareto optimal solutions is updated and stored. Quagliarella and Vicini [1997] used this approach for airfoil design.

Criticism. Since this approach does not apply Pareto selection to the entire population, but only to a segment at each run, the technique is very fast and produces good nondominated runs that can be kept for a large number of generations [Coello 1996]. However, to perform well, besides requiring a sharing factor, this approach also requires a good choice of the value t_{dom}, complicating its appropriate use in practice.

7. FUTURE RESEARCH

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

—Since the size of the Pareto set is normally quite large, and in the particular case of the genetic algorithm depends on the size of the population, it may be desirable to devise ways of reducing the number of elements in such a set in order to facilitate analysis for the decision maker. Kunha et al. [1997] proposed incorporating Roseman and Gero's algorithm [1985] 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 determining how to measure the quality of a solution. So far, visual inspection is the only technique used, unless there is some previous knowledge of the points that lie in the Pareto front (in which case there is obviously no need for a multiobjective optimization technique). Fonseca and Fleming [1996b] proposed the definition of certain (arbitrary) goals for the GA to attain; we can then perform multiple runs and apply standard nonparametric statistical procedures to evaluate the quality of the solutions (i.e., the nondominated 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, and more work needs to be done to develop a good and fair way of measuring the quality of the solutions produced by different multiobjective optimization approaches.

—In some cases it may be necessary to assign more importance to certain 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 so directly. Fonseca and Fleming [1993] proposed the use of a utility function combined with MOGA [Fonseca and Fleming 1994; 1995c] to produce a method for the progressive articulation of preferences. They proposed a feedback loop between the decision maker and the GA, so that some solutions (from the Pareto set) are given preference over others. Ideally, such a process could be done automatically by replacing the decision maker with an expert system [Fonseca and Fleming 1993]. 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 in some cases.

In an attempt to overcome the problems with Fonseca's approach, Bentley and Wakefield [1997] proposed the use of weights to estimate the importance of solutions that are already identified as Pareto optimal. Also, in a more elaborate approach, Greenwood et al. [1997] 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. [1997] used an approach imprecisely called the *specified multiattribute value theory* (ISMAUT) [White et al. 1984] which, combined with a GA, allows the definition of preferences by the GA itself, rather than the intervention of the decision maker. However, the decision maker still gets to decide what particular area of the trade-off surface he wants to explore, so that the GA constrains the search to that specific area. Additionally, Greenwood et al. [1997] defined a metric that allows us to obtain a single value (or utility function) that will guide the search to the particular Pareto region of interest to the decision maker.

Finally, Voget and Kolonko [1998] proposed 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 [Voget and Kolonko 1998].

These three proposals are quite interesting, but more work needs to be done in this area, preferably with real-world problems (Fonseca's approach is an appropriate choice for optimizing a gas turbine engine [Fonseca and Fleming 1993]). Greenwood et al. [1997] showed that their approach performs well in two hardware/software codesign problems, so that more general guidelines can be derived from the various approaches.

—The problem of measuring the quality of the solutions found with a multiobjective optimization technique is directly related to the need for a set of benchmark problems for testing existing and new approaches. This set should include both constrained and unconstrained problems,[6] examples with few objectives (two or three) suitable for graphical inspection, problems with few and several variables, and problems in which it is possible to achieve the ideal vector (for tuning up any technique to be tried). There is also a need to perform detailed studies of various GAs (assuming certain quality measures) using these benchmark problems and to derive more accurate information on the behavior of each of the algorithms. Coello [1996] conducted a study of this type using several engineering design problems; but it is necessary to design more general test problems.

—It is also important to define stopping criteria for a GA-based multiobjective optimization technique, since it is not easy to know when the population has reached a point at which no further improvements can be made. Currently, the main ways to stop this kind of GA are to either use a fixed number of generations or to monitor the population at certain intervals and visually interpret the results to determine when to halt the evolution process.

—Sharing in these techniques introduces another problem, since the value of σ_{share} becomes another parameter with which the user has to experiment until a reasonable setting is found. Even though important work has been done in this area (for example, Deb and Goldberg [1989] and Fonseca and Fleming [1993]), most of it is focused on single-objective or multimodal optimization.

[6]Most current papers that introduce new GA-based multiobjective optimization techniques use two or three simple unconstrained biobjective functions, particularly those originally used by Schaffer [1985].

—Some researchers also found quite interesting alternative applications of multiobjective optimization techniques. Perhaps the most remarkable is the attempt to use ranking techniques, or similar approaches, to handle constraints in a single objective optimization problem in order to avoid using penalty functions. Surry et al. [1995] proposed the COMOGA approach (constrained optimization by multiobjective genetic algorithms), which treats each constraint as a separate objective. Thus, it transforms a constrained single objective optimization problem into an unconstrained multiobjective optimization problem, which is solved using Fonseca's MOGA [Fonseca and Fleming 1993]. This approach is used by Surry et al. [1995] to optimize gas supply networks. Fonseca and Fleming [1995a] also proposed treating constraints as objectives, and applied their approach to the design of a gas turbine [Fonseca and Fleming 1995b]. Finally, Stanley and Mudge [1995] used Pareto ranking to handle constraints treated as objectives in a combinatorial optimization problem (microprocessor design).

—Finally, a very important topic, scarcely addressed by researchers in multiobjective optimization, is the development of a theory that can explain issues such as the effect of parameters (i.e., population size, crossover and mutation rates, and niche sizes) and the way in which the selection technique affects algorithm performance.

8. CONCLUSIONS

This paper attempts to provide a comprehensive review of the most popular evolutionary-based approaches to multiobjective optimization, giving some insights into their operations research roots, a brief description of their main algorithms, their advantages and disadvantages, and possible range of applicability. Additionally, some representative real-world applications of each approach (when found) are also included, together with a very rich bibliography (which should be enough to guide a newcomer in this important and growing area of research).

In the final section, the most promising areas for future research (in the author's opinion) are briefly described, and some of the work already done is briefly addressed.

ACKNOWLEDGMENTS

The author would like to thank the anonymous reviewers for their valuable comments, which helped to improve this paper.

REFERENCES

AHERNE, F. J., THACKER, N. A., AND ROCKETT, P. I. 1997. Optimal pairwise geometric histograms. In *Electronic Proceedings of the Eighth British Conference on Machine Vision* (BMVC97), A. F. Clark, Ed. University of Essex, Colchester, UK. http://peipa.essex.ac.uk/bmva/bmvc97/papers/071/bmvc.html.

ALLENSON, R. 1992. Genetic algorithms with gender for multi-function optimisation. EPCC-SS92-01. University of Edinburgh, Edinburgh, UK.

BAIER, H. 1977. Uber algorithmen zur ermittlung und charakterisierung pareto-optimaler losungen bei entwurfsaufgaben elastischer tragwerke. *Zamm 57*, 22, 318–320.

BELEGUNDU, A. D., MURTHY, D. V., SALAGAME, R. R., AND CONSTANTS, E. W. 1994. Multiobjective optimization of laminated ceramic composites using genetic algorithms. In *Proceedings of the Fifth AIAA/USAF/NASA Symposium on Multidisciplinary Analysis and Optimization* (Panama City, FL). 1015–1022.

BENTLEY, P. J. AND WAKEFIELD, J. P. 1997. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. In *Proceedings of Second On-Line World Conference on Soft Computing in Engineering Design and Manufacturing* (WSC2, June). http://users.aol.com/docbentley/dispaper.htm

BOYCHUK, L. M. AND OVCHINNIKOV, V. O. 1973. Principal methods of solution of multicriterial optimization problems (survey). *Soviet Automatic Control 6*, 1–4.

CHARNES, A. AND COOPER, W. W. 1961. *Management Models and Industrial Applications of Linear Programming*. John Wiley and Sons, Inc., New York, NY.

CHEN, Y. L. AND LIU, C. C. 1994. Multiobjective VAR planning using the goal-attainment method. In *Proceedings of IEEE Conference on Generation, Transmission and Distribution*. IEEE Computer Society Press, Los Alamitos, CA, 227–232.

CHIPPERFIELD, A. AND FLEMING, P. 1995. Gas turbine engine controller design using multiobjective genetic algorithms. In *Proceedings of First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications* (GALESIA'95, Halifax Hall, University of Sheffield, UK), A. M. S. Zalzala, Ed. 214–219.

COELLO, C. A. C. AND CHRISTIANSEN, A. D. 1999. MOSES: A multiobjective optimization tool for engineering design. *Eng. Optim. 31*, 3 (Feb.), 337–368.

COELLO, C. A. C. AND CHRISTIANSEN, A. D. 1998. Two new GA-based methods for multiobjective optimization. *Civil Eng. Syst. 15*, 3, 207–243.

COELLO, C. A. C., CHRISTIANSEN, A. D., AND AGUIRRE, A. H. 1998. Using a new GA-based multiobjective optimization technique for the design of robot arms. *Robotica 16*, 401–414.

COELLO, C. A. C., HERNANDEZ, F. S., AND FARRERA, F. A. 1997. Optimal design of reinforced concrete beams using genetic algorithms. *Expert Syst. Appl. Int. J. 12*, 1, 101–108.

COELLO, C. A. C. 1996. An empirical study of evolutionary techniques for multiobjective optimization in engineering design. Ph.D. Dissertation. Tulane University, New Orleans, LA.

COHON, J. L. AND MARKS, D. H. 1975. A review and evaluation of multiobjective programming techniques. *Water Resources Res. 11*, 2 (Apr.), 208–220.

CVETKOVIC, D., PARME, I., AND WEBB, E. 1998. Multi-objective optimisation and preliminary airframe design. In *The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realization*, I. Parme, Ed. Springer-Verlag, New York, NY, 255–267.

DEB, K. AND GOLDBERG, D. E. 1989. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms* (George Mason University, June 4–7), J. D. Schaffer, Ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, 42–50.

DEJONG, A. K. 1975. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation. University of Michigan, Ann Arbor, MI.

DUCKSTEIN, L. 1984. Multiobjective optimization in structural design: The model choice problem. In *New Directions in Optimum Structural Design*, E. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz, Eds. John Wiley and Sons, Inc., New York, NY, 459–481.

ESHELMAN, L. J. 1997. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *Foundations of Genetic Algorithms*, G. J. Rawlins, R. K. Belew, and M. D. Vose, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, 265–283.

ESHELMAN, L. J. AND SCHAFFER, J. D. 1991. Preventing premature convergence in genetic algorithms by preventing incest. In *Proceedings of the Fourth International Conference on Genetic Algorithms* (Univ. of California, San Diego, July 13-16,), R. K. Belew and L. B. Booker, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, 115–122.

EVANS, G. W. 1984. An overview of techniques for solving multiobjective mathematical programs. *Manage. Sci. 30*, 11, 1268–1282.

FISHBURN, P. C. 1978. A survey of multiattribute/multicriterion evaluation theories. In *Multiple Criteria Problem Solving*, S. Zionts, Ed. Springer-Verlag, New York, NY, 181–224.

FONSECA, C. M. AND FLEMING, P. J. 1996a. Nonlinear system identification with multiobjective genetic algorithms. In *Proceedings of the 13th World Congress on IFAC* (San Francisco, CA). 187–192.

FONSECA, C. M. AND FLEMING, P. J. 1996b. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Parallel Problem Solving from Nature: PPSN IV*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Springer-Verlag, New York, NY, 584–593.

FONSECA, C. M. AND FLEMING, P. J. 1995a. Multiobjective optimization and multiple constraint handling with evolutionary algorithms I: A unified formulation. Tech. Rep. 564. University of Sheffield, Sheffield, UK.

FONSECA, C. M. AND FLEMING, P. J. 1995b. Multiobjective optimization and multiple constraint handling with evolutionary algorithms II: Application example. Tech. Rep. 565. University of Sheffield, Sheffield, UK.

FONSECA, C AND FLEMING, P. J. 1995c. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Comput. 3*, 1, 1–16.

FONSECA, C. M. AND FLEMING, P. J. 1994. An overview of evolutionary algorithms in multiobjective optimization. Tech. Rep.. University of Sheffield, Sheffield, UK.

FONSECA, C. M. AND FLEMING, P. J. 1993. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms* (University of Illinois at Urbana-Champaign), S. Forrest, Ed. Morgan Kaufmann, San Mateo, CA, 416–423.

FOURMAN, M. P. 1985. Compaction of symbolic layout using genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications:*

Proceedings of the First International Conference on Genetic Algorithms. Lawrence Erlbaum Associates Inc., Hillsdale, NJ, 141–153.

GEN, M. AND CHENG, R. 1997. *Genetic Algorithms and Engineering Design*. John Wiley and Sons, Inc., New York, NY.

GEN, M., IDA, K., LI, Y., AND KUBOTA, E. 1995. Solving bicriteria solid transportation problem with fuzzy numbers by a genetic algorithm. *Comput. Industrial Eng. 29*, 1-4 (Sept.), 537–541.

GOLDBERG, D. E. AND DEB, K. 1997. A comparison of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, G. J. Rawlins, R. K. Belew, and M. D. Vose, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, 69–93.

GOLDBERG, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Inc., Redwood City, CA.

GOLDBERG, D. E. AND RICHARDSON, J. 1987. Genetic algorithm with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, J. J. Grefenstette, Ed. Lawrence Erlbaum Associates Inc., Hillsdale, NJ, 41–49.

GREENWOOD, G. W., HU, X. S., AND D'AMBROSIO, J. G. 1997. Fitness functions for multiple objective optimization problems: Combining preferences with Pareto rankings. In *Foundations of Genetic Algorithms*, G. J. Rawlins, R. K. Belew, and M. D. Vose, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, 437–455.

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

HAIMES, Y. Y., HALL, W., AND FREEDMAN, H. 1975. *Multi-Objective Optimization In Water Resources Systems: Surrogate Trade-off Method*. Elsevier, Amsterdam, The Nethlands.

HAJELA, P., YOO, J., AND LEE, J. 1997. GA based simulation of immune networks: Applications in structural optimization. *Eng. Optim. 29*, 131–149.

HAJELA, P. AND LEE, J. 1996. Constrained genetic search via scheme adaptation: An immune network solution. *Structural Optim. 12*, 1, 11–15.

HAJELA, P. AND LIN, C. Y. 1992. Genetic search strategies in multicriterion optimal design. *Structural Optim. 4*, 99–107.

HILLIARD, M. R., LIEPINS, G. E., PALMER, M., AND RANGARAJEN, G. 1989. The computer as a partner in algorithmic design: Automated discovery of parameters for a multiobjective scheduling heuristic. In *Impacts of Recent Computer Advances on Operations Research*, R. Sharda, B. L. Golden, E. Wasil, O. Balci, and W. Stewart, Eds. North-Holland Publishing Co., Amsterdam, The Netherlands.

HOLLAND, J. H. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. 2nd ed. MIT Press, Cambridge, MA.

HOLLAND, J. H. 1975. *Adaption in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor, MI.

HORN, J. AND NAFPLIOTIS, N. 1993. Multiobjective optimization using the niched Pareto genetic algorithm. IlliGAl Rep. 93005. University of Illinois at Urbana-Champaign, Champaign, IL.

HWANG, C. L., PAIDY, S. R., AND YOON, K. 1980. Mathematical programming with multiple objectives: A tutorial. *Comput. Oper. Res. 7*, 5–31.

IGNIZIO, J. P. 1981. The determination of a subset of efficient solutions via goal programming. *Comput. Oper. Res. 3*, 9–16.

IGNIZIO, J. P. 1976. *Goal Programming and Extensions*. D. C. Heath and Company, Lexington, MA.

IJIRI, Y. 1965. *Management Goals and Accounting for Control*. North-Holland Publishing Co., Amsterdam, The Netherlands.

ISHIBUCHI, H. 1996. Multi-objective genetic local search algorithm. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation* (Nagoya, Japan), T. Fukuda and T. Furuhashi, Eds. 119–124.

JAKOB, W., GORGES-SCHLEUTER, M., AND BLUME, C. 1992. Application of genetic algorithms to task planning and learning. In *Proceedings of the Second Workshop on Parallel Problem Solving From Nature*, R. Männer and B. Manderick, Eds. North-Holland Publishing Co., Amsterdam, The Netherlands, 291–300.

JONES, G., BROWN, R. D., CLARK, D. E., WILLETT, P., AND GLEN, R. C. 1993. Searching databases of two-dimensional and three-dimensional chemical structures using genetic algorithms. In *Proceedings of the Fifth International Conference on Genetic Algorithms* (University of Illinois at Urbana-Champaign), S. Forrest, Ed. Morgan Kaufmann, San Mateo, CA, 597–602.

JUTLER, H. 1967. Liniejnaja modiel z nieskolkimi celevymi funkcjami (linear model with several objective functions). *Ekonomika i matematiceckije Metody 3*, 397–406.

KITA, H., YABUMOTO, Y., MORI, N., AND NISHIKAWA, Y. 1996. Multi-objective optimization by means of the thermodynamical genetic algorithm. In *Parallel Problem Solving from Nature: PPSN IV*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Springer-Verlag, New York, NY, 504–512.

KOZA, J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA.

KUHN, H. W. AND TUCKER, A. W. 1951. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, J. Neyman, Ed. University of California at Berkeley, Berkeley, CA, 481–492.

KUNDU, S. 1996. A multicriteria genetic algithm to solve optimization problems in structural engineering design. In *Information Processing in Civil and Structural Engineering Design*, B. Kumar, Ed. 225–233.

KUNDU, S., KAWATA, S., AND WATANABE, A. 1996. A multicriteria approach to control system design with genetic algorithm. In *Proceedings of the International Federation 13th World Congress Conference on Automatic Control* (IFAC'96, Klidington, UK). Elsevier Science Publishers Ltd., Essex, UK, 315–320.

KUNDU, S. AND OSYCZKA, A. 1996. The effect of genetic algorithm selection mechanisms on multicriteria optimization using the distance method. In *Proceedings of the Fifth International Conference on Intelligent Systems* (Reno, NV). ISCA, 164–168.

KUNHA, A. G., OLIVEIRA, P., AND COVAS, J. A. 1997. Use of genetic algorithms in multicriteria optimization to solve industrial problems. In *Proceedings of the Seventh International Conference on Genetic Algorithms* (Michigan State University), T. Bäck, Ed. Morgan Kaufmann Publishers Inc., San Francisco, CA.

KURSAWE, F. 1991. A variant of evolution strategies for vector optimization. In *Proceedings of the First Workshop on Parallel Problem Solving from Nature* (PPSN 1, Dortmund, Germany, Oct. 1–3), H.-P. Schwefel and R. Männer, Eds. Springer Lecture Notes in Computer Science. Springer-Verlag, New York, NY, 193–197.

LIEBERMAN, E. R. 1991. Soviet multi-objective mathematical programming methods: an overview. *Manage. Sci. 37*, 9 (Sept.), 1147–1165.

LIEPINS, G. E., HILLIARD, M. R., RICHARDSON, J., AND PALMER, M. 1990. Genetic algorithms application to set covering and travelling salesman problems. In *Operations Research and Artificial Intelligence: The Integration of Problem-Solving Strategies*, D. E. Brown and C. C. White, Eds. Kluwer Academic Publishers, Hingham, MA, 29–57.

LIN, J. G. 1976. Maximal vectors and multi-objective optimization. *J. Optim. Theory Appl. 18*, 1 (Jan.), 41–64.

LIS, J. AND EIBEN, A. E. 1996. A multi-sexual genetic algorithm for multiobjective optimization. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation* (Nagoya, Japan), T. Fukuda and T. Furuhashi, Eds. 59–64.

LIU, X., BEGG, D. W., AND FISHWICK, R. J. 1998. Genetic approach to optimal topology/controller design of adaptive structures. *Int. J. Numer. Methods Eng. 41*, 815–830.

LOUGHLIN, D. H. AND RANJITHAN, S. 1997. The neighborhood constraint method: A genetic algorithm-based multiobjective optimization technique. In *Proceedings of the Seventh International Conference on Genetic Algorithms* (Michigan State University), T. Bäck, Ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, 666–673.

MAHFOUD, S. M. 1992. Crowding and preselection revisited. In *Proceedings of the Second Workshop on Parallel Problem Solving From Nature*, R. Männer and B. Manderick, Eds. North-Holland Publishing Co., Amsterdam, The Netherlands.

MICHALEWICZ, Z. 1994. *Genetic Algorithms + Data Structures = Evolution Programs*. 2nd extended ed. Springer-Verlag, New York, NY.

MICHIELSSEN, E. AND WEILE, D. S. 1995. Electromagnetic system design using genetic algorithms. In *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. John Wiley and Sons Ltd., Chichester, UK, 267–288.

MITCHELL, M. 1996. *An Introduction to Genetic Algorithms*. Complex adaptive series. MIT Press, Cambridge, MA.

NASH, J. 1950. The bargaining problem. *Econometrica 18*, 155–162.

OSYCZKA, A. 1978. An approach to multicriterion optimization problems for engineering design. *Comput. Methods Appl. Mech. Eng. 15*, 309–333.

OSYCZKA, A. 1981. An approach to multicriterion optimization for structural design. In *Proceedings of the International Symposium on Optimal Structural Design* (University of Arizona).

OSYCZKA, A. 1984. *Multicriterion Optimization in Engineering with FORTRAN Programs*. Ellis Horwood, Upper Saddle River, NJ.

OSYCZKA, A. 1985. Multicriteria optimization for engineering design. In *Design Optimization*, J. S. Gero, Ed. Academic Press, Inc., New York, NY, 193–227.

OSYCZKA, A. AND KUNDU, S. 1995. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural Optim. 10*, 94–99.

PARETO, V. 1896. *Cours D'Economie Politique*. Rouge, Lausanne, Switzerland.

PÉRIAUX, J., SEFRIOUI, M., AND MANTEL, B. 1995. Ga multiple objective optimization strategies for electromagnetic backscattering. In *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. John Wiley and Sons Ltd., Chichester, UK, 225–243.

POLONI, C. AND PEDIRODA, V. 1995. GA coupled with computationally expensive simulations: Tools to improve efficiency. In *Genetic Algorithms and Evolution Strategies in Engineering* *and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. John Wiley and Sons Ltd., Chichester, UK, 267–288.

POWELL, D. AND SKOLNICK, M. M. 1993. Using genetic algorithms in engineering design optimization with non-linear constraints. In *Proceedings of the Fifth International Conference on Genetic Algorithms* (University of Illinois at Urbana-Champaign), S. Forrest, Ed. Morgan Kaufmann, San Mateo, CA, 424–431.

QUAGLIARELLA, D. AND VICINI, A. 1995. Coupling genetic algorithms and gradient based optimization techniques. In *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. John Wiley and Sons Ltd., Chichester, UK, 289–309.

QUAGLIARELLA, D., PÉRIAUX, J., POLONI, C., AND WINTER, G., EDS. 1995. *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications*. John Wiley and Sons Ltd., Chichester, UK.

RANJITHAN, S., EHEART, J. W., AND LIEBMAN, J. C. 1992. Incorporating fixed-cost component of pumping into stochastic groundwater management: A genetic algorithm-based optimization approach. *Eos Trans. AGU 73*, 14.

RAO, S. 1986. Game theory approach for multiobjective structural optimization. *Comput. Structures 25*, 1, 119–127.

RAO, S. S. 1984. Multiobjective optimization in structural design with uncertain parameters and stochastic processes. *AIAA J. 22*, 11, 1670–1678.

RAO, S. S. 1987. Game theory approach for multiobjective structural optimization. *Comput. Structures 25*, 1, 119–127.

RICHARDSON, J. T., PALMER, M. R., LIEPINS, G. E., AND HILLIARD, M. 1989. Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the Third International Conference on Genetic Algorithms* (George Mason University, June 4–7), J. D. Schaffer, Ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, 191–197.

RITZEL, B. J., EHEART, J. W., AND RANJITHAN, S. 1994. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Res. 30*, 5, 1589–1603.

RODRÍGUEZ-VÁZQUEZ, K., FONSECA, C. M., AND FLEMING, P. J. 1997. Multiobjective genetic programming: A nonlinear system identification application. In *Proceedings of the 1997 Conference on Genetic Programming: Late Breaking Papers*, J. R. Koza, Ed. 207–212.

ROSEMAN, M. A. AND GERO, J. S. 1985. Reducing the Pareto optimal set in multicriteria optimization. *Eng. Optim. 8*, 189–206.

ROSENBERG, R. S. 1967. Simulation of genetic populations with biochemical properties. Ph.D. Dissertation. University of Michigan Press, Ann Arbor, MI.

SANDGREN, E. 1994. Multicriteria design optimization by goal programming. In *Advances in Design Optimization*, H. Adeli, Ed. Chapman & Hall, London, UK, 225–265.

SCHAFFER, J. D. 1985. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates Inc., Hillsdale, NJ, 93–100.

SCHWEFEL, H. P. 1981. *Numerical Optimization of Computer Models*. John Wiley and Sons, Inc., New York, NY.

SMITH, R. E., FORREST, S., AND PERELSON, A. S. 1993. Population diversity in an immune system model: Implications for genetic search. In *Foundations of Genetic Algorithms 2*, L. D. Whitley, Ed. Morgan Kaufmann, San Mateo, CA, 153–165.

SOLICH, R. 1969. Zadanie programowania liniowego z wieloma funkcjami celu (linear programming problem with several objective functions). *Przeglad Statystyczny 16*, 24–30.

SRINIVAS, N. AND DEB, K. 1993. Multiobjective optimization using nondominated sorting in genetic algorithms. Tech. Rep.. Indian Institute of Technology, Delhi, India.

SRINIVAS, N. AND DEB, K. 1994. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Comput. 2*, 3, 221–248.

STADLER, W. 1984. A survey of multicriteria optimization or the vector maximum problem, Part I: 1776-1960. *J. Optim. Theory Appl. 29*, 1 (Sept.), 1–52.

STANLEY, T. J. AND MUDGE, T. 1995. A parallel genetic algorithm for multiobjective microprocessor design. In *Proceedings of the Sixth International Conference on Genetic Algorithms* (University of Pittsburgh), L. J. Eshelman, Ed. Morgan Kaufmann, San Mateo, CA, 597–604.

SURRY, P. D., RADCLIFFE, N. J., AND BOYD, I. D. 1995. A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method. In *Evolutionary Computing, AISB Workshop. Selected Papers*, T. C. Fogarty, Ed. Springer-Verlag, New York, NY, 166–180.

SYSWERDA, G. AND PALMUCCI, J. 1991. The application of genetic algorithms to resource scheduling. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, R. K. Belew and L. B. Booker, Eds. Morgan Kaufmann, San Mateo, CA, 502–508.

SZIDAROVSZKY, F. AND DUCKSTEIN, L. 1982. Basic Properties of MODM problems. *Classnotes:*

Department of Systems and Industrial Engineering, University of Arizona 82, 1.

TAMAKI, H., KITA, H., AND KOBAYASHI, S. 1996. Multi-objective optimization by genetic algorithms: A review. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation* (Nagoya, Japan), T. Fukuda and T. Furuhashi, Eds. 517–522.

TAMAKI, H., MORI, M., ARAKI, M., AND OGAI, H. 1995. Multicriteria optimization by genetic algorithms: A case of scheduling in hot rolling process. In *Proceedings of the Third Conference on APORS*. 374–381.

TAN, K. C. AND LI, Y. 1997. Multi-objective genetic algorithm based time and frequency domain design unification of linear control systems. Tech. Rep. CSC-97007. University of Glasgow, Glasgow, Scotland, UK.

TODD, D. S. AND SEN, P. 1997. A multiple criteria genetic algorithm for containership loading. In *Proceedings of the Seventh International Conference on Genetic Algorithms* (Michigan State University), T. Bäck, Ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, 674–681.

TSENG, C. H. AND LU, T. W. 1990. Minimax multiobjective optimization in structural design. *Int. J. Numer. Method. Eng. 30*, 1213–1228.

VALENZUELA-RENDÓN, M. AND URESTI-CHARRE, E. 1997. A non-generational genetic algorithm for multiobjective optimization. In *Proceedings of the Seventh International Conference on Genetic Algorithms* (Michigan State University), T. Bäck, Ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, 658–665.

VEDARAJAN, G., CHAN, L. C., AND GOLDBERG, D. E. 1997. Investment portfolio optimization using genetic algorithms. In *Proceedings of the 1997 Conference on Genetic Programming: Late Breaking Papers*, J. R. Koza, Ed. 255–263.

VOGET, S. AND KOLONKO, M. 1998. Multidimensional optimization with a fuzzy genetic algorithm. *J. Heuristics 4*, 3 (Sept.), 221–244.

WHITE, C., SAGE, A., AND DOZONO, S. 1984. A model of multiattribute decision-making and tradeoff weight determination under uncertainty. *IEEE Trans. Syst. Man Cybern. SMC-14*, 223–229.

WIENKE, P. B., LUCASIUS, C., AND KATEMAN, G. 1992. Multicriteria target optimization of analytical procedures using a genetic algorithm. *Anal. Chimica Acta 265*, 2, 211–225.

WILSON, P. B. AND MACLEOD, M. D. 1993. Low implementation cost IIR digital filter design using genetic algorithms. In *Proceedings of the IEE/IEEE Workshop on Natural Algorithms in Signal Processing*.

YANG, X. AND GEN, M. 1994. Evolution program for bicriteria transportation problem. In *Proceedings of the 16th International Conference on Computers and Industrial Engineering* (Ashikaga, Japan), M. Gen and T. Kobayashi, Eds. 451–454.

Received: April 1998; revised: April 1999; accepted: August 1999