

# Current and Future Research Trends in Evolutionary Multiobjective Optimization

Carlos A. Coello Coello, Gregorio Toscano Pulido and Efrén Mezura Montes

CINVESTAV-IPN  
Evolutionary Computation Group  
Departamento de Ingeniería Eléctrica  
Sección de Computación  
Av. Instituto Politécnico Nacional No. 2508  
Col. San Pedro Zacatenco  
México, D.F. 07300, MEXICO  
ccoello@cs.cinvestav.mx  
{gtoscano, emezura}@computacion.cs.cinvestav.mx

**Abstract.** In this chapter we present a brief analysis of the current research performed on evolutionary multiobjective optimization. After analyzing first and second generation multiobjective evolutionary algorithms, we address two important issues: the role of elitism in evolutionary multiobjective optimization and the way in which concepts from multiobjective optimization can be applied to constraint-handling techniques. We conclude with a discussion of some of the most promising research trends in the years to come.

## 1 Introduction

Evolutionary algorithms have become an increasingly popular design and optimization tool in the last few years, with a constantly growing development of new algorithms and applications [1]. Despite this considerably large volume of research, new areas remain to be explored with sufficient depth. One of them is the use of evolutionary algorithms to solve multiobjective optimization problems.

The first implementation of a multi-objective evolutionary algorithm (MOEA) dates back to the mid-1980s [45, 46]. Since then, a considerable amount of research has been done in this area, now known as evolutionary multi-objective optimization (EMO for short). The growing importance of this field is reflected by a significant increment (mainly during the last eight years) of technical papers in international conferences and peer-reviewed journals, books, special sessions in international conferences and interest groups on the Internet [13].<sup>1</sup>

Evolutionary algorithms seem also particularly desirable for solving multiobjective optimization problems because they deal simultaneously with a set of possible solutions (the so-called population) which allows us to find several members of the Pareto optimal

---

<sup>1</sup> The first author maintains an EMO repository with over 900 bibliographical entries at: <http://delta.cs.cinvestav.mx/~ccoello/EMOO>, with mirrors at <http://www.lania.mx/~ccoello/EMOO/> and <http://www.jeo.org/emo/>

set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous and concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques [8].

This chapter deals with some of the current and future research trends in evolutionary multiobjective optimization. The perspective adopted is derived from our own research experience in the area and therefore the bias towards certain particular topics of interest. The chapter is organized as follows. Section 2 presents some basic concepts used in multiobjective optimization. Section 3 briefly describes the origins of evolutionary multiobjective optimization. Section 4 introduces the so-called first generation multiobjective evolutionary algorithms. Second generation multiobjective evolutionary algorithms are discussed in Section 5, emphasizing the role of elitism in evolutionary multiobjective optimization. Section 6 discusses ways in which multiobjective optimization concepts have been and could be incorporated into constraint-handling techniques (both for single and for multiobjective optimization). Finally, Section 7 discusses some of the research trends that are likely to be predominant in the next few years.

## 2 Basic Concepts

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

$$\text{minimize } [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})] \quad (1)$$

subject to the  $m$  inequality constraints:

$$g_i(\mathbf{x}) \geq 0 \quad i = 1, 2, \dots, m \quad (2)$$

and the  $p$  equality constraints:

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, p \quad (3)$$

where  $k$  is the number of objective functions  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ . We call  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  the vector of decision variables. We wish to determine from among the set  $\mathcal{F}$  of all vectors which satisfy (2) and (3) the particular set of values  $x_1^*, x_2^*, \dots, x_n^*$  which yield the optimum values of all the objective functions.

### 2.1 Pareto optimality

It is rarely the case that there is a single point that simultaneously optimizes all the objective functions. Therefore, we normally look for “trade-offs”, rather than single solutions when dealing with multiobjective optimization problems. The notion of “optimality” is therefore, different in this case. The most commonly adopted notion of optimality is that originally proposed by Francis Ysidro Edgeworth [21] and later generalized by

Vilfredo Pareto [39]. Although some authors call this notion *Edgeworth-Pareto optimality* (see for example [49]), we will use the most commonly accepted term: *Pareto optimality*.

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

In words, this definition says that  $\mathbf{x}^*$  is Pareto optimal if there exists no feasible vector of decision variables  $\mathbf{x} \in \mathcal{F}$  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  $\mathbf{x}^*$  corresponding to the solutions included in the Pareto optimal set are called *nondominated*. The image of the Pareto optimal set under the objective functions is called *Pareto front*.

### 3 How it all started

The potential of evolutionary algorithms for solving multiobjective optimization problems was hinted as early as the late 1960s by Rosenberg in his PhD thesis [42]. Rosenberg’s study contained a suggestion that would have led to multiobjective 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.

The first actual implementation of what it is now called a multi-objective evolutionary algorithm (or MOEA, for short) was Schaffer’s *Vector Evaluation Genetic Algorithm* (VEGA), which was introduced in the mid-1980s, mainly aimed for solving problems in machine learning [45–47].

VEGA basically consisted of a simple genetic algorithm (GA) with a modified selection mechanism. At each generation, a number of sub-populations were generated by performing proportional selection according to each objective function in turn. Thus, for a problem with  $k$  objectives,  $k$  sub-populations of size  $N/k$  each would be generated (assuming a total population size of  $N$ ). These sub-populations would then be shuffled together to obtain a new population of size  $N$ , on which the GA would apply the crossover and mutation operators in the usual way. Schaffer realized that the solutions generated by his system were nondominated in a local sense, because their nondominance was limited to the current population, which was obviously not appropriate. Also, he noted a problem that in genetics is known as “speciation” (i.e., we could have the evolution of “species” within the population which excel on different aspects of performance). This problem arises because this technique selects individuals who excel in one dimension of performance, without looking at the other dimensions. The potential danger doing that is that we could have individuals with what Schaffer called “middling” performance<sup>2</sup> in all dimensions, which could be very useful for compromise solutions,

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

but which will not survive under this selection scheme, since they are not in the extreme for any dimension of performance (i.e., they do not produce the best value for any objective function, but only moderately good values for all of them). Speciation is undesirable because it is opposed to our goal of finding Pareto optimal solutions. Although VEGA's speciation can be dealt with using heuristics or other additional mechanisms, it remained as the main drawback of VEGA.

From the second half of the 1980s up to the first half of the 1990s, few other researchers developed MOEAs. Most of the work reported back then involves rather simple evolutionary algorithms that use an aggregating function (linear in most cases) [33, 54], lexicographic ordering [24], and target-vector approaches [28]. All of these approaches were strongly influenced by the work done in the operations research community and in most cases did not require any major modifications to the evolutionary algorithm adopted.

The algorithms proposed in this initial period are rarely referenced in the current literature except for VEGA (which is still used by some researchers). However, the period is of great importance because it provided the first insights into the possibility of using evolutionary algorithms for multiobjective optimization. The fact that only relatively naive approaches were developed during this stage is natural considering that these were the initial attempts to develop multiobjective extensions of an evolutionary algorithm. Such approaches kept most of the original evolutionary algorithm structure intact (only the fitness function was modified in most cases) to avoid any complex additional coding. The emphasis in incorporating the concept of Pareto dominance into the search mechanism of an evolutionary algorithm would come later.

## 4 MOEAs: First Generation

The major step towards the first generation of MOEAs was given by David Goldberg on pages 199 to 201 of his famous book on genetic algorithms published in 1989 [25]. In his book, Goldberg analyzes VEGA and proposes a selection scheme based on the concept of Pareto optimality. Goldberg not only suggested what would become the standard first generation MOEA, but also indicated that stochastic noise would make such algorithm useless unless some special mechanism was adopted to block convergence. First generation MOEAs typically adopt niching or fitness sharing for that sake. The most representative algorithms from the first generation are the following:

1. **Nondominated Sorting Genetic Algorithm (NSGA)**: This algorithm was proposed by Srinivas and Deb [48]. The approach is based on several layers of classifications of the individuals as suggested by Goldberg [25]. Before selection is performed, the population is ranked on the basis of nondomination: all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population are classified.

Stochastic remainder proportionate selection is adopted for this technique. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows to search for nondominated regions, and results in convergence of the population toward such regions. Sharing, by its part, helps to distribute the population over this region (i.e., the Pareto front of the problem).

2. **Niched-Pareto Genetic Algorithm** (NPGA): Proposed by Horn et al. [32]. The NPGA uses tournament selection scheme based on Pareto dominance. The basic idea of the algorithm is the following: Two individuals are randomly chosen and compared against a subset from the entire population (typically, around 10% of the population). If one of them is dominated (by the individuals randomly chosen from the population) and the other is not, then the nondominated individual wins. When both competitors are either dominated or nondominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing [27].
3. **Multi-Objective Genetic Algorithm** (MOGA): Proposed by Fonseca and Fleming [23]. In MOGA, 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.  
The rank of an individual is given by [23]:

$$\text{rank}(x_i, t) = 1 + p_i^{(t)} \quad (4)$$

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 [23]:

- (a) Sort population according to rank.
- (b) Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank  $n \leq M$ ) in the way proposed by Goldberg (1989), according to some function, usually linear, but not necessarily.
- (c) Average the fitnesses of individuals with the same rank, so that all of them are sampled at the same rate. This procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used.

The main questions raised during the first generation were:

- Are aggregating functions (so common before and even during the golden years of Pareto ranking) really doomed to fail when the Pareto front is non-convex [16]? Are there ways to deal with this problem? Is it worth trying? Some recent work seems to indicate that aggregating functions are not death yet [35].
- Can we find ways to maintain diversity in the population without using niches (or fitness sharing), which requires a process  $O(M^2)$  where  $M$  refers to the population size?

- If assume that there is no way of reducing the  $O(kM^2)$  process required to perform Pareto ranking ( $k$  is the number of objectives and  $M$  is the population size), how can we design a more efficient MOEA?
- Do we have appropriate test functions and metrics to evaluate quantitatively an MOEA? Not many people worried about this issue until near the end of the first generation. During this first generation, practically all comparisons were done visually (plotting the Pareto fronts produced by different algorithms) or were not provided at all (only the results of the proposed method were reported).
- When will somebody develop theoretical foundations for MOEAs?

Summarizing, the first generation was characterized by the use of selection mechanisms based on Pareto ranking and fitness sharing was the most common approach adopted to maintain diversity. Much work remained to be done, but the first important steps towards a solid research area had been already taken.

## 5 MOEAs: Second Generation

The second generation of MOEAs was born with the introduction of the notion of elitism. In the context of multiobjective optimization, elitism usually (although not necessarily) refers to the use of an external population (also called secondary population) to retain the nondominated individuals. The use of this external file raises several questions:

- How does the external file interact with the main population?
- What do we do when the external file is full?
- Do we impose additional criteria to enter the file instead of just using Pareto dominance?

Elitism can also be introduced through the use of a  $(\mu + \lambda)$ -selection in which parents compete with their children and those which are nondominated (and possibly comply with some additional criterion such as providing a better distribution of solutions) are selected for the following generation.

The previous points bring us to analyze in more detail the true role of elitism in evolutionary multiobjective optimization. For that sake, we will review next the way in which some of the second-generation MOEAs implement elitism:

1. **Strength Pareto Evolutionary Algorithm (SPEA):** This algorithm was introduced by Zitzler and Thiele [57]. This approach was conceived as a way of integrating different MOEAs. 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. It should be obvious that the external nondominated set is in this case the elitist mechanism adopted.

In SPEA, the fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. Additionally, a clustering technique called “average linkage method” [37] is used to keep diversity.

2. **Strength Pareto Evolutionary Algorithm 2 (SPEA2)**: SPEA2 has three main differences with respect to its predecessor [56]: (1) it incorporates a fine-grained fitness assignment strategy which takes into account for each individual the number of individuals that dominate it and the number of individuals by which it is dominated; (2) it uses a nearest neighbor density estimation technique which guides the search more efficiently, and (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions. Therefore, in this case the elitist mechanism is just an improved version of the previous.
3. **Pareto Archived Evolution Strategy (PAES)**: This algorithm was introduced by Knowles and Corne [36]. PAES consists of a (1+1) evolution strategy (i.e., a single parent that generates a single offspring) in combination with a historical archive that records some of the nondominated solutions previously found. This archive is used as a reference set against which each mutated individual is being compared. Such a historical archive is the elitist mechanism adopted in PAES. However, an interesting aspect of this algorithm is the procedure used to maintain diversity which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its “coordinates” or “geographical location”). A map of such grid is maintained, indicating the number of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space).
4. **Nondominated Sorting Genetic Algorithm II (NSGA-II)**: Deb et al. [18–20] proposed a revised version of the NSGA [48], 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 NSGA-II does not use an external memory as the previous algorithms. Instead, the elitist mechanism consists of combining the best parents with the best offspring obtained (i.e., a  $(\mu + \lambda)$ -selection).
5. **Niched Pareto Genetic Algorithm 2 (NPGA 2)**: Erickson et al. [22] proposed a revised version of the NPGA [32] called the NPGA 2. This algorithm uses Pareto ranking but keeps tournament selection (solving ties through fitness sharing as in the original NPGA). In this case, no external memory is used and the elitist mechanism is similar to the one adopted by the NSGA-II. Niche counts in the NPGA 2 are calculated using individuals in the partially filled next generation, rather than using the current generation. This is called continuously updated fitness sharing, and was proposed by Oei et al. [38].

6. **Pareto Envelope-based Selection Algorithm (PESA):** This algorithm was proposed by Corne et al. [15]. This approach uses a small internal population and a larger external (or secondary) population. PESA uses the same hyper-grid division of phenotype (i.e., objective function) space adopted by PAES to maintain diversity. However, its selection mechanism is based on the crowding measure used by the hyper-grid previously mentioned. This same crowding measure is used to decide what solutions to introduce into the external population (i.e., the archive of non-dominated vectors found along the evolutionary process). Therefore, in PESA, the external memory plays a crucial role in the algorithm since it determines not only the diversity scheme, but also the selection performed by the method. There is also a revised version of this algorithm, called PESA-II [14]. This algorithm is identical to PESA, except for the fact that region-based selection is used in this case. In region-based selection, the unit of selection is a hyperbox rather than an individual. The procedure consists of selecting (using any of the traditional selection techniques [26]) a hyperbox and then randomly select an individual within such hyperbox. The main motivation of this approach is to reduce the computational costs associated with traditional MOEAs (i.e., those based on Pareto ranking). Again, the role of the external memory in this case is crucial to the performance of the algorithm.

7. **Micro Genetic Algorithm:** This approach was introduced by Coello Coello & Toscano Pulido [11, 12]. A micro-genetic algorithm is a GA with a small population and a reinitialization process. The way in which the micro-GA works is illustrated in Figure 1. First, a random population is generated. This random population feeds the population memory, which is divided in two parts: a replaceable and a non-replaceable portion. The non-replaceable portion of the population memory never changes during the entire run and is meant to provide the required diversity for the algorithm. In contrast, the replaceable portion experiences changes after each cycle of the micro-GA.

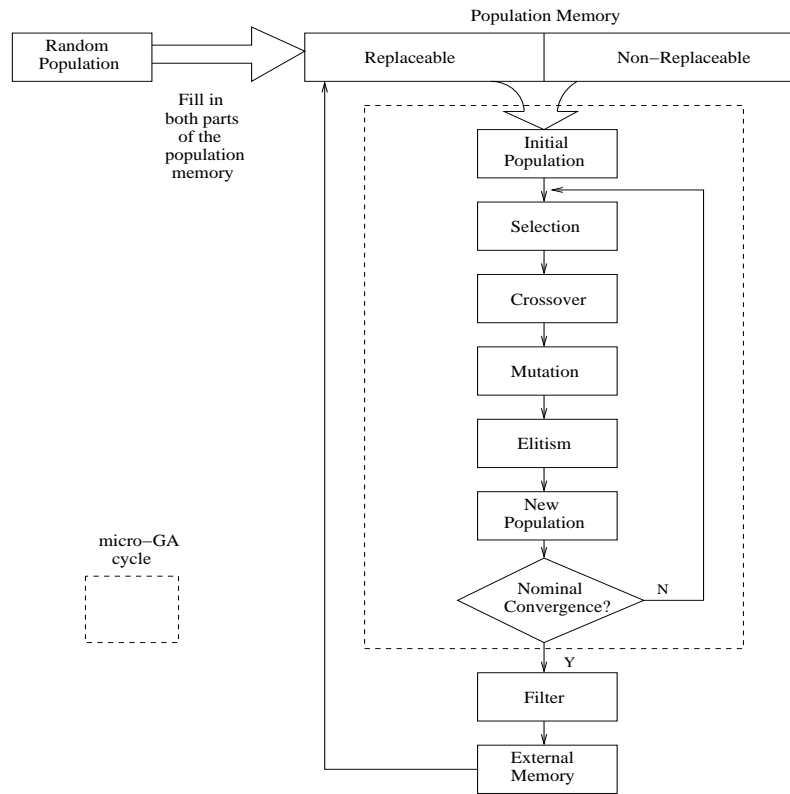
The population of the micro-GA at the beginning of each of its cycles is taken (with a certain probability) from both portions of the population memory so that there is a mixture of randomly generated individuals (non-replaceable portion) and evolved individuals (replaceable portion). During each cycle, the micro-GA undergoes conventional genetic operators. After the micro-GA finishes one cycle, two nondominated vectors are chosen<sup>3</sup> from the final population and they are compared with the contents of the external memory (this memory is initially empty). If either of them (or both) remains as nondominated after comparing it against the vectors in this external memory, then they are included there (i.e., in the external memory). This is the historical archive of nondominated vectors. All dominated vectors contained in the external memory are eliminated.

The micro-GA uses then three forms of elitism: (1) it retains nondominated solutions found within the internal cycle of the micro-GA, (2) it uses a replaceable memory whose contents is partially “refreshed” at certain intervals, and (3) it re-

---

<sup>3</sup> This is assuming that there are two or more nondominated vectors. If there is only one, then this vector is the only one selected.





**Fig. 1.** Diagram that illustrates the way in which the micro-GA for multiobjective optimization works [12].

places the population of the micro-GA by the nominal solutions produced (i.e., the best solutions found after a full internal cycle of the micro-GA). Therefore, the micro-GA is another example of how elitism can play a vital role to improve the performance of an evolutionary algorithm used for multiobjective optimization.

Second generation MOEAs can be characterized by an emphasis on efficiency and by the use of elitism (in the two main forms previously described). During the second generation, some important theoretical work also took place, mainly related to convergence [43, 44, 29, 30, 53]. Also, metrics and standard test functions were developed to validate new MOEAs [55, 52].

The main concerns during the second generation (which we are still living nowadays) are the following:

- Are our metrics reliable? What about our test functions? We have found out that developing good metrics is in itself a multiobjective optimization problem, too. In fact, it is ironic that nowadays we are going back to trusting more visual comparisons than metrics as during the first generation.
- Are we ready to tackle problems with more than two objective functions efficiently? Is Pareto ranking doomed to fail when dealing with too many objectives? If so, then what is the limit up to which Pareto ranking can be used to select individuals reliably?
- What are the most relevant theoretical aspects of evolutionary multiobjective optimization that are worth exploring in the short-term?

## 6 Relating constraint-handling and multiobjective optimization

Another research area within evolutionary multiobjective optimization that has not been explored in enough detail in the current literature is constraint-handling (particularly for single-objective optimization). We believe that it is important to study the relationship between constraint-handling and multiobjective optimization because of two main reasons: (1) constrained single-objective optimization problems can be re-stated as multi-objective optimization problems in a natural way, and (2) this sort of constrained single-objective optimization problems can be used to measure performance of MOEAs on a more quantitative basis than when using conventional multiobjective test functions.

The most straightforward approach to use multiobjective optimization techniques to solve a single-objective optimization problems is to redefine the single-objective optimization of  $f(\mathbf{x})$  as a multiobjective optimization problem in which we will have  $m + 1$  objectives, where  $m$  is the number of constraints<sup>4</sup>. Then, we can apply any MOEA to the new vector  $\bar{v} = (f(\mathbf{x}), f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ , where  $f_1(\mathbf{x}), \dots, f_m(\mathbf{x})$  are the original constraints of the problem. An ideal solution  $\mathbf{x}$  would thus have  $f_i(\mathbf{x})=0$  for  $1 \leq i \leq m$  and  $f(\mathbf{x}) \leq f(\mathbf{y})$  for all feasible  $\mathbf{y}$  (assuming minimization).

However, it should be clear that in single-objective optimization problems we do not want just good trade-offs; we want to find the best possible solutions that do not violate any constraints. Therefore, a mechanism such as Pareto ranking may be useful

---

<sup>4</sup> The assumption that we have  $m$  constraints will hold throughout this section.

to approach the feasible region, but once we arrive to it, we will need to guide the search with a different mechanism so that we can reach the global optimum. In order to achieve this goal, we should also be able to maintain diversity in the population. Some of the most representative attempts to use multiobjective optimization techniques (or concepts) to handle constraints in single-objective optimization problems are the following:

1. **COMOGA:** Surry et al. [50] proposed the use of Pareto ranking and VEGA to handle constraints. In their approach, called COMOGA, the population is ranked based on constraint violations (counting the number of individuals dominated by each solution). Then, one portion of the population is selected based on constraint ranking, and the rest based on real cost (fitness) of the individuals. COMOGA compared fairly with a penalty-based approach in a pipe-sizing problem, and was less sensitive to changes in the parameters, but the results achieved were not better than those found with a penalty function [50]. It should be added that COMOGA requires several extra parameters, although its authors argue that the technique is not particularly sensitive to the values of such parameters.
2. **VEGA:** Parmee and Purchase [40] implemented a version of VEGA that handled the constraints of a gas turbine problem as objectives to allow a GA to locate a feasible region within the highly constrained search space of this application. However, VEGA was not used to further explore the feasible region, and instead the authors used specialized operators that would create a variable-size hypercube around each feasible point to help the GA to remain within the feasible region at all times. It is important to notice that no real attempt to reach the global optimum was made in this case. Coello [6] also proposed the use of a population-based multiobjective optimization technique such as VEGA to handle each of the constraints of a single-objective optimization problem as an objective. In this case, however, the goal was to approximate the global optimum. At each generation, the population is split into  $m + 1$  sub-populations ( $m$  is the number of constraints), so that a fraction of the population is selected using the (unconstrained) objective function as its fitness and another fraction uses the first constraint as its fitness and so on. This approach provided good results in several optimization problems [6]. Its main disadvantage was related to scalability issues. However, in a recent application in combinational circuit design we were able to successfully deal with up to 49 objective functions [7]. Furthermore, the approach showed an important improvement (in terms of efficiency) with respect to a previous GA-based approach developed by us for the same task [4].
3. **Line Search and Pareto Dominance:** Camponogara & Talukdar [2] proposed to restate a single objective optimization problem in such a way that two objectives would be considered: the first would be to optimize the original objective function and the second would be to minimize:

$$\Phi(\mathbf{x}) = \sum_{i=1}^m \max[0, g_i(\mathbf{x})]^\beta \quad (5)$$

where  $\beta$  is normally 1 or 2. Once the problem is redefined, nondominated solutions with respect to the two new objectives are generated. The solutions found define a search direction  $d = (x_i - x_j)/|x_i - x_j|$ , where  $x_i \in S_i$ ,  $x_j \in S_j$ , and  $S_i$  and  $S_j$  are Pareto sets. The direction search  $d$  is intended to simultaneously minimize all the objectives. Line search is performed in this direction so that a solution  $x$  can be found such that  $x$  dominates  $x_i$  and  $x_j$  (i.e.,  $x$  is a better compromise than the two previous solutions found). Line search takes the place of crossover in this approach, and mutation is essentially the same, where the direction  $d$  is projected onto the axis of one variable  $j$  in the solution space. Additionally, a process of eliminating half of the population is applied at regular intervals (only the less fitted solutions are replaced by randomly generated points). This approach has obvious problems to keep diversity, as it is reflected by the need to discard the worst individuals at each generation. Also, the use of line search increases the computational cost of the approach and it is not clear what is the impact of the segment chosen to search in the overall performance of the algorithm.

4. **Min-Max:** Jiménez et al. [34] proposed the use of a min-max approach [3] to handle constraints. The main idea of this technique is to apply a set of simple rules to decide the (binary tournament) selection process:
  - (a) If the two individuals being compared are both feasible, then select based on the minimum value of the objective function.
  - (b) If one of the two individuals being compared is feasible and the other one is infeasible, then select the feasible individual.
  - (c) If both individuals are infeasible, then select based on the maximum constraint violation ( $\max g_j(x)$ , for  $j = 1, \dots, m$ ). The individual with the lowest maximum violation wins.

A subtle problem with this approach is that the evolutionary process first concentrates only on the constraint satisfaction problem and therefore it samples points in the feasible region essentially at random [51]. This means that in some cases (e.g., when the feasible region is disjoint) we might land in an inappropriate part of the feasible region from which we will not be able to escape. However, this approach may be a good alternative to find a feasible point in a heavily constrained search space. Deb [17] proposed a similar approach but using tournament selection based on feasibility. However, niching was required to maintain diversity in the population.

5. **MOGA:** Coello [5] explored the use of selection based on dominance (defined in terms of feasibility) to handle constraints. In this case, ranking is performed at three different levels: from two feasible individuals the one with the highest fitness is preferred; if one is feasible and the other infeasible, then the first is chosen; if both are infeasible, then the individual with the lowest amount of constraint violation is chosen. This approach uses stochastic universal sampling so that the selection pressure is not too high and no extra procedures are required to maintain diversity. Also, adaptive crossover and mutation rates were adopted as part of the approach.

6. **NPGA:** Coello & Mezura [10] proposed the use of tournaments based on nondominance (as in the NPGA [32]) to handle constraints. An additional parameter, called selection rank ( $R_s$ ) is added to control the selection pressure of the approach. This parameter makes unnecessary to use equivalent class sharing (as in the NPGA) to maintain diversity and also decreases the (normally high) selection pressure originated from using tournament selection.
7. **Domain Knowledge and Ranking:** Ray et al. [41] proposed a technique to handle constraints in which the population is ranked both in objective function space and in constraint space. The selection strategy adopted eliminates weaknesses from both spaces and ensures a better constraint satisfaction in the offspring produced. The approach uses niches to maintain diversity with Euclidean distances being the similarity measure adopted. It also incorporates mating restrictions based on the information that each individual has of its own feasibility (this idea was inspired on an earlier approach by Hinterding and Michalewicz [31]), so that the global optimum can be reached through cooperative learning.

Some of the possible trends in this area are the following:

- Use of other MOEAs to handle constraints. Some of these techniques may be rather simple and still remain highly competitive. See for example [9].
- Use of online and self-adaptation in constraint-handling techniques both for single- and for multiobjective optimization.
- Extraction and reuse of knowledge obtained from the evolutionary process in order to guide more efficiently the search.
- Design (single-objective optimization) test functions that are particularly difficult for MOEAs to tackle and devise appropriate metrics to measure their performance in this context.

## 7 Where are we heading?

Once we have been able to distinguish between the first and second generations in evolutionary multiobjective optimization, a reasonable question is: where are we heading now?

In the last few years, there has been a considerable growth in the number of publications related to evolutionary multiobjective optimization. However, the variety of topics covered is not as rich as the number of publications released each year. The current trend is to either develop new algorithms (validating them with some of the metrics and test functions available) or to develop interesting applications of existing algorithms.

We will finish this section with a list of some of the research topics that we believe that will keep researchers busy during the next few years:

- New metrics with some insightful analysis of their behavior and limitations. Also, metrics that measure not only offline performance, but also online performance are expected to arise.

- More test functions with more than two objectives and with high dimensionality. Concerns about epistasis, deception, dynamic functions, uncertainty and noise should also be reflected in the upcoming work in this topic.
- Development of a theoretical framework that allows to analyze the behavior of MOEAs. Topics such as the run-time analysis and bounded convergence times of an MOEA are expected to be tackled in the next few years. We should also expect more work on convergence and on modelling MOEAs using statistical tools.

## 8 Conclusions

This chapter has provided a rather brief and general picture of the past, current and future research in evolutionary multiobjective optimization. A brief analysis of some of the most popular algorithms reported in the literature has also been provided together with a summary of the main contributions made in this area in the last few years. Finally, some promising areas of future research were also provided.

Our main goal was to provide a motivation for researchers and students to get into this exciting research discipline to tackle the problems that are our main concern right now. Being a young research area, evolutionary multiobjective optimization still has a lot of opportunities to offer to newcomers and we expect many of them to join us in the next few years.

## 9 Acknowledgments

This paper is representative of the research performed by the Evolutionary Computation Group at CINVESTAV-IPN (EVOCINV). The first author acknowledges support from the mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 34201-A. The second and third authors acknowledge support from CONACyT through a scholarship to pursue graduate studies at CINVESTAV-IPN's Electrical Engineering Department.

## References

1. Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, 1997.
2. Eduardo Camponogara and Sarosh N. Talukdar. A genetic algorithm for constrained and multiobjective optimization. In Jarmo T. Alander, editor, *3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA)*, pages 49–62, Vaasa, Finland, August 1997. University of Vaasa.
3. V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. Systems Science and Engineering. North-Holland, 1983.
4. Carlos A. Coello, Alan D. Christiansen, and Arturo Hernández Aguirre. Use of evolutionary techniques to automate the design of combinational circuits. *International Journal of Smart Engineering System Design*, 2(4):299–314, June 2000.
5. Carlos A. Coello Coello. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering Systems*, 17:319–346, 2000.

6. Carlos A. Coello Coello. Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308, 2000.
7. Carlos A. Coello Coello, Arturo Hernández Aguirre, and Bill P. Buckles. Evolutionary multiobjective design of combinational logic circuits. In Jason Lohn, Adrian Stoica, Didier Keymeulen, and Silvano Colombano, editors, *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pages 161–170, Los Alamitos, CA, July 2000. IEEE Computer Society.
8. 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.
9. Carlos A. Coello Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.
10. Carlos A. Coello Coello and Efrén Mezura Montes. Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In I.C. Parmee, editor, *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, volume 5, pages 273–284, University of Exeter, Devon, UK, April 2002. Springer-Verlag.
11. Carlos A. Coello Coello and Gregorio Toscano Pulido. A Micro-Genetic Algorithm for Multiobjective Optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 126–140. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
12. Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.
13. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.
14. David W. Corne, Nick R. Jerram, Joshua D. Knowles, and Martin J. Oates. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 283–290, San Francisco, California, 2001. Morgan Kaufmann Publishers.
15. David W. Corne, Joshua D. Knowles, and Martin J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
16. 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.
17. Kalyanmoy Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2000.
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 Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
20. Kalyanmoy Deb, Amrit Pratab, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
21. F. Y. Edgeworth. *Mathematical Physics*. P. Keagan, London, England, 1881.
22. Mark Erickson, Alex Mayer, and Jeffrey Horn. The Niche Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 681–695. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
23. Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, CA, 1993. Morgan Kaufmann Publishers.
24. Michael P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153, Hillsdale, NJ, 1985. Lawrence Erlbaum.
25. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, MA, 1989.
26. David E. Goldberg and Kalyanmoy Deb. A comparison of selection schemes used in genetic algorithms. In G.J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, San Mateo, CA, 1991.
27. David E. Goldberg and Jon Richardson. Genetic algorithm with sharing for multimodal function optimization. In John J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, NJ, 1987. Lawrence Erlbaum.
28. P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.
29. T. Hanne. On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, 117(3):553–564, September 2000.
30. Thomas Hanne. Global multiobjective optimization using evolutionary algorithms. *Journal of Heuristics*, 6(3):347–360, August 2000.
31. Robert Hinterding and Zbigniew Michalewicz. Your brains and my beauty: Parent matching for constrained optimisation. In *Proceedings of the 5th International Conference on Evolutionary Computation*, pages 810–815, Anchorage, Alaska, May 1998.
32. Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, NJ, June 1994. IEEE Service Center.
33. W. Jakob, M. Gorges-Schleuter, and C. Blume. Application of genetic algorithms to task planning and learning. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2nd Workshop*, Lecture Notes in Computer Science, pages 291–300, Amsterdam, 1992. North-Holland Publishing Company.
34. Fernando Jiménez, José L. Verdegay, and Antonio F. Gómez-Skarmeta. Evolutionary techniques for constrained multiobjective optimization problems. In Annie S. Wu, editor, *Pro-*



- ceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, pages 115–116, Orlando, Florida, July 1999.
35. Yaochu Jin, Tatsuya Okabe, and Bernhard Sendhoff. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How? In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 1042–1049, San Francisco, California, 2001. Morgan Kaufmann Publishers.
  36. Joshua D. Knowles and David W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
  37. J.N. Morse. Reducing the size of the nondominated set: Pruning by clustering. *Computers and Operations Research*, 7(1–2):55–66, 1980.
  38. Christopher K. Oei, David E. Goldberg, and Shau-Jin Chang. Tournament Selection, Niching, and the Preservation of Diversity. Technical Report 91011, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois, December 1991.
  39. Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.
  40. I. C. Parmee and G. Purchase. The development of a directed genetic search technique for heavily constrained design spaces. In I. C. Parmee, editor, *Adaptive Computing in Engineering Design and Control-'94*, pages 97–102, Plymouth, UK, 1994. University of Plymouth, University of Plymouth.
  41. Tapabrata Ray, Tai Kang, and Seow Kian Chye. An evolutionary algorithm for constrained optimization. In Darrell Whitley, David Goldberg, Erick Cantú-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pages 771–777, San Francisco, California, 2000. Morgan Kaufmann.
  42. R. S. Rosenberg. *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Arbor, MI, 1967.
  43. Günter Rudolph. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 511–516, Piscataway, NJ, 1998. IEEE Press.
  44. Günter Rudolph and Alexandru Agapie. Convergence Properties of Some Multi-Objective Evolutionary Algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, volume 2, pages 1010–1016, Piscataway, NJ, July 2000. IEEE Press.
  45. J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Nashville, TN, 1984.
  46. J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, NJ, 1985. Lawrence Erlbaum.
  47. 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, CA, 1985. AAAI.
  48. N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.
  49. W. Stadler. Fundamentals of multicriteria optimization. In W. Stadler, editor, *Multicriteria Optimization in Engineering and the Sciences*, pages 1–25. Plenum Press, New York, NY, 1988.
  50. Patrick D. Surry and Nicholas J. Radcliffe. The COMOGA method: Constrained optimisation by multiobjective genetic algorithms. *Control and Cybernetics*, 26(3):391–412, 1997.
  51. Patrick D. Surry, Nicholas J. Radcliffe, and Ian D. Boyd. A multi-objective approach to constrained optimisation of gas supply networks : The COMOGA method. In Terence C.

- Fogarty, editor, *Evolutionary Computing. AISB Workshop. Selected Papers*, pages 166–180, Sheffield, U.K., 1995. Springer-Verlag. Lecture Notes in Computer Science No. 993.
52. David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, May 1999.
  53. David A. Van Veldhuizen and Gary B. Lamont. Evolutionary computation and convergence to a pareto front. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, Stanford, CA, July 1998. Stanford University Bookstore.
  54. 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.
  55. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
  56. Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.
  57. 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.