Chapter 1

Multi-Objective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering

Antonio López Jaimes and Carlos A. Coello Coello
CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D.F. 07360, MEXICO

Abstract

In this chapter, we provide a general overview of evolutionary multi-objective optimization, with a particular emphasis on the algorithms in current use. Several applications of these algorithms in chemical engineering are also discussed and analyzed. We also provide some additional information about public-domain resources available for those interested in pursuing research in this area. In the final part of the chapter, some potential areas for future research in this area are briefly described.

keywords: evolutionary multi-objective optimization, chemical engineering, metaheuristics, evolutionary algorithms.

1.1 Introduction

The solution of problems having two or more (normally conflicting) objectives has become very common in the last few years, within a wide variety of disciplines. Such problems are called "multi-objective", and can be solved using either mathematical programming techniques (Miettinen, 1999) or using metaheuristics (Coello Coello *et al.*, 2002). In either case, normally

the concept of Pareto optimality is adopted. When using this concept, we aim to obtain the best possible trade-offs among all the objectives.

From the many metaheuristics available, evolutionary algorithms (EAs) have become very popular because of their ease of implementation and high effectiveness. EAs are based on an emulation of the natural selection mechanism (Goldberg, 1989). EAs are particularly suitable for solving multi-objective problems because of their ability to handle a set of solutions in a simultaneous manner, and their capability to deal with problems of different types, without requiring any specific problem-domain information (e.g., derivatives) (Deb, 2001).

The first multi-objective evolutionary algorithms (MOEAs) were introduced in the 1980s (Schaffer, 1985), but they only became popular until the mid-1990s. Nowadays, the use of MOEAs in all disciplines has become widespread (see for example (Coello Coello and Lamont, 2004)), and chemical engineering is, by no means, an exception.

This chapter provides a short introduction to MOEAs, which is presented from a historical perspective. It also reviews some of the most representative work regarding their use in chemical engineering applications. Finally, it provides a short description of some of the main Internet resources currently available for those interested in pursuing research in this area.

The rest of this chapter is organized as follows. Section 1.2 presents some basic concepts related to multi-objective optimization (MOO), which aim to make this chapter self-contained. In Section 1.3, we describe the early MOEAs developed from the mid-1980s up to the late 1990s. Section 1.4 describes the major modern MOEAs in current use. Section 1.5 provides a short introduction regarding the use of MOEAs in chemical engineering. Then, in Section 1.6, some of the most significant MOEAs that were originated in the chemical engineering literature are described. Section 1.7 focuses on chemical engineering applications that rely on well-known MOEAs. Section 1.8 presents some of our final remarks regarding the application of MOEAs in chemical engineering applications, based on the papers reviewed in this chapter. Some additional resources available for those interested in this area are briefly discussed in Section 1.9. In Section 1.10, we present some potential paths for future research in this area. Finally, our conclusions are provided in Section 1.11.

1.2 Basic Concepts

We are interested in the solution of MOO problems (MOOPs) of the form:

$$\text{minimize } [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \tag{1.1}$$

subject to the m inequality constraints:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{1.2}$$

and the p equality constraints:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{1.3}$$

where k is the number of objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. We call $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We wish to determine from among the set $\mathcal{F}$ of all vectors which satisfy (1.2) and (1.3) the particular set of values $x_1^*, x_2^*, \ldots, x_n^*$ which yield the optimum values of all the objective functions.

1.2.1 *Pareto optimality*

The most commonly notion of optimality adopted in multiobjetive optimization is the so-called *Pareto optimality* (Pareto, 1896).

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 $f_j(\vec{x}) < f_j(\vec{x}^*)$ for at least one j.

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 $\vec{x}^*$ correspoding 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*.

1.3 The Early Days

Apparently, Rosenberg's PhD thesis (Rosenberg, 1967) contains the first reference regarding the possible use of an evolutionary algorithm in a

MOOP. 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. This is then, a MOOP. However, Rosenberg's actual implementation contained a single property. Thus, no actual multi-objective evolutionary algorithm (MOEA) is developed in his thesis.

Despite the existence of an early paper by Ito *et al.* (1983), the first actual implementation of a MOEA is normally attributed to David Schaffer, who developed the *Vector Evaluated Genetic Algorithm* (VEGA) in the mid-1980s (Schaffer, 1985). VEGA was designed to solve machine learning problems.

During the period from the mid-1980s up to the first half of the 1990s, few other MOEAs were developed. Most of these approaches had a clear influence of the mathematical programming techniques developed within the operations research community and their implementations were straightforward, since they required very few (and simple) changes in the original structure of their underlying EAs.

In his famous book on genetic algorithms, Goldberg (1989), analyzes VEGA, and indicates its main limitations. Goldberg also proposes a ranking scheme based on Pareto optimality. Such mechanism, which was called *Pareto ranking* would soon become standard within modern MOEAs. The basic idea of Pareto ranking is to find the set of individuals in the population that are Pareto nondominated with respect to the rest of the population. These individuals are then assigned the highest rank and 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 highest rank. This process continues until the population is suitably ranked. Goldberg also suggested the use of some kind of diversification technique to keep the EA from converging to a single Pareto optimal solution. A niching mechanism such as fitness sharing (Goldberg and Richardson, 1987) was suggested for this sake. Three major MOEAs would soon be developed based on these ideas. Each of them is briefly described next.

Fonseca and Fleming (1993) proposed the **Multi-Objective Genetic Algorithm** (MOGA), which soon became a very popular MOEA, because of its effectiveness and ease of use. In MOGA, the rank of an individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual x_i at generation t, which is dominated by $p_i^{(t)}$ individuals in the current generation.

The rank of an individual is given by (Fonseca and Fleming, 1993):

$$\text{rank}(x_i, t) = 1 + p_i^{(t)} \tag{1.4}$$

All nondominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region to which they belong. An interesting aspect of MOGA, is that the ranking of the entire population is done in one pass, instead of having to reclassify the same individuals several times (as suggested by Goldberg (1989)).

Srinivas and Deb (1994) proposed the **Nondominated Sorting Genetic Algorithm** (NSGA) which is based on several layers of classifications of the individuals as suggested by Goldberg (1989). 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. Stochastic remainder proportional 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.

Horn *et al.* (1994) proposed the **Niched-Pareto Genetic Algorithm** (NPGA), which uses a tournament selection scheme based on Pareto dominance. The basic idea of the algorithm is the following: Two individuals are randomly chosen and compared against a subset from the entire population (typically, around 10% of the population). 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 (Goldberg and Richardson, 1987). 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) (Van Veldhuizen, 1999). MOGA was also the most popular MOEA of its time, mainly within the automatic control community.

1.4 Modern MOEAs

During the mid-1990s, several researchers considered a notion of elitism in their MOEAs (Husbands, 1994). 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. In MOO, 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 MOO was taken seriously. This was due to two main factors: the first was the proof of convergence of a MOEA developed by Rudolph (1998), which requires elitism. The second was the publication of the **Strength Pareto Evolutionary Algorithm** (SPEA) (Zitzler and Thiele, 1999) in the *IEEE Transactions on Evolutionary Computation*, which became a landmark in the field.

SPEA was conceived as a way of integrating different MOEAs. It incorporates elitism through the usage of an archive containing nondominated solutions previously found (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set. For each individual in this external set, a *strength* value is computed. This strength is similar to the ranking value of MOGA (Fonseca and Fleming, 1993), since it is proportional to the number of solutions to which a certain individual dominates. In SPEA, the fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. 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.

After the publication of SPEA's paper, most researchers in the field started to incorporate external populations in their MOEAs, as their elitist mechanism. In 2001, a revised version of SPEA (called **SPEA2**) was introduced. SPEA2 has three main differences with respect to its predecessor (Zitzler *et al.*, 2001): (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 **Pareto Archived Evolution Strategy** (PAES) is another major MOEA that was introduced at about the same time that SPEA (Knowles and Corne, 2000). PAES consists of a (1+1) evolution strategy (i.e., a single parent that generates a single offspring) in combination with a historical archive that records the nondominated solutions previously found. This archive is used as a reference set against which each mutated individual is being compared. Such a historical archive is the elitist mechanism adopted in PAES. However, an interesting aspect of this algorithm is the procedure used to maintain diversity which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its "coordinates" or "geographical location"). A map of such grid is maintained, indicating the number of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space).

The **Nondominated Sorting Genetic Algorithm II** (NSGA-II) was introduced as an upgrade of the NSGA (Srinivas and Deb, 1994), although it is easier to identify their differences than their similarities (Deb *et al.*, 2002). 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. 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.

Many other MOEAs exist (see for example (Coello Coello *et al.*, 2002)), but they will not be discussed due to obvious space limitations. In any case, the MOEAs previously discussed are among the most popular within the current literature.

1.5 MOEAs in Chemical Engineering

A wide variety of techniques have been used to solve MOOPs in chemical engineering, including mathematical programming techniques (e.g., goal programming and the ε-constraint method) and MOEAs. This chapter only focuses on MOEAs, but readers interested in the first type of methods should refer to Bhaskar *et al.* (2000) for a review. It is worth noting, however, that since the late 1990s, MOEAs seem to be the preferred choice of practitioners to tackle MOO chemical engineering applications.

After reviewing the relevant literature, we found two types of papers:

(1) Those focusing on novel MOEAs or MOEA components.
(2) Those focusing on novel applications using an existing MOEA.

Section 1.6 briefly describes the most significant MOEAs that originated in the chemical engineering literature. For each algorithm, we mention some of their known applications and their advantages and disadvantages. Section 1.7, on the other hand, presents a selection of some representative MOO applications in chemical engineering that make use of well-known MOEAs. This selection is not meant to be exhaustive but attempt to delineate the current research trends in the area.

1.6 MOEAs Originated in Chemical Engineering

As indicated before, this section is devoted to review works whose main goal is to propose a new MOEA (or an important component of it). Among these novel contributions we can find, for instance, an evolutionary operator, a constraint-handling technique and a proposal to extend a single objective technique in order to deal with multiple objectives. It is important to emphasize that all of the MOEAs discussed in this section originated in the chemical engineering community and have been mainly used to solve chemical engineering problems, although most of them are applicable to other domains.

1.6.1 *Neighborhood and Archived Genetic Algorithm*

Sheng-jing *et al.* (2003) proposed the Neighborhood and Archived Genetic Algorithm (NAGA), whose main goals are to provide (*i*) a new method to check for nondominance and (*ii*) a new technique to keep diversity in the Pareto front produced by the algorithm. In order to fulfill these goals, NAGA carries out neighborhood comparisons. The procedure to check for nondominance in the current population is divided in two stages. First, each new solution is locally compared to its neighbors. If the solution is locally dominated, then it is discarded since it will be globally dominated as well. On the other hand, if it is locally nondominated, then the solution is retained for the second stage. At this stage, only the locally nondominated solutions are compared with the current approximation set stored in a historical archive, using Pareto dominance again. After checking for nondominance, the new nondominated solutions are compared again using a crowding neighborhood process aimed to keep diversity. The implementation described by Sheng-jing *et al.* (2003) only considers one neighbor for each solution $\mathbf{x}$, namely the point resulting from a small perturbation in only one variable of $\mathbf{x}$.

Regarding the crowding neighborhood process, if a new nondominated solution, $\mathbf{x}$, is in the neighborhood of some solution $\mathbf{x}^P$ in the archive, i.e., if $x_i \in [x_i^{(j)} - \varepsilon, x_i^{(j)} + \varepsilon]$, for each variable i and each archive solution j, then the solution is discarded; otherwise, it is added to the archive. The parameter ε is defined by $\varepsilon = d \times (x_i^U - x_i^L)$, where d is a user-defined parameter, and x_i^U and x_i^L are the upper and lower bounds of the i-th variable, respectively. Although, on average, the time required to identify the nondominated solutions is reduced in comparison with the standard

Pareto ranking approach, the neighborhood comparisons introduce an extra evaluation of the objective functions per each individual. That is to say, the number of evaluations per generation is doubled with respect to the standard Pareto ranking approach. This is an important drawback of this approach, particularly in chemical engineering applications where the time required to evaluate the objective functions is usually high. This is, with no doubt, an important issue that must be taken into account before deciding to adopt NAGA in an application.

Applications

NAGA was used by Sheng-jing *et al.* (2004) to optimize the operation of a purified terephthalic acid oxidation process. Here, they consider two objectives: minimization of the concentration of 4-carboxy-benzaldehyde (4-CBA) in the crude terephthalic acid and maximization of the feed flow rate of the reactor vector. They consider four optimization problems according to the number of decision variables used (1, 2, 4 and 6 variables). The problem has 2 constraints. The plot of the Pareto front obtained presented a convex and continuous curve.

Recently, Weifeng *et al.* (2007) used NAGA to maximize the aromatic yield and minimize the yield of heavy aromatics in an industrial naphtha continuous catalytic reforming process (that aims to obtain aromatic products). The authors considered six decision variables: four reactor inlet temperatures, the reactor pressure and the hydrogen-to-oil molar ratio. As in the previous problem, they obtained a Pareto front which is convex and continuous.

1.6.2 *Criterion Selection MOEAs*

Dedieu *et al.* (2003) proposed an algorithm that can be considered as a criterion selection technique (Coello Coello *et al.*, 2002). That is to say, an algorithm where the solutions are selected based on separate objective performance. The main idea is to optimize separately each objective using a single objective genetic algorithm (SOGA). At the end of the single optimizations, the populations are merged to obtain the nondominated individuals. The authors proposed a variant where all populations generated through all generations of the SOGAs are merged. It is interesting to note that the proposed optimization algorithm was coupled with a discrete event simulator (DES), which was used to evaluate the different objectives and

the technical feasibility of the proposed solutions. A detailed description of the DES used can be found in (Bernal-Haro *et al.*, 2002). Since in this case, the objective functions are not defined explicitly, this makes this kind of application an excellent candidate to be solved by an evolutionary algorithm which, in contrast to gradient-based techniques, only needs objective function evaluations. As pointed out by Dietz *et al.* (2006), a drawback of this approach is that it is not able to produce a good distribution of solutions along the Pareto front, since it focuses on finding only a few solutions around the optima of each objective considered separately.

In a further paper, Dietz *et al.* (2006) proposed an approach similar to VEGA (Schaffer, 1985) in order to overcome the disadvantages of the previous proposal. In this new algorithm, k subpopulations of the whole population are ranked and selected according to a different objective (assuming k objective functions). After shuffling the subpopulations together, the crossover and mutation operators are applied in the usual way. This procedure is repeated until the stopping criterion is reached. At the end of the search, a procedure to check dominance is applied to obtain the Pareto set approximation. This algorithm showed a generated Pareto front with better distribution than that of its predecessor.

Applications

The first algorithm (i.e., the one proposed in (Dedieu *et al.*, 2003)) was applied to optimize the design of a multi-objective batch plant for manufacturing four products by using three types of equipment. The problem considers two objectives: minimization of both the investment cost of the plant and the number of sizes for each unit operation. The second algorithm (i.e., the one proposed in (Dietz *et al.*, 2006)) was used to optimize the design of a multi-product batch plant for the production of proteins (human insulin, vaccine for Hepatitis B, chymosine and cryophilic protease). This combinatorial problem presents three objectives, the investment cost and two objectives concerning the environmental impact (total biomass quantity released and volume of polyethylene glycol used), and 44 decision variables: 16 continuous variables (operating conditions) and 28 integer variables (batch plant configuration). The cost objective involves investment cost for both equipment and storage vessels, whereas the evaluation of environmental impact combines three methodologies, namely, the life cycle assessment, the pollution balance principle and the pollution vector methodology. The study considers one problem using the three objectives

mentioned above and two bicriteria problems (alternating the environmental objective).

1.6.3 *The Jumping Gene Operator*

Kasat and Gupta (2003) proposed two new binary mutation operators whose main goal is to accelerate the convergence of the search in terms of the number of generations. These operators, called jumping genes (JG), are the following: (*i*) *replacement operator*, where a randomly selected l-length substring of the chromosome is replaced by a new string with length l generated at random; and (*ii*) *reversion operator*, where a randomly selected substring is reversed. The JG operator is applied to a fraction P_{jump} of the current population after the mutation phase. In order to evaluate the performance of the JG operators, the authors used the NSGA-II (Deb *et al.*, 2002). According to their results, the reversion operator yields similar results than those obtained by NSGA-II without the proposed operator. However, the results of NSGA-II with the replacement operator (NSGA-II-JG) outperform those obtained by the standard NSGA-II. In the three test problems considered and based on visual inspections, NSGA-II-JG showed better convergence and distribution than NSGA-II.

Guria *et al.* (2005) proposed an adaptation of the JG operators aimed to solve network problems like the design of froth flotation circuits (discussed below). In network or circuit optimization problems, usually the optimal configuration includes variable values exactly at their lower or upper bounds. The modified jumping gene operator (mJG) takes this peculiarity into account and does not select a substring at random, but a substring associated to one of the variables (*i.e.*, a gene). The gene selected is then replaced by a new gene that contains all zeros or all ones in accordance with a certain probability.

It is noteworthy that the JG operator has also been successfully incorporated into a Multi-objective Simulated Annealing technique. This algorithm was employed in the optimization of an industrial fluized-bed catalytic cracking unit and the performance assessment included three well-known test problems commonly used in the evolutionary MOO field.

Applications

Kasat and Gupta (2003) used the JG operator in the optimization of an industrial fluized-bed catalytic cracking unit. The objectives considered

are the maximization of the yield of gasoline and the minimization of the coked formed on the catalyst during the cracking of heavy compounds. The decision variables included the feed preheat temperature, the air preheat temperature, the catalyst flow rate, and the air flow rate. The CPU time required for running the NSGA-II-JG for 50 generations on a Pentium IV, 1.7 MHz PC was 48 hours. In order to evaluate the performance of the algorithm, the optimization problem was also solved using sequential quadratic programming (SQP) with the ε-constraint method. According to the results obtained using six different values of ε, SQP failed to converge to the correct solution.

The modified JG (mJG) operator was used by Guria *et al.* (2005) to optimize the design of froth flotation circuits for mineral processing. In particular, they optimized a circuit with two flotation cells and two species. This problem involves the maximization of the recovery (ratio of the flow rates of the solid in the concentrate stream to that in the feed stream) and maximization of the grade (the fraction of the valuable mineral in the concentrate stream). The problem comprises 16 decision variables, namely, 14 cell linkage parameters and 2 mean residence times. The problem contains 3 constraints related to the streams and 1 constraint related to the total volume of the cells. Recently, Guria *et al.* (2006) applied the mJG operator to optimize circuits with more than two cells and also considering problems with three and four objectives. A three-objective problem (maximization of the overall recovery of the concentrate, maximization of the number of non-linking streams and minimization of the total cell volume) is then solved. All the problems constrain the grade of the product to lie at a fixed value. Finally, a complex and computationally intensive four-objective optimization problem is solved.

1.6.4 *Multi-Objective Differential Evolution*

Differential evolution (DE) is a branch of evolutionary algorithms developed by Storn and Price (1997) for optimization problems over continuous domains. DE is characterized by representing the variables by real numbers and by its three-parents crossover. At the selection stage, three parents are chosen and they generate a single offspring by adding the weighted difference vector between two parents to a third parent. The offspring is compared with a parent to determine who passes to the following generation. DE has been very successful in the solution of a variety of continuous single-objective optimization problems in which it has shown a great ro-

bustness and a very fast convergence. Recently, there have been several proposals to extend DE to MOO (Robič and Filipič, 2005). This section is devoted to present one of these extensions which has been used mainly to solve chemical engineering applications. The Multi-Objective Differential Evolution algorithm (MODE) was proposed by Babu and Jehan (2003). Its general framework is very similar to that of the standard DE. The main differences are: (*i*) the F parameter is generated from a random generator between 0 and 1; (*ii*) only the nondominated solutions are retained for recombination; (*iii*) the generated offspring is placed into the population if it dominates the first selected parent; and (*iv*) the constraints are handled using a penalty function approach.

Applications

MODE was used by Babu *et al.* (2005) to optimize the operation of an adiabatic styrene reactor. This work concerns a comparative study between the performance of MODE and the results of NSGA reported in a previous paper (Yee *et al.*, 2003). This application is described in Section 1.7.3. For comparative purposes, this study adopts the same formulation used by Yee *et al.* (2003). That is to say, the objectives are productivity, selectivity and yield of styrene; the variables are ethyl benzene feed temperature, pressure, steam over reactant ratio and initial ethyl benzene flow rate; and two constraints are also considered.

On the one hand, the results obtained by MODE agreed with those obtained by the NSGA, in particular the behavior of the variables in the Pareto optimal set. On the other hand, based on visual inspections, it was revealed that, in some cases, the Pareto fronts obtained by MODE were better than those obtained by NSGA, while in other cases the Pareto fronts seemed nearly identical (no performance indicators were adopted in this case).

1.7 Some Applications Using Well-known MOEAs

The aim of this section is to present a selection of MOO chemical engineering applications that were solved using a well-known MOEA (e.g., MOGA (Fonseca and Fleming, 1993) or NSGA-II (Deb *et al.*, 2002)) with some small adaptations suitable to the given application. We also found that some authors developed their own approaches based on mechanisms of existing MOEAs (for example, the nondominated sorting mechanism of the

NSGA-II (Deb *et al.*, 2002)).

The applications treated in this section are divided in four types:

(1) TYPE I: Related to polymerization processes.
(2) TYPE II: Involve catalytic reactors.
(3) TYPE III: Related to catalytic processes.
(4) TYPE IV: General applications.

As we will see later on, in some cases, more than one work was found to address the same application.

1.7.1 *TYPE I: Optimization of an Industrial Nylon 6 Semibatch Reactor*

Mitra *et al.* (1998) employed the NSGA (Srinivas and Deb, 1994) to optimize the operation of an industrial nylon 6 semibatch reactor. The two objectives considered in this study were the minimization of the total reaction time and minimization of the concentration of undesirable cyclic dimer in the polymer produced. The problem involves two equality constraints: one to ensure a desired degree of polymerization and the other to ensure a desired value of monomer conversion. The former was handled using a penalty function approach whereas the latter was used as a stopping criterion of the integration of the model equations. The decision variables were the vapor release rate history from the semibatch reactor and the jacket fluid temperature. It is important to note that the former variable is a function of time. Therefore to encode it properly as a sequence of variables, the continuous rate history was discretized into several equally spaced time points. The experimental study showed that NSGA solutions were superior to the solutions obtained by the Pontryagin's minimum principle.

1.7.2 *TYPE I: Optimization of an Industrial Ethylene Reactor*

Tarafder *et al.* (2005b) applied the NSGA-II (Deb *et al.*, 2002) to study an industrial ethylene reactor following a MOO approach. The authors selected a free-radical mechanism to model the reactor. Three objectives were considered in this study, namely ethane conversion, ethylene selectivity and flow rate of ethylene. From these objectives four maximization problems were formulated. First, a bi-objective optimization problem including ethane conversion and ethylene selectivity since these objectives had a con-

flicting behavior. Further, the flow rate, which depends on conversion and selectivity was included in two bi-objective problems: flow rate–conversion and flow rate–selectivity. Finally a three-objective problem was formulated including all three objectives. The problem involved 9 decision variables (7 continuous and 2 discrete). In order to verify the quality of the obtained Pareto front, an ε-constraint method was applied to generate some solutions in the middle and in the extreme of the non-convex front. The results showed these solutions lie on the pareto front obtained by NSGA-II. Furthermore, the results obtained by NSGA-II were compared, in all cases, with the industrial solution predicted. According to the results, the industrial data solution was dominated by some NSGA-II solutions in all the optimization problems.

1.7.3 *TYPE II: Optimization of an Industrial Styrene Reactor*

Yee *et al.* (2003) make use of the original NSGA (Srinivas and Deb, 1994) to optimize both adiabatic and steam-injected styrene reactors. The reactor model employed was a pseudo-homogeneous model. This study comprises three objectives: maximization of the amount of styrene produced, maximization of selectivity of styrene and maximization of yield of styrene. Two- and three-objective optimization problems are studied using combinations from these objectives. The variables for the adiabatic configuration are ethyl benzene feed temperature, inlet pressure, steam to ethyl benzene molar ratio and initial ethyl benzene flow rate. The problem considers three constraints related to temperatures which are handled using penalty functions. According to the plot of the Pareto front obtained, in two of the bi-objective cases the solutions obtained by NSGA are better than the known industrial operation point. For the rest of the cases, the industrial operation point seems to lie on the Pareto front. Concerning the adiabatic configuration, one of the most interesting findings of this study is that in all the bi-objective problems only one variable changes while the others remain nearly constant. Additionally, the results confirm that the steam injection is better than the adiabatic operation. In a further paper, Tarafder *et al.* (2005a) carried out the optimization of the entire styrene manufacturing process, which besides the styrene monomer reactor, includes the heat-exchangers and the separation units. Instead of adopting the original NSGA, in this case the more recent NSGA-II is used (Deb *et al.*, 2002). In this study, three objectives are considered: the maximization of styrene

production, maximization of selectivity of styrene and the minimization of the total heat duty. This last objective reduces the emission of gases such as CO_x, SO_x and NO_x to the environment. Apart from the four variables included in the previous work, four more variables are added. Although three different reactor designs are studied (single bed, double bed and stream injection), only the double bed reactor was considered for the three-objective optimization. The problem comprises six constraints which are handled using the technique already incorporated in the NSGA-II. According to the authors, this approach achieved better results than those obtained by the penalty function approach.

1.7.4 *TYPE II: Optimization of an Industrial Hydrocracking Unit*

Bhutani *et al.* (2006) carried out the optimization of an industrial hydrocracking unit using the NSGA-II (Deb *et al.*, 2002). Hydrocracking is a catalytic cracking process for the conversion of feedstock into more valuable lower boiling products. The optimization of a hydrocracking unit involves many objectives and variables. In this study, the authors considered three optimization problems depending, mainly, on the objectives chosen. The first case comprises the maximization of kerosene and the minimization of the makeup of hydrogen. The decision variables are the flow rate of feed, recycle gas mass flow rate and temperature, recycle oil temperature, recycle oil mass fraction and quench flow rates to the catalyst beds. The maximum allowable inlet and exit temperatures for the hydrocracking are taken as the constraints. The second case considers the maximization of heavy diesel and minimization of makeup of hydrogen. The third case involves the maximization of high-value end products and minimization of low-value end products. The hydrocracking unit was modeled using a discrete lumped model approach, where the individual components in the reaction mixture are divided into discrete pseudo-compounds (lumps). Interestingly, the calibration of the model parameters was carried out employing a genetic algorithm. The Pareto optimal set obtained by NSGA-II shows the conflict between the objectives in the three cases studied. Also, the results show that the current industrial solution is clearly dominated by the Pareto set obtained by NSGA-II.

1.7.5 *TYPE III: Optimization of Semi-batch Reactive Crystallization Process*

Reactive crystallization is a production step for a wide range of chemical and pharmaceutical industries to produce solid particles with desirable characteristics, such as large crystal size, narrow crystal-size distribution, high-yield and so on. The feed flow rate of the reactants is key control variable to improve the quality of the product crystals. Sarkar *et al.* (2007) carried out the optimization of a semi-batch reactive crystallization process using the NSGA-II (Deb *et al.*, 2002). Since the quality of the product crystals is usually defined by the weight mean size and coefficient variation, the authors selected these parameters as the two objectives of the problem. The amount of reactants added at certain intervals was used as the decision that defines the feed addition profile. In order to define a feed profile in this way, the total time was divided into P equal-length intervals and each of these intervals has associated an amount of reactant added. Thus, the number of decision variables to define a feed profile is P. The optimization problem presents three inequality constraints which are managed with the constraint-handling mechanism embedded into the NSGA-II. The particular instance of the problem studied involves the precipitation of barium sulfate from aqueous feed stream of barium chloride and sodium sulfate. The total batch time (180 seconds) was divided into ten intervals. In order to verify the closeness of the obtained solutions to the true Pareto front, some solutions are generated executing repeatedly a weighted sum approach. Based on a visual comparison, the authors conclude that the solutions obtained using the weighted sum approach lie near the Pareto front obtained by the NSGA-II.

1.7.6 *TYPE III: Optimization of Simulated Moving Bed Process*

Yu *et al.* (2003a) carried out the optimization in the design and operation of reactive simulated moving bed (RSMB) using the NSGA (Srinivas and Deb, 1994) for the synthesis of methyl acetate (MeOAc). The model adopted was defined by Yu *et al.* (2003b). The study considers three optimization problems. The first problem concerns the maximization of purity and yield of methyl acetate and has two constraints. The decision variables are the switching time and the eluent flow rate. The results showed that the switching time plays a key role in determining the obtained Pareto

front. The second problem is to minimize the absorbent requirement and the eluent flow. The variables are the length of each column and the column configuration (i.e., the number of columns in each section). The last problem involves the maximization of purity and yield of methyl acetate and the minimization of eluent consumption.

In a previous work, Zhang *et al.* (2002) used a MOO approach to compare the performance between the SMB process and a recently developed variant of SMB called the VARICOL process (Ludemann-Hombourger *et al.*, 2000). For comparison purposes, the authors use the same model for the SMB and VARICOL processes developed by Ludemann-Hombourger *et al.* (2000). Besides the single-optimization problem used for verification purposes, the study considered two MOOPs. The first problem involves the maximization of purity of both raffinate and extract streams (using the feed flow rate and eluent compsumption as parameters). The variables are then, the fluid flow rate in section one of the SMB system, the switching time and the column configuration. This case considers two inequality constraints which are handled with penalty functions. The second case involves the maximization of throughput and minimization of eluent consumption. The decision variables are the same as in the last case adding the feed flow rate and eluent consumption. This case considers two equality constraints. The authors concluded that the performance of a VARICOL process is superior to that of a SMB process in terms of treating more feed using less eluent or producing better product quality for fixed productivity and solvent consumption. Recently, Yu *et al.* (2005) carried out a study that also compares the SMB and VARICOL process, but for the hydrolysis of methyl acetate. In this study, the superiority of the VARICOL over the SMB process is confirmed.

1.7.7 *TYPE IV: Optimization of a Waste Incineration Plant*

Anderson *et al.* (2005) applied MOGA (Fonseca and Fleming, 1993) to optimize the operation of a waste incineration plant. In order to guarantee profitability and taking into account environmental concerns, the objectives of this problem comprise, respectively, the maximization of waste feed rate and minimization of unburnt carbon in the combustion ash. The decision variables considered are the waste feed rate and the residence time of the waste on the burning bed. The chamber's temperature was chosen as a constraint. The variant of MOGA used in this study allows the user to

define goal values and priorities for the objectives in order to articulate preferences (Fonseca and Fleming, 1998). This modification of MOGA also incorporates a methodology to handle constraints-related information. MOGA performed well in this application, since it converged (as expected) to values of high residence time and a range of feed rate values.

1.7.8 *TYPE IV: Chemical Process Systems Modelling*

From the three evolutionary techniques, genetic algorithms are by far the most commonly applied in chemical engineering. However, genetic programming (GP) embodies a powerful technique that has a lot of potential within chemical engineering (e.g., for modelling chemical process systems). In this direction, Hinchliffe *et al.* (1998) proposed a multi-objective genetic programming (MOGP) algorithm to model steady-state chemical process. This technique is based on MOGA (Fonseca and Fleming, 1993) which means that uses fitness sharing to keep diversity and the concept of preferability based on a given goal vector. This work includes two case studies: an inferential estimator for bottom product composition in a vacuum distillation column and a model for the degree of starch gelatinization in an industrial cooking extruder. The four objectives considered for both case studies include: (*i*) root mean square error on the training data set; (*ii*) residual variance, which aggregates a credit to models that produce accurate approximations apart from a constant offset; (*iii*) correlation between individuals and the process output; and (*iv*) model string length, which helps to avoid a complex models leading to overfitting. The study compares the performance of the MOGP with that of a single objective genetic programming (SOGP) algorithm proposed in a previous work (Hinchliffe *et al.*, 1996). The comparison was based on the RMS error on the validation data set and on the lengths of the models with lowest RMS error. The comparison involves the distribution of model prediction errors resulting from multiple runs. In the case of the distillation column no significant difference between the distributions of SOGP and MOGP was observed neither in RMS error nor in string length. This is due, according to the authors, to the fact that modelling of the column data is not a particularly difficult problem from a GP point of view. With regard to the cooking extruder, the MOGP obtained the best minimum RMS error and the best mean RMS value. However the distribution analysis did not reveal a significance difference between the distributions.

In a more recent study, Hinchliffe and Willis (2003) model dynamic

systems using genetic programming. The new approach is evaluated using two case studies, a test system with a time delay and an industrial cooking extruder. The objectives considered are the minimization of the root mean square error and minimization of the correlation and autocorrelations between residuals. The residuals of a model represent the difference between the predicted and actual values of the process output. In this work two MOGPs are compared, one based on Pareto ranking but without preferences, and another one, also based on Pareto ranking but with goal and priority information. From the results obtained in both case studies, the authors conclude that the MOGP with preference information was able to evolve a greater number of acceptable solutions than the algorithm that used conventional Pareto ranking.

1.7.9 *TYPE V: Biological and Bioinformatics Problems*

Bioinformatics is the interdisciplinary field that encompasses the analysis of large volumes of biological data to generate useful information and knowledge. This knowledge can be used for applications such as drug discovery, disease diagnosis and prognosis, and determination of the relationship between species.

Some bioinformatics problems can be formulated as MOOPs, for instance the sequence alignment of DNA or protein sequences, protein structure prediction and design, and inference of gene regulatory networks, just to mention a few. The interested reader is referred to the review of Handl *et al.* (2007) that covers in detail more MOO applications in bioinformatics. The next paragraphs describe some of the current applications in bioinformatics of interest for the chemical engineering community.

Sequence and structure alignment. Malard *et al.* (2004) formulate the *de novo* peptide identification as a constrained MOOP. The objectives considered in the study were the maximization of the similarity between portions of two peptides, and the maximization of the likelihood ratio between the null hypothesis and the alternative hypothesis. The algorithm was implemented using the island parallel model, in which some subpopulations evolve independently of each other although periodically individuals migrate between neighboring islands.

Calonder *et al.* (2006) address the problem of identifying gene modules on the basis of different types of biological data such as gene expression and protein-protein interaction data. The module identifications refers to the identification of groups of genes similar with respect to its function or

regulation mechanism.

Protein and structure prediction. Chen *et al.* (2005) proposed a method to solve the structure alignment problem for homologous proteins. This problem can be formulated as a MOOP where the objectives are maximize the number of aligned atoms and minimize their distance.

Shin *et al.* (2005) use the controlled NSGA-II (Deb and Goel, 2001) to generate a set of quality DNA sequences. In this study the quality of a sequence was achieved by minimizing four objectives: the similarity between two sequences in the set, the possible hybridization between sequences in a set, the continuous occurrence of the same base and the possible occurrence of the complementary substring in a sequence.

Gene regulatory networks. Spieth *et al.* (2005) address the problem of finding gene regulatory networks using a evolutionary algorithm combined with a local search method. The global optimizer is a genetic algorithm whereas an evolutionary strategy plays the role of the local optimizer.

Recently, Keedwell and Narayanan (2005) combined a genetic algorithm with a neural network to elucidate gene regulatory networks. The genetic algorithm has the goal of evolving a population of genes, while the neural network is used to evaluate how well the expression of the set of genes affect the expression values of other genes.

1.8 Critical Remarks

Most of the works reviewed in Section 1.6 rely on visual inspections to compare the generated Pareto fronts from different algorithms in order to show which algorithm performs better. However, graphical plots have some drawbacks for comparative purposes. One of the most serious drawbacks is that given the stochastic nature of MOEAs, a unique graphical plot is not enough to state that one algorithm outperforms another since in each run a different Pareto fronts may be generated. Furthermore, even if we can state that one algorithm is better than another using only visual inspections, it is better to be able to determine, in a quantitative way, how much better it is. MOEA researchers have developed a variety of performance measures for this sake (see (Coello Coello *et al.*, 2002; Zitzler *et al.*, 2003) for further information) and a more extended use of them is expected to occur in future chemical engineering applications of MOEAs.

It is also worth indicating that the chemical engineering applications

reviewed in this chapter tend to select a MOEA from a very reduced set (MOGA (Fonseca and Fleming, 1993), NSGA (Srinivas and Deb, 1994) and NSGA-II (Deb *et al.*, 2002)). However there are many other MOEAs that may be worth exploring: for example SPEA2 (Zitzler *et al.*, 2001), PAES (Knowles and Corne, 2000) and ε-MOEA (Deb *et al.*, 2005), which have all been successfully applied in other domains.

Finally, it is important to emphasize that comparative evaluations of MOEAs for any application should be done based on results obtained by the algorithms using the exact same process model, rather than based on previously reported results (particularly when they may have been originated by different researchers). This is because of the potential (and unknown) differences in the process models used by different researchers and their effect on optimization results.

1.9 Additional Resources

Launched in 1998, the EMOO repository[1] (Coello Coello, 2006) is one of the main resources for those interested in pursuing research in evolutionary MOO. The EMOO repository contains:

- Public-domain software.
- Test functions (either academic or real-world problems).
- URLs of events of interest for the EMOO community.
- Contact information of those who want to be added to the database of EMOO researchers (name, affiliation, postal address, email, web page and photo, if available).

As of June 2007, the EMOO repository contains:

- Over 2870 bibliographic references, which include 175 Ph.D. theses, 24 Masters theses, more than 750 journal papers and more than 1550 conference papers.
- Contact information of 66 EMOO researchers.
- Statistics of the EMOO repository.
- Public domain implementations of several MOEAs.
- Links to PISA (Bleuler *et al.*, 2003) and ParadisEO-MOEO (Liefooghe *et al.*, 2007), which are modern platforms that facilitate the use and development of MOEAs.

[1]The EMOO repository is located at: `http://delta.cs.cinvestav.mx/~ccoello/EMOO`

1.10 Future Research

There are several potential areas of future research regarding the use of MOEAs in chemical engineering applications. Some of them are the following:

- **Use of relaxed forms of dominance:** Some researchers have proposed the use of relaxed forms of Pareto dominance as a way of regulating convergence of a MOEA. Laumanns *et al.* (2002) proposed a relaxed form of dominance for MOEAs called ε-dominance. This mechanism acts as an archiving strategy to ensure both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found. Several modern MOEAs have adopted the concept of ε-dominance (see for example (Deb *et al.*, 2005)), because of its several advantages. Its use within chemical engineering, however, remains to be explored.
- **Incorporation of user's preferences:** Although many of the current MOEA-related work assumes that the user is interested in generating the entire Pareto front of a problem, in practice, normally only a small portion (or even a few solutions) is required. The incorporation of user's preferences is a problem that has been long studied by operation researchers (Figueira *et al.*, 2005). However, relatively little work has been done in this regard by MOEA researchers (Coello Coello, 2000; Branke and Deb, 2005). Nevertheless, this is a topic that certainly deserves attention from practitioners in chemical engineering.

1.11 Conclusions

This chapter has provided a brief introduction to MOEAs and their use in chemical engineering. Both algorithms and applications have been described and analyzed. From the review of the literature that was undertaken to write this chapter, it became evident that chemical engineering practitioners are already familiar with MOEAs, thus no attempt was made to raise their interest any further.

Thus, this chapter has attempted to provide a critical review of the current work done with MOEAs in chemical engineering, from a MOEA researchers' perspective. The intention, however, was not to minimize or disregard the important work already done. Instead, the aim was to bring practitioners close to the MOEA community so that both can interact and

mutually benefit. If some of the ideas presented in this chapter are incorporated by chemical engineering practitioners in the years to come, we will then know that the goals of this chapter have been fulfilled.

Acknowledgements

The second author acknowledges support from CONACyT project no. 45683-Y.

Bibliography

Anderson, S., Kadirkamanathan, V., Chipperfield, A., Sharifi, V. and Swithenbank, J. (2005). Multi-objective optimization of operational variables in a waste incineration plant, *Computers & Chemical Engineering* **29**, 5, pp. 1121–1130.

Babu, B., Chakole, P. and Mubeen, J. (2005). Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor, *Chemical Engineering Science* **60**, 17, pp. 4822–4837.

Babu, B. and Jehan, M. M. L. (2003). Differential Evolution for Multi-Objective Optimization, in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, Vol. 4 (IEEE Press, Canberra, Australia), pp. 2696–2703.

Bernal-Haro, L., Azzaro-Pantel, C., Pibouleau, L. and Domenech, S. (2002). Multiobjective Batch Plant Design: A Two-Stage Methodology. 1. Development of a Design-Oriented Discrete-Event Simulation Model, *Ind. Eng. Chem. Res* **41**, 23, pp. 5727–5742.

Bhaskar, V., Gupta, S. and Ray, A. (2000). Applications of multiobjective optimization in chemical engineering, *Reviews in Chemical Engineering* **16**, 1, pp. 1–54.

Bhutani, N., Ray, A. and Rangaiah, G. (2006). Modeling, Simulation and Multiobjective Optimization of an Industrial Hydrocracking Unit, *Industrial and Engineering Chemistry Research* **45**, pp. 1354–1372.

Bleuler, S., Laumanns, M., Thiele, L. and Zitzler, E. (2003). PISA—A Platform and Programming Language Independent Interface for Search Algorithms, in C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb and L. Thiele (eds.), *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003* (Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal), pp. 494–508.

Branke, J. and Deb, K. (2005). Integrating User Preferences into Evolutionary Multi-Objective Optimization, in Y. Jin (ed.), *Knowledge Incorporation in Evolutionary Computation* (Springer, Berlin Heidelberg), pp. 461–477, iSBN 3-540-22902-7.

Calonder, M., Bleuler, S. and Zitzler, E. (2006). Module Identification from Het-

erogeneous Biological Data Using Multiobjective Evolutionary Algorithms, in *Parallel Problem Solving from Nature (PPSN IX)*, no. 4193 in LNCS (Springer), pp. 573–582.

Chen, L., Wu, L., Wang, R., Wang, Y., Zhang, S. and Zhang, X. (2005). Comparison of Protein Structures by Multi-Objective Optimization, *Genome Informatics* **16**, 2, pp. 114–124.

Coello Coello, C. A. (2000). Handling Preferences in Evolutionary Multiobjective Optimization: A Survey, in *2000 Congress on Evolutionary Computation*, Vol. 1 (IEEE Service Center, Piscataway, New Jersey), pp. 30–37.

Coello Coello, C. A. (2006). The EMOO repository: a resource for doing research in evolutionary multiobjective optimization, *IEEE Computational Intelligence Magazine* **1**, 1, pp. 37–45.

Coello Coello, C. A. and Lamont, G. B. (eds.) (2004). *Applications of Multi-Objective Evolutionary Algorithms* (World Scientific, Singapore), iSBN 981-256-106-4.

Coello Coello, C. A., Van Veldhuizen, D. A. and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems* (Kluwer Academic Publishers, New York), iSBN 0-3064-6762-3.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms* (John Wiley & Sons, Chichester, UK), iSBN 0-471-87339-X.

Deb, K. and Goel, T. (2001). Controlled elitist non-dominated sorting genetic algorithms for better convergence, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization* , pp. 67–81.

Deb, K., Mohan, M. and Mishra, S. (2005). Evaluating the ϵ-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions, *Evolutionary Computation* **13**, 4, pp. 501–525.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II, *IEEE Transactions on Evolutionary Computation* **6**, 2, pp. 182–197.

Dedieu, S., Pibouleau, L., Azzaro-Pantel, C. and Domenech, S. (2003). Design and retrofit of multiobjective batch plants via a multicriteria genetic algorithm, *Computers & Chemical Engineering* **27**, 12, pp. 1723–1740.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L. and Domenech, S. (2006). Multiobjective optimization for multiproduct batch plant design under economic and environmental considerations, *Computers & Chemical Engineering* **30**, 4, pp. 599–613.

Figueira, J., Greco, S. and Erghott, M. (eds.) (2005). *Multiple Criteria Decision Analysis. State of the Art Surveys* (Springer, USA).

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

Fonseca, C. M. and Fleming, P. J. (1998). Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation, *IEEE Transactions on Systems, Man, and Cybernetics, Part*

A: Systems and Humans **28**, 1, pp. 26–37.

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

Goldberg, D. E. and Richardson, J. (1987). Genetic algorithm with sharing for multimodal function optimization, in J. J. Grefenstette (ed.), *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (Lawrence Erlbaum, Hillsdale, New Jersey), pp. 41–49.

Guria, C., Varma, M., Mehrotra, S. P. and Gupta, S. K. (2006). Simultaneous optimization of the performance of flotation circuits and their simplification using the jumping gene adaptations of genetic algorithm-ii: More complex problems, *International Journal of Mineral Processing* **79**, 3, pp. 149–166.

Guria, C., Verma, M., Mehrotra, S. and Gupta, S. (2005). Multi-objective optimal synthesis and design of froth flotation circuits for mineral processing, using the jumping gene adaptation of genetic algorithm, *Industrial & Engineering Chemistry Research* **44**, 8, pp. 2621–2633.

Handl, J., Kell, D. and Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology, *IEEE Trans Comput Biol Bioinformatics* .

Hinchliffe, M., Hiden, H., McKay, B., Willis, M., Tham, M. and Barton, G. (1996). Modelling chemical process systems using a multi-gene genetic programming algorithm, in J. R. Koza (ed.), *Late Breaking Papers at the Genetic Programming 1996 Conference, Stanford University July 28-31, 1996* (Stanford Bookstore, Stanford University, CA, USA), ISBN 0-18-201031-7, pp. 56–65.

Hinchliffe, M., Willis, M. and Tham, M. (1998). Chemical Process Systems Modelling using Multi-Objective Genetic Programming, in J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba and R. L. Riolo (eds.), *Proceedings of the Third Annual Conference on Genetic Programming*, University of Wisconsin at Madison (Morgan Kaufmann Publishers, San Mateo, California), pp. 134–139.

Hinchliffe, M. P. and Willis, M. J. (2003). Dynamic systems modelling using genetic programming, *Computers & Chemical Engineering* **27**, 12, pp. 1841–1854.

Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A Niched Pareto Genetic Algorithm for Multiobjective Optimization, in *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Vol. 1 (IEEE Service Center, Piscataway, New Jersey), pp. 82–87.

Husbands, P. (1994). Distributed Coevolutionary Genetic Algorithms for multi-Criteria and Multi-Constraint Optimisation, in T. C. Fogarty (ed.), *Evolutionary Computing. AIS Workshop. Selected Papers*, Lecture Notes in Computer Science Vol. 865 (Springer Verlag, Leeds, UK), pp. 150–165.

Ito, K., Akagi, S. and Nishikawa, M. (1983). A Multiobjective Optimization Approach to a Design Problem of Heat Insulation for Thermal Distribution

Piping Network Systems, *Journal of Mechanisms, Transmissions and Automation in Design (Transactions of the ASME)* **105**, pp. 206–213.

Kasat, R. B. and Gupta, S. K. (2003). Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm (GA) with the jumping genes operator, *Computers & Chemical Engineering* **27**, 12, pp. 1785–1800.

Keedwell, E. and Narayanan, A. (2005). Discovering gene networks with a neural-genetic hybrid, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **02**, 3, pp. 231–242, doi:http://doi.ieeecomputersociety.org/10.1109/TCBB.2005.40.

Knowles, J. D. and Corne, D. W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy, *Evolutionary Computation* **8**, 2, pp. 149–172.

Laumanns, M., Thiele, L., Deb, K. and Zitzler, E. (2002). Combining Convergence and Diversity in Evolutionary Multi-objective Optimization, *Evolutionary Computation* **10**, 3, pp. 263–282.

Liefooghe, A., Basseur, M., Jourdan, L. and Talbi, E.-G. (2007). ParadisEO-MOEO: A Framework for Evolutionary Multi-objective Optimization, in S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu and T. Murata (eds.), *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007* (Springer. Lecture Notes in Computer Science Vol. 4403, Matshushima, Japan), pp. 386–400.

Ludemann-Hombourger, O., Nicoud, R. M. and Bailly, M. (2000). The VARICOL Process: A New Multicolumn Continuous Chromatographic Process, *Separation Science and Technology* **35**, 12, pp. 1829–1862.

Malard, J., Heredia-Langner, A., Baxter, D., Jarman, K. and Cannon, W. (2004). Constrained de novo peptide identification via multi-objective optimization, *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International* .

Miettinen, K. M. (1999). *Nonlinear Multiobjective Optimization* (Kluwer Academic Publishers, Boston, Massachusetts).

Mitra, K., Deb, K. and Gupta, S. K. (1998). Multiobjective Dynamic Optimization of an Industrial Nylon 6 Semibatch Reactor Using Genetic Algorithm, *Journal of Applied Polymer Science* **69**, 1, pp. 69–87.

Pareto, V. (1896). *Cours D'Economie Politique*, Vol. I and II (F. Rouge, Lausanne).

Robič, T. and Filipič, B. (2005). DEMO: Differential Evolution for Multiobjective Optimization, in C. A. Coello Coello, A. Hernández Aguirre and E. Zitzler (eds.), *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005* (Springer. Lecture Notes in Computer Science Vol. 3410, Guanajuato, México), pp. 520–533.

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

Rudolph, G. (1998). On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set, in *Proceedings of the 5th IEEE Conference on Evolutionary Computation* (IEEE Press, Piscataway, New Jersey), pp.

511–516.

Sarkar, D., Rohani, S. and Jutan, A. (2007). Multiobjective optimization of semi-batch reactive crystallization processes, *AIChE Journal* **53**, 7, pp. 1164–1177.

Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, in *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms* (Lawrence Erlbaum, Hillsdale, New Jersey), pp. 93–100.

Sheng-jing, M., Hong-ye, S., Jian, C. and Yue-xuan, W. (2003). An New Evolutionary Multi-objective Optimization algorithm, in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, Vol. 2 (IEEE Press, Canberra, Australia), pp. 914–920.

Sheng-jing, M., Hong-ye, S., Tao, J., Yong, G. and Jian, C. (2004). Scalable multi-objective optimization of industrial purified terephthalic acid (PTA) oxidation process, *Computers & Chemical Engineering* **28**, 11, pp. 2219–2231.

Shin, S., Lee, I., Kim, D. and Zhang, B. (2005). Multiobjective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing, *Evolutionary Computation, IEEE Transactions on* **9**, 2, pp. 143–158.

Spieth, C., Streichert, F., Supper, J., Speer, N. and Zell, A. (2005). Feedback Memetic Algorithms for Modeling Gene Regulatory Networks, *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on* , pp. 1–7.

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

Storn, R. and Price, K. (1997). Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *Journal of Global Optimization* **11**, 4, pp. 341–359.

Tarafder, A., Lee, B., Ray, A. K. and Rangaiah, G. (2005a). Multi-objective optimization of an Industrial Ethylene Reactor using a Non-dominated Sorting Genetic Algorithm, *Industrial and Engineering Chemistry Research* **44**, pp. 124–141.

Tarafder, A., Rangaiah, G. P. and Ray, A. K. (2005b). Multiobjective optimization of an industrial styrene monomer manufacturing process, *Chemical Engineering Science* **60**, 2, pp. 347–363.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Ph.D. thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Weifeng, H., Hong-ye, S., Sheng-jing, M. and Jian, C. (2007). Multiobjective optimization of the industrial naphtha catalytic reforming process, *Chinese Journal of Chemical Engineering* **15**, 1, pp. 75–80.

Yee, A. K., Ray, A. K. and Rangaiah, G. P. (2003). Multiobjective optimization of an industrial styrene reactor, *Computers & Chemical Engineering* **27**, 1, pp. 111–130.

Yu, W., Hidajat, K. and Ray, A. (2003a). Application of multiobjective optimization in the design and operation of reactive SMB and its experimental verification, *Industrial & Engineering Chemistry Research* **42**, 26, pp. 6823–6831.

Yu, W., Hidajat, K. and Ray, A. (2003b). Modeling, simulation, and experimental study of a simulated moving bed reactor for the synthesis of methyl acetate ester, *Industrial & Engineering Chemistry Research* **42**, 26, pp. 6743–6754.

Yu, W., Hidajat, K. and Ray, A. (2005). Optimization of reactive simulated moving bed and Varicol systems for hydrolysis of methyl acetate, *Chemical Engineering Journal* **112**, pp. 57–72.

Zhang, Z., Hidajat, K., Ray, A. K. and Morbidelli, M. (2002). Multiobjective Optimization of SMB and Varicol Process for Chiral Separation, *AIChE Journal* **48**, 12, pp. 2800–2816.

Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm, in K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty (eds.), *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems* (Athens, Greece), pp. 95–100.

Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation* **3**, 4, pp. 257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and da Fonseca, V. G. (2003). Performance Assessment of Multiobjective Optimizers: An Analysis and Review, *IEEE Transactions on Evolutionary Computation* **7**, 2, pp. 117–132.