CHAPTER 1

AN INTRODUCTION TO MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS AND THEIR APPLICATIONS

Carlos A. Coello Coello[1] and Gary B. Lamont[2]

[1] *CINVESTAV-IPN*
Evolutionary Computation Group
Dpto. de Ing. Elect./Secc. Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D.F. 07300, MEXICO
E-mail: ccoello@cs.cinvestav.mx
[2] *Department of Electrical and Computer Engineering*
Graduate School of Engineering and Management
Air Force Institute of Technology
WPAFB, Dayton, Ohio, 45433, USA
E-mail: Gary.Lamont@afit.edu

This chapter provides the basic concepts necessary to understand the rest of this book. The introductory material provided here includes some basic mathematical definitions related to multi-objective optimization, a brief description of the most representative multi-objective evolutionary algorithms in current use and some of the most representative work on performance measures used to validate them. In the final part of this chapter, we provide a brief description of each of the chapters contained within this volume.

1.1. Introduction

Early analogies between the mechanism of natural selection and a learning (or optimization) process led to the development of the so-called "evolutionary algorithms" (EAs)[3], in which the main goal is to simulate the evolutionary process in a computer. The use of EAs for optimization tasks has become very popular in the last few years, spanning virtually every application domain[22,44,25,4].

From the several emergent research areas in which EAs have become increasingly popular, multi-objective optimization has had one of the fastest growing in recent years[12]. A multi-objective optimization problem (MOP) differs from a single-objective optimization problem because it contains several objectives that require optimization. When optimizing a single-objective problem, the best single design solution is the goal. But for multi-objective problems, with several (possibly conflicting) objectives, there is usually no single optimal solution. Therefore, the decision maker is required to select a solution from a finite set by making compromises. A suitable solution should provide for acceptable performance over all objectives[40]. Many fields continue to address complex real-world multi-objective problems using search techniques developed within computer engineering, computer science, decision sciences, and operations research[10]. The potential of evolutionary algorithms for solving multi-objective optimization problems was hinted as early as the late 1960s by Rosenberg[47]. However, the first actual implementation of a multi-objective evolutionary algorithm (MOEA) was produced until the mid-1980s[48,49]. Since then, a considerable amount of research has been done in this area, now known as evolutionary multi-objective optimization (EMOO)[12]. The growing importance of this field is reflected by a significant increment (mainly during the last ten years) of technical papers in international conferences and peer-reviewed journals, special sessions in international conferences and interest groups in the Internet[b].

The main motivation for using EAs to solve multi-objective optimization problems is because EAs deal simultaneously with a set of possible solutions (the so-called population) which 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[40]. Additionally, EAs 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 known problems with mathematical programming techniques[7,18,12,61].

This monograph attempts to present an extensive variety of high-dimensional MOPs and their acceptable statistical solutions using MOEAs as exercised by numerous researchers. The intent of our discussion then is to promote a wider understanding and an ability to use MOEAs in order

[b]The first author maintains an EMOO repository with over 1700 bibliographical entries at: `http://delta.cs.cinvestav.mx/~ccoello/EMOO`, with a mirror at `http://www.lania.mx/~ccoello/EMOO/`

to find "good" solutions in a wide spectrum of high-dimensional real-world applications.

1.2. Basic concepts

In order to provide a common basis for understanding the rest of this book, we provide next a set of basic definitions normally adopted both in single-objective and in multi-objective optimization[12]:

Definition 1 (Global Minimum): *Given a function $f : \Omega \subseteq \mathcal{S} = \mathbb{R}^n \rightarrow \mathbb{R}$, $\Omega \neq \emptyset$, for $\vec{x} \in \Omega$ the value $f^* \triangleq f(\vec{x}^*) > -\infty$ is called a global minimum if and only if*

$$\forall \vec{x} \in \Omega : \quad f(\vec{x}^*) \leq f(\vec{x}) \; . \tag{1}$$

Then, $\vec{x}^*$ is the global minimum solution(s), f is the objective function, and the set Ω is the feasible region ($\Omega \subset \mathcal{S}$). The problem of determining the global minimum solution(s) is called the ***global optimization*** *problem.*
$\square$

Although single-objective optimization problems may have a unique optimal solution, MOPs (as a rule) present a possibly uncountable *set* of solutions, which when evaluated, produce vectors whose components represent trade-offs in objective space. A decision maker then implicitly chooses an acceptable solution (or solutions) by selecting one or more of these vectors. MOPs are mathematically defined as follows:

Definition 2 (General MOP): *In general, an MOP minimizes $F(\vec{x}) = (f_1(\vec{x}), \ldots, f_k(\vec{x}))$ subject to $g_i(\vec{x}) \leq 0$, $i = 1, \ldots, m$, $\vec{x} \in \Omega$. An MOP solution minimizes the components of a vector $F(\vec{x})$ where $\vec{x}$ is an n-dimensional decision variable vector ($\vec{x} = x_1, \ldots, x_n$) from some universe Ω.* $\square$

Definition 3 (Pareto Dominance): *A vector $\vec{u} = (u_1, \ldots, u_k)$ is said to dominate $\vec{v} = (v_1, \ldots, v_k)$ (denoted by $\vec{u} \preceq \vec{v}$) if and only if u is partially less than v, i.e., $\forall i \in \{1, \ldots, k\},\ u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$.* $\square$

Definition 4 (Pareto Optimality): *A solution $x \in \Omega$ is said to be Pareto optimal with respect to Ω if and only if there is no $x' \in \Omega$ for which $\vec{v} = F(x') = (f_1(x'), \ldots, f_k(x'))$ dominates $\vec{u} = F(x) = (f_1(x), \ldots, f_k(x))$. The phrase "Pareto optimal" is taken to mean with respect to the entire decision variable space unless otherwise specified.* $\square$

Definition 5 (Pareto Optimal Set): *For a given MOP $F(x)$, the Pareto optimal set ($\mathcal{P}^*$) is defined as:*

$$\mathcal{P}^* := \{x \in \Omega \mid \neg\exists\, x' \in \Omega \;\; F(x') \preceq F(x)\}. \tag{2}$$

□

Definition 6 (Pareto Front): *For a given MOP $F(x)$ and Pareto optimal set $\mathcal{P}^*$, the Pareto front ($\mathcal{PF}^*$) is defined as:*

$$\mathcal{PF}^* := \{\vec{u} = F(x) = (f_1(x), \ldots, f_k(x)) \mid x \in \mathcal{P}^*\}. \tag{3}$$

□

The Pareto optimal solutions are ones within the search space whose corresponding objective vector components cannot be improved simultaneously. These solutions are also known as *non-inferior*, *admissible*, or *efficient* solutions, with the entire set represented by $\mathcal{P}^*$ or P_{true}. Their corresponding vectors are known as *nondominated*; selecting a vector(s) from this vector set (the Pareto Front set $\mathcal{PF}^*$ or PF_{true}) implicitly indicates acceptable Pareto optimal solutions (**genotypes**). These are the set of all solutions whose vectors are nondominated; these solutions are classified based on their *phenotypical* expression. Their expression (the nondominated vectors), when plotted in criterion (**phenotype**) space, is known as the *Pareto front*[55,68]. With these basic MOP definitions, we are now ready to delve into the structure of MOPs and the specifics of various MOEAs.

1.3. Basic operation of a MOEA

The objective of a MOEA is to converge to the true Pareto front of a problem which normally consists of a diverse set of points. MOPs (as a rule) can present an uncountable set of solutions, which when evaluated produce vectors whose components represent trade-offs in decision space. During MOEA execution, a "local" set of Pareto optimal solutions (with respect to the *current* MOEA generational population) is determined at each EA generation and termed $P_{current}(t)$, where t represents the generation number. Many MOEA implementations also use a *secondary population*, storing all/some Pareto optimal solutions found through the generations[55]. This secondary population is termed $P_{known}(t)$, also annotated with t (representing completion of t generations) to reflect possible changes in its membership during MOEA execution. $P_{known}(0)$ is defined as $\emptyset$ (the empty set) and P_{known} alone as the *final*, overall set of Pareto optimal solutions returned by a MOEA. Of course, the *true* Pareto optimal solution set (termed

P_{true}) is not explicitly known for MOPs of any difficulty. P_{true} is defined by the functions composing an MOP; it is fixed and does not change.

$P_{current}(t)$, P_{known}, and P_{true} are sets of MOEA genotypes where each set's phenotypes form a Pareto front. We term the associated Pareto front for each of these solution sets as $PF_{current}(t)$, PF_{known}, and PF_{true}. Thus, when using a MOEA to solve MOPs, one implicitly assumes that one of the following conditions holds: $PF_{known} \subseteq PF_{true}$ or that over some norm (Euclidean, RMS, etc.), $PF_{known} \in [PF_{true}, PF_{true} + \epsilon]$, where ϵ is a small value.

Generally speaking, a MOEA is an extension on an EA in which two main issues are considered:

- How to select individuals such that nondominated solutions are preferred over those which are dominated.
- How to maintain diversity as to be able to maintain in the population as many elements of the Pareto optimal set as possible.

Regarding selection, most current MOEAs use some form of Pareto ranking. This approach was originally proposed by Goldberg[25] and it sorts the population of an EA based on Pareto dominance, such that all nondominated individuals are assigned the same rank (or importance). The idea is that all nondominated individuals get the same probability to reproduce and that such probability is higher than the one corresponding to individuals which are dominated. Although conceptually simple, several possible ways exist to implement a MOEA using Pareto ranking[18,12].

The issue of how to maintain diversity in an EA as been addressed by a extensive number of researchers[39,27]. The approaches proposed include fitness sharing and niching[19], clustering[54,65], use of geographically-based schemes to distribute solutions[36,14,13], and the use of entropy[32,16], among others. Additionally, some researchers have also adopted mating restriction schemes[51,63,41]. More recently, the use of relaxed forms of Pareto dominance has been adopted as a mechanism to encourage more exploration and, therefore, to provide more diversity. From these mechanisms, ϵ-dominance has become increasingly popular, not only because of its effectiveness, but also because of its sound theoretical foundation[38].

In the last few years, the use of elitist schemes has also become common among MOEA researchers. Such schemes tend to consist of the use of an external archive (normally called "secondary population") that may interact in different ways with the main (or "primary") population of the MOEA. Despite storing the nondominated solutions found along the evolutionary process, secondary populations have also been used to improve the distribution of the solutions[35] and to regulate the selection pressure of a MOEA[65]. Alternatively, a few algorithms use a plus (+) selection mechanism by which parents are combined with their offspring in a single population from which a subset of the "best" individuals is retained. The most popular from these algorithms is the Nondominated Sorting Genetic Algorithm-II (NSGA-II)[21].

1.4. Classifying MOEAs

There are several possible ways to classify MOEAs. The following taxonomy is perhaps the most simple and is based on the type of selection mechanism adopted:

- Aggregating Functions
- Population-based Approaches
- Pareto-based Approaches

We will briefly discuss each of them in the following subsections.

1.4.1. *Aggregating Functions*

Perhaps the most straightforward approach to deal with multi-objective problems is to combine them into a single scalar value (e.g., adding them together). These techniques are normally known as "aggregating functions", because they combine (or "aggregate") all the objectives of the problem into a single one. An example of this approach is a fitness function in which we aim to solve the following problem:

$$\min \sum_{i=1}^{k} w_i f_i(\vec{x}) \tag{4}$$

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

$$\sum_{i=1}^{k} w_i = 1 \tag{5}$$

Aggregating functions may be linear (as the previous example) or nonlinear[46,59,26]. Aggregating functions have been largely underestimated by MOEA researchers mainly because of the well-known limitation of linear aggregating functions (i.e., they cannot generate non-convex portions of the Pareto front regardless of the weight combination used[17]). Note however that nonlinear aggregating functions do not necessarily present such limitation[12], and they have been quite successful in multi-objective combinatorial optimization[30].

1.4.2. *Population-based Approaches*

In this type of approach, the population of an EA is used to diversify the search, but the concept of Pareto dominance is not directly incorporated into the selection process. The classical example of this sort of approach is the Vector Evaluated Genetic Algorithm (VEGA), proposed by Schaffer[49]. VEGA basically consists of a simple genetic algorithm with a modified selection mechanism. At each generation, a number of sub-populations are generated by performing proportional selection according to each objective function in turn. Thus, for a problem with k objectives, k sub-populations of size M/k each are generated (assuming a total population size of M). These sub-populations are then shuffled together to obtain a new population of size M, on which the genetic algorithm applies the crossover and mutation operators. VEGA has several problems, from which the most serious is that its selection scheme is opposed to the concept of Pareto dominance. If, for example, there is an individual that encodes a good compromise solution for all the objectives (i.e., a Pareto optimal solution), but it is not the best in any of them, it will be discarded. Schaffer suggested some heuristics to deal with this problem. For example, to use a heuristic selection preference approach for nondominated individuals in each generation, to protect individuals that encode Pareto optimal solutions but are not the best in any single objective function. Also, crossbreeding among the "species" could be encouraged by adding some mate selection heuristics instead of using the random mate selection of the traditional genetic algorithm. Nevertheless, the fact that Pareto dominance is not directly incorporated into the selection process of the algorithm remains as its main disadvantage.

One interesting aspect of VEGA is that despite its drawbacks it remains in current use by some researchers mainly because it is appropriate for problems in which we want the selection process to be biased and in which we have to deal with a large number of objectives (e.g., when handling constraints as objectives in single-objective optimization[9] or when solving problems in which the objectives are conceptually identical[11]).

1.4.3. *Pareto-based Approaches*

Under this category, we consider MOEAs that incorporate the concept of Pareto optimality in their selection mechanism. A wide variety of Pareto-based MOEAs have been proposed in the last few years and it is not the intent of this section to provide a comprehensive survey of them since such a review is available elsewhere[12]. In contrast, this section provides a brief discussion of a relatively small set of Pareto-based MOEAs that are representative of the research being conducted in this area.

Goldberg's Pareto Ranking: Goldberg suggested moving the population toward PF_{true} by using a selection mechanism that favors solutions that are nondominated with respect to the current population[25]. He also suggested the use of fitness sharing and niching as a diversity maintenance mechanism[19].

Multi-Objective Genetic Algorithm (MOGA): Fonseca and Fleming[23] proposed a ranking approach different from Goldberg's scheme. In this case, each individual in the population is ranked based on how many other points dominate them. All the nondominated individuals in the population are assigned the same rank and obtain the same fitness, so that they all have the same probability of being selected. MOGA uses a niche-formation method in order to diversify the population, and a relatively simple methodology is proposed to compute the similarity threshold (called σ_{share}) required to determine the radius of each niche.

The Nondominated Sorting Genetic Algorithm (NSGA): This method[53] 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. An offshoot of this approach, the **NSGA-II**[21], uses elitism and a crowded comparison operator that ranks the population based on both Pareto dominance and region density. This crowded comparison operator makes the NSGA-II considerably faster than its predecesor while producing very good results.

Niched Pareto Genetic Algorithm (NPGA): This method employs an interesting form of tournament selection called Pareto domination tournaments. Two members of the population are chosen at random and they are each compared to a subset of the population. If one is nondominated and the other is not, then the nondominated one is selected. If there is a tie (both are either dominated or nondominated), then fitness sharing decides the tourney results[28].

Strength Pareto Evolutionary Algorithm (SPEA): This method attempts to integrate different MOEAs[65]. The algorithm uses a "strength" value that is computed in a similar way to the MOGA ranking system. Each member of the population is assigned a fitness value according to the strengths of all nondominated solutions that dominate it. Diversity is maintained through the use of a clustering technique called the "average linkage method."

A revision of this method, called **SPEA2**[62], adjusts slightly the fitness strategy and uses nearest neighbor techniques for clustering. In addition, archiving mechanism enhancements allow for the preservation of boundary solutions that are missed with SPEA.

Multi-Objective Messy Genetic Algorithm (MOMGA): This method extends the mGA[20] to solve multi-objective problems. The MOMGA[55] is an explicit building block GA that produces all building blocks of a user specified size. The algorithm has three phases: Initialization, Primordial, and Juxtapositional. The **MOMGA-II** algorithm was developed by Zydallis as an extension of the MOMGA[67]. It was developed in order to expand the state of the art for explicit building-block MOEAs. While there has been a lot of research done for single objective explicit building-block EAs, this was a first attempt at using the concept for MOPs. Exponential growth of the population as the building block size grows may be a disadvantage of this approach in some applications.

Multi-Objective Hierarchical Bayesian Optimization Algorithm (hBOA): This search technique is a conditional model builder. It expands the idea of the compact genetic algorithm and the stud genetic algorithm. The hBOA defines a Bayesian model that represents "small" building blocks (BBs) reflecting genotypical epistasis using a hierarchical Bayesian network[45]. The mhBOA[31] is in essence a linkage learning algorithm that extends the hBOA and attempts to define tight and loose linkages to building blocks in the chromosome over a Pareto front. In particular, this method uses a Bayesian network (a conditional probabilistic model) to guide the search toward a solution. A disadvantage of this algorithm is the time it takes to generate results for a relatively small number of linkages.

Pareto Archived Evolution Strategy (PAES): This method, formulated by Knowles and Corne[34], uses a (1+1) evolution strategy, where each parent generates one offspring through mutation. The method uses an archive of nondominated solutions to compare with individuals in the current population. For diversity, the algorithm generates a grid overlaid on the search space and counts the number of solutions in each grid. A disadvantage of this method is its performance on disconected Pareto Fronts.

Micro-Genetic Algorithm for Multi-Objective Optimization: The micro-genetic algorithm was introduced by Coello Coello and Toscano Pulido[10] and, by definition, has a small population requiring a reinitialization technique. An initial random population flows into a population memory which has two parts: a replaceable and a non-replaceable portion. The non-replaceable part provides the population diversity. The replaceable portion of course changes at the end of each generation where this population undergoes crossover and mutation. Using various elitism selection operators, the non-dominated individuals compose the replaceable portion.

General Multi-Objective Program (GENMOP): This method is a parallel, real-valued MOEA initially used for bioremediation research[33]. This method archives all previous population members and ranks them. Archived individuals with the highest ranks are used as a mating pool to mate with the current generation. The method uses equivalence class sharing for niching to allow for diversity in the mating pool. A disadvantage of this algorithm is the Pareto ranking of the archived individuals at each generation.

Other researchers have combined elements of these MOEAs to develop unique MOEAs for their specific problem domain with excellent results.

1.5. MOEA Performance Measures

The use of performance measures (or metrics) allows a researcher or computational scientist to assess (in a quantitative way) the performance of their algorithms. The MOEA field is no different. MOEA performance measures tend to focus on the phenotype or objective domain as to the accuracy of the results. This is different to what most operations researchers do. They tend to use metrics in the genotype domain. But since there is an explicit mapping between the two, it doesn't really matter in which domain you define your metrics[12,57].

MOEA metrics can be used to measure final performance or track the generational performance of the algorithm. This is important because it allows the researcher to manage the algorithm convergence process during execution. This section presents a variety of MOEA metrics, yet, no attempt is made to be comprehensive. For a more detailed treatment of this topic, the interested reader should consult additional references[12,60,66].

Error Ratio (ER): This metric reports the number of vectors in PF_{known} that are not members of PF_{true}. This metric requires that the researcher knows PF_{true}. The mathematical representation of this metric is shown in equation 6:

$$ER \triangleq \frac{\sum_{i=1}^{n} e_i}{n} \tag{6}$$

where n is the number of vectors in PF_{known} and e_i is a zero when the i vector is an element of PF_{true} or a 1 if i is not an element. So when $ER = 0$, the PF_{known} is the same as PF_{true}; but when $ER = 1$, this indicates that none of the points in PF_{known} are in PF_{true}.

Two Set Coverage (CS): This metric[60] compares the coverage of two competing sets and outputs the percentage of individuals in one set dominated by the individuals of the other set. This metric does not require that the researcher has knowledge of PF_{true}. The equation for this metric is shown in equation 7:

$$CS(X', X'') \triangleq \frac{|a'' \in X''; \forall a' \in X' : a' \succeq a''|}{|X''|} \tag{7}$$

where $X', X'' \subseteq X$ are two sets of phenotype decision vectors, and (X', X'') are mapped to the interval $[0, 1]$. This means that $CS = 1$ when X' dominates or equals X''.

Generational Distance (GD): This metric was proposed by Van Veldhuizen and Lamont[56]. It reports how far, on average, PF_{known} is from PF_{true}. This metric requires that the researcher knows PF_{true}. It is mathematically defined in equation

$$GD \triangleq \frac{(\sum_{i=1}^{n} d_i^p)^{1/p}}{n} \tag{8}$$

where n is the number of vectors in PF_{known}, $p = 2$, and D_i is the Euclidean distance between each member and the closest member of PF_{true}, in the phenotype space. When $GD = 0$, $PF_{known} = PF_{true}$.

Hyperarea and Ratio (H,HR): These metrics, introduced by Zitzler & Thiele[64], define the area of coverage that PF_{known} has with respect to the objective space. This would equate to the summation of all the areas of rectangles, bounded by the origin and $(f_1(\vec{x}), f_2(\vec{x}))$, for a two-objective MOEA. Mathematically, this is described in equation 9:

$$H \triangleq \left\{ \bigcup_i a_i | v_i \in PF_{known} \right\} \tag{9}$$

where v_i is a nondominated vector in PF_{known} and a_i is the hyperarea calculated between the origin and vector v_i. But if PF_{known} is not convex, the results can be misleading. It is also assumed in this model that the origin is $(0, 0)$.

The hyperarea ratio metric definition can be seen in equation 10:

$$HR \triangleq \frac{H_1}{H_2} \tag{10}$$

where H_1 is the PF_{known} hyperarea and H_2 is the hyperarea of PF_{true}. This results in $HR \geq 1$ for minimization problems and $HR \leq 1$ for maximization problems. For either type of problem, $PF_{known} = PF_{true}$ when $HR = 1$. This metric requires that the researcher knows PF_{true}.

Spacing (S): This metric was proposed by Schott[50] and it measures the distance variance of neighboring vectors in PF_{known}. Equation 11 defines this metric.

$$S \triangleq \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2} \tag{11}$$

and

$$d_i = min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|) \tag{12}$$

where $i, j = 1\ldots, n$, $\bar{d}$ is the mean of all d_i, and n is the number of vectors in PF_{known}. When $S = 0$, all members are spaced evenly apart. This metric does not require the researcher to know PF_{true}.

Overall Nondominated Vector Generation Ratio (ONVGR): This metric measures the total number of nondominated vectors during MOEA execution and divides it by the number of vectors found in PF_{true}. This metric is defined as shown in equation 13:

$$ONVG \triangleq \frac{PF_{false}}{PF_{true}} \tag{13}$$

When $ONVGR = 1$ this states only that the same number of points have been found in both PF_{true} and PF_{known}. It does not infer that $PF_{true} = PF_{known}$. This metric requires that the researcher knows PF_{true}.

Progress Measure RP: For single-objective EAs, Bäck[3] defines a metric that measures convergence velocity. This single-objective metric is applied to multi-objective MOEAs[55], and is reflected in equation 14:

$$RP \triangleq ln\sqrt{\frac{G_1}{G_T}} \tag{14}$$

where G_1 is the generational distance for the first generation and G_T is the distance for generation T. Recall that generational distance was defined in equation 8 and it measures the average distance from PF_{true} to PF_{known}. This metric requires that the researcher knows PF_{true}.

Generational Nondominated Vector Generation (GNVG): This is a simple metric, introduced by Van Veldhuizen[55] that lists the number of nondominated vectors produced for each generation. This is defined in equation 15

$$GNVG \stackrel{\triangle}{=} |PF_{current}(t)| \tag{15}$$

This metric does not require the researcher knows PF_{true}.

Nondominated Vector Addition (NVA): This metric, introduced by Van Veldhuizen[55], calculates the number of nondominated vectors gained or lost from the previous PF_{known} generation. Equation 16 defines this metric.

$$NVA \stackrel{\triangle}{=} |PF_{known}(t)| - |PF_{known}(t-1)| \tag{16}$$

But this metric can be misleading when a new vector dominates two or more vectors from the previous generation. In addition, this metric may remain static over the course of several generations while new points are added that dominate others from the previous generation. This metric does not require the researcher knows PF_{true}.

As to what metrics are appropriate, it of course depends upon the MOEA application to the given MOP. Since in real-world applications, the true Pareto Front is unknown, relative metrics are usually selected. It is also worth observing that recent research has shed light on the limitations of unary metrics (i.e., performance measures that assign each approximation of the Pareto optimal set a number that reflects a certain quality aspect)[66]. Such study favors the use of binary metrics. As a consequence of this study, it is expected that in the next few years MOEA researchers will eventually adopt binary metrics on a regular basis, but today, the use of unary metrics (such as error ratio and many of the others discussed in this section) is still common.

1.6. Design of MOEA Experiments

To conduct a thorough evaluation of the performance of any MOEA, a design of experiments or methodology should be common practice prior to testing and evaluating the search results. The main goal of MOEA research is the creation of an effective and efficient algorithm that renders good solutions. But to achieve that goal, several smaller goals need to be addressed. These goals can be classified under two categories: effectiveness goals and efficiency goals. *Effectiveness goals* should list the effectiveness goals and the experimental design employed to validate that these goals are met. *Efficiency Goals* should list the experimental design employed to validate the efficiency goals. Also, a section on the *Computing Environment* should indicate the computing environment for ease of repeatability. Finding good solutions is the top priority for any MOEA application research. Therefore one has to validate that their algorithm does indeed find good solutions. Benchmarks can also used. In addition, comparison with current MOP designs is appropriate for validation. Once a baseline set of runs are completed and analyzed, algorithm parameters can be tweaked to possibly improve effectiveness. The various application chapters in this text have attempted to adhere to such an experimental design and follow the reporting techniques of the next section.

1.6.1. *Reporting MOEA Computational Results*

Before the advent of the "Scientific Method", many engineers and scientists merely used the trial and error method in an attempt to gain insight into a particular problem. The scientific method is the process by which engineers and scientists, collectively and over time, endeavor to construct an accurate (that is, reliable, consistent and non-arbitrary) representation of the world or the problem which they study. Recognizing that personal and cultural beliefs influence both our perception and our interpretation of natural phenomena, we aim through the use of standard procedures and criteria to minimize those influences when developing a conjecture or a theory or a qualification. In summary, the scientific method attempts to minimize the influence of bias or prejudice in the experimenter.

Each application chapter reporting computational experiments attempt to follow the above objective since they use computer generated evidence to compare or rank competing MOEA software techniques and Pareto Front solutions. Chapter authors consider various classical references that can direct computational experimentation[6,15,29]. According to Jackson, et al.[29], the researcher should always keep in mind various elements identified as to "What to Keep In Mind When Testing:"

- Are the results presented statistically sufficient to justify the claims made?

- Is there sufficient detail to reproduce the results?
- When should a statistically-based experiment be done–usually when a claim such as "this method is better (i.e. faster, more accurate, more efficient, easier-to-use, etc.)"?
- Are the proper test problems being used?
- Are all possible performance measures (efficiency, robustness, reliability, ease-of-use, etc.) addressed?
- Is enough information provided with respect to the architecture of the hardware being used?

One should organize the design of experiments. For example, one should discuss input and output data, the identification of all parameters available during testing (for all tests the parameters are the same unless otherwise indicated), a discussion of the random number generators and seeds and other topics that are pertinent to the set of experiments. Following the general information, each individual experiment is presented with the objective and methodology of the experiment identified. For each experiment any parameter settings or environmental settings that differ from the generalized discussion are duly noted. Various statistical methods should be addressed such as mean, average, max, min, student t-test, Kruskal-Wallis test, and others as appropriate for the computational experiment.

1.7. Layout of the Book

After presenting some basic concepts, terminology and a brief discussion on methodological aspects related to the use of MOEAs, we devote this last section to discuss briefly each of the remaining chapters that are integrated into this book or monograph. As indicated in the preface, these 29 chapters are divided in four application collections. The specific chapters that compose each of these parts are summarized in the following subsections. Note that many authors use specific MOEAs that are summarized in Section 1.4. Also, observe that some of the various metrics discussed in Section 1.5 are employed in statistical MOEA evaluation employing the experimental testing techniques of Section 1.6.

1.7.1. *Part I: Engineering Applications*

Considering that the use of MOEAs in engineering has been very extensive, this first part is the largest in the book, as it includes chapters from 2 to 13.

In Chapter 2, Ray adopts a scheme that handles objectives and constraints separately. Nondominance is used not only for selecting individuals but also to handle constraints. The MOEA adopted in this work is the NSGA[53] with elitism. The approach is applied to some engineering design problems (a welded beam, a bulk carrier and an airfoil).

Farina and Di Barba apply in Chapter 3 several approaches to the design of industrial electromagnetic devices (the case studies consist of a magnetic reactor and an inductor for transverse-flux heating of a metal strip). The authors consider the use of the Non-dominated Sorting Evolutionary Strategy Algorithm[c] (NSESA), a Pareto Gradient Based Algorithm (PGBA), a Pareto Evolution Strategy Algorithm (PESTRA), and a Multi Directional Evolution Strategy Algorithm (MDESTRA). At the end, they decide to adopt hybrid approaches in which NSESA is combined with both a deterministic and a local-global strategy.

Reed & Devireddy use in Chapter 4 the NSGA-II[21] enhanced with the ϵ-dominance archiving and automatic parameterization techniques[38] to optimize groundwater monitoring networks. The authors indicate that the use of ϵ-dominance not only eliminated the empirical fine-tuning of parameters of their MOEA, but also reduced the computational demands by more than 70% with respect to some of their previous work.

In Chapter 5, Hernández Luna and Coello Coello use a particle swarm optimizer with a population-based selection scheme (similar to VEGA[49]) to design combinational logic circuits. One of the relevant aspects of this work is that the problem to be solved is actually mono-objective. However, the use of a multi-objective selection scheme improves both the robustness and the quality of the results obtained.

Furukawa et al. present in Chapter 6 the application of two MOEAs to the sensor and vehicle parameter determination for successful autonomous vehicles navigation. The MOEAs adopted are: (1) the Multi-objective Continuous Evolutionary Algorithm (MCEA) and (2) the Multi-Objective Gradient-based Method (MOGM). Due to space limitations, only the results produced by the MCEA are presented in the chapter, although the authors indicate that both MOEAs reach the same final results. It is worth noticing the use of the so-called Center-of-Gravity Method (CoGM) to select a single solution from the Pareto optimal set produced by the MCEA.

In Chapter 7, Tan and Li use a MOEA to design optimal unified linear time-invariant control (ULTIC) systems. The proposal consists of a method-

[c]This algorithm is a variation of the NSGA[53].

ology for performance-prioritized computer aided control system design in which a MOEA toolbox previously designed by the authors is used as an optimization engine. An interesting aspect of this work is that the user is allowed to set his/her goals on-line (without having to restart the entire design cycle) and can visualize (in real-time) the effect of such goal setting on the results. The proposed methodology is applied to a non-minimal phase plant control system.

Gaspar-Cunha and Covas in Chapter 8 apply a MOEA to solve polymer extrusion problems. The authors optimize the performance of both single-screw and co-rotating twin-screw extruders. The MOEA adopted is called Reduced Pareto Set Genetic Algorithm with Elitism (RPSGAe) and was previously proposed by the same authors[24]. An interesting aspect of this work is that the RPSGAe uses a clustering technique not to maintain diversity as is normally done, but to reduce the number of Pareto optimal solutions. The problems solved are formulated as multi-objective traveling salesperson problems (i.e., they are actually dealing with multi-objective combinatorial optimization problems).

In Chapter 9, Hernández Aguirre and Botello Rionda propose an extension of the Pareto Archived Evolution Strategy (PAES)[36] which is able to deal with both single-objective and multi-objective optimization problems. The proposed approach is called Inverted and Shrinkable Pareto Archived Evolutionary Strategy (ISPAES), and is used to solve several truss optimization problems (a common problem in structural and mechanical engineering). The main differences between ISPAES and PAES are in the selection mechanism and the implementation of the adaptive grid. The test problems adopted include both single and multiple objective problems as well as discrete and continuous search spaces.

Balling presents in Chapter 10 an interesting application of MOEAs to city and regional planning. The MOEA adopted uses the maximin fitness function previously proposed by the author[5]. The approach has been applied to plan the Wasatch Front Metropolitan Region in Utah (in the USA). An interesting aspect of this work is the discussion presented by the author regarding the reluctance from the authorities to actually implement some of the plans produced by the MOEA. The author attributes this reluctance both to the high number of (nondominated) plans produced (no scheme to incorporate user's preferences[8] was adopted by Balling) and to the psychological impact that this sort of (radically new) approach has on people.

Jozefowiez et al. present in Chapter 11 a MOEA to solve the bi-objective covering tour problem. The MOEA adopted is the NSGA-II[21], and the results are compared with respect to an exact algorithm based on a branch-and-bound approach which can be applied only to relatively small instances of the problem. The chapter also presents a thorough review of multi-objective routing problems reported in the specialized literature.

Chapter 12, by Künzli et al., presents a benchmark problem in computer engineering (the design space exploration of packet processor architectures). Besides describing several details related to the proposed benchmark problem, the authors also refer to the text-based interface developed by them which is platform and programming language independent. This aims to facilitate the use of different MOEAs (across different platforms) to solve such problem.

In the last chapter of the first part (Chapter 13), Obayashi and Sasaki present the use of a MOEA for aerodynamic design of supersonic wings. The MOEA adopted is the Adaptive Range Multiobjective Genetic Algorithm (ARMOGA), which is based on an approach originally developed by Arakawa and Hagiwara[2]. The multi-objective extensions are based on MOGA[23]. An interesting aspect of this work is the use of Self-Organizing Maps (SOMs) both to visualize trade-offs among the objectives of the problem and to perform some sort of data mining of the designs produced.

1.7.2. *Part II: Scientific Applications*

The second part of the book, which focuses on scientific applications of MOEAs, includes chapters from 14 to 19.

In Chapter 14, Ray presents the use of a MOEA to optimize gas-solid separation devices used for particulate removal from air (namely, the design of cyclone separators and venturi scrubbers). The author used the NSGA[53], mainly because of her previous experience with such algorithm.

Mancini et al. present in Chapter 15 the use of a MOEA for an application in Physics: the spectroscopic data analysis of inertial confinement fusion implosion cores based on the self-consistent analysis of simultaneous narrow-band X-ray images and X-ray line spectra. The MOEA adopted is the Niched-Pareto Genetic Algorithm (NPGA)[28].

Chapter 16, by Lahanas, presents a survey of the use of MOEAs in medicine. The types of problems considered include medical image processing, computer-aided diagnosis, treatment planning, and data mining.

In Chapter 17, Kumar describes the use of a MOEA in the solution of high-dimensional and complex domains of machine learning. The MOEA is used as a pre-processor for partitioning these complex learning tasks into simpler domains that can then be solved using traditional machine learning approaches. The MOEA adopted is the Pareto Converging Genetic Algorithm (PCGA), which was proposed by the author[37].

Romero Zaliz et al. describe in Chapter 18 an approach for identifying interesting qualitative features in biological sequences. The approach is called Generalized Analysis of Promoters (GAP) and is based on the use of generalized clustering techniques where the features being sought correspond to the solutions of a multiobjective optimization problem. A MOEA is then used to identify multiple promoters occurrences within genomic regulatory regions. The MOEA adopted is a Multi-Objective Scatter Search (MOSS) algorithm.

Lamont et al. present in Chapter 19 an application of the multi-objective messy genetic algorithm-II (MOMGA-II) to two NP-complete problems: the multi-objective Quadratic Assignment Problem (mQAP) and the Modified Multi-objective Knapsack Problem (MMOKP).

1.7.3. *Part III: Industrial Applications*

The third part of the book, which focuses on real-world industrial applications of MOEAs, includes chapters from 20 to 24.

In Chapter 20, Anderson uses a MOEA to design fluid power systems. The MOEA adopted is called multi-objective struggle genetic algorithm (MOSGA) and was proposed by the same author[1]. The approach is further extended so that it can deal with mixed variable design problems (i.e., with both continuous and discrete variables).

Mansouri presents in Chapter 21 the application of a MOEA in cellular manufacturing systems. The problem tackled consists of deciding on which parts to subcontract and which machines to duplicate in a cellular manufacturing system wherein some exceptional elements exist. The MOEA adopted is the NSGA[53].

Chapter 22, by Ishibuchi and Shibata, presents the solution of flowshop scheduling problems (both single- and multi-objective) using genetic algorithms. The multi-objective instances are solved using the NSGA-II[21]. The authors recommend the use of mating restrictions and a hybridization with local search in order to improve the performance of the MOEA adopted.

Gandibleux et al. deal in Chapter 23 with multi-objective combinatorial optimization problems. The approach adopted in this case is peculiar, since it is a population-based heuristic that uses three operators: crossover, pathrelinking and a local search on elite solutions. However, this approach differs from a MOEA in two main aspects: (1) it does not use Pareto ranking, and (2) it performs no direction searches to drive the approximation process. The authors apply their approach to the bi-objective assignment problem and to the bi-objective knapsack problem.

In Chapter 24, Watanabe and Hiroyasu apply a MOEA to the solution of the multi-objective rectangular packing problem, which is a discrete combinatorial optimization problem that arises in many applications (e.g., truck packing and floor planning, among others). The MOEA adopted is the Neighborhood Cultivation Genetic Algorithm (NCGA) which was proposed by the authors[58].

1.7.4. *Part IV: Miscellaneous Applications*

The fourth and last part of the book, deals with miscellaneous applications of MOEAs in a variety of domains, and includes chapters from 25 to 30.

Pappa et al. present in Chapter 25 the use of MOEAs to select attributes in data mining. The authors use two approaches that were previously proposed by them: (1) an elitist multi-objective genetic algorithm (which uses Pareto dominance) in which all the nondominated solutions found pass unaltered to the next generation[42] and (2) a multi-objective forward sequential selection method[43].

In Chapter 26, Schlottmann and Seese present a fairly detailed survey of the use of MOEAs in portfolio management problems. The authors emphasize the importance of the incorporation of problem-specific knowledge into a MOEA as to improve its performance in such financial applications. The authors also identify some other potential applications of MOEAs in finance.

Chapter 27, by Jin et al., describes the application of a MOEA to the evolution of both the weights and the structure of neural networks used for regression and prediction. The MOEA adopted is the NSGA-II[21], expanded with Lamarckian inheritance. The authors report success of the MOEA to generate diverse neural network ensemble members, which significantly improves the regression accuracy, particularly in cases in which a single network is not able to predict reliably.

In Chapter 28, Fieldsend and Singh use a MOEA to train neural networks used for time series forecasting. The MOEA adopted is a variation of PAES[36]. The most interesting aspect of this work is that the use of a multi-objective approach allows the user to get a good representation of the complexity/accuracy trade-off of the problem being solved. This may lead to the selection of neural networks with very low complexity.

Chapter 29, by Ducheyne et al., presents the application of MOEAs in forest management problems (particularly forest scheduling problems). Two MOEAs are studied by the authors: MOGA[23] and the NSGA-II[21]. An interesting aspect of this work is the use of fitness inheritance[52] to speed up the optimization process.

Finally, in Chapter 30, Landa Silva and Burke propose the use of diversity measures to guide a MOEA's search. Such an approach is used to solve space allocation problems arising in academic institutions. The MOEA adopted is called Population-based Hybrid Annealing Algorithm and was previously proposed by the same authors. In this approach, each individual is evolved by means of local search and a specialized mutation operator. This MOEA combines concepts of simulated annealing, tabu search, evolutionary algorithms and hillclimbing.

1.8. General Comments

As has been seen in the previous presentation, this book includes a wide variety of applications of MOEAs. Nevertheless, if we consider the important growth of the number of publications related to MOEAs in the last few years, it is likely that we will see more novel applications in the near future. As a matter of fact, there are still several areas in which applications of MOEAs are rare (e.g., computer vision, operating systems, compiler design, computer architecture, and business activities among others).

The application of MOEAs to increasingly challenging problems is triggering more research on MOEA algorithmic design as well as influencing developmental trends. For example, the hybridization of MOEAs with other mechanisms (e.g., local search) may become standard practice in complex MOP application domains.

This volume constitutes an initial attempt to collect a representative sample of contemporary MOEA applications, thus providing insight to their efficient and effective use. Of course, it is expected that more and more specialized monographs and textbooks will include the use of MOEAs in diverse problem domains because of the expanding understanding and utility of MOEA concepts in solving complex high-dimensional MOPs.

References

1. Johan Andersson. *Multiobjective Optimization in Engineering Design—Applications to Fluid Power Systems*. PhD thesis, Division of Fluid and Mechanical Engineering Systems. Department of Mechanical Engineering. Linköping Universitet, Linköping, Sweden, 2001.
2. Masao Arakawa and Ichiro Hagiwara. Development of Adaptive Real Range (ARRange) Genetic Algorithms. *JSME International Journal, Series C*, 41(4):969–977, 1998.
3. Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.
4. Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, New York, 1997.
5. Richard Balling. The Maximin Fitness Function; Multiobjective City and Regional Planning. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 1–15, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.
6. Richard S. Barr, Bruce L. Golden, James P. Kelly, Mauricio G. C. Resende, and Jr. William R. Stewart. Designing and Reporting on Computational Experiments with Heuristic Methods. *Journal of Heuristics*, 1:9–32, 1995.
7. 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.
8. 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.
9. Carlos A. Coello Coello. Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308, 2000.
10. Carlos A. Coello Coello. A Short Tutorial on Evolutionary 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 21–40. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
11. Carlos A. Coello Coello and Arturo Hernández Aguirre. Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization Approach. *Artificial Intelligence for Engineering, Design, Analysis and Manufacture*, 16(1):39–53, January 2002.
12. 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.
13. 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.

14. 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, Juan Julian 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.
15. H. Crowder, R. S. Demo, and J. H. Mulvey. On Reporting Computational Experiments with Mathematical Software. *ACM Transactions on Mathematical Software*, 5(2):193–203, June 1979.
16. Xunxue Cui, Miao Li, and Tingjian Fang. Study of Population Diversity of Multiobjective Evolutionary Algorithm Based on Immune and Entropy Principles. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 1316–1321, Piscataway, New Jersey, May 2001. IEEE Service Center.
17. 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.
18. Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.
19. 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.
20. Kalyanmoy Deb and David E. Goldberg. mga in C: A Messy Genetic Algorithm in C. Technical Report 91008, Illinios Genetic Algorithms Laboratory (IlliGAL), September 1991.
21. Kalyanmoy Deb, Amrit Pratap, 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.
22. David B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. The Institute of Electrical and Electronic Engineers, New York, 1995.
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, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
24. António Gaspar-Cunha and José A. Covas. RPSGAe–Reduced Pareto Set Genetic Algorithm: Application to Polymer Extrusion. In Xavier Gandibleux,

Marc Sevaux, Kenneth Sörensen, and Vincent T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 221–249, Berlin, 2004. Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535.

25. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
26. P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.
27. 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.
28. 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.
29. Richard H. F. Jackson, Paul T. Boggs, Stephen G. Nash, and Susan Powell. Guidelines for Reporting Results of Computational Experiments – Report of the Ad Hoc Committee. *Mathematical Programming*, 49:413–425, 1991.
30. Andrzej Jaszkiewicz. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—a comparative experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, August 2002.
31. Nazan Khan. Bayesian optimization algorithms for multiobjective and heirarchically difficult problems. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, July 2003.
32. Hajime Kita, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. 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 504–512, Berlin, Germany, September 1996. Springer-Verlag.
33. Mark R. Knarr, Mark N. Goltz, Gary B. Lamont, and Junqi Huang. *In Situ* Bioremediation of Perchlorate-Contaminated Groundwater using a Multi-Objective Parallel Evolutionary Algorithm. In *Congress on Evolutionary Computation (CEC'2003)*, volume 1, pages 1604–1611, Piscataway, New Jersey, December 2003. IEEE Service Center.
34. Joshua Knowles and David Corne. M-PAES: A Memetic Algorithm for Multiobjective Optimization. In *2000 Congress on Evolutionary Computation*, volume 1, pages 325–332, Piscataway, New Jersey, July 2000. IEEE Service Center.
35. Joshua Knowles and David Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.
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. Rajeev Kumar and Peter Rockett. Improved Sampling of the Pareto-Front in Multiobjective Genetic Optimizations by Steady-State Evolution: A Pareto Converging Genetic Algorithm. *Evolutionary Computation*, 10(3):283–314, Fall 2002.
38. Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.
39. Samir W. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Department of General Engineering, Urbana, Illinois, May 1995.
40. Kaisa M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.
41. Tadahiko Murata, Hisao Ishibuchi, and Mitsuo Gen. Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 82–95. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
42. Gisele L. Pappa, Alex A. Freitas, and Celso A.A. Kaestner. Attribute Selection with a Multiobjective Genetic Algorithm. In G. Bittencourt and G.L. Ramalho, editors, *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence (SBIA-2002)*, pages 280–290. Springer-Verlag. Lecture Notes in Artificial Intelligence Vol. 2507, 2002.
43. Gisele L. Pappa, Alex A. Freitas, and Celso A.A. Kaestner. A Multiobjective Genetic Algorithm for Attribute Selection. In *Proceedings of the 4th International Conference on Recent Advances in Soft Computing (RASC-2002)*, pages 116–121, Nottingham, UK, December 2002. Nottingham Trent University.
44. Ian C. Parmee. *Evolutionary and Adaptive Computing in Engineering Design*. Springer, London, 2001.
45. Martin Pelikan and David E. Goldberg. Heirarchical problem solving and the bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 267–274, 2000.
46. Jacques Périaux, Mourad Sefrioui, and Bertrand Mantel. GA Multiple Objective Optimization Strategies for Electromagnetic Backscattering. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, chapter 11, pages 225–243. John Wiley and Sons, West Sussex, England, 1997.
47. R. S. Rosenberg. *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Harbor, Michigan, 1967.
48. J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, 1984.
49. 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.

50. Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.
51. K. J. Shaw and P. J. Fleming. Initial Study of Practical Multi-Objective Genetic Algorithms for Scheduling the Production of Chilled Ready Meals. In *Proceedings of Mendel'96, the 2nd International Mendel Conference on Genetic Algorithms*, Brno, Czech Republic, September 1996.
52. Robert E. Smith, Bruce A. Dike, and S. A. Stegmann. Fitness Inheritance in Genetic Algorithms. In *Proceedings of the 1995 ACM Symposium on Applied Computing*, pages 345–350, Nashville, Tennessee, USA, February 1995. ACM.
53. N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.
54. Gregorio Toscano Pulido and Carlos A. Coello Coello. Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer. In Kalyanmoy Deb et al., editor, *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference*, pages 225–237, Seattle, Washington, USA, June 2004. Springer-Verlag, Lecture Notes in Computer Science Vol. 3102.
55. 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, Ohio, May 1999.
56. 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.
57. David A. Van Veldhuizen and Gary B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. In *2000 Congress on Evolutionary Computation*, volume 1, pages 204–211, Piscataway, New Jersey, July 2000. IEEE Service Center.
58. Shinya Watanabe, Tomoyuki Hiroyasu, and Mitsunori Miki. Neighborhood Cultivation Genetic Algorithm for Multi-Objective Optimization Problems. In Lipo Wang, Kay Chen Tan, Takeshi Furuhashi, Jong-Hwan Kim, and Xin Yao, editors, *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, volume 1, pages 198–202, Orchid Country Club, Singapore, November 2002. Nanyang Technical University.
59. 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.
60. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computa-*

tion, 8(2):173–195, Summer 2000.

61. Eckart Zitzler, Marco Laumanns, and Stefan Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In Xavier Gandibleux, Marc Sevaux, Kenneth Sörensen, and Vincent T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 3–37, Berlin, 2004. Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535.
62. Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece, September 2001.
63. 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.
64. Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.
65. 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.
66. Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.
67. Jesse Zydallis. *Explicit Building-Block Multiobjective Genetic Algorithms: Theory, Analysis, and Development*. PhD thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, March 2003.
68. Jesse B. Zydallis, David A. Van Veldhuizen, and Gary B. Lamont. A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA–II. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 226–240. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.