A Short Tutorial on Evolutionary Multiobjective Optimization

Carlos A. Coello Coello*

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

Abstract. This tutorial will review some of the basic concepts related to evolutionary multiobjective optimization (i.e., the use of evolutionary algorithms to handle more than one objective function at a time). The most commonly used evolutionary multiobjective optimization techniques will be described and criticized, including some of their applications. Theory, test functions and metrics will be also discussed. Finally, we will provide some possible paths of future research in this area.

1 Introduction

Most real-world engineering optimization problems are multiobjective in nature, since they normally have several (possibly conflicting) objectives that must be satisfied at the same time. The notion of "optimum" has to be re-defined in this context and instead of aiming to find a single solution, we will try to produce a set of good compromises or "trade-offs" from which the decision maker will select one.

Over the years, the work of a considerable amount of operational researchers has produced an important number of techniques to deal with multiobjective optimization problems [46]. However, it was until relatively recently that researchers realized of the potential of evolutionary algorithms in this area.

The potential of evolutionary algorithms in multiobjective optimization was hinted by Rosenberg in the 1960s [52], but this research area, later called Evolutionary Multi-Objective Optimization (EMOO for short) remained unexplored for almost twenty five years. However, researchers from many different disciplines have shown an increasing interest in EMOO in recent years. The considerable amount of research related to EMOO currently reported in the literature (over 630 publications[1]) is a clear reflection of such interest.

* This work was done while the author was at the Laboratorio Nacional de Informática Avanzada, Rébsamen 80, Xalapa, Veracruz 91090, México.

[1] The author maintains a repository on Evolutionary Multiobjective Optimization at: http://www.lania.mx/~ccoello/EMOO/ with a mirror at http://www.jeo.org/emo/

This paper will provide a short tutorial on EMOO, including a review of the main existing approaches (a description of the technique, together with its advantages and disadvantages and some of its applications) and of the most significant research done in theory, test functions and metrics. We will finish with a short review of two promising areas of future research.

2 Basic Definitions

Multiobjective optimization (also called multicriteria optimization, multiperformance or vector optimization) can be defined as the problem of finding [49]:

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

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

the p equality constraints

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

and optimizes the vector function

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

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ 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.

It is rarely the case that there is a single point that simultaneously optimizes all the objective functions. Therefore, we normally look for "trade-offs", rather than single solutions when dealing with multiobjective optimization problems. The notion of "optimum" is therefore, different. The most commonly adopted notion of optimality is that originally proposed by Francis Ysidro Edgeworth [22], and later generalized by Vilfredo Pareto [50]. Although some authors call *Edgeworth-Pareto optimum* to this notion (see for example Stadler [61]), we will use the most commonly accepted term: *Pareto optimum*.

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

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

2.1 An Example

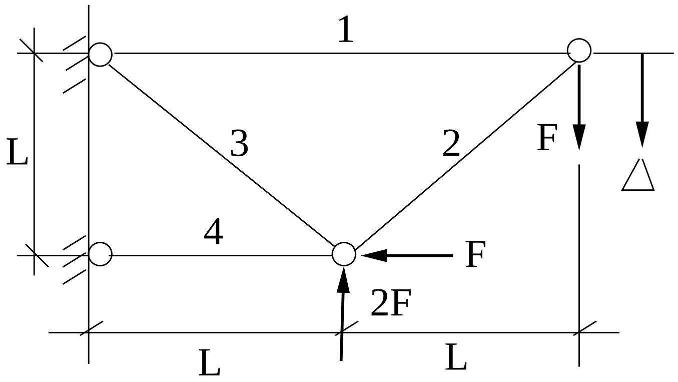

Fig. 1. A four-bar plane truss.

Let us analyze a simple example of a multiobjective optimization problem, that has been studied by Stadler & Dauer [62]. We want to design the four-bar plane truss shown in Figure 1. We will consider two objective functions: minimize the volume of the truss (f_1) and minimize its joint displacement Δ (f_2). The mathematical definition of the problem is:

$$\text{Minimize} \begin{cases} f_1(\boldsymbol{x}) = L\left(2x_1 + \sqrt{2}x_2 + \sqrt{x_3} + x_4\right) \\ f_2(\boldsymbol{x}) = \frac{FL}{E}\left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}\right) \end{cases} \tag{4}$$

such that:

$$\begin{array}{l} (F/\sigma) \leq x_1 \leq 3(F/\sigma) \\ \sqrt{2}(F/\sigma) \leq x_2 \leq 3(F/\sigma) \\ \sqrt{2}(F/\sigma) \leq x_3 \leq 3(F/\sigma) \\ (F/\sigma) \leq x_4 \leq 3(F/\sigma) \end{array} \tag{5}$$

where $F = 10$ kN, $E = 2 \times 10^5$ kN/cm^2, $L = 200$ cm, $\sigma = 10$ kN/cm^2.

The global Pareto front of this problem can be obtained by enumeration. The process consists of iterating on the four decision variables (with a reasonable granularity) to get a set of points representing the search space. Then, we apply the concept of Pareto optimality previously defined to the points generated. The result of this procedure, plotted on objective function space is shown in Figure 2. This is the true (or global) Pareto front of the problem.

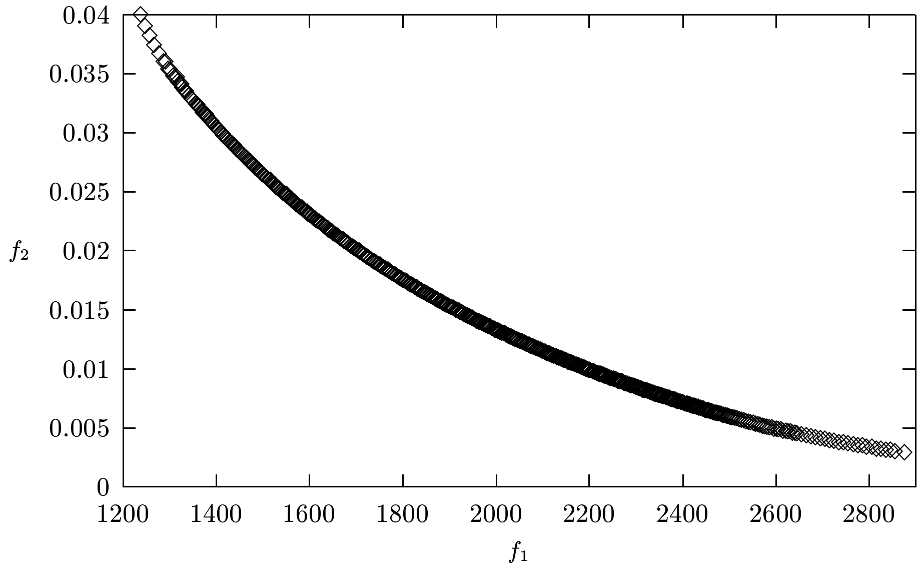

Fig. 2. True Pareto front of the four-bar plane truss problem.

3 Why Evolutionary Algorithms?

The first implementation of an EMOO approach was Schaffer's *Vector Evaluation Genetic Algorithm* (VEGA), which was introduced in the mid-1980s, mainly intended for solving problems in machine learning [57–59].

Schaffer's work was presented at the *First International Conference on Genetic Algorithms* in 1985 [58]. Interestingly, his simple unconstrained two-objective functions became the usual test suite to validate most of the evolutionary multiobjective optimization techniques developed during the following years [60, 38].

Evolutionary algorithms seem particularly suitable to solve multiobjective optimization problems, because they deal simultaneously with a set of possible solutions (the so-called population). This allows us to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques [5]. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous or concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques.

4 Reviewing EMOO Approaches

There are several detailed surveys of EMOO reported in the literature [5, 27, 64] and this tutorial does not intend to produce a new one. Therefore, we will limit ourselves to a short discussion on the most popular EMOO techniques currently in use, including two recent approaches that look very promising.

4.1 Aggregating functions

A genetic algorithm relies on a scalar fitness function to guide the search. Therefore, the most intuitive approach to deal with multiple objectives would be to combine them into a single function. The approach of combining objectives into a single (scalar) 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.

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

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

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

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

Since the results of solving an optimization model using (6) can vary significantly as the weighting coefficients change, and since very little is usually known about how to choose these coefficients, a necessary approach is to solve the same problem for many different values of w_i.

4.1.1 Advantages and Disadvantages This approach does not require any changes to the basic mechanism of a genetic algorithm and it is therefore very simple, easy to implement and efficient. The approach can work properly in simple multiobjective optimization problems with few objective functions and convex search spaces. One obvious problem of this approach is that it may be difficult to generate a set of weights that properly scales the objectives when little is known about the problem. However, its most serious drawback is that it cannot generate proper members of the Pareto optimal set when the Pareto front is concave regardless of the weights used [13].

4.1.2 Sample Applications

- Truck packing problems [30].
- Real-time scheduling [47].
- Structural synthesis of cell-based VLSI circuits [1].

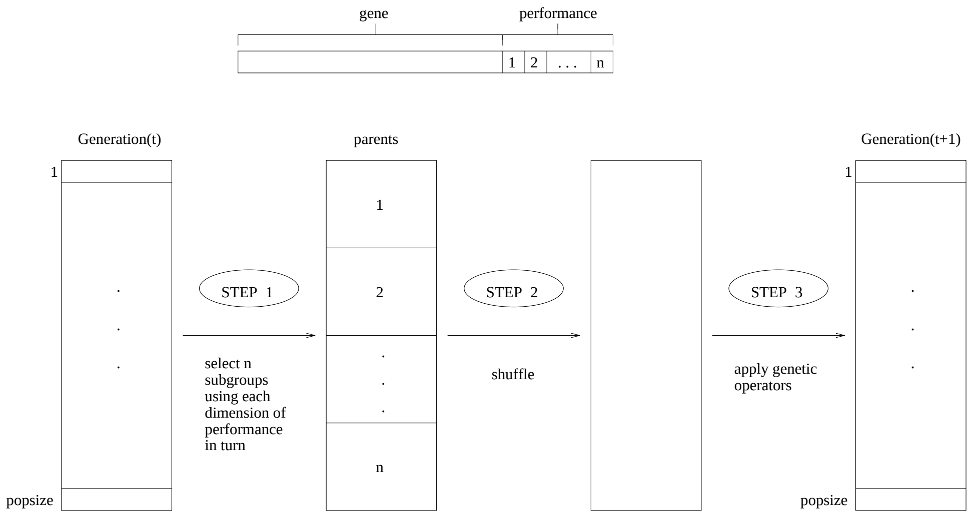

Fig. 3. Schematic of VEGA selection.

4.2 VEGA

Schaffer [58] proposed an approach that he called the *Vector Evaluated Genetic Algorithm* (VEGA), and that differed of the simple genetic algorithm (GA) 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 and a population size of M, k sub-populations of size M/k each would be generated. These sub-populations would be shuffled together to obtain a new population of size M, on which the GA would apply the crossover and mutation operators in the usual way. This process is illustrated in Figure 3.

The solutions generated by VEGA are locally nondominated, but not necessarily globally nondominated. VEGA presents the so-called "speciation" problem (i.e., we could have the evolution of "species" within the population which excel on different objectives). This problem arises because this technique selects individuals who excel in one objective, without looking at the others. The potential danger doing that is that we could have individuals with what Schaffer [58] called “middling” performance[2] 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 compromise solutions.

4.2.1 Advantages and Disadvantages

Since only the selection mechanism of the GA needs to be modified, the approach is easy to implement and it is quite efficient. However, the “middling” problem prevents the technique from finding the compromise solutions that we normally aim to produce. In fact, if proportional selection is used with VEGA (as Schaffer did), the shuffling and merging of all the sub-populations corresponds to averaging the fitness components associated with each of the objectives [51]. In other words, under these conditions, VEGA behaves as an aggregating approach and therefore, it is subject to the same problems of such techniques.

4.2.2 Sample Applications

- Optimal location of a network of groundwater monitoring wells [4].
- Combinational circuit design [8].
- Design multiplierless IIR filters [71].

4.3 MOGA

Fonseca and Fleming [25] proposed the *Multi-Objective Genetic Algorithm* (MOGA). The approach consists of a scheme in which the rank of a certain individual corresponds to the number of individuals in the current population by which it is dominated. All nondominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface.

Fitness assignment is performed in the following way [25]:

1. Sort population according to rank.
2. Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $n \leq M$) in the way proposed by Goldberg [29] (the so-called Pareto ranking assignment process), 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.

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

Since the use of a blocked fitness assignment scheme as the one indicated before is likely to produce a large selection pressure that might produce premature convergence [29], the authors proposed the use of a niche-formation method to distribute the population over the Pareto-optimal region [20]. Sharing is performed on the objective function values, and the authors provided some guidelines to compute the corresponding niche sizes. MOGA also uses mating restrictions.

4.3.1 Advantages and Disadvantages

The main strengths of MOGA is that is efficient and relatively easy to implement [11]. Its main weakness is that, as with all the other Pareto ranking techniques[3], its performance is highly dependent on an appropriate selection of the sharing factor.

MOGA has been a very popular EMOO technique (particularly within the control community), and it normally exhibits a very good overall performance [11].

4.3.2 Some Applications

- Fault diagnosis [45].
- Control system design [3, 69, 21].
- Wing planform design [48].
- Design of multilayer microwave absorbers [68].

4.4 NSGA

The *Nondominated Sorting Genetic Algorithm* (NSGA) was proposed by Srinivas and Deb [60], and is based on several layers of classifications of the individuals. Before selection is performed (stochastic remainder proportionate selection was used), the population is ranked on the basis of domination (using Pareto ranking): all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size). To maintain the diversity of the population, these classified individuals are shared (in decision variable space) with their dummy fitness values. Then this group of classified individuals is removed from the population and another layer of nondominated individuals is considered (i.e., the remainder of the population is re-classified). The process continues until all individuals in the population are classified. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows us to search for nondominated regions, and results in convergence of the population toward such regions. Sharing, on its part, helps to distribute the population over this region. Figure 4 (taken from Srinivas and Deb [60]) shows the general flow chart of this approach.

[3] The use of a ranking scheme based on the concept of Pareto optimality was originally proposed by Goldberg [29].

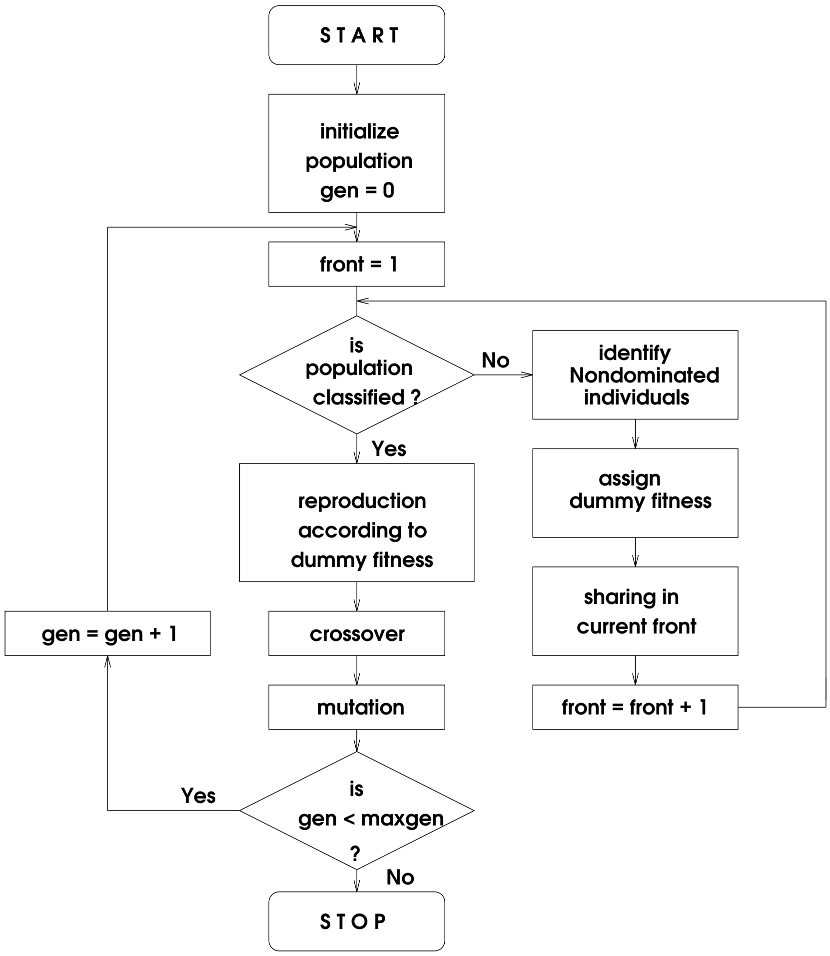

Fig. 4. Flowchart of the Nondominated Sorting Genetic Algorithm (NSGA).

4.4.1 Advantages and Disadvantages

Some researchers have reported that NSGA has a lower overall performance than MOGA (both computationally and in terms of quality of the Pareto fronts produced), and it seems to be also more sensitive to the value of the sharing factor than MOGA [11]. However, Deb et al. [18, 19] have recently proposed a new version of this algorithm, called NSGA-II, which is more efficient (computationally speaking), uses elitism and a crowded comparison operator that keeps diversity without specifying any additional parameters. The new approach has not been extensively tested yet, but it certainly looks promising.

4.4.2 Sample Applications

- Airfoil shape optimization [43].
- Scheduling [2].
- Minimum spanning tree [73].

4.5 NPGA

Horn et al. [38] proposed the *Niched Pareto Genetic Algorithm*, which uses a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals (as normally done with traditional GAs), a higher number of individuals is involved in the competition (typically around 10% of the population size). When both competitors are either dominated or nondominated (i.e., when there is a tie), the result of the tournament is decided through fitness sharing in the objective domain (a technique called *equivalent class sharing* was used in this case) [38].

The pseudocode for Pareto domination tournaments assuming that all of the objectives are to be maximized is presented below [37]. 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
```

This technique normally requires population sizes considerably larger than usual with other approaches, so that the noise of the selection method can be tolerated by the emerging niches in the population [26].

4.5.1 Advantages and Disadvantages

Since this approach does not apply Pareto ranking to the entire population, but only to a segment of it at each run, its main strength are that it is faster than MOGA and NSGA[4]. Furthermore, it also produces good nondominated fronts that can be kept for a large number of generations [11]. However, its main weakness is that besides requiring a sharing factor, this approach also requires an additional parameter: the size of the tournament.

4.5.2 Sample Applications

- Automatic derivation of qualitative descriptions of complex objects [55].
- Feature selection [24].
- Optimal well placement for groundwater containment monitoring [37, 38].
- Investigation of feasibility of full stern submarines [63].

4.6 Target Vector Approaches

Under this name we will consider approaches in which the decision maker has to assign targets or goals that wishes to achieve for each objective. The GA in this case, tries to minimize the difference between the current solution found and the vector of goals (different metrics can be used for that purpose). The most popular techniques included here are hybrids with: Goal Programming [16, 70], Goal Attainment [71, 72] and the min-max approach [32, 9].

4.6.1 Advantages and Disadvantages

The main strength of these methods is their efficiency (computationally speaking) because they do not require a Pareto ranking procedure. However, their main weakness is the definition of the

[4] Pareto ranking is $O(kM^2)$, where k is the number of objectives and M is the population size.

desired goals which requires some extra computational effort (normally, these goals are the optimum of each objective function, considered separately). Furthermore, these techniques will yield a nondominated solution only if the goals are chosen in the feasible domain, and such condition may certainly limit their applicability.

4.6.2 Some Applications

- Truss design [56, 7].
- Design of a robot arm [10].
- Synthesis of low-power operational amplifiers [72].

4.7 Recent approaches

Recently, several new EMOO approaches have been developed. We consider important to discuss briefly at least two of them: PAES and SPEA.

The *Pareto Archived Evolution Strategy* (PAES) was introduced by Knowles & Corne [42]. This approach is very simple: it uses a (1+1) evolution strategy (i.e., a single parent that generates a single offspring) together with a historical archive that records all the nondominated solutions previously found (such archive is used as a comparison set in a way analogous to the tournament competitors in the NPGA). PAES also uses a novel approach to keep diversity, which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives. A map of such grid is maintained, indicating the amount of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space). Furthermore, the procedure has a lower computational complexity than traditional niching methods. PAES has been used to solve the off-line routing problem [41] and the adaptive distributed database management problem [42].

The *Strength Pareto Evolutionary Algorithm* (SPEA) was introduced by Zitzler & Thiele [78]. This approach was conceived as a way of integrating different EMOO techniques. SPEA uses an archive containing nondominated solutions previously found (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set. For each individual in this external set, a strength value is computed. This strength is similar to the ranking value of MOGA, since it is proportional to the number of solutions to which a certain individual dominates. The fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. Additionally, a clustering technique is used to keep diversity. SPEA has been used to explore trade-offs of software implementations for DSP algorithms [76] and to solve 0/1 knapsack problems [78].

5 Theory

The most important theoretical work related to EMOO has concentrated on two main issues:

- Studies of convergence towards the Pareto optimum set [53, 54, 33, 34, 65].
- Ways to compute appropriate sharing factors (or niche sizes) [36, 35, 25].

Obviously, a lot of work remains to be done. It would be very interesting to study, for example, the structure of fitness landscapes in multiobjective optimization problems [40, 44]. Such study could provide some insights regarding the sort of problems that are particularly difficult for an evolutionary algorithm and could also provide clues regarding the design of more powerful EMOO techniques.

Also, there is a need for detailed studies of the different aspects involved in the parallelization of EMOO techniques (e.g., load balancing, impact on Pareto convergence, performance issues, etc.), including new algorithms that are more suitable for parallelization than those currently in use.

6 Test Functions

The design of test functions that are appropriate to evaluate EMOO approaches was disregarded in most of the early research in this area. However, in recent years, there have been several interesting proposals. Deb [14, 15] proposed ways to create controllable test problems for evolutionary multiobjective optimization techniques using single-objective optimization problems as a basis. He proposed to transform deceptive and massively multimodal problems into very difficult multiobjective optimization problems. More recently, his proposal was extended to constrained multiobjective optimization problems [17] (in most of the early papers on EMOO techniques, only unconstrained test functions were used).

Van Veldhuizen and Lamont [66, 67] have also proposed some guidelines to design a test function suite for evolutionary multiobjective optimization techniques, and have included in a technical report some sample test problems (mainly combinatorial optimization problems) [66]. In this regard, the literature on multiobjective combinatorial optimization can be quite useful [23]. The benchmarks available for problems like the multiobjective 0/1 knapsack can be used to validate EMOO approaches. Such idea has been explored by a few EMOO researchers (for example [78, 39]), but more work in this direction is still necessary.

7 Metrics

Assuming that we have a set of test functions available, the next issue is how to compare different EMOO techniques. The design of metrics has been studied recently in the literature. The main proposals so far are the following:

- Van Veldhuizen and Lamont [65] proposed the so-called *generational distance*, which is a measure of how close our current Pareto front is from the true Pareto front (assuming we know where it lies).
- Srinivas and Deb [60] proposed the use of an statistical measure (the chi-square distribution) to estimate the spread of the population on the Pareto front with respect to the sharing factor used.
- Zitzler and Thiele [77] proposed two measures: the first concerns the size of the objective value space which is covered by a set of nondominated solutions and the second compares directly two sets of nondominated solutions, using as a metric the fraction of the Pareto front covered by each of them. Several other similar metrics have been also suggested recently by Zitzler et al. [75].
- Fonseca and Fleming [28] 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 Pareto fronts) produced by the EMOO technique under study, and/or compare it against other similar techniques.

There are few comparative studies of EMOO techniques where these metrics have been used and more comprehensive comparisons are still lacking in the literature [75, 64, 74]. Also, it is important to consider that most of the previously mentioned metrics assume that the user can generate the global Pareto front of the problem under study (using, for example, an enumerative approach), and that will not be possible in most real-world applications.

8 Promising areas of future research

There are at least two areas of future research that deserve more attention in the next few years:

- **Incorporation of preferences**: We should not ignore the fact that the solution of a multiobjective optimization problem really involves three stages: measurement, search, and decision making. Most EMOO research tends to concentrate on issues related to the search of nondominated vectors. However, these nondominated vectors do not provide any insight into the process of decision making itself (the decision maker still has to choose manually one of the several alternatives produced), since they are really a useful generalization of a utility function under the conditions of minimum information (i.e., all attributes are considered as having equal importance; in other words, the decision maker does not express any preferences of the attributes). Thus, the issue is how to incorporate the decision maker's preferences into an EMOO approach as to guide the search only to the regions of main interest. There are a few recent proposals in this area [12, 6], but more research is still needed. Issues such as scalability of the preferences' handling mechanism and capability of the approach to incorporate preferences from several decision makers deserve special attention.

– **Emphasis on efficiency**: Efficiency has been emphasized in EMOO research until recently, mainly regarding the number of comparisons performed for ranking the population [18], ways to maintain diversity [42], and procedures to reduce the computational cost involved in evaluating several (expensive) objective functions [21]. However, more work is still needed. For example, EMOO researchers have paid little attention to the use of efficient data structures. In contrast, operational researchers have used, for example, domination-free quad trees where a nondominated vector can be retrieved from the tree very efficiently. Checking if a new vector is dominated by the vectors in one of these trees can also be done very efficiently [31]. It is therefore necessary to pay more attention to efficiency issues in the design of new EMOO approaches, to make them more suitable for real-world applications.

9 Conclusions

This paper has attempted to provide a short tutorial of evolutionary multiobjective optimization. Our discussion has covered the main EMOO approaches currently in use, their advantages and disadvantages, and some of their applications reported in the literature.

We have also discussed briefly the theoretical work done in this area, as well as some of the research that has attempted to produce benchmarks that are appropriate to validate EMOO approaches. We also discussed another problem related to this last issue: the definition of appropriate metrics that allow us to compare several EMOO techniques. Such metrics should evaluate the capability of an EMOO approach to produce a sufficient amount of elements of the Pareto optimal set of the problem as well as to spread them appropriately.

Our discussion finishes with a short description of two possible areas of future research in EMOO: mechanisms that facilitate the incorporation of user's preferences and the search for efficient procedures and algorithms for evolutionary multiobjective optimization and to keep diversity.

Acknowledgements

The author gratefully acknowledges support from CONACyT through project 34201-A.

References

1. T. Arslan, D. H. Horrocks, and E. Ozdemir. Structural Synthesis of Cell-based VLSI Circuits using a Multi-Objective Genetic Algorithm. *IEE Electronic Letters*, 32(7):651–652, March 1996.
2. Tapan P. Bagchi. *Multiobjective Scheduling by Genetic Algorithms.* Kluwer Academic Publishers, Boston, 1999.

3. A. J. Chipperfield and P. J. Fleming. Gas Turbine Engine Controller Design using Multiobjective Genetic Algorithms. In A. M. S. Zalzala, editor, *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems : Innovations and Applications, GALESIA'95*, pages 214–219, Halifax Hall, University of Sheffield, UK, September 1995. IEEE.
4. Scott E. Cieniawski, J. W. Eheart, and S. Ranjithan. Using Genetic Algorithms to Solve a Multiobjective Groundwater Monitoring Problem. *Water Resources Research*, 31(2):399–409, February 1995.
5. Carlos A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, August 1999.
6. Carlos A. Coello Coello. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *2000 Congress on Evolutionary Computation*, volume 1, pages 30–37, Piscataway, New Jersey, July 2000. IEEE Service Center.
7. Carlos A. Coello Coello. Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308, 2000.
8. Carlos A. Coello Coello, Arturo Hernández Aguirre, and Bill P. Buckles. Evolutionary Multiobjective Design of Combinational Logic Circuits. In Jason Lohn, Adrian Stoica, Didier Keymeulen, and Silvano Colombano, editors, *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pages 161–170, Los Alamitos, California, July 2000. IEEE Computer Society.
9. Carlos A. Coello Coello and Alan D. Christiansen. Two New GA-based methods for multiobjective optimization. *Civil Engineering Systems*, 15(3):207–243, 1998.
10. Carlos A. Coello Coello, Alan D. Christiansen, and Arturo Hernández Aguirre. Using a New GA-Based Multiobjective Optimization Technique for the Design of Robot Arms. *Robotica*, 16(4):401–414, July–August 1998.
11. Carlos Artemio Coello Coello. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*. PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, April 1996.
12. Dragan Cvetković. *Evolutionary Multi–Objective Decision Support Systems for Conceptual Design*. PhD thesis, School of Computing, University of Plymouth, Plymouth, UK, November 2000.
13. Indraneel Das and John Dennis. A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Structural Optimization*, 14(1):63–69, 1997.
14. Kalyanmoy Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, 1998.
15. Kalyanmoy Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, chapter 8, pages 135–161. John Wiley & Sons, Ltd, Chichester, UK, 1999.
16. Kalyanmoy Deb. Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms. In *1999 Congress on Evolutionary Computation*, pages 77–84, Washington, D.C., July 1999. IEEE Service Center.
17. Kalyanmoy Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 2000. (in Press).
18. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

19. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858. Springer, 2000.
20. Kalyanmoy Deb and David E. Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.
21. N.M. Duarte, A. E. Ruano, C.M. Fonseca, and P.J. Fleming. Accelerating Multi-Objective Control System Design Using a Neuro-Genetic Approach. In *2000 Congress on Evolutionary Computation*, volume 1, pages 392–397, Piscataway, New Jersey, July 2000. IEEE Service Center.
22. F. Y. Edgeworth. *Mathematical Physics.* P. Keagan, London, England, 1881.
23. Matthias Ehrgott and Xavier Gandibleux. An Annotated Bibliography of Multiobjective Combinatorial Optimization. Technical Report 62/2000, Fachbereich Mathematik, Universitat Kaiserslautern, Kaiserslautern, Germany, 2000.
24. C. Emmanouilidis, A. Hunter, and J. MacIntyre. A Multiobjective Evolutionary Setting for Feature Selection and a Commonality-Based Crossover Operator. In *2000 Congress on Evolutionary Computation*, volume 1, pages 309–316, Piscataway, New Jersey, July 2000. IEEE Service Center.
25. Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
26. Carlos M. Fonseca and Peter J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. Technical report, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U. K., 1994.
27. Carlos M. Fonseca and Peter J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995.
28. Carlos M. Fonseca and Peter J. Fleming. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 584–593, Berlin, Germany, September 1996. Springer-Verlag.
29. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
30. Pierre Grignon, J. Wodziack, and G. M. Fadel. Bi-Objective optimization of components packing using a genetic algorithm. In *NASA/AIAA/ISSMO Multidisciplinary Design and Optimization Conference*, pages 352–362, Seattle, Washington, September 1996. AIAA-96-4022-CP.
31. W. Habenicht. Quad trees, A data structure for discrete vector optimization problems. In *Lecture notes in economics and mathematical systems*, volume 209, pages 136–145, 1982.
32. P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.
33. T. Hanne. On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, 117(3):553–564, September 2000.

34. Thomas Hanne. Global Multiobjective Optimization Using Evolutionary Algorithms. *Journal of Heuristics*, 6(3):347–360, August 2000.
35. Jeffrey Horn. Multicriterion Decision Making. In Thomas Bäck, David Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, volume 1, pages F1.9:1 – F1.9:15. IOP Publishing Ltd. and Oxford University Press, 1997.
36. Jeffrey Horn. *The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Populations.* PhD thesis, University of Illinois at Urbana Champaign, Urbana, Illinois, 1997.
37. Jeffrey Horn and Nicholas Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
38. Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.
39. Andrzej Jaszkiewicz. On the performance of multiple objective genetic local search on the 0/1 knapsack problem. a comparative experiment. Technical Report RA-002/2000, Institute of Computing Science, Poznan University of Technology, Poznań, Poland, July 2000.
40. S. Kaufmann. Adaptation on rugged fitness landscapes. In D. Stein, editor, *Lectures in the Sciences of Complexity*, pages 527–618. Addison-Wesley, Reading, Massachusetts, 1989.
41. Joshua D. Knowles and David W. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In *1999 Congress on Evolutionary Computation*, pages 98–105, Washington, D.C., July 1999. IEEE Service Center.
42. Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
43. R. Mäkinen, P. Neittaanmäki, J. Périaux, and J. Toivanen. A genetic Algorithm for Multiobjective Design Optimization in Aerodynamics and Electromagnetics. In K. D. Papailiou et al., editor, *Computational Fluid Dynamics '98, Proceedings of the ECCOMAS 98 Conference*, volume 2, pages 418–422, Athens, Greece, September 1998. Wiley.
44. Bernard Manderick, Mark de Weger, and Piet Spiessens. The Genetic Algorithm and the Structure of the Fitness Landscape. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150, San Mateo, California, 1991. Morgan Kaufmann.
45. Teodor Marcu. A multiobjective evolutionary approach to pattern recognition for robust diagnosis of process faults. In R. J. Patton and J. Chen, editors, *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes: SAFEPROCESS'97*, pages 1183–1188, Kington Upon Hull, United Kingdom, August 1997.
46. Kaisa M. Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, Boston, Massachusetts, 1998.
47. David Montana, Marshall Brinn, Sean Moore, and Garrett Bidwell. Genetic Algorithms for Complex, Real-Time Scheduling. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2213–2218, La Jolla, California, October 1998. IEEE.

48. S. Obayashi, S. Takahashi, and Y. Takeguchi. Niching and Elitist Models for MOGAs. In A. E. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving From Nature — PPSN V*, pages 260–269, Amsterdam, Holland, 1998. Springer-Verlag.
49. Andrzej Osyczka. Multicriteria optimization for engineering design. In John S. Gero, editor, *Design Optimization*, pages 193–227. Academic Press, 1985.
50. Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.
51. Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197, George Mason University, 1989. Morgan Kaufmann Publishers.
52. R. S. Rosenberg. *Simulation of genetic populations with biochemical properties.* PhD thesis, University of Michigan, Ann Harbor, Michigan, 1967.
53. Günter Rudolph. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 511–516, Piscataway, New Jersey, 1998. IEEE Press.
54. Günter Rudolph and Alexandru Agapie. Convergence Properties of Some Multi-Objective Evolutionary Algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, volume 2, pages 1010–1016, Piscataway, New Jersey, July 2000. IEEE Press.
55. Enrique H. Ruspini and Igor S. Zwir. Automated Qualitative Description of Measurements. In *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference*, Venice, Italy, 1999.
56. Eric Sandgren. Multicriteria design optimization by goal programming. In Hojjat Adeli, editor, *Advances in Design Optimization*, chapter 23, pages 225–265. Chapman & Hall, London, 1994.
57. J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* PhD thesis, Vanderbilt University, 1984.
58. J. David Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.
59. J. David Schaffer and John J. Grefenstette. Multiobjective Learning via Genetic Algorithms. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 593–595, Los Angeles, California, 1985. AAAI.
60. N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.
61. W. Stadler. Fundamentals of multicriteria optimization. In W. Stadler, editor, *Multicriteria Optimization in Engineering and the Sciences*, pages 1–25. Plenum Press, New York, 1988.
62. W. Stadler and J. Dauer. Multicriteria optimization in engineering: A tutorial and survey. In *Structural Optimization: Status and Future*, pages 209–249. American Institute of Aeronautics and Astronautics, 1992.
63. Mark W. Thomas. *A Pareto Frontier for Full Stern Submarines via Genetic Algorithm.* PhD thesis, Ocean Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, june 1998.
64. David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.* PhD thesis, Department of Electrical and Com-

puter Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.
65. David A. Van Veldhuizen and Gary B. Lamont. Evolutionary Computation and Convergence to a Pareto Front. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, Stanford University, California, July 1998. Stanford University Bookstore.
66. David A. Van Veldhuizen and Gary B. Lamont. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.
67. David A. Van Veldhuizen and Gary B. Lamont. Multiobjective Evolutionary Algorithm Test Suites. In Janice Carroll, Hisham Haddad, Dave Oppenheim, Barrett Bryant, and Gary B. Lamont, editors, *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351–357, San Antonio, Texas, 1999. ACM.
68. D. S. Weile, E. Michielssen, and D. E. Goldberg. Genetic algorithm design of pareto optimal broad-band microwave absorbers. Technical Report CCEM-4-96, Electrical and Computer Engineering Department, Center for Computational Electromagnetics, University of Illinois at Urbana-Champaign, May 1996.
69. J. F. Whidborne, D.-W. Gu, and I. Postlethwaite. Algorithms for the method of inequalities – a comparative study. In *Procedings of the 1995 American Control Conference*, pages 3393–3397, Seattle, Washington, 1995.
70. P. B. Wienke, C. Lucasius, and G. Kateman. Multicriteria target optimization of analytical procedures using a genetic algorithm. *Analytical Chimica Acta*, 265(2):211–225, 1992.
71. P. B. Wilson and M. D. Macleod. Low implementation cost IIR digital filter design using genetic algorithms. In *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, pages 4/1–4/8, Chelmsford, U.K., 1993.
72. R. S. Zebulum, M. A. Pacheco, and M. Vellasco. A multi-objective optimisation methodology applied to the synthesis of low-power operational amplifiers. In Ivan Jorge Cheuri and Carlos Alberto dos Reis Filho, editors, *Proceedings of the XIII International Conference in Microelectronics and Packaging*, volume 1, pages 264–271, Curitiba, Brazil, August 1998.
73. Gengui Zhou and Mitsuo Gen. Genetic Algorithm Approach on Multi-Criteria Minimum Spanning Tree Problem. *European Journal of Operational Research*, 114(1), April 1999.
74. Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.
75. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
76. Eckart Zitzler, Jürgen Teich, and Shuvra S. Bhattacharyya. Multidimensional Exploration of Software Implementations for DSP Algorithms. *VLSI Signal Processing Systems*, 1999. (To appear).
77. Eckart Zitzler and Lothar Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998.
78. Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.