Evolutionary Multi-Objective Optimization in Materials Science and Engineering

Carlos A. Coello Coello and Ricardo Landa Becerra
CINVESTAV-IPN
Evolutionary Computation Group
Departamento de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D. F. 07360, MEXICO
ccoello@cs.cinvestav.mx, rlanda@computacion.cs.cinvestav.mx

Abstract

This paper provides a short introduction to the evolutionary multi-objective optimization field. The first part of the paper discusses the most representative multi-objective evolutionary algorithms that have been developed, from a historical perspective. In the second part of the paper, some representative applications within materials science and engineering are reviewed. In the final part of the paper, some potential areas for future research in this area are briefly described.

1 Introduction

In engineering (as well as in many other disciplines), it is very common to face problems having two or more objectives that we want to optimize at the same time and which are normally expressed in different units and are in conflict with each other. These are the so-called "multi-objective optimization problems", and have been studied since the XIX century, in which a notion of optimality was specifically stated for them, in the context of economics [1, 2]. Multi-objective optimization problems have been studied within Operations Research since then, and a variety of mathematical programming techniques currently exist to solve them [3].

On the other hand, evolutionary algorithms are metaheuristics[1] inspired on the "survival of the fittest"

[1] The term *metaheuristic* refers to an approach which tries to combine basic *heuristic* methods in higher level frameworks aimed at efficiently and effectively exploring a search space [4]. The term "meta" means "beyond" (in a sense of "higher level") and "heuristic" means "to find". A *heuristic* is a technique which seeks good (i.e., near optimal) solutions at a reasonable computational cost without being able to guarantee optimality of the solutions that it obtains [5].

principle from Darwin's evolutionary theory, that have been used as search and optimization techniques since the 1960s, in a wide variety of disciplines. Although the first hint regarding the potential use of evolutionary algorithms arose in the late 1960s, it was until the mid-1980s that the first actual implementation of a multi-objective evolutionary algorithm (MOEA) took place [6]. Since then, this area, which is now called "evolutionary multi-objective optimization" (EMO for short) has grown in a very significant manner.[2]

This paper presents a general overview of the EMO field, including some of its applications in materials science and engineering. The remainder of this paper is organized as follows. Section 2 presents some basic concepts required to make this paper self-contained. In Section 3, we provide a brief description of the historical origins of multi-objective optimization, as well as on the use of evolutionary algorithms in this field. Section 4 briefly describes the first development period of the EMO field, which covers approximately 14 years (from 1984 up to late 1998). Section 5 briefly describes the most representative algorithms from the second period of this field (which we are currently living). Section 6 reviews some representative applications of MOEAs in materials science and engineering. Some potential paths for future research are sketched in Section 7. Finally, Section 8 contains our conclusions.

2 Basic Concepts

In this paper, we are interested in the solution of multi-objective optimization problems (MOPs) of the form:

$$\text{minimize } \left[f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})\right] \tag{1}$$

subject to the m inequality constraints:

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

and the p equality constraints:

$$h_i(\vec{x}) = 0 \quad i = 1,2,\ldots,p \tag{3}$$

where k is the number of objective functions $f_i : \Re^n \rightarrow \Re$. We call $\vec{x} = \left[x_1, x_2, \ldots, x_n\right]^T$ 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 $x^*_1, x^*_2, \ldots, x^*_n$ which yield the optimum values of all the objective functions.

2.1 Pareto optimality

[2] The first author maintains the EMOO repository, which, as of July 2008, contains over 3400 bibliographic entries. The EMOO repository is located at: `http://delta.cs.cinvestav.mx/~ccoello/EMOO`

It is rarely the case that there exists a single point that simultaneously optimizes all the objective functions.[3] Therefore, we normally look for "trade-offs", rather than single solutions when dealing with multi-objective optimization problems. The notion of "optimality" is therefore, different. The most commonly adopted notion of optimality is that originally proposed by Francis Ysidro Edgeworth [1] and later generalized by Vilfredo Pareto [2]. Although it is more appropriate to call this notion *Edgeworth-Pareto optimality*, we will use the most commonly accepted term: *Pareto optimality*.

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

In words, this definition says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\vec{x} \in \mathcal{F}$ 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 $\vec{x}^*$ 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 Historical Origins of the Field

The interest in solving problems with multiple objectives is not new, since such problems have been studied since the XIX Century. However, the formal development of mathematical programming techniques capable of dealing with multi-objective optimization problems dates back to the late 1950s [7]. Today, an important number of mathematical programming techniques are available in the Operations Research literature in order to deal with multi-objective optimization problems [3, 8]. However, these techniques have a number of limitations. For example, some of them require differentiability and continuity of both the objective functions and the constraints of the problem. Others are sensitive to the shape of the Pareto front. Additionally, in general, these techniques require an initial point to start the search and, therefore, the algorithm needs to be executed several times from different starting points in order to generate different nondominated solutions.[4]

Evolutionary Algorithms (EAs) have been very popular as single-objective optimizers for the last 20 years [9, 10]. Because of their main features, EAs are indeed, very suitable to deal with multi-objective optimization problems. For example, EAs do not require differentiability or continuity of the objective functions or the constraints of a problem. Also, they operate on a set of solutions (the so-called "population"), which makes it possible (if appropriately managed) to generate a set of (different) nondominated solutions in

[3] This would be possible only if there is no conflict among the objectives. In such case, the optimization of each objective, considered separately, would produce this single solution and, therefore, no actual MOEA would be needed.
[4] Note however that the use of different initial points does not, in general, guarantee convergence to different nondominated solutions in each run.

a single run, departing from a random set of initial points. Additionally, the practice has shown that EAs are normally little susceptible to the shape or continuity of the Pareto front [11, 12].

4 Early MOEAs

Rosenberg's PhD thesis, which dates back to the late 1960s [13] contains what seems to be the first reference regarding the use of an EA to solve a MOP. In his PhD thesis, Rosenberg suggests the use of multiple *properties* (nearness to some specified chemical composition) in his simulation of the genetics and chemistry of a population of single-celled organisms. Unfortunately, the genetic algorithm implemented by Rosenberg contained only a single property. The first actual implementation of a MOEA had to wait for almost 20 years until David Schaffer developed (in 1984) the *Vector Evaluated Genetic Algorithm* (VEGA) [6, 14]. VEGA was designed to solve machine learning problems [15].

During the early days of the EMO field (which goes from 1984 up to late 1998), several MOEAs were developed, many of which were straightforward EA extensions of well-known mathematical programming techniques (e.g., linear aggregating functions). However, in 1989, David Goldberg analyzed VEGA in his famous book on genetic algorithms [9] and proposed a ranking scheme based on Pareto optimality. This scheme, which is now called *Pareto ranking*, consists of identifying the set of individuals which are nondominated with respect to the current population. These individuals are assigned the highest (perhaps artificial) rank value and are eliminated from further contention. Another set of individuals which are nondominated with respect to the remainder of the population is then determined and these individuals are assigned the next (possibly artificial) highest rank. This process continues until the entire population is suitably ranked. Goldberg indicated that a traditional EA tends to converge to a single solution if run a sufficiently large number of iterations, because of stochastic noise. Thus, he suggested to use a mechanism to avoid this convergence to a single solution. He suggested the use of fitness sharing [16] or a similar mechanism. The two components suggested by Goldberg (i.e., Pareto ranking and a density estimator) are today part of every modern MOEA.

From the several MOEAs developed in the early days of the EMO field, only three became popular among practitioners (i.e., they were extensively used by researchers different from those who developed them):

1. **The Multi-Objective Genetic Algorithm (MOGA)**: Proposed by Fonseca and Fleming [17] in 1993. In this approach, the rank of an individual corresponds to the number of individuals in the current population by which it is dominated. An interesting aspect of MOGA, is that the ranking of the entire population is done in a single pass, instead of having to reclassify the same individuals several times (as suggested by Goldberg [9]). MOGA used fitness sharing, but its authors proposed a relatively simple way of calculating the niche radius, instead of requiring the user to empirically set up its value.

2. **The Nondominated Sorting Genetic Algorithm (NSGA)**: Proposed by Srinivas and Deb [18], this approach is based on several layers of classifications of the individuals, following Goldberg's idea [9]. 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, fitness sharing is applied to these classified individuals using 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.

3. **The Niched-Pareto Genetic Algorithm (NPGA)**: Proposed by Horn et al. [19], this approach uses a binary tournament selection scheme based on Pareto dominance. Two individuals are randomly chosen and compared against a subset from the entire population (typically, around 10% of the population). There are only two possible outcomes: (1) one of them is dominated (by the individuals randomly chosen from the population) and the other is not; in this case, the nondominated individual wins. The second possible outcome is that the two competitors are either dominated or nondominated (i.e., there is a tie); in that case, the result of the tournament is decided through fitness sharing [16]. Since the NPGA does not rank the entire population, but only a sample of it, it is more efficient (algorithmically) than MOGA and the NSGA.

The few comparative studies among these 3 MOEAs (MOGA, NPGA, and NSGA) performed during the mid and late 1990s, indicated that MOGA was the most effective and efficient approach, followed by the NPGA and by the NSGA (in a distant third place) [20, 21]. MOGA was also the most popular MOEA of its time, mainly within the automatic control community.

5 Modern MOEAs

Elitism in a single-objective EA consists on retaining the best individual from the current generation, and passing it intact (i.e., without being affected by crossover or mutation) to the following generation. Elitism is important, because it allows us to (mathematically) prove convergence [22]. In multi-objective optimization, however, elitism is not straightforward, since all the Pareto optimal solutions are equally good and, in theory, all of them should be retained. Elitism was not emphasized (or even considered) in the early MOEAs described in the previous section. It was until the late 1990s, when elitism in the context of multi-objective

optimization was taken seriously.[5] This was due to two main factors: the first was the proof of convergence of a MOEA developed by [25], which, analogously to the single-objective case, requires elitism. The second was the publication of an elitist MOEA: the **Strength Pareto Evolutionary Algorithm** (SPEA) [26] in the *IEEE Transactions on Evolutionary Computation*, which became a landmark in the field.

This second period of development of the EMO field (that started in the late 1990s and continues until today) has seen the rise of a wide variety of MOEAs. However, as in the early days, very few of these MOEAs have become popular. Next, we will briefly discuss three MOEAs that are representative of this era:

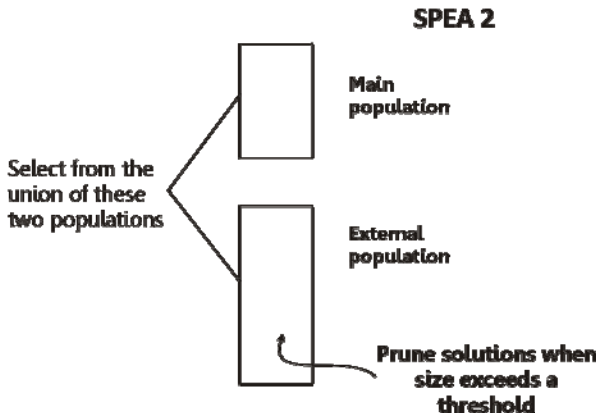

Figure 1: A graphical illustration of the way in which SPEA 2 operates.

1. **The Strength Pareto Evolutionary Algorithm (SPEA)**: Proposed by Zitzler and Thiele [26], this approach was conceived as a way of integrating different MOEAs. It incorporates elitism through the use of an archive containing the nondominated solutions previously found (this is 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 [17], since it is proportional to the number of solutions to which a certain individual dominates. The fitness of each member of the current population is computed according to the strengths of all the external nondominated solutions that dominate it. The fitness assignment process of SPEA considers both closeness to the true Pareto front and even distribution of solutions at the same time. Thus, instead of using niches based on distance, Pareto dominance is used to ensure that the solutions are properly distributed along the Pareto front. Although this approach does not require a niche radius, its effectiveness relies on the size of the external nondominated set. In fact, since the external nondominated set participates in the selection process of SPEA, if its size grows too large, it might reduce the selection pressure, thus slowing down the search. Because of this, the authors decided to adopt a clustering technique that prunes the contents of the external nondominated set so that its size remains below a certain threshold. In 2001, a revised version of SPEA (called **SPEA2**) was introduced. SPEA2 has three

[5] It is worth noting, however, that several researchers had already adopted elitism in the mid-1990s. See for example [23, 24].

main differences with respect to its predecessor [27]: (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. The way in which SPEA2 operates is graphically shown in Figure 1.

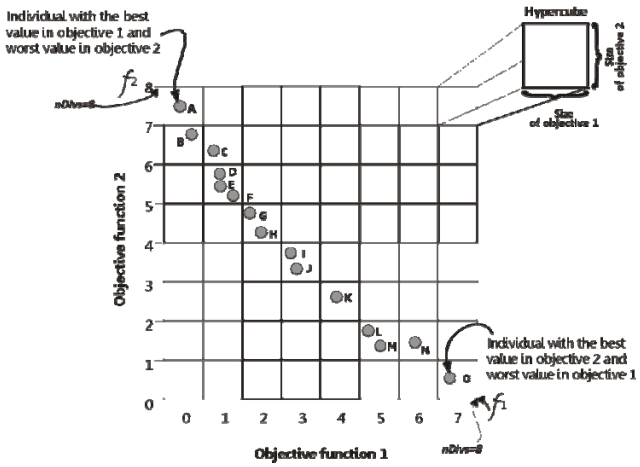

Figure 2: A graphical representation of the external population adopted by PAES.

2. **The Pareto Archived Evolution Strategy (PAES)**: This approach was proposed by Knowles and Corne [28], and it 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 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. The procedure used to maintain diversity 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, as shown in Figure 2. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions to be applied in objective function space).

3. **The Nondominated Sorting Genetic Algorithm II (NSGA-II)**: This approach was proposed by Deb et al. [29] as an upgrade of the NSGA [18], although it shares few similarities with it. In the NSGA-II, for each solution one has to determine how many solutions dominate it and the set of solutions to which it dominates. The NSGA-II estimates the density of solutions surrounding a particular solution in the population by computing the average distance of two points on either side of this point along each of the objectives of the problem. This value is the so-called *crowding*

distance. During selection, the NSGA-II uses a crowded-comparison operator which takes into consideration both the nondomination rank of an individual in the population and its crowding distance (i.e., nondominated solutions are preferred over dominated solutions, but between two solutions with the same nondomination rank, the one that resides in the less crowded region is preferred). The NSGA-II does not implement an elitist mechanism based on an external archive. Instead, the elitist mechanism of the NSGA-II consists of combining the best parents with the best offspring obtained. The way in which the NSGA-II operates is graphically illustrated in Figure 3. Due to its clever mechanisms, the NSGA-II is much more efficient (computationally speaking) than its predecessor, and its performance is so good, that it has gained a lot of popularity in the last few years, becoming a landmark against which other MOEAs are often compared.

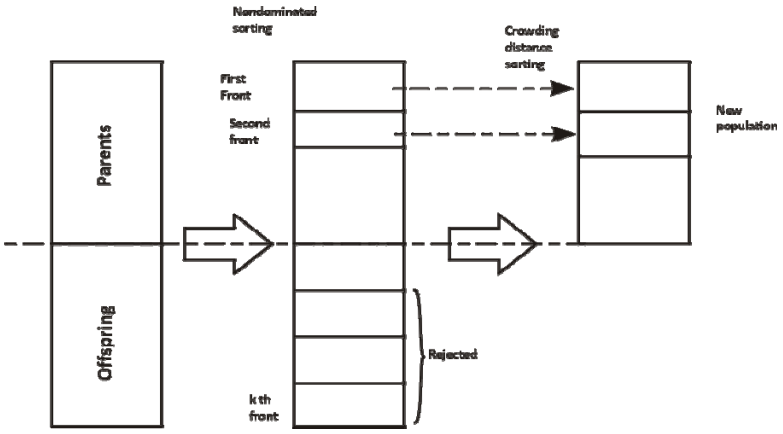

Figure 3: Graphical illustration of the way in which the NSGA-II works.

Many other MOEAs exist (see for example [30, 31, 32, 33]), but they will not be discussed here due to obvious space limitations. For more information on alternative MOEAs and other aspects of the EMO field (for example, density estimators, incorporation of user's preferences and theoretical foundations), interested readers are referred to [11].

6 Applications of MOEAs in Materials Science and Engineering

The use of MOEAs in materials science and engineering has become increasingly popular in the last few years [34, 35, 36, 37, 38, 39, 40]. However, in the review of applications presented next, no attempt was made to be comprehensive for the sake of brevity. Nevertheless, in order to group together the applications reviewed, the following taxonomy (based on the contents of the papers analyzed) was adopted:

- Iron-making and steel-making

- Casting of metals
- Polymerization
- Hot and cold rolling
- Design of alloys
- Reheating and heat treatment

6.1 Iron-making and steel-making

Kumar et al. [41] use SPEA [26] to optimize the gas injection in steelmaking vessels. The authors consider a bottom blown gas-stirred system involving an ascending gas bubble stream in a vessel containing molten metal. Two objectives are considered: (1) maximize the degree of mixing and (2) minimize the wall stress. In order to evaluate these two objective functions, the authors adopt a complex mathematical model for fluid analysis, considering Reynolds averaged transient Navier-Stokes equations, energy and species conservation equations, and equations for turbulent kinetic energy and its dissipation rate. Since each evaluation of the objective functions is very expensive (computationally speaking), SPEA is run only during 20 generations (with a main population size of 60 individuals and an external population size of 30 individuals). The authors indicate, however, that this is enough to achieve reasonably stable solutions.

Saxén et al. [42] evolve neural networks using a genetic algorithm implemented in a predator-prey framework [43]. The objectives minimized are the approximation error and the number of weights of the network (i.e., its size). The authors use a multilayer perceptron, treating the architecture of the lower part of the network and its corresponding weight matrix as decision variables. In contrast, the weights in the upper part of the network were determined by linear least squares. Then, the authors propose an extension of this approach for online learning, in which the weights in the upper layer of the networks are recursively estimated by a Kalman filter. Finally, they apply this approach to the prediction of the silicon content of hot metal produced in a blast furnace, which is a challenging problem in the ironmaking industry. The authors found that their approach could predict the silicon content very accurately, if we consider the fact that the models lack autoregressive terms. They also indicate that an analysis of the final models obtained revealed that the inputs that were retained by the approach correspond to variables that are known to affect (or strongly correlate), in practice, the hot metal silicon content.

6.2 Casting of metals

Miettinen [44] adopts an interactive mathematical programming method[6] (NIMBUS [3]) in which the single-objective optimizations are performed with a real-coded genetic algorithm using an adaptive penalty approach to handle constraints. This hybrid approach is applied to the continuous casting of steel where secondary cooling plays an important role. The main objective is to find a control that produces steel of the best possible quality (i.e., with minimum defects). However, the constraints imposed on the cooling process are so difficult to satisfy that the feasible region turns out to be empty. Thus, the author adopts a model in which the constraints violations are also minimized. The decision variables are the intensities of the water sprays, since they strongly affect the solidification rate of steel, which in turn regulates the quality of the steel (e.g., overcooling can lead to the formation of cracks). The results were very satisfactory, since three of the original constraints could be satisfied relatively well.

Filipič et al. [45] use an approach called Differential Evolution for Multiobjective Optimization [46] (DEMO) for the multi-objective optimization of secondary coolant flows in continuous casting of steel. The authors defined two cost functions related to the deviations of the actual temperatures from the (predefined) targets and to the core length, which is the distance between the mold exit and the point of complete solidification of the slab. These two cost functions are to be minimized. In their model, the authors consider a casting machine with the secondary cooling area divided into nine zones, through which cooling water is dispersed. The authors adopt a mathematical model of the process with a finite element method discretization of the temperature field in order to calculate the temperature field in the slab as a function of the process parameters. The corresponding nonlinear equations are solved using relaxation iterative methods. The authors used four instances of an optimization problem consisting of the casting process for a selected steel grade with a given slab cross-section. These four instances correspond to different casting speeds. In each instance, the speed was kept constant, but at a different value. The values adopted went from 1.2 m/min up to 1.8 m/min. The authors clarify that their study is meant to provide a better understanding of the process and for assessing the casting machine performance, instead of being intended for control purposes. In the study, the authors showed 18 cooling water flows with respect to the two cost functions previously indicated, on an industrial casting machine. It is worth noting that despite the high computational cost of the approach, the authors perform 50,000 solution evaluations (each run required about 55 hours) and adopt performance measures (namely, hypervolume and attainment surfaces) in order to assess the quality of their results. Some of the findings of this study were already known by practitioners, but others indicated the need to ensure uniform distribution of temperature differences over the different zones defined, and also raised the need to add one extra constraint to the model.

6.3 Polymerization

Mitra and Majumdar [47] use the NSGA-II [29] for the performance improvement of the Poly (propylene terepthalate) (PPT) polymerization process. The authors adopt a model in which the aim is to maximize the

[6] Interactive mathematical programming methods are based on linear combinations of the objective functions and the user's preferences (which are iteratively asked). This produces a new single-objective optimization problem that can be solved using any search engine available.

degree of polymerization while minimizing the processing time for the esterification step, since this leads to achieving the desired polymer quality with maximum possible productivity. This leads them to two instances, one of which has three objectives and the other one has four objectives; both problems are nonlinear and have several constraints. The authors adopt a large population size (500 individuals) and run the NSGA-II for 300 generations. The authors indicate that the esterification process shows a reasonable good balance of the required functional groups, while maintaining the goal of having a better polymer product within a reasonable processing time.

Babu et al. [48] use Multi-Objective Differential Evolution (MODE) [49] for the optimization of wet film Poly-Ethylene Terephthalate (PET) reactors. Two objectives are minimized: (1) acid end group concentration and (2) vinyl end group concentration. Results are compared with respect to those reported by Bhaskar et al. [50], which used the NSGA [18], but were able to find only a single nondominated solution. In contrast, MODE produces the Pareto front of this problem (which represent different operating conditions) in the five different cases studied. The two MOEAs are run during 600 generations, but the authors do not indicate the population size nor the values of any of the other parameters adopted.

6.4 Hot and cold rolling

Nandan et al. [51] use two MOEAs for the hot rolling process practiced in a steel plant, considering two objectives. The first is used to ensure that there will be no abrupt jumps in the dimensions of the slabs used in the schedule[7], and the second implies that the total crown in the rolled strip from all sources is kept to a minimum. The two MOEAs adopted are: SPEA [26] and the distance-based Pareto genetic algorithm (DPGA) [52]. The authors adopted position based crossover and position-based mutation [53]. For their experiments, the authors used a batch of 50 different strips actually rolled at an Indian company. The authors only report the use of 500 generations to run their algorithms, but not the population size nor any of the values of the other parameters adopted in their experiments. SPEA was found to have a better performance than DPGA in the problem studied.

Wang et al. [54] use a genetic algorithm with a weighted linear aggregating function to optimize the rolling schedule for a tandem cold strip mill operation. The authors aim to maximize rolling mill throughput while minimizing processing costs and crop losses. The rolling schedule is optimized through the use of a cost function that combines tension, shape and power distribution costs. The constraints considered include the roll force and torque limitations, work roll speed references, strip exit thickness, threading conditions, and tension limitations. The genetic algorithm adopts binary encoding, one-point crossover and bit-flip mutation. The parameters generated by the genetic algorithm are further checked with respect to practical rolling constraints, in order to ensure that they are not unrealistic values. The results indicate that the genetic algorithm can produce an effective improvement with respect to the empirical models used as a reference.

6.5 Design of alloys

[7] The rolling schedule is the sequence in which the strips are to be rolled during a particular campaign.

Mahfouf et al. [55] use both a single-objective genetic algorithm and SPEA2 [27] for the optimal design of alloy steels. The problem consists on determining the optimal heat treatment regime and the required weight percentages for the chemical composites, in order to obtain certain desired mechanical properties in a steel alloy. Since the available physical knowledge of the heat treatment process is not enough to allow the computation of the mechanical properties, the authors adopt elicited data-driven models. Namely, the authors use neural networks to predict the mechanical properties of steel, such as the tensile strength, the reduction of area and the elongation. These are precisely the mechanical properties that were optimized, and two objectives were considered for each of them: the mechanical property itself and its standard deviation. This aims to take into consideration the reliability of the predictions performed. It is worth indicating, however, that three different bi-objective optimization problems are solved in the paper, rather than a single one that considers the six objective functions of the problem. First, a single-objective genetic algorithm is adopted, using a weighted linear aggregating function. The results were acceptable, but the problem of obtaining the appropriate weighting between the two objectives became evident. Thus, in the second part of the paper, the authors adopt SPEA2 to optimize simultaneously each pair of objectives. Much better trade-offs were obtained this time, showing the advantages of using a Pareto-based MOEA in this application.

Egorov-Yegorov and Dulikravich [56] use a semi-stochastic evolutionary algorithm to determine the optimum concentrations of alloying elements in heat-resistant austenitic stainless steel alloys and superalloys that maximize certain mechanical properties of such alloys. The authors adopt an approach called "Indirect Optimization Based on the Self-Organization (IOSO) Algorithm". IOSO consists of two stages. In the first stage, an approximate model of the objective functions is created. In the second stage, this approximate model is optimized. IOSO incorporates evolutionary algorithms, and artificial neural networks with radial basis functions that are used to build the response surfaces. The idea is to use this metamodel (or approximate model) to perform a very reduced number of evaluations of the actual objective functions of the problem. The authors conducted an experiment in which they simultaneously maximized three objectives (strength, temperature and time-to-ruptute) for alloys having 17 chemical elements. In the final part of the paper, the authors deal with the inverse design of alloy compositions. In this case, the idea is to determine chemical compositions of alloys that are able to sustain a specified level of stress at a given temperature for a specified length of time. An example of such inverse design is also included in the paper. The resultant alloys were tested used experimental techniques, which confirmed their characteristics and, in consequence, the validity of the results obtained.

6.6 Reheating and heat treatment

Chakraborti et al. [57] adopt three optimization heuristics (a simple genetic algorithm, a micro-genetic algorithm and differential evolution) with a heat transfer model, for the design of a reheating furnace. In such work, the furnace is modeled with 1D and 2D heat conduction differential equations, which are solved using finite differences approach. The problem is subject to the dropout temperature constraint, which is handled in all the algorithms using the bracket operator penalty parameter approach; this constraint seems to be very

difficult to satisfy for the approaches adopted in this paper. All the algorithms implement tournament selection, and use standard parameters (a population size of 100 for the genetic algorithm and the differential evolution, and of 5 for the micro-genetic algorithm, crossover probability of 0.9, mutation probability of 0.02 but no mutation in the micro-genetic algorithm, crossover constant of 0.5 and scaling factor of 0.8 for the differential evolution). Three decision variables are considered: the three burner temperatures. The authors report similar good results for the genetic algorithm and the differential evolution, which are better, in general, than the results of the micro-genetic algorithm.

Broughton et al. [58] adopt a genetic algorithm for scheduling in a reheat furnace. Two objectives are considered in an aggregative function: the average temperature and the difference between the highest and lowest temperature. A set of heuristic rules is applied for pre-processing and post-processing the schedules, using previous domain knowledge about the problem. A mutation operator based on swapping is applied, to maintain the feasibility of the schedules, in addition to a specialized crossover operator, which is applied partially to the chromosomes. Multiple offspring are produced, and the authors use a population size of 30, and a maximum number of generations of 150; the algorithm is run in real time, so the parameter values must not result in excessive running times. Also, the authors implement a mechanism for the reduction of errors when delays and unexpected events occur.

Sahay et al. [59] perform the optimization in a process of age-hardening of aluminum rods. The first part of this work presents a formal modeling approach for these age-hardening processes. Then, the NSGA-II [29] is adopted, and two objectives are considered: the minimization of yield strength variation and the maximization of furnace productivity; additionally, some experiments are performed with an extra objective: to minimize the furnace temperature. Such experiments show that the Pareto front covers a wide variety of temperatures, and solutions with lower temperatures (low energy consuming) are feasible. The authors also explore the use of multizone furnaces, which increase the number of variables from 2 to 10 (they model a 5-segment furnace). The optimization simulations using such furnaces present an important reduction in energy consumption, while maintaining or improving productivity and quality variations.

7 Future Areas of Research

As we have seen, MOEAs have been successfully applied to different problems in materials science and engineering. However, there are other possible paths for future research that may be worth exploring. For example:

- **Use of Mechanisms to Increase Efficiency:** Several of the applications reviewed have a very high computational cost, which makes it difficult (or even impossible) to perform a large number of objective function evaluations. The use of MOEAs hybridized with local search mechanisms (the so-called memetic MOEAs [60]), the use of approaches that speed up convergence and then reconstruct the Pareto front from a

handful of nondominated points (see for example [61]), and the use of metamodels (see for example [62]) are possible alternatives to explore the search space of this computationally expensive problems in a more efficient way.

- **Use of parallelism:** Considering the high computational cost required by some of the applications in materials science and engineering, it is evidently important to use parallel MOEAs (see for example [63, 64, 65, 66]). It is worth emphasizing that the careful design of such parallel MOEAs is important in order to obtain the maximum possible benefit by improving not only the running time of the algorithm, but also the quality of the solutions obtained.

- **Incorporation of User's Preferences:** It is normally the case that in a real-world problem only a small portion of the Pareto front is needed. Thus, it is very useful when the user can identify regions of interest within the Pareto front that can then be explored in more detail. A number of mechanisms that allow the incorporation of user's preferences into a MOEA already exist (see for example [11, 67, 68, 69]), but their use is not widespread yet.

- Alternative Metaheuristics: There are several other bio-inspired metaheuristics that have been extended for multi-objective optimization and that are becoming increasingly popular in the specialized literature. Three of them deserve special attention, from the authors' perspective, due to their high potential in the materials science and engineering field:

1) **Particle swarm optimization (PSO)**: This metaheuristic was proposed by Kennedy and Eberhard [70], and simulates the movements of a flock of birds that look for food. It is a population-based approach, but does not adopt a crossover or mutation operator. Instead, the search is conducted through the adjustment of the velocity and position of a set of particles that move in the search space following one or more leaders (i.e., those particles with the best performance so far). PSO has become very popular because of its simplicity and effectiveness. Several multi-objective extensions of this metaheuristic already exist [71,72], but relatively few applications of it have been reported in the materials science and engineering field until now.
2) **Artificial immune systems (AIS)**: Our immune system can be seen as a highly parallel intelligent system, which is able to learn and retrieve previous knowledge to solve a variety of recognition and classification tasks. Due to its interesting properties, researchers have proposed computational models of artificial immune systems since the early 1990s [73,74]. Several multi-objective extensions of AIS exist [75,76], but their potential for solving multi-objective classification and pattern recognition problems has not been sufficiently exploited so far.
3) **Ant colony optimization (ACO)**: This metaheuristic is inspired by colonies of real ants, which deposit a chemical substance in the ground, called "pheromone", and which influences the behavior of the ants (they tend to take paths containing a larger amount of pheromone). Computational

models of these ant colonies have been effectively used for solving combinatorial optimization problems [77]. Several multi-objective extensions of ACO already exist [78,79], but their use has been restricted mainly to solve portfolio optimization problems and the traveling salesman problem.

8 Conclusions

This chapter has provided a brief introduction to MOEAs and some of their applications in materials science and engineering. We have briefly described both, the most representative algorithms and some of their applications. Clearly, researchers in materials science and engineering already know and have indeed used MOEAs in several of their applications. Thus, we considered unnecessary to attempt to motivate any further their interest in the EMO field. Also, no attempt was made to provide a critical review of the use of MOEAs in this area, since any of such criticism would be from a purely MOEA designer's perspective. In fact, the aim of this paper is to try to reduce the gap between MOEA designers and the practitioners who apply such algorithms in a variety of complex problems. The feedback that practitioners could provide to the MOEA designers would be, with no doubt, very valuable, and could certainly change the research trends in the EMO field for the years to come.

Acknowledgements

The author acknowledges support from CONACyT through project 45683-Y.

Keywords: evolutionary algorithms, multi-objective optimization, evolutionary multi-objective optimization, metaheuristics, applications of evolutionary algorithms, casting of metals, iron-making, steel-making, polymerization, hot rolling, cold rolling, alloys design, reheating, heat treatment, multi-objective evolutionary algorithms.

References

[1] Francis Ysidro Edgeworth. *Mathematical Psychics*. P. Keagan, London, England, 1881.

[2] Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.

[3] Kaisa M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[4] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.

[5] Colin B. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Great Britain, 1993.

[6] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, 1984.

[7] W. Stadler. Initiators of Multicriteria Optimization. In J. Jahn and W. Krabs, editors, *Recent Advances and Historical Development of Vector Optimization*, pages 3–47. Springer-Verlag, Berlin, 1986.

[8] Matthias Ehrgott. *Multicriteria Optimization*. Springer, Berlin, second edition, 2005. ISBN 3-540-21398-8.

[9] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[10] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin, 2003. ISBN 3-540-40184-9.

[11] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[12] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[13] R. S. Rosenberg. *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Harbor, Michigan, 1967.

[14] 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.

[15] J. David Schaffer and John J. Grefenstette. Multiobjective Learning via Genetic Algorithms. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 593–595, Los Angeles, California, 1985. AAAI.

[16] 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, New Jersey, 1987. Lawrence Erlbaum.

[17] 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.

[18] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[19] 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.

[20] 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.

[21] Carlos Artemio Coello Coello. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*. PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, April 1996.

[22] Günter Rudolph. Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks*, 5:96–101, January 1994.

[23] Phil Husbands. Distributed Coevolutionary Genetic Algorithms for multi-Criteria and Multi-Constraint Optimisation. In Terence C. Fogarty, editor, *Evolutionary Computing. AIS Workshop. Selected Papers*, Lecture Notes in Computer Science Vol. 865, pages 150–165. Springer Verlag, April 1994.

[24] Andrzej Osyczka and Sourav Kundu. A modified distance method for multicriteria optimization using genetic algorithms. *Computers in Industrial Engineering*, 30(4):871–882, 1996.

[25] Günter Rudolph. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 511–516, Piscataway, New Jersey, 1998. IEEE Press.

[26] 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.

[27] 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*, pages 95–100, Athens, Greece, 2002.

[28] Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[29] 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.

[30] 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.

[31] 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.

[32] Sanyou Y. Zeng, Lishan S. Kang, and Lixin X. Ding. An Orthogonal Multi-objective Evolutionary

Algorithm for Multi-objective Optimization Problems with Constraints. *Evolutionary Computation*, 12(1):77–98, Spring 2004.

[33] Kalyanmoy Deb, Manikanth Mohan, and Shikhar Mishra. Evaluating the ε-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation*, 13(4):501–525, Winter 2005.

[34] N. Chakraborti, R. Kumar, and D. Jain. A study of the continuous casting mold using a pareto-converging genetic algorithm. *Applied Mathematical Modelling*, 25:287–297, 2001.

[35] Brian J. Reardon. Optimizing the Hot Isostatic Pressing Process. *Materials and Manufacturing Processes*, 18(3):493–508, 2003.

[36] Pasu Sirisalee, Michael F. Ashby, Geoffrey T. Parks, and P. John Clarkson. Multi-Criteria Material Selection in Engineering Design. *Advanced Engineering Materials*, 6(1–2):84–92, February 2004.

[37] K. Weinert, J. Mehnen, and M. Stautner. The Application of Multiobjective Evolutionary Algorithms to the Generation of Optimized Tool Paths for Multi-Axis Die and Mould Making. In *Intelligent Computation in Manufacturing Engineering, 4th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (CIRP ICME'04)*, pages 405–412, Sorrento, Naples, Italy, July 2004.

[38] S. Tiwari and N. Chakraborti. Multi-objective optimization of a two-dimensional cutting problem using genetic algorithms. *Journal of Materials Processing Technology*, 173:384–393, 2006.

[39] Shubhabrata Datta, Frank Pettersson, Subhas Ganguly, Henrik Saxén, and Niruopam Chakraborti. Designing high strength multi-phase steel for improved strength-ductility balance using neural networks and multi-objective genetic algorithms. *ISIJ International*, 47(8):1195–1203, 2007.

[40] S. Ganguly, S. Datta, and N. Chakraborti. Genetic Algorithms in Optimization of Strength and Ductility of Low-Carbon Steels. *Materials and Manufacturing Processes*, 22(5):650–658, 2007.

[41] A. Kumar, D. Sahoo, S. Chakraborty, and N. Chakraborti. Gas injection in steel-making vessels: Coupling a fluid dynamic analysis with a genetic algorithms-based pareto-optimality. *Materials and Manufacturing Processes*, 20(3):363–379, 2005.

[42] Henrik Saxén, Frank Pettersson, and Kiran Gunturu. Evolving Nonlinear Time-Series Models of the Hot Metal Silicon Content in the Blast Furnace. *Materials and Manufacturing Processes*, 22(5):577–584, 2007.

[43] Xiaodong Li. A Real-Coded Predator-Prey Genetic Algorithm for Multiobjective Optimization. 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 207–221, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[44] Kaisa Miettinen. Using Interactive Multiobjective Optimization in Continuous Casting of Steel. *Materials and Manufacturing Processes*, 22(5):585–593, 2007.

[45] Bogdan Filipič, Tea Tušar, and Erkki Laitinen. Preliminary Numerical Experiments in Multiobjective Optimization of a Metallurgical Production Process. *Informatica*, 31(2):233–240, 2007.

[46] Tea Robič and Bogdan Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 520–533, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[47] Kishalay Mitra and Saptarishi Majumdar. Multicriteria Optimal Control of Polypropylene Terepthalate Polymerization Reactor. *Materials and Manufacturing Processes*, 22(5):532–540, 2007.

[48] B.V. Babu, J.H. Syed Mubeen, and Pallavi G. Chakole. Simulation and Optimization of Wiped-Film Poly-Ethylene Terephthalate (PET) Reactor using Multiobjective Differential Evolution (MODE). *Materials and Manufacturing Processes*, 22(5):541–552, 2007.

[49] B.V. Babu, P.G. Chakole, and J.H.S. Mubeen. Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor. *Chemical Engineering Science*, 60(17):4822–4837, September 2005.

[50] V. Bhaskar, S.K. Gupta, and A.K. Ray. Multiobjective optimization of an industrial wiped film poly(ethylene terephthalate) reactor: some further insights. *Computers & Chemical Engineering*, 25(2–3):391–407, March 2001.

[51] R. Nandan, R. Rai, R. Jayakanth, S. Moitra, N. Chakraborti, and A. Mukhopadhyay. Regulating crown and flatness during hot rolling: A multiobjective optimization study using genetic algorithms. *Materials and Manufacturing Processes*, 20(3):459–478, 2005.

[52] Sourav Kundu and Andrzej Osyczka. The effect of genetic algorithm selection mechanisms on multicriteria optimization using the distance method. In *Proceedings of the Fifth International Conference on Intelligent Systems*, pages 164–168, Reno, Nevada, 1996. International Society for Computers and Their Applications (ISCA).

[53] Mitsuo Gen and Runwei Cheng. *Genetic Algorithms and Engineering Optimization*. Wiley Series in Engineering Design and Automation. John Wiley & Sons, New York, 2000.

[54] D.D. Wang, A.K. Tieu, and Giovanni D'Alessio. Computational intelligence-based process optimization for tandem cold rolling. *Materials and Manufacturing Processes*, 20(3):479–496, 2005.

[55] M. Mahfouf, M. Jamei, and D.A. Linkens. Optimal design of alloy steels using multiobjective genetic algorithms. *Materials and Manufacturing Processes*, 20(3):553–567, 2005.

[56] Igor N. Egorov-Yegorov and George S. Dulikravich. Chemical composition design of superalloys for maximum stress, temperature, and time-to-rupture using self-adaptive surface optimization. *Materials and Manufacturing Processes*, 20(3):569–590, 2005.

[57] Nirupam Chakraborti, Kalyanmoy Deb and Avinash Jha. Genetic algorithm based heat transfer analysis of a bloom re-heating furnace. *Steel Research*, 71(10):396–402, 2000.

[58] Jonathan S. Broughton, Mahdi Mahfouf and Derek A. Linkens. A Paradigm for the Scheduling of a Continuous Walking Beam Reheat Furnace Using a Modified Genetic Algorithm. *Materials and Manufacturing Processes*, 22(5):607-614, 2007.

[59] Satyam S. Sahay, Rajesh Mehta and Karthik Krishnan. Genetic-Algorithm-Based Optimization of an Industrial Age-Hardening Operation for Packed Bundles of Aluminum Rods. *Materials and*

Manufacturing Processes, 22(5):615-622, 2007.

[60] Joshua Knowles and David Corne. Memetic Algorithms for Multiobjective Optimization: Issues, Methods and Prospects. In William E. Hart, N. Krasnogor, and J.E. Smith, editors, *Recent Advances in Memetic Algorithms*, pages 313–352. Springer. Studies in Fuzziness and Soft Computing, Vol. 166, 2005.

[61] Alfredo G. Hernández-Díaz, Luis V. Santana-Quintero, Carlos Coello Coello, Rafael Caballero, and Julián Molina. A New Proposal for Multi-Objective Optimization using Differential Evolution and Rough Sets Theory. In Maarten Keijzer et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 675–682, Seattle, Washington, USA, July 2006. ACM Press. ISBN 1-59593-186-4.

[62] Joshua Knowles. ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, February 2006.

[63] David A. Van Veldhuizen, Jesse B. Zydallis, and Gary B. Lamont. Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):144–173, April 2003.

[64] A.J. Nebro, F. Luna, E.-G. Talbi, and E. Alba. Parallel Multiobjective Optimization. In Enrique Alba, editor, *Parallel Metaheuristics*, pages 371–394. Wiley-Interscience, New Jersey, USA, 2005. ISBN 13-978-0-471-67806-9.

[65] Antonio J. Nebro, Juan J. Durillo, Francisco Luna, Bernabé Dorronsoro, and Enrique Alba. Design Issues in a Multiobjective Cellular Genetic Algorithm. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 126–140, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[66] Antonio López Jaimes and Carlos A. Coello Coello. MRMOGA: A New Parallel Multi-Objective Evolutionary Algorithm Based on the Use of Multiple Resolutions. *Concurrency and Computation: Practice and Experience*, 19(4):397–441, March 2007.

[67] Jürgen Branke and Kalyanmoy Deb. Integrating User Preferences into Evolutionary Multi-Objective Optimization. In Yaochu Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 461–477. Springer, Berlin Heidelberg, 2005. ISBN 3-540-22902-7.

[68] Jiachuan Wang and Janis P. Terpenny. Interactive Preference Incorporation in Evolutionary Engineering Design. In Yaochu Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 525–543. Springer, Berlin Heidelberg, 2005. ISBN 3-540-22902-7.

[69] L. Rachmawati and D. Srinivasan. Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 3385–3391, Vancouver, BC, Canada, July 2006. IEEE.

[70] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.

[71] Carlos A. Coello Coello, Gregorio Toscano Pulido and Maximino Salazar Lechuga. Handling Multiple Objectives with Particle Swarm Optimization, *IEEE Transactions on Evolutionary Computation*, 8(3):256-279, June 2004.

[72] Shubham Agrawal, Yogesh Dashora, Manoj Kumar Tiwari and Young-Jun Son. Interactive Particle Swarm: A Pareto-Adaptive Metaheuristic to Multiobjective Optimization, *IEEE Transactions on Systems, Man, and Cybernetics Part A--Systems and Humans*, 38(2):258-277, March 2008.

[73] Dipankar Dasgupta, editor. *Artificial Immune Systems and Their Applications*. Springer-Verlag, Berlin, 1999.

[74] Leandro N. de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer, London, 2002.

[75] Carlos A. Coello Coello and Nareli Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune System, *Genetic Programming and Evolvable Machines*, 6(2):163-190, June 2005.

[76] Fabio Freschi and Maurizio Repetto. VIS: an artificial immune network for multi-objective optimization, *Engineering Optimization*, 38(8):975-996, December 2006.

[77] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. The MIT Press, USA, 2004.

[78] Inès Alaya and Christine Solnon and Khaled Ghédira. Ant Colony Optimization for Multi-objective Optimization Problems, in *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, Vol. 1, pp. 450-457, IEEE Computer Society Press, October 2007.

[79] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss and C. Stummer. Pareto ant colony optimization with ILP preprocessing in multiobjective portfolio selection, *European Journal of Operational Research*, 171(3):830-841, 16th June, 2006.