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 aconsequence, many new evolutionary-based approaches and variations of existing techniques have been recently 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, their degree of applicability and some of their known applications. Finally, the future trends in this area and some possible paths of 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 with no doubt a very important research topic both for scientists and engineers, not only because of the multiobjective nature of most real-world problems, but also because there are still many open questions in this area. In 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 a naïve combination of objectives into a single one to the use of game theory to coordinate the relative importance of each objective. However, the fuzziness of this area lies on the fact that there is no accepted definition of "optimum" as in single-objective optimization, and therefore is difficult to even compare results of one method to another, because normally the decision about what the "best" answer corresponds to the so-called (human) *decision maker*.

In the past, there have been other surveys of multiobjective optimization techniques in the mathematical programming literature, from which the papers by

`ccoello@xalapa.lania.mx`
Rébsamen No. 80, Apartado Postal 696
Xalapa, Veracruz 91090
México

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

This paper emphasizes the importance of looking at the previous work done 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 mentioned, 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 is given after the description of each technique, mentioning their advantages, their possible drawbacks and limitations, and (in some cases) possible ways to exploit their characteristics or even improve their performance.

2. STATEMENT OF THE PROBLEM

Multiobjective optimization (also called multicriteria optimization, multiperformance or vector optimization) can be defined as the problem of finding [Osyczka 1985]:

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

Formally, we can state it as follows:

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

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

the p equality constraints

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

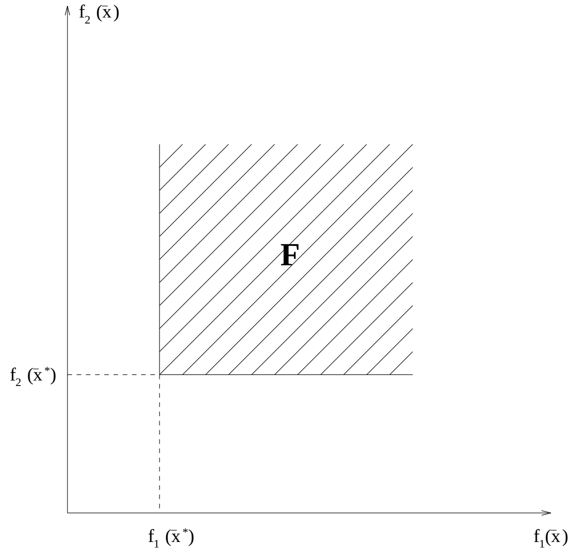

Fig. 1. Ideal solution in which all our functions have their minimum at a common point.

and optimizes the vector function

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

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

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

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

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

If this was the case, then $\bar{x}^*$ would be a desirable solution, but we normally never have a situation like this, in which all the $f_i(\bar{x})$ have a minimum in $\mathcal{F}$ at a common point x^*. An example of this ideal situation is shown in Figure 1. However, since

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

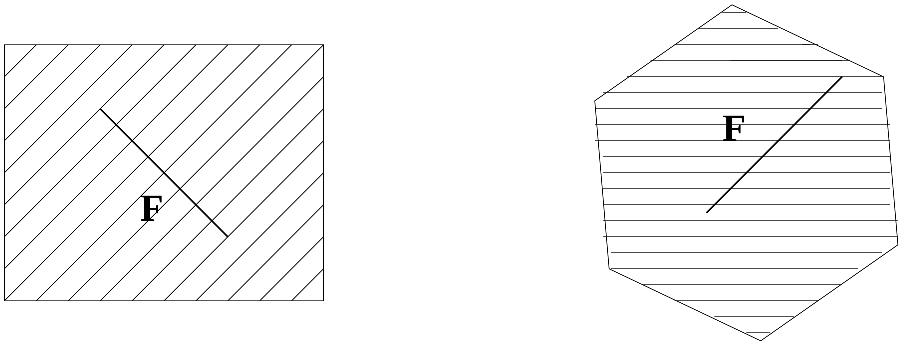

Fig. 2. Two examples of convex sets.

this situation rarely happens in real-world problems, then we have to establish a certain criteria to determine what would be considered as an "optimal" solution.

2.1 Ideal Vector

Let us 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 which optimizes (either minimizes or maximizes) the ith objective function $f_i(x)$. In other words, the vector $\bar{x}^{0(i)} \in \mathcal{X}$ is such that

$$f_i(\bar{x}^{0(i)}) = \underset{x \,\in\, \mathcal{F}}{opt} \quad f_i(\bar{x}) \tag{6}$$

In general, there will be a unified criterion with respect to "opt". Most authors prefer to treat it as a minimum. In that case, $f_i(\bar{x}^{0(i)})$ or simply f_i^0 (more convenient notation) will denote 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, and the point in $\mathcal{R}^n$ which determined this vector is the ideal (utopical) solution, and is consequently called the *ideal vector*.

2.2 Convexity and Non-Convexity

The set $\mathcal{F}$ is convex if for every $\bar{a}^1$, $\bar{a}^2 \in \mathcal{F}$ and every $\theta \in [0, 1]$

$$\bar{f}(\theta\bar{a}^1 + (1-\theta)\bar{a}^2) \leq \theta\bar{f}(\bar{a}^1) + (1-\theta)\bar{f}(\bar{a}^2) \tag{7}$$

In other words, the set $\mathcal{F}$ is convex if for any points $\bar{a}^1$ and $\bar{a}^2$ in the set, the line segment joining these points is also in the set. So, for example, the sets shown in Figure 2 are convex, but the sets shown in Figure 3 are not.

2.3 Pareto Optimum

The concept of *Pareto optimum* was formulated by Vilfredo Pareto in the XIX century [Pareto 1896], and constitutes by itself the origin of research in multiobjective

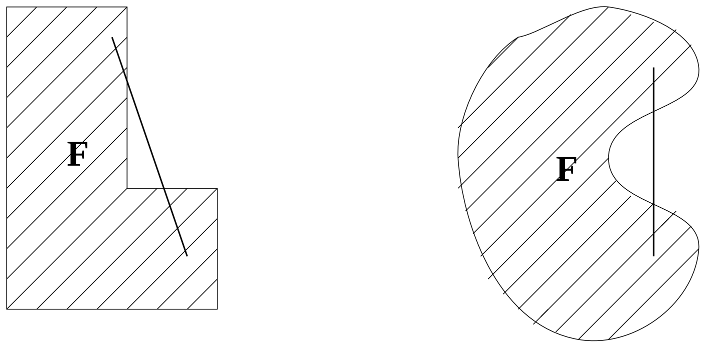

Fig. 3. Two examples of non-convex sets.

optimization. We say that a point $\bar{x}^* \in \mathcal{F}$ is *Pareto optimal* if for every $\bar{x} \in \mathcal{F}$ either,

$$\bigwedge_{i \in I} (f_i(\bar{x}) = f_i(\bar{x}^*)) \tag{8}$$

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

$$f_i(\bar{x}) > f_i(\bar{x}^*) \tag{9}$$

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

2.4 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 4, 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 $\mathcal{F}^k$ and their corresponding $f(\mathcal{F}^k)$. When we have a sufficient amount of these, we may proceed to take the final decision.

A point $\bar{x}^* \in \mathcal{F}$ is a *weakly non-dominated solution* if there is no $\bar{x} \in \mathcal{F}$ such that $f_i(\bar{x}) < f_i(\bar{x}^*)$, for $i = 1, \ldots, n$. A point $\bar{x}^* \in \mathcal{F}$ is a *strongly non-dominated solution* if there is no $\bar{x} \in \mathcal{F}$ 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 non-dominated*, it is also *weakly non-dominated*, but the converse is not necessarily true. Non-dominated solutions for the biobjective case can readily be represented graphically by passing into the objective function space $\{f_1(\bar{x}), f_2(\bar{x})\}$. To the locus of strongly non-dominated points corresponds the so-

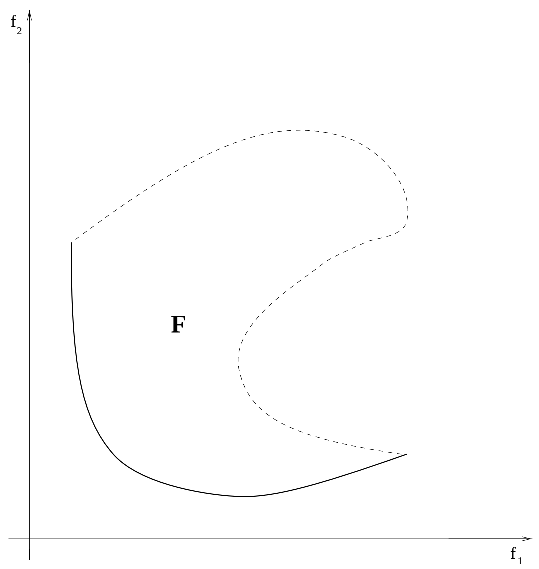

Fig. 4. An example of a problem with two objective functions. The Pareto front is marked with a bold line.

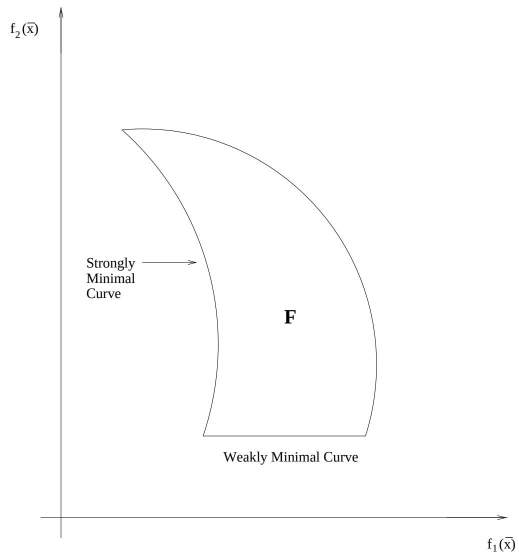

Fig. 5. Weakly and strongly non-dominated curves on the biobjective case.

called *minimal curve*, and to the locus of weakly non-dominated points, the *weakly minimal curve* [Baier 1977]. These two curves are sketched in Figure 5 for a simple bi-objective 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 the entire Pareto front of a problem, and not only a single non-dominated solution. The question is then how to keep the GA from converging to a single solution.

Early researchers in genetic algorithms identified this convergence phenomenon of the GA, called *genetic drift* [DeJong 1975], and found that it happens in Nature as well. They correctly stated that the key to solve this problem is to find a way of preserving diversity in the population, and several proposals, modelled after natural systems were made. Holland [1975] suggested the use of a "crowding" operator, which was intended to identify situations in which more and more individuals dominate an environmental niche, since in those cases the competition for limited resources increases rapidly, which will result in lower life expectancies and birth rate. DeJong [1975] experimented with such a *crowding* operator, which was implemented by having a newly formed offspring to replace the existing individual more similar to itself. The similarity between two individuals was measured in the genotype, by counting the number of bits along each chromosome that were equal in the two individuals being compared. DeJong used two parameters in his model: generation gap (G) and crowding factor (CF) [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 [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 replace one another [DeJong 1975].

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

$$\phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{sh}}\right)^{\alpha}, & d_{ij} < \sigma_{sh} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

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 which controls the extent of sharing allowed. The fitness of a design i is then modified as:

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{M} \phi(d_{ij})} \tag{11}$$

where M is the number of designs located in 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 2 individuals is measured in decoded parameter space, and can be calculated with a simple Euclidean distance in an p-dimensional space, where p refers to the number of variables encoded in the GA; the value of d_{ij} can then be calculated as:

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{k,i} - x_{k,j})^2} \tag{12}$$

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 to use the expression:

$$\sigma_{share} = \frac{r}{\sqrt[p]{q}} = \frac{\sqrt{\sum_{k=1}^{p} (x_{k,max} - x_{k,min})^2}}{\sqrt[p]{2q}} \tag{13}$$

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 to form separate niches in the population. The experiments performed by Deb and Goldberg [1989] showed sharing as a better way of keeping diversity than crowding, and indicated that phenotypic sharing was better than genotypic sharing.

It should be added that much further work has been done regarding keeping the diversity in the population. Deb and Goldberg [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]. Other researchers such as Eshelman [1991] and Schaffer [1991] did exactly the opposite: they didn't allow mating between individuals that were too similar (they said to be "preventing incest").

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

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

Kita et al. [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

$$F = \langle E \rangle - HT \tag{14}$$

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

4. NAÏVE APPROACHES TO MULTIOBJECTIVE OPTIMIZATION

The notion of genetic search in a multicriteria problem dates back to the late 60s, in which Rosenberg's [1967] study contained a suggestion that would have led to multicriteria optimization if he had carried it out as presented. His suggestion was to use multiple *properties* (nearness to some specified chemical composition) in his simulation of the genetics and chemistry of a population of single-celled organisms. Since his actual implementation contained only one single property, the multiobjective approach could not be shown in his work, but it was nevertheless a good starting point for researchers interested in this topic.

Knowing that a genetic algorithm needs scalar fitness information to work, probably the simplest idea that we could come up with would be to combine all the objectives into a single one using either an 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 to dominate the others. This implies that we should know, to a certain extent, the behavior of each of the objective functions, which is normally (at least in most real-world applications) a very expensive process (computationally speaking) that we can not afford in most cases. Obviously, if this combination of objectives is possible (and it is in some applications), this is not only the simplest approach, but also is one of the most efficient procedures, because no further interaction with the decision maker is required, and if the GA succeeds at optimizing the resulting fitness function, then the results will be at least sub-optimum in most cases.

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

4.1 Weighted sum approach

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

$$min \sum_{i=1}^{k} w_i f_i(\bar{x}) \tag{15}$$

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

$$\sum_{i=1}^{k} w_i = 1 \tag{16}$$

Since the results of solving an optimization model using (15) can vary significantly as the weighting coefficients change, and since very little is usually known about how to choose these coefficients, a necessary approach is to solve the same problem for many different values of w_i. But in this case, the designer is still, of course, confronted with the decision of having to choose the most appropriate solution based on his/her intuition.

Note that the weighting coefficients do not reflect proportionally the relative importance of the objectives, but are only factors which, when varied, locate points in the Pareto set. For the numerical methods that can be used to seek the minimum of (15), 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 reflect closely the importance of the objectives, all functions should be expressed in units of approximately the same numerical values. Additionally, we can also transform (15) to the form:

$$min \sum_{i=1}^{k} w_i f_i(\bar{x}) c_i \tag{17}$$

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

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

Applications

—Syswerda and Palmucci [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 was applied to generate hyperstructures from a set of chemical structures.

—Wilson & Macleod [1993] used this approach as one of the methods incorporated into a GA to design multiplierless IIR filters in which the two conflicting objectives were to minimize the response error and the implementation cost of the filter.

—Liu et al. [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, the robustness and the 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 involved in the decision making process. A weighted sum is still used in this approach, but it is combined with a fuzzy ranking technique that helps to identify Pareto solutions, since the coefficients of the objectives are represented with fuzzy numbers reflecting the existing uncertainty regarding their relative importance.

Criticism

This method was the first technique developed for the generation of non-inferior solutions for multiobjective optimization. This is an obvious consequence of the fact that it was implied by Kuhn and Tucker in their seminal work on numerical optimization [1951]. This method is very efficient computationally speaking, and can be applied to generate a strongly non-dominated solution that can be used as an initial solution for other techniques. The problem with this approach is how to determine the appropriate weights when we do not have enough information about the problem. In this case, any optimal point obtained will be a function of the coefficients used to combine the objectives. Most researchers 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, the decision maker has to assign targets or goals that he/she wishes to achieve for each objective. These values are incorporated into the problem as additional constraints. The objective function will then try to minimize the absolute deviations from the targets to the objectives. The simplest form of this method may be formulated as follows [1984]:

$$min \sum_{i=1}^{k} |f_i(\bar{x}) - T_i|, \quad \text{subject to } \bar{\mathrm{x}} \in \mathcal{F} \tag{18}$$

where T_i denotes the target or goal set by the decision maker for the ith objective function $f_i(\bar{x})$, and $\mathcal{F}$ represents the feasible region. The criterion, then, is to

[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 non-convex search spaces.

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; Ignizio 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 optimize simultaneously 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 that we wish to achieve, and if they are in the feasible region. However, the decision maker is given the task of devising the appropriate weights or priorities for the objectives that will eliminate the non-commensurable characteristics of the problem, which in most cases is difficult unless there is prior knowledge about the shape of the search space. Also, if the feasible region is difficult to approach, this method could become very inefficient. Nevertheless, this technique may be useful in cases in which a linear or piecewise-linear approximation of the objective functions can be made, because of the availability of excellent computer programs for that, and the possibility of eliminating dominated goal points easily. On the other hand, in non-linear cases, other approaches may be more efficient.

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:

$$Minimize \;\; \alpha \tag{19}$$

subject to:

$$\begin{aligned} g_j(\bar{x}) \leq 0; &\quad j = 1, 2, \ldots, m \\ b_i + \alpha \cdot w_i \geq f_i(\bar{x}); &\quad i = 1, 2, \ldots, k \end{aligned} \tag{20}$$

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

$$\sum_{i=1}^{k} |w_i| = 1 \tag{21}$$

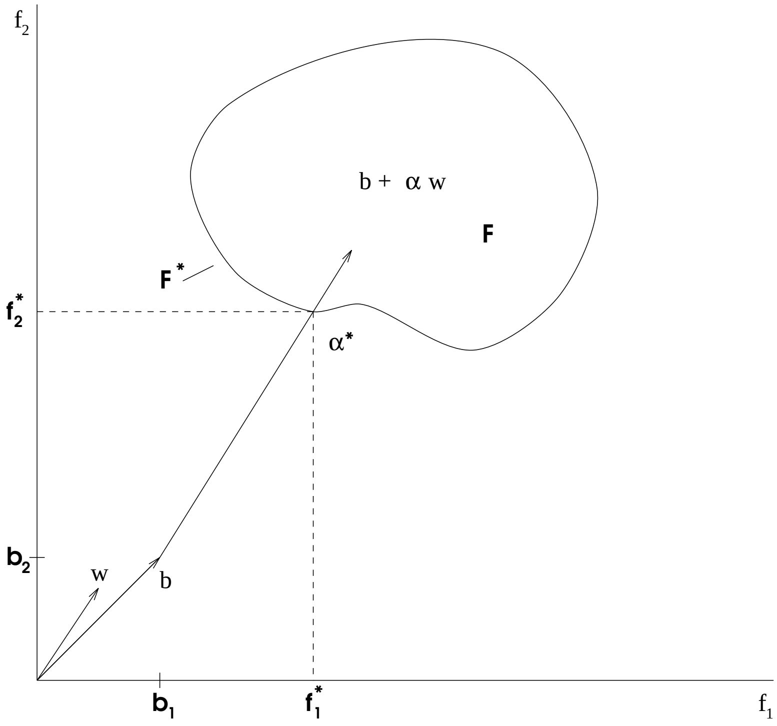

Fig. 6. Goal-attainment method with two objective functions.

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

It can be easily shown [Chen and Liu 1994] that the set of non-dominated solutions for a problem can be generated by varying the weights, with $w_i \geq 0$ $(i = 1, 2, \ldots, k)$ even for nonconvex problems. The mechanism by which this method operates is illustrated in Figure 6. 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 stated by equation (19) is equivalent to finding a feasible point on this vector in objective space which is closest to the origin. It is obvious that the optimal solution of equation (19) will be the first point at which $\bar{b} + \alpha \cdot \bar{w}$ intersects the feasible region $\mathcal{F}$ in the objective space. Should this point of intersection exist, it would clearly be a noninferior (or non-dominated) solution.

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

Applications

Wilson & MacLeod [1993] used this approach as another of the methods incorporated into their GA to design multiplierless IIR filters.

Criticism

As Wilson and MacLoud [1993] indicate, goal attainment has several problems, from which probably the main one is the misleading selection pressure that it can generate in some cases. For example, if we have two candidate solutions which are the same in one objective function value but different in the other, they will still have the same goal-attainment value for their two objectives, which means that for the GA none of them will be better than the other.

4.4 The ε-constraint Method

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

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

$$f_r(\bar{x}^*) = \min_{x \in \mathcal{F}} f_r(\bar{x}) \tag{22}$$

subject to additional constraints of the form

$$f_i(\bar{x}) \leq \varepsilon_i \quad for\ i = 1, 2, \ldots, k \ \ \text{and i} \neq \text{r} \tag{23}$$

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

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

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

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

$$\varepsilon_i \geq f_i(\bar{x}_i^*) \quad i = 1, 2, \ldots, r-1, r+1, \ldots, k \tag{24}$$

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

$$\varepsilon_i \leq f_i(\bar{x}_r^*) \quad i = 1, 2, \ldots, r-1, r+1, \ldots, k \tag{25}$$

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

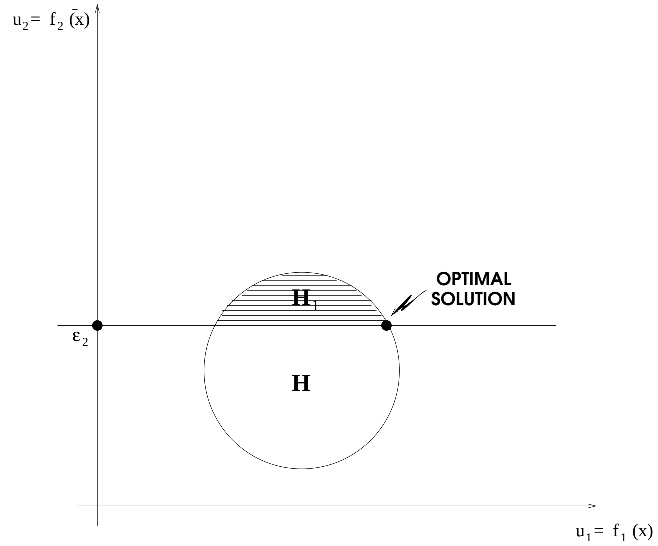

Fig. 7. The ε-constraint method for a maximizing problem.

Figure 7 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 handled as additional constraints.

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

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

In the mathematical programming literature, this approach is also known as the *trade-off method*, because of 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 used gradient based optimization techniques to speed up the search in order to reduce the computational cost required in a real-world application) to solve multiobjective optimization problems. Ranjithan et al. [1992] used this approach to solve groundwater pollution containment problems.

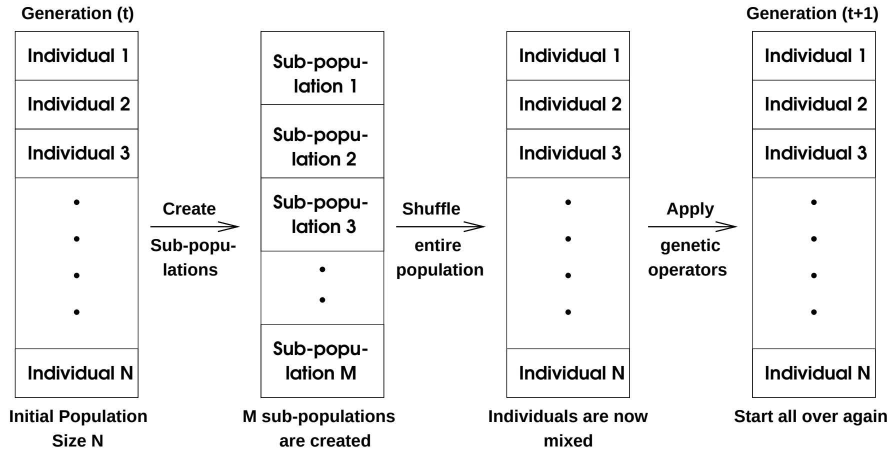

Fig. 8. Schematic of VEGA selection. It is assumed that the population size is N and that there are M objective functions.

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

Criticism

The obvious drawback of this approach is that it is time-consuming, and the coding of the objective functions may be difficult or even impossible for certain problems, particularly if there are too many objectives. Furthermore, 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. NON-AGGREGATING APPROACHES THAT ARE NOT PARETO-BASED

To overcome the difficulties involved in the aggregating approaches, much work has been devoted to the development of alternative techniques based on population policies or special handling of the objectives [Powell and Skolnick 1993]. Some of the most popular approaches that fall into this category will be examined in this section.

5.1 VEGA

David Schaffer [1985] extended Grefenstette's GENESIS program [Grefenstette 1984] to include multiple objective functions. Schaffer's approach was to use an extension of the Simple Genetic Algorithm (SGA) that he called the *Vector Evaluated Genetic Algorithm* (VEGA), and that differed of the first only in the way in which selection was performed. This operator was modified so that at each generation a number of sub-populations was generated by performing proportional selection according to each objective function in turn. Thus, for a problem with k objectives, k sub-populations of size N/k each would be generated (assuming a total population size of N). These sub-populations would be shuffled together to obtain a new population of size N, on which the GA would apply the crossover and mutation operators in the usual way. This process is illustrated in Figure 8. Schaffer realized that the solutions generated by his system were non-dominated in a local sense, because their non-dominance was limited to the current population, and while a locally dominated individual is also globally dominated, the converse is not necessarily true [Schaffer 1985]. An individual who is not dominated in one generation may become dominated by an individual who emerges in a later generation. Also, he noted a problem that in genetics is known as "speciation" (i.e., we could have the evolution of "species" within the population which excel on different aspects of performance). This problem arises because this technique selects individuals who excel in one dimension of performance, without looking at the other dimensions. The potential danger doing that is that we could have individuals with what Schaffer calls "middling" performance[4] in all dimensions, which could be very useful for compromise solutions, but that will not survive under this selection scheme, since they are not in the extreme for any dimension of performance (i.e., they do not produce the best value for any objective function, but only moderately good values for all of them). Speciation is undesirable because it is opposed to our goal of finding a compromise solution. Schaffer suggested some heuristics to deal with this problem. For example, to use a heuristic selection preference approach for non-dominated individuals in each generation, to protect our "middling" chromosomes. Also, crossbreeding among the "species" could be encouraged by adding some mate selection heuristics instead of using the random mate selection of the traditional GA.

Applications

Ritzel and Wayland [1994] used a variation of VEGA in which they incorporated a parameter to control the selection ratio. In the case of the groundwater pollution containment problem that Ritzel and Wayland solved, there were only two objectives, and the selection ratio was defined as the ratio of the fraction of strings selected on the basis of the first objective (reliability) to the fraction selected via the second objective (cost). Surry et al. [1995] proposed an interesting application of VEGA to model constraints in a single-objective optimization problem to avoid the need of a penalty function. Surry et al., however, modified the standard procedure of VEGA and introduced a form of ranking based on the number of constraints vio-

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

lated by a certain solution, and they reported that their approach worked well in the optimization of gas supply networks, since the tendency of VEGA to favor certain solutions can actually be an advantage when handling constraints, because in that case we want to favor precisely any solution that does not violate any constraint over those which do.

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

Tamaki et al. [1995, 1996] developed a technique in which at each generation, non-dominated individuals in the current population are kept for the following generation. This approach is really a mixture of Pareto selection (see next section) and VEGA, because if the number of non-dominated individuals is less that the population size, the remainder of the population in the following generation is filled applying VEGA to the dominated individuals. On the other hand, if the number of the non-dominated individuals exceeds the population size, individuals in the following generation are selected among the non-dominated individuals using VEGA. In a later version of this algorithm, called Pareto Reservation strategy, Tamaki et al. [1996] used also fitness sharing among the non-dominated individuals to maintain diversity in the population.

Criticism

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

5.2 Lexicographic ordering

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

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

$$Minimize \quad f_1(\bar{x}) \tag{26}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{27}$$

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

$$Minimize \quad f_2(\bar{x}) \tag{28}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{29}$$

$$f_1(\bar{x}) = f_1^* \tag{30}$$

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

$$Minimize \quad f_i(\bar{x}) \tag{31}$$

subject to

$$g_j(\bar{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{32}$$

$$f_l(\bar{x}) = f_l^*, \quad l = 1, 2, \ldots, i-1 \tag{33}$$

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 were assigned different priorities by the user and each pair of individuals were compared according to the objective with the highest priority. If this resulted in a tie, the objective with the second highest priority was used, and so on. In another version of this algorithm (that apparently worked quite well), an objective was randomly selected at each run. Fourman used this approach to design compact symbolic layouts [1985].

Kursawe [1991] formulated a multiobjective version of evolution strategies [Schwefel 1981] (ESs) based on lexicographic ordering. Selection consisted of as many steps as objective functions had the problem. At each step, one of these objectives was selected randomly according to a probability vector, and used to delete a fraction of the current population. After selection, the survivors became the parents of the next generation. The map of the trade-off surface was produced from the points evaluated during the run. Since the environment was allowed to change over time, diploid individuals were necessary to keep recessive information stored.

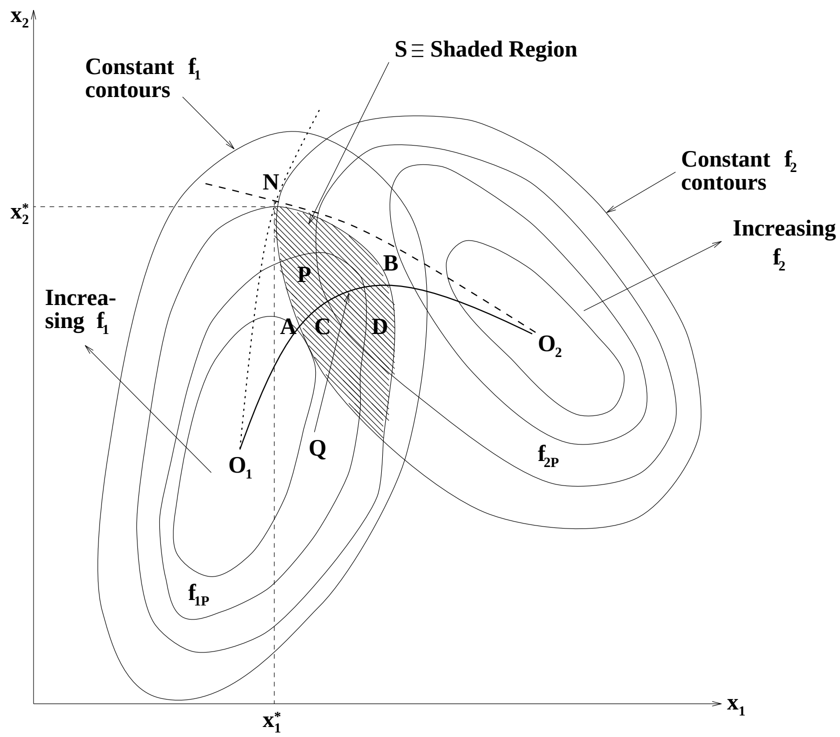

Fig. 9. Example of cooperative and non-cooperative game solutions.

Criticism

Selecting randomly an objective is equivalent to a weighted combination of objectives, in which each weight is defined in terms of the probability that each objective has of being selected. However, the use of tournament selection with this approach makes an important difference with respect to other approaches such as VEGA, because the pairwise comparisons of tournament selection will make scaling information neggligible [Fonseca and Fleming 1994; Fonseca and Fleming 1995c]. This means, that this approach may be able to see as convex a concave trade-off surface, although that really depends on the distribution of the population and on the problem itself. Its main drawback is that this approach will tend to favor more certain objectives when many are present in the problem, because of the randomness involved in the process, and this will have the undesirable consequence of making the population to converge to a particular part of the Pareto front rather than to delineate it completely [Coello 1996].

5.3 Use of 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 9. Let $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ represent two scalar objectives and x_1 and

x_2 two scalar design variables. It is assumed that one player is associated with each objective. The first player wants to select a design variable x_1 which will minimize his objective function f_1, and similarly the second player seeks a variable x_2 which will minimize his objective function f_2. If f_1 and f_2 are continuous, then the contours of constant values of f_1 and f_2 appear as shown in Figure 9. The dotted lines passing through O_1 and O_2 represent the loci of rational (minimizing) choices for the first and second player for a fixed value of x_2 and x_1, respectively. The intersection of these two lines, if it exists, is a candidate for the two objective minimization problem, assuming that the players do not cooperate with each other (*non-cooperative game*). In Figure 9, the point $N(x_1^*, x_2^*)$ represents such intersection point. This point, known as a *Nash equilibrium* solution, represents a stable equilibrium condition in the sense that no player can deviate unilaterally from this point for further improvement of his/her own criterion [Nash 1950].

This point has the characteristic that

$$f_1(x_1^*, x_2^*) \leq f_1(x_1, x_2^*) \tag{34}$$

and

$$f_2(x_1^*, x_2^*) \leq f_2(x_1^*, x_2) \tag{35}$$

where x_1 may be to the left or right of x_1^* in Equation (34) and x_2 may lie above or below x_2^* in Equation (35).

Applications

Périaux et al. [1997] proposed a GA-based approach that uses the concept of Nash equilibrium to solve a bi-objective optimization problem (the optimal distribution of active control elements which minimizes the backscaterring of aerodynamic reflectors). The main idea of this work was to use 2 non-cooperative players represented by 2 independent sub-populations in a genetic algorithm, and then make them to interact in the following way:

If f_1 and f_2 are the 2 objectives to be optimized, let P_1 and P_2 represent the 2 non-cooperative players. We start at generation zero with P_1 trying to optimize f_1 while f_2 remains fixed and P_2 trying to optimize f_2 while f_1 remains fixed. After one generation is over (i.e., when all the individuals in both populations have been evaluated and the genetic operators have been applied independently to each of those populations), we send (or migrate) the best individual from population 1 to population 2 and the best individual from population 2 to population 1. This process is repeated for as many generations as needed, until the Nash equilibrium is reached.

Criticism

This approach seems to be computationally very efficient, but in the state presented in the work by Périaux et al. [1997] is not possible to generate more than one non-dominated solution which, hopefully will be the best overall solution to the problem[5]. However, it is indeed possible to extend this approach to k players

[5] Périaux et al. didn't succeed at that in the example presented in their paper.

(where k is the number of objectives of a problem), and to have several Nash equilibrium points, with which the Pareto front can actually be found, although a *cooperative game* may be preferred in that case over a *non-cooperative* approach. For more information on cooperative games, refer to Rao [1987, 1984] and Coello [1996].

5.4 Using Gender to identify objectives

Robin Allenson [1992] used a population-based approach modelled after VEGA in which gender was used to distinguish between the two objective functions of a problem consisting of the planning of a route composed of a number of straight pipeline segments. With this approach, only male-female mating was allowed, and such gender was randomly assigned at birth. In the initial population, Allenson made sure that there was an equal number of males and females, but such balance was not kept after applying the genetic operators. At each generation, the worst individual (chosen according to one of the two genders) was eliminated and replaced by another (randomly picked) individual of the same gender. Allenson used evolution strategies to implement some form of sexual attractors that would modify the way in which mating was performed. The idea was to model the sexual attraction that some individuals have over others in nature, which determines a not so random mating.

Lis and Eiber [1996] also incorporated gender in their GA, but in a more general sense. In this case, the number of genders (or sexes), was not limited to two, but it could be as many as objectives we had. Another distinction of this approach is that the crossover operator was modifed as to allow panmitic reproduction, in which several parents generate a single child (instead of having two parents generate two children as in the traditional genetic algorithm). The idea was to select one parent from each sex to contribute to the generation of a child. This child will have the sex of the parent that contributed with the largest amount of genes (if there is a tie, then the sex is randomly chosen from the parents that contributed the same amount of genes). If no crossover takes place, then one of the individuals in the old generation is copied exactly the same (including its sex) to the following generation. In this approach, individuals are evaluated using different fitness functions (according to their corresponding sex). The mutation operator is only slightly restricted to avoid changes in the sex of an individual. As generations progress, a list of non-dominated individuals is updated, removing from it any individual that is no longer non-dominated after the list is modified. At the end, this list will contain the Pareto optimal solutions.

Applications

Lis and Eiber [1996] tested successfully their approach with the two multiobjective optimization problems provided in the paper by Srinivas and Deb [1993], but no further applications of this technique seem to be available at the moment.

Criticism

The use of genders is really another way of defining separate subpopulations for each objective. The difference of this approach with VEGA [Schaffer 1985] lies on the fact that Lis and Eiber used panmitic crossover, which imposes certain mating restrictions, avoiding the random crossing among different subpopulations performed by Schaffer. However, as we increase the number of objectives (or genders), we will have many subpopulations and panmitic crossover will become more inefficient (computationally speaking), because we will need to use more parents to generate a child. Additionally, the population size will have to be large enough as we increase the number of objectives, to keep a reasonably diverse spread of genders, across the entire population.

5.5 Weighted Min-Max Approach

The idea of stating the *min-max optimum* and applying it to multiobjective optimization problems, was taken from game theory, which deals with solving conflicting situations. The min-max approach to a linear model was proposed by Jutler [1967] and Solich [1969], and was further developed by Osyczka [1978, 1981, 1984], Rao [1986] and Tseng and Lu [1990].

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

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

or from

$$z_i^{''}(\bar{x}) = \frac{|f_i(\bar{x}) - f_i^0)|}{|f_i(\bar{x})|} \tag{37}$$

It should be clear that for (36) and (37) we have to assume that for every $i \in I$ and for every $\bar{x} \in \mathcal{F}$, $f_i(\bar{x}) \neq 0$.

If all the objective functions are going to be minimized, then equation (36) defines function relative increments, whereas if all of them are going to be maximized, it defines relative decrements. Equation (37) works conversely.

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

$$\forall_{i \in I}(z_i(\bar{x})) = max\ \{z_i^{'}(\bar{x}), z_i^{''}(\bar{x})\} \tag{38}$$

Now we define the min-max optimum as follows [1984]:

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

<u>Step 1:</u>

$$v_1(\bar{x}^*) = \min_{x \in \mathcal{F}} \ \max_{i \in I} \{z_i(\bar{x})\} \tag{39}$$

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

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

<u>Step 2:</u>

$$v_2(\bar{x}^*) = \min_{x \in x_1} \max_{i \in I, i \notin I_1} \{z_i(\bar{x})\} \tag{40}$$

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

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

Step r:

$$v_r(\bar{x}^*) = \min_{x \in x_{r-1}} \max_{i \in I, i \notin I_{r-1}} \{z_i(\bar{x})\} \tag{41}$$

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

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

Step k:

$$v_k(\bar{x}^*) = \min_{\bar{x} \in x_{k-1}} z_i(\bar{x}) \max_{i \in I, i \notin I_{k-1}} \quad \text{for i} \in \text{I and i} \notin \text{I}_{\text{k}-1} \tag{42}$$

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

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

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

Applications

Hajela and Lin [1992] included the weights of each objective in the chromosome, and promoted their diversity in the population through fitness sharing. Their goal was to be able to simultaneously generate a family of Pareto optimal designs corresponding to different weighting coefficients in a single run of the GA. Besides using sharing, Hajela and Lin used a vector evaluated approach based on VEGA to achieve their goal. They proposed the use of a utility function of the form:

$$\bar{U} = \sum_{i=1}^{l} W_i \frac{F_i}{F_i^*} \tag{43}$$

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 expressed in equation (10), 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 represents not 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 was imposed, to avoid members within a radius σ_{mat} to cross. The value of $\sigma_{mat} = 0.15$ was suggested by Hajela and Lin in their paper [1992].

Hajela and Lin [1992] used their approach to optimize a 10-bar plane truss in which weight and displacemente were to be minimized, and a wing-box structure in which they wanted to minimize its weight while maximizing its natural frequency.

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]. The use of sharing will avoid to a certain extent to have a premature convergence, but the use of a sharing factor (which is not easy to determine) increases the number of parameters required by the GA, and is therefore subject to further experimenting.

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

In a second approach, Coello [1996] proposed the use of a local ideal vector that was computed at each generation, and the selection process was modified as to allow the incorporation of min-max dominance. That means that a certain individual would be considered the winner of a tournament if its maximum deviation from the ideal vector was the smallest from the set under competition. Also, mating restrictions were imposed to keep only feasible solutions at all generations. Finally, sharing had to be used to overcome the high selection pressure introduced by the use of min-max tournament selection.

Applications

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

Criticism

The use of weights obviously represents a problem, because it is not always easy to find an appropriate set that can delineate correctly the part of the Pareto region that we wish to find. However, Coello [1996] showed through several engineering design examples that it was actually possible to find a good approximation of the Pareto front with a relatively small amount of weights chosen systematically (using a deterministic technique). The use of mating restrictions and feasibility checks during the entire evolution process may be seen as an important drawback, since it has been shown that such constraint-handling approach will not work in concave search surfaces. However, this was an attempt to incorporate the handling of constraints into the search process in another way different from the traditional penalty approach, and it does not preclude the algorithm from handling constraints in a different manner.

The second approach, in which weights are not used, is much more efficient and produces good Pareto fronts [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 Use of the Contact Theorem to detect Pareto Optimal Solutions

Osyczka and Kundu [1995] proposed the use of an algorithm based on the contact theorem (one of the main theorems in multiobjective optimization [1976]) to determine relative distances of a solution vector with respect to the Pareto set. In this paper [1995], the contact theorem was used to determine the fitness of each individual in the population.

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

The algorithm to identify Pareto optimal solutions is the following[6] [Osyczka 1984]:

(1) Read k (number of objective functions), n (number of decision variables), l^a (number of solutions available).

(2) Set $f_{i1}^p = \infty$ for $i = 1, 2, \ldots, k$ and $j^a = 1$.

(3) Set $l = 1$.

(4) Read $\bar{x}^{(l)}$ and $f(\bar{x}^{(l)})$.

(5) Set $j = 1$.

(6) If for every $i \in I$ we have $f_i(\bar{x}^{(l)}) < f_{ij}^p$ then substitute $\bar{x}_j^p = \bar{x}^{(l)}$ and $f_j^p = f(\bar{x}^{(l)})$ and go to 10, otherwise go to 7.

(7) If for every $i \in I$ we have $f_i(\bar{x}^{(l)}) > f_{ij}^p$ then go to 10 otherwise go to 8.

(8) Set $j = j + 1$.

(9) If $j > j^a$ then $j^a = j^a + 1$ and $x_{ja}^p = \bar{x}^{(l)}$ and $\bar{f}_j^p = \bar{f}(\bar{x}^{(l)})$ and go to 10, otherwise go to 6.

(10) Set $l = l + 1$.

(11) If $l \leq l^a$, then go to 4, otherwise go to 12.

(12) Print $\bar{x}_j^p$ and $\bar{f}_j^p$ for $j = 1, 2, \ldots, j^a$.

Applications

Although the GA-based technique presented by Osyczka and Kundu [1995] has not been applied to real-world problems (they used only two simple bi-objective optimization functions in their paper), Osyczka's algorithm for detecting Pareto optimality has been applied before to several problems, mainly in machine design

[6] The contact theorem is described in detail in Osyczka's book [1984].

[Osyczka 1975; Osyczka 1978; Osyczka 1981; Osyczka and Koski 1982; Osyczka 1984].

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, and its performance relies heavily on the so-called starting distance, which is some sort of scaling factor used to compare relative quality among the different solutions. If any of these 2 values is not chosen properly, too much selection pressure may be generated, or the GA may often jump back and forth between the feasible and unfeasible region at any given generation, producing too many dominated points in the process, and consequently losing portions of the Pareto front.

5.8 A Non-Generational Genetic Algorithm

Valenzuela-Rendón & Uresti-Charre [1997] proposed a GA that uses non-generational selection and in which the fitness of an individual is calculated incrementally. The idea comes from Learning Classifier Systems (LCS) [Goldberg 1989], in which it has been shown that a simple replacement of the worst individual in the population followed by an update of fitnesses of the rest of the population works better than a traditional (generational) GA. In the context of multiobjective optimization, what Valenzuela-Rendón and Uresti-Charre did was to transform the problem with N objectives into another one with only two objectives : the minimization of domination count (weighted average of the number of individuals that have dominated this individual so far) and the minimization of the moving niche count (weighted average of the number of individuals that lie close according to a certain sharing function). Then, this bi-objective optimization problem is transformed into a single objective optimization problem by taking a linear combination of these 2 objectives. The basic algorithm is the following [Valenzuela-Rendón and Uresti-Charre 1997]:

(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 that dominated each other individual in the group. Similarly, the moving niche count is updated, using a certain measurement of closeness (normally a distance among their fitness values) 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:

$$fitness_i = c_d d_i + c_w w_i \tag{44}$$

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:

$$d(t+1) = d(t) - k_d d(t) + D(t) \tag{45}$$

where k_d was set to zero in Valenzuela's experiments, and $D(t)$ was 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:

$$w(t+1) = w(t) - k_w w(t) + sharing(d) \tag{46}$$

where k_w was set to $\frac{1}{P}$ in Valenzuela's experiments and $sharing(d)$ refers to the sharing expression used, based on the distance d allowed among individuals to consider 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] that has been explained before.

—Perform proportional selection according to the maximum fitness in the population.

—Apply crossover and mutation, and produce a single new individual that 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 [1993] (see below) in 3 bi-objective optimization problems, both in terms of the number of points in the Pareto front at the final iteration, and in terms of the total number of function evaluations. However, no further comparisons with other methods or in problems with more objectives was provided.

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 called weighted average ranking (WAR). Even when this approach seems to provide good distributions, it does not seem feasible to incorporate in it preferences of objectives defined by decision maker, which may be a drawback in real-world applications. Also, it does not seem to be clear how to define the additional parameters required by this algorithm, which seem to be subject to empirical fine tuning as the other normal parameters of the GA (e.g., crossover and mutation rates).

5.9 Use of randomly generated weights and elitism

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

(1) Generate the initial population randomly.

(2) Compute the values of the p objectives for each individual in the population. Then determine which are the non-dominated solutions and store them in a separate population that we will call NOND to distinguish it from the current population, that we will denominate CURRENT.
(3) If N represents the number of non-dominated 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, ..., r_k be k random numbers in the interval [0,1]. The fitness function used for each individual is:

$$f(\bar{x}) = \sum_{i=1}^{p} w_i f_i(\bar{x}) \tag{47}$$

where p is the number of objectives, and

$$w_i = \frac{r_i}{(r_1 + r_2 + \ldots + r_p)} \tag{48}$$

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

$$\sum_{i}^{p} w_i = 1 \tag{49}$$

—Select a parent with probability:

$$P(\bar{x}) = \frac{f(\bar{x}) - f_{min}(CURRENT)}{\sum_{x \in CURRENT}\{f(\bar{x}) - f_{min}(CURRENT)\}} \tag{50}$$

where f_{min} is the minimum fitness in the current population.
(4) Apply crossover to the selected $(M-N)$ pairs of parents. Apply then mutation to the new solutions generated.
(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 was 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 were examined and if any of them was better than the current individual, then it would replace it. Local search was applied to the M individuals in CURRENT.
(7) Finish if a pre-specified stopping criterion is satisfied (e.g., the pre-defined maximum number of generations has been reached). Otherwise, return to step 2.

Applications

Ishibuchi and Murata [1996] used this technique to solve bi-objective optimization flowshop scheduling problems in which the makespan and maximum tardiness were to be minimized.

Criticism

This approach is very similar to the technique called Sum of Weighted Rations (SWR) by 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 as to keep a wide spread of solutions through many generations. However, Coello [1996] has shown (using a similar approach), that such spread may not be kept in problems in which there is an objective in the ideal vector that can be easily achieved by a wide set of solutions. In such case, it is necessary to use sharing techniques or a local search technique (as proposed by Ishibuchi and Murata [1996]) to keep diversity. Bentley and Wakefield [1997] showed also another variation of this algorithm called Sum of Weighted Global Ratios (SWGR) which visibly reduces the spread of solutions produced (i.e., the size of the Pareto set) by using the globally best and worst values instead of the current ones. The idea is nevertheless interesting and the implementation of this algorithm seems to be not only easy, but also quite efficient with respect to most of the Pareto-based approaches described next.

6. PARETO-BASED APPROACHES

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

Applications

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

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

Criticism

The main problem with Pareto ranking in general is that there is no efficient algorithm to check for non-dominance in a set of feasible solutions [Coello 1996]. Traditional algorithms have serious degradation in performance as we increase the size of the population and the number of objectives. Also, the use of sharing makes necessary to be able to estimate the value of σ_{share}, which is not easy, and the performance of the method relies a lot on such value.

6.1 Multiple Objective Genetic Algorithm

Fonseca and Fleming [1993] have proposed a scheme in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual x_i at generation t, which is dominated by $p_i^{(t)}$ individuals in the current generation. Its current position in the individuals' rank can be given by [Fonseca and Fleming 1993]:

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{51}$$

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

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

(1) Sort population according to rank.

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

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

As Goldberg and Deb [1991] point out, this type of blocked fitness assignment is likely to produce a large selection pressure that might produce premature convergence. To avoid that, Fonseca and Fleming used a niche-formation method to distribute the population over the Pareto-optimal region, but instead of performing sharing on the parameter values, they have used sharing on the objective function values [Srinivas and Deb 1994].

In 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. While the basic ranking scheme remains unaltered, as we perform a Pareto comparison of the individuals, then those objectives which already satisfy their goals 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, Chen Tan and Li [1997] reported success in the use of MOGA for the multiobjective optimization of ULTIC controllers that satisfy a number of time domain and frequency domain specifications. Also, 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 non-linear model structures, as an alternative that the authors reported to work better (in terms of representation power) than the use of the conventional linear representation of MOGA that they had attempted before [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 for the preplanning of containership layouts (a large scale combinatorial problem). In Todd and Sen's approach, a population of non-dominated individuals is kept and updated at each generation, removing individuals that become dominated and duplicates. The traditional genetic operators and sharing are applied only to this population. Niche sizes are computed automatically for each criterion by substracting the maximum value for that criterion from the minimum and dividing it by the population size. Crossover was restricted so that only individuals that were very similar could mate, and because of the permutations encoded, a repair algorithm had to be used afterwards. Finally, a heuristic mutation that basically defined rules to exchange bit positions had to be used to avoid premature convergence of the population.

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 under this scheme. This is apparently undesirable, because these are precisely the kind of solutions that the user normally wants, although it should be said that the method works quite well in practice [Coello 1996].

MOGA is a good approach, efficient and relatively easy to implement, but as 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 using their approach.

6.2 Non-dominated Sorting Genetic Algorithm

The Non-dominated Sorting Genetic Algorithm (NSGA) was proposed by Srinivas and Deb [1993], and is based on several layers of classifications of the individuals. Before the selection is performed, the population is ranked on the basis of non-

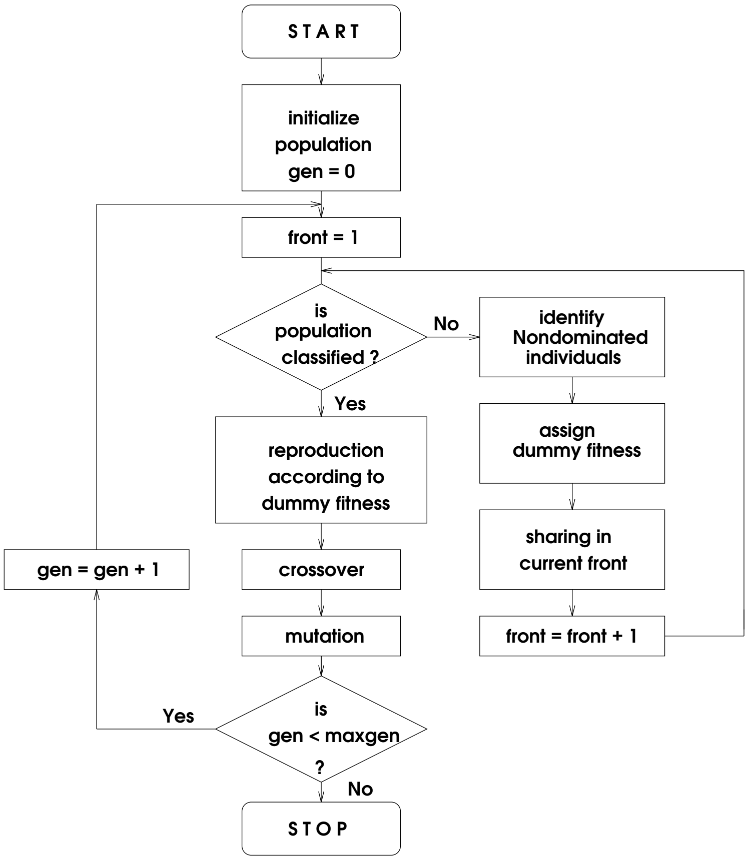

Fig. 10. Flowchart of the Nondominated Sorting Genetic Algorithm (NSGA).

domination: all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population are classified. A stochastic remainder proportionate selection was used for this approach. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows to search for nondominated regions, and results in quick convergence of the population toward such regions. Sharing, by its part, helps to distribute it over this region. The efficiency of NSGA lies in the way in which multiple objectives are reduced to a dummy fitness function using a nondominated sorting procedure. With this approach, any number of objectives can be solved [Srinivas and Deb 1994], and both maximimization and minimization problems can be handled. Figure 10 shows the general flow chart of this approach.

Applications

Périaux et al. [1997] used the NSGA to find an optimal distribution of active control elements which minimizes the backscattering of aerodynamic reflectors. Vedarajan et al. [1997] used the NSGA for investment portfolio optimization, but interestingly they used binary tournament selection instead of stochastic remainder selection as suggested by Srinivas and Deb [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 had been absorbed by the extra individuals in the population. Michielssen and Weile [1995] used also the 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 terms of quality of the 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 other individuals in the population was used to help determine dominance (typically around 10). When both competitors were either dominated or non-dominated (i.e., there was a tie), the result of the tournament was decided through fitness sharing [Goldberg and Richardson 1987]. Population sizes considerably larger than usual with other approaches were used so that the noise of the selection method could be tolerated by the emerging niches in the population [Fonseca and Fleming 1994].

The pseudocode for Pareto domination tournaments assuming that all of the objectives are to be 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); /* Re-randomize random index array */
    candidate_1 = random_pop_index[1];
    candidate_2 = random_pop_index[2];
    candidate_1_dominated = false;
    candidate_2_dominated = false;
    for comparison_set_index = 3 to t_dom + 3 do
    /* Select t_dom individuals randomly from S */
        begin
            comparison_individual = random_pop_index[comparison_set_index];
            if S[comparison_individual] dominates S[candidate_1]
                then candidate_1_dominated = true;
            if S[comparison_individual] dominates S[candidate_2]
                then candidate_2_dominated = true;
        end /* end for loop */
    if ( candidate_1_dominated AND ¬ candidate_2_dominated )
        then return candidate_2;
    else if ( ¬ candidate_1_dominated AND candidate_2_dominated )
        then return candidate_1;
    else
        do sharing;

end
```

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 called *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 of the problem in the selection mechanism (as in the NPGA), but then selected the individuals to be reproduced to generate the following population using a random walk operator. This obviously produces a locally dominating individual rather than a globally dominating one. Additionally, if more than one non dominated individual is found, then the first one encountered is selected (instead of doing sharing like in the NPGA). At the end of every new generation, the set of Pareto optimal solutions is updated and stored. They used this approach for airfoil design [Quagliarella and Vicini 1997].

Criticism

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

7. FUTURE RESEARCH PATHS

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

—Since the size of the Pareto set is normally considerably large, and in the particular case of using a genetic algorithm, depends on the size of the population, it may be desirable in some cases to devise ways of reducing the number of elements in such set, in order to facilitate the analysis for the decision maker. Kunha, Oliveira and Covas [1997] proposed the incorporation of 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 to determine how to measure the quality of a solution. So far, practically visual inspection is the only technique used, unless there is some previous knowledge of the points which lie in the Pareto front (in which case there is obviously no need for a multiobjective optimization technique). Fonseca and Fleming [1996b] proposed the definition of certain (arbitrary) goals that we wish the GA to attain; then we can perform multiple runs and apply standard non-parametric statistical procedures to evaluate the quality of the solutions (i.e., the non-dominated fronts) produced by the technique under study, and/or compare it against other similar techniques. However, these arbitrary goals are not easy to define either, 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 be able to assign more importance to certain objective or objectives. Interestingly, in such cases, an aggregating approach allows us to change the importance of the objectives very easily, as opposed to any ranking technique (i.e., Pareto-based approaches) which normally do not provide the means to do it directly. Fonseca and Fleming [1993] proposed the use of a utility function combined with MOGA [Fonseca and Fleming 1994; Fonseca and Fleming 1995c] to produce a method for the progressive articulation of preferences. The idea that they proposed was to have a feedback loop between the decision maker and the GA, so that certain solutions (from the Pareto set) are given more preference than others. Ideally, such process could be done automatically by replacing the decision maker with an expert system [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 to obtain in some cases.

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

Finally, Voget and Kolonko [1998] proposed the use of a fuzzy controller that regulates the selection pressure automatically by using a set of predefined goals that define the 'desirable' behavior of the population. An interesting aspect of this work is that they actually combine Pareto ranking with VEGA during the same run of the GA, to allow the desired reduction of deviations from the goals specified by the authors [1998].

These 3 proposals are quite interesting, but still more work needs to be done in this area, preferrently with real-world problems (Fonseca's approach was an appropriate choice for the optimization of a gas turbine engine [1993], and Greenwood et al. [1997] showed that their approach performed well in two hardware/software codesign problems), so that more general guidelines can be derived from the different approaches proposed.

—Directly related to the problem of measuring the quality of the solutions found with a multiobjective optimization technique lies the need to have a set of benchmark problems that can be used to test existing and new approaches. This set should include both constrained and unconstrained problems[7], examples with few objectives (2 or 3) suitable for graphical inspection, problems with few and several variables, and problems in which is possible to achieve the ideal vector (these could be used to tune up any technique to be tried). Also, there is a need to perform detailed studies of performance of different GAs (assuming certain quality measures) using these benchmark problems, and derive more accurate information on the behavior of each of the algorithms used. Coello [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, because it is not obvious to know when the population has reached a point from which no further improvement can be reached. Currently, the main approaches used to stop this kind of GA haven been to either use a fixed number of generations, or to monitor the population at certain intervals and interpret visually the results to determine when to halt the evolution process.

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

—Some researchers have also found alternative applications of multiobjective optimization techniques that are quite interesting. The most remarkable is perhaps the attempt to use ranking techniques or similar approaches to handle constraints in a single objective optimization problem, as to avoid the use of penalty functions. Surry et al. [1995] proposed the COMOGA (Constrained Optimization by Multi-Objective Genetic Algorithms) approach, which treats each constraint as a separate objective and therefore transforms a constrained single objective optimization problem into an unconstrained multi-objective optimization problem, which is solved using Fonseca's MOGA [1993]. This approach was used by Surry et al. to optimize gas supply networks [1995]. Fonseca and Fleming [1995a] also proposed to handle constraints as objectives, and applied their approach to the design of a gas turbine [1995b]. Finally, Stanley and Mudge [1995] used also Pareto ranking to handle constraints treated as objectives in a combinatorial optimization problem (microprocessor design).

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

[7]Most current papers that introduce new GA-based multiobjective optimization techniques, use 2 or 3 simple unconstrained bi-objective functions, particularly those used originally by Schaffer [1985].

8. CONCLUSIONS

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

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

REFERENCES

AHERNE, F. J., THACKER, N. A., AND ROCKETT, P. I. 1997. Optimal Pairwise Geometric Histograms. In A. F. CLARK Ed., *Electronic Proceedings of the Eighth British Machine Vision Conference, BMVC97* (University of Essex, United Kingdom, 1997), pp. http://peipa.essex.ac.uk/bmva/bmvc97/papers/071/bmvc.html.

ALLENSON, R. 1992. Genetic algorithms with gender for multi-function optimisation. Technical Report EPCC-SS92-01, Edinburgh Parallel Computing Centre, Edinburgh, Scotland.

BAIER, H. 1977. Über 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 *Fifth AIAA/USAF/NASA Symposium on Multidisciplinary Analysis and Optimization*, Volume Paper 84-4363-CP (Panama City, Florida, 1994), pp. 1015–1022. AIAA.

BENTLEY, P. J. AND WAKEFIELD, J. P. 1997. Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. In *Proceedings of the 2nd On-Line World Conference on Soft Computing in Engineering Design and Manufacturing (WSC2)* (June 1997), pp. `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*, Volume 1. John Wiley, New York.

CHEN, Y. L. AND LIU, C. C. 1994. Multiobjective VAR planning using the goal-attainment method. *IEE Proceedings on Generation, Transmission and Distribution 141*, 3 (may), 227–232.

CHIPPERFIELD, A. AND FLEMING, P. 1995. Gas Turbine Engine Controller Design using Multiobjective Genetic Algorithms. In A. M. S. ZALZALA Ed., *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems : Innovations and Applications, GALESIA'95* (Halifax Hall, University of Sheffield, UK, September 1995), pp. 214–219. IEEE.

COELLO, C. A. C. 1996. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design.* Ph. D. thesis, Department of Computer Science, Tulane University, New Orleans, LA.

COELLO, C. A. C. AND CHRISTIANSEN, A. D. 1998a. MOSES : A Multiobjective Optimization Tool for Engineering Design. *Engineering Optimization*. (Accepted for publication).

COELLO, C. A. C. AND CHRISTIANSEN, A. D. 1998b. Two new GA-based methods for multiobjective optimization. *Civil Engineering Systems*. (In Press).

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*. (In Press).

Coello, C. A. C., Hernández, F. S., and Farrera, F. A. 1997. Optimal design of reinforced concrete beams using genetic algorithms. *Expert Systems with Applications : An International Journal 12*, 1 (January).

Cohon, J. L. and Marks, D. H. 1975. A review and evaluation of multiobjective programming techniques. *Water Resources Research 11*, 2 (apr), 208–220.

Cvetković, D., Parmee, I., and Webb, E. 1998. Multi-objective Optimisation and Preliminary Airframe Design.

Deb, K. and Goldberg, D. E. 1989. An investigation of niche and species formation in genetic function optimization. In J. D. Schaffer Ed., *Proceedings of the Third International Conference on Genetic Algorithms* (San Mateo, California, jun 1989), pp. 42–50. George Mason University: Morgan Kaufmann Publishers.

DeJong, A. K. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* Ph. D. thesis, University of Michigan.

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

Eshelman, L. J. 1991. The CHC Adaptive Search Algorithm: How to Have Safe Search when Engaging in Nontraditional Genetic Recombination. In G. E. Rawlins Ed., *Foundations of Genetic Algorithms*, pp. 265–283. San Mateo, California: Morgan Kaufmann Publishers.

Eshelman, L. J. and Schaffer, J. D. 1991. Preventing premature convergence in genetic algorithms by preventing incest. In R. K. Belew and L. B. Booker Eds., *Proceedings of the Fourth International Conference on Genetic Algorithms* (San Mateo, California, 1991), pp. 115–122. Morgan Kaufmann Publishers.

Evans, G. W. 1984. An overview of techniques for solving multiobjective mathematical programs. *Management Science 30*, 11 (nov), 1268–1282.

Fishburn, P. C. 1978. A survey of multiattribute/multicriterion evaluation theories. In S. Zionts Ed., *Multiple Criteria Problem Solving* (Berlin, 1978), pp. 181–224. Springer-Verlag.

Fonseca, C. M. and Fleming, P. J. 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest Ed., *Proceedings of the Fifth International Conference on Genetic Algorithms* (San Mateo, California, 1993), pp. 416–423. University of Illinois at Urbana-Champaign: Morgan Kauffman Publishers.

Fonseca, C. M. and Fleming, P. J. 1994. An overview of evolutionary algorithms in multiobjective optimization. Technical report, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U. K.

Fonseca, C. M. and Fleming, P. J. 1995a. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms I: A Unified Formulation. Technical Report 564 (January), 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. Technical Report 565 (January), University of Sheffield, Sheffield, UK.

Fonseca, C. M. and Fleming, P. J. 1995c. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation 3*, 1 (Spring), 1–16.

Fonseca, C. M. and Fleming, P. J. 1996a. Nonlinear System Identification with Multiobjective Genetic Algorithms. In *Proceedings of the 13th World Congress of IFAC* (San Francisco, California, 1996), pp. 187–192.

Fonseca, C. M. and Fleming, P. J. 1996b. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel Eds., *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pp. 584–593. Berlin, Germany: Springer-Verlag.

Fourman, M. P. 1985. Compaction of symbolic layout using genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms* (1985), pp. 141–153. Lawrence Erlbaum.

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

GEN, M., IDA, K., AND LI, Y. 1995. Solving bicriteria solid transportation problem with fuzzy numbers by genetic algorithm. *International Journal of Computers and Industrial Engineering 29*, 537–543.

GOLDBERG, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts.

GOLDBERG, D. E. AND DEB, K. 1991. A comparison of selection schemes used in genetic algorithms. In G. E. RAWLINS Ed., *Foundations of Genetic Algorithms*, pp. 69–93. San Mateo, California: Morgan Kaufmann.

GOLDBERG, D. E. AND RICHARDSON, J. 1987. Genetic algorithm with sharing for multimodal function optimization. In J. J. GREFENSTETTE Ed., *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (1987), pp. 41–49. Lawrence Erlbaum.

GREENWOOD, G. W., HU, X. S., AND D'AMBROSIO, J. G. 1997. Fitness Functions for Multiple Objective Optimiztion Problems: Combining Preferences with Pareto Rankings. In R. K. BELEW AND M. D. VOSE Eds., *Foundations of Genetic Algorithms 4*, pp. 437–455. San Mateo, California: Morgan Kaufmann.

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

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

HAJELA, P. AND LEE, J. 1996. Constrained Genetic Search Via Scheme Adaptation: An Immune Network Solution. *Structural Optimization 12*, 1, 11–15.

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

HAJELA, P., YOO, J., AND LEE, J. 1997. GA Based Simulation of Immune Networks—Applications in Structural Optimization. *Journal of Engineering Optimization 29*, 131–149.

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 R. SHARDA, B. L. GOLDEN, E. WASIL, O. BALCI, AND W. STEWART Eds., *Impacts of Recent Computer Advances on Operations Research*. New York: North-Holland Publishing Company.

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

HOLLAND, J. H. 1992. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence* (Second ed.). MIT Press, Cambridge, Massachusetts.

HORN, J. AND NAFPLIOTIS, N. 1993. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.

HWANG, C. L., PAIDY, S. R., AND YOON, K. 1980. Mathematical Programming with Multiple Objectives: A Tutorial. *Computing and Operational Research 7*, 5–31.

IGNIZIO, J. P. 1976. *Goal Programming and Extensions*. Heath, Lexington, Massachusetts.

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

IJIRI, Y. 1965. *Management Goals and Accounting for Control*. North-Holland, Amsterdan.

ISHIBUCHI, H. AND MURATA, T. 1996. Multi-Objective Genetic Local Search Algorithm. In T. FUKUDA AND T. FURUHASHI Eds., *Proceedings of the 1996 International Conference on Evolutionary Computation* (Nagoya, Japan, 1996), pp. 119–124. IEEE.

JAKOB, W., GORGES-SCHLEUTER, M., AND BLUME, C. 1992. Application of genetic algorithms to task planning and learning. In R. MÄNNER AND B. MANDERICK Eds., *Parallel*

Problem Solving from Nature, 2nd Workshop, Lecture Notes in Computer Science (Amsterdam, 1992), pp. 291–300. North-Holland Publishing Company.

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 S. Forrest Ed., *Proceedings of the Fifth International Conference on Genetic Algorithms* (San Mateo, California, 1993), pp. 597–602. Morgan Kaufmann.

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

Kita, H., Yabumoto, Y., Mori, N., and Nishikawa, Y. 1996. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel Eds., *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pp. 504–512. Berlin, Germany: Springer-Verlag.

Koza, J. R. 1992. *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* The MIT Press.

Kuhn, H. W. and Tucker, A. W. 1951. Nonlinear programming. In J. Neyman Ed., *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (Berkeley, California, 1951), pp. 481–492. University of California Press.

Kunha, A. G., Oliveira, P., and Covas, J. A. 1997. Use of Genetic Algorithms in Multicriteria Optimization to Solve Industrial Problems. In T. Bäck Ed., *Proceedings of the Seventh International Conference on Genetic Algorithms* (San Mateo, California, July 1997), pp. 682–688. Michigan State University: Morgan Kaufmann Publishers.

Kursawe, F. 1991. A variant of evolution strategies for vector optimization. In H. P. Schwefel and R. Männer Eds., *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, Volume 496 of *Lecture Notes in Computer Science* (Berlin, Germany, oct 1991), pp. 193–197. Springer-Verlag.

Lieberman, E. R. 1991. Soviet multi-objective mathematical programming methods: An overview. *Management Science 37*, 9 (sep), 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 D. E. Brown and C. C. White Eds., *Operations research and Artificial Intelligence: The integration of problem-solving strategies*, pp. 29–57. Norwell, Massachusetts: Kluwer Academic.

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

Lis, J. and Eiben, A. E. 1996. A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. In T. Fukuda and T. Furuhashi Eds., *Proceedings of the 1996 International Conference on Evolutionary Computation* (Nagoya, Japan, 1996), pp. 59–64. IEEE.

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

Loughlin, D. H. and Ranjithan, S. 1997. The Neighborhood constraint method: A Genetic Algorithm-Based Multiobjective Optimization Technique. In T. Bäck Ed., *Proceedings of the Seventh International Conference on Genetic Algorithms* (San Mateo, California, July 1997), pp. 666–673. Michigan State University: Morgan Kaufmann Publishers.

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

Michalewicz, Z. 1992. *Genetic Algorithms + Data Structures = Evolution Programs* (Second ed.). Springer-Verlag.

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

Mitchell, M. 1996. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, Massachusetts.

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

OBAYASHI, S. 1997. Pareto Genetic Algorithm for Aerodynamic Design using the Navier-Stokes Equations. In D. QUAGLIARELLA, J. PÉRIAUX, C. POLONI, AND G. WINTER Eds., *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, Chapter 12, pp. 245–266. West Sussex, England: John Wiley and Sons.

OSYCZKA, A. 1975. Optimization of the steady state parameters for machine tool gear trains. *International Journal of Machine Tool Design and Research 15*, 31–68.

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

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

OSYCZKA, A. 1984. *Multicriterion Optimization in Engineering with FORTRAN programs.* Ellis Horwood Limited.

OSYCZKA, A. 1985. Multicriteria optimization for engineering design. In J. S. GERO Ed., *Design Optimization*, pp. 193–227. Academic Press.

OSYCZKA, A. AND KOSKI, J. 1982. Selected works related to multicriterion optimization methods for engineering design. In *Proceedings of Euromech Colloquium* (University of Siegen, 1982).

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

PARETO, V. 1896. *Cours D'Economie Politique*, Volume I and II. F. Rouge, Lausanne.

PÉRIAUX, J., SEFRIOUI, M., AND MANTEL, B. 1997. Ga Multiple Objective Optimization Strategies for Electromagnetic Backscattering. In D. QUAGLIARELLA, J. PÉRIAUX, C. POLONI, AND G. WINTER Eds., *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, Chapter 11, pp. 225–243. West Sussex, England: John Wiley and Sons.

POLONI, C. AND PEDIRODA, V. 1997. GA coupled with computationally expensive simulations: tools to improve efficiency. In D. QUAGLIARELLA, J. PÉRIAUX, C. POLONI, AND G. WINTER Eds., *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, Chapter 13, pp. 267–288. West Sussex, England: John Wiley and Sons.

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

QUAGLIARELLA, D. AND VICINI, A. 1997. Coupling Genetic Algorithms and Gradient Based Optimization Techniques. In D. QUAGLIARELLA, J. PÉRIAUX, C. POLONI, AND G. WINTER Eds., *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, Chapter 14, pp. 289–309. West Sussex, England: John Wiley and Sons.

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 Transactions AGU 73*, 14, 125. Spring meeting supplement.

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

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

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

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

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

Rodríguez-Vázquez, K., Fonseca, C. M., and Fleming, P. J. 1997. Multiobjective Genetic Programming : A Nonlinear System Identification Application. In J. R. Koza Ed., *Late Breaking Papers at the Genetic Programming 1997 Conference* (Stanford University, California, July 1997), pp. 207–212. Stanford Bookstore.

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

Rosenberg, R. S. 1967. *Simulation of genetic populations with biochemical properties*. Ph. D. thesis, University of Michigan, Ann Harbor, Michigan.

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

Schaffer, J. D. 1985. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms* (1985), pp. 93–100. Lawrence Erlbaum.

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

Smith, R. E., Forrest, S., and Perelson, A. S. 1993. Population diversity in an immune system model: Implications for genetic search. In L. D. Whitley Ed., *Foundations of Genetic Algorithms 2*, pp. 153–165. San Mateo, California: Morgan Kaufmann Publishers.

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

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

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

Stadler, W. 1984. A Survey of Multicriteria Optimization or the Vector Maximum Problem, Part I : 1776-1960. *Journal of Optimization Theory and Applications 29*, 1 (sep), 1–52.

Stanley, T. J. and Mudge, T. 1995. A Parallel Genetic Algorithm for Multiobjective Microprocessor Design. In L. J. Eshelman Ed., *Proceedings of the Sixth International Conference on Genetic Algorithms* (San Mateo, California, July 1995), pp. 597–604. University of Pittsburgh: Morgan Kaufmann Publishers.

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 T. C. Fogarty Ed., *Evolutionary Computing. AISB Workshop. Selected Papers*, Lecture Notes in Computer Science, pp. 166–180. Sheffield, U.K.: Springer-Verlag.

Syswerda, G. and Palmucci, J. 1991. The Application of Genetic Algorithms to Resource Scheduling. In R. K. Belew and L. B. Booker Eds., *Proceedings of the Fourth International Conference on Genetic Algorithms* (San Mateo, California, 1991), pp. 502–508. Morgan Kaufmann.

Szidarovszky, F. and Duckstein, L. 1982. Basic Properties of MODM problems. In *Classnotes 82-1*. Tucson, Arizona: Department of Systems and Industrial Engineering, University of Arizona.

Tamaki, H., Kita, H., and Kobayashi, S. 1996. Multi-Objective Optimization by Genetic Algorithms : A Review. In T. Fukuda and T. Furuhashi Eds., *Proceedings of the 1996 International Conference on Evolutionary Computation* (Nagoya, Japan, 1996), pp. 517–522. IEEE.

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 3rd APORS* (1995), pp. 374–381.

TAN, K. C. AND LI, Y. 1997. Multi-Objective Genetic Algorithm Based Time and Frequency Domain Design Unification of Linear Control Systems. Technical Report CSC-97007, Department of Electronics and Electrical Engineering, University of Glasglow, Glasglow, Scotland.

TODD, D. S. AND SEN, P. 1997. A Multiple Criteria Genetic Algorithm for Containership Loading. In T. BÄCK Ed., *Proceedings of the Seventh International Conference on Genetic Algorithms* (San Mateo, California, July 1997), pp. 674–681. Michigan State University: Morgan Kaufmann Publishers.

TSENG, C. H. AND LU, T. W. 1990. Minimax multiobjective optimization in structural design. *International Journal for Numerical Methods in Engineering 30*, 1213–1228.

VALENZUELA-RENDÓN, M. AND URESTI-CHARRE, E. 1997. A Non-Generational Genetic Algorithm for Multiobjective Optimization. In T. BÄCK Ed., *Proceedings of the Seventh International Conference on Genetic Algorithms* (San Mateo, California, July 1997), pp. 658–665. Michigan State University: Morgan Kaufmann Publishers.

VEDARAJAN, G., CHAN, L. C., AND GOLDBERG, D. E. 1997. Investment Portfolio Optimization using Genetic Algorithms. In J. R. KOZA Ed., *Late Breaking Papers at the Genetic Programming 1997 Conference* (Stanford University, California, July 1997), pp. 255–263. Stanford Bookstore.

VOGET, S. AND KOLONKO, M. 1998. Multidimensional Optimization with a Fuzzy Genetic Algorithm. *Journal of Heuristics*. (In Press).

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

WIENKE, P. B., LUCASIUS, C., AND KATEMAN, G. 1992. Multicriteria target optimization of analytical procedures using a genetic algorithm. *Analytical 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 *IEE/IEEE Workshop on Natural Algorithms in Signal Processing* (Chelmsford, U.K., 1993), pp. 4/1–4/8.

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