Chapter 12
Recent Developments in Evolutionary Multi-Objective Optimization

Kalyanmoy Deb

Abstract By now evolutionary multi-objective optimization (EMO) is an established and a growing field of research and application with numerous texts and edited books, commercial software, freely downloadable codes, a biannual conference series running successfully since 2001, special sessions and workshops held at all major evolutionary computing conferences, and full-time researchers from universities and industries from all around the globe. In this chapter, we discuss the principles of EMO through an illustration of one specific algorithm and an application to an interesting real-world bi-objective optimization problem. Thereafter, we provide a list of recent research and application developments of EMO to paint a picture of some salient advancements in EMO research. Some of these descriptions include hybrid EMO algorithms with mathematical optimization and multiple criterion decision-making procedures, handling of a large number of objectives, handling of uncertainties in decision variables and parameters, solution of different problem-solving tasks better by converting them into multi-objective problems, runtime analysis of EMO algorithms, and others. The development and application of EMO to multi-objective optimization problems and their continued extensions to solve other related problems has elevated the EMO research to a level which may now undoubtedly be termed as an active field of research with a wide range of theoretical and practical research and application opportunities.

Keywords Evolutionary optimization · Multi-objective optimization · Evolutionary multi-objective optimization · EMO

K. Deb (✉)
Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur, PIN 208016, India
e-mail: deb@iitk.ac.in
and
Department of Business Technology, Helsinki School of Economics, FIN-00101, Helsinki, Finland
e-mail: Kalyanmoy.Deb@hse.fi

M. Ehrgott et al. (eds.), *Trends in Multiple Criteria Decision Analysis*,
International Series in Operations Research and Management Science 142,
DOI 10.1007/978-1-4419-5904-1_12,

12.1 Introduction

Since the middle of Nineties, evolutionary multi-objective optimization (EMO) has become a popular and useful field of research and application. In a recent survey announced during the World Congress on Computational Intelligence (WCCI) in Vancouver 2006, EMO has been judged as one of the three fastest growing fields of research and application among all computational intelligence topics. Evolutionary optimization (EO) algorithms use a population-based approach in which more than one solution participates in an iteration and evolves a new population of solutions in each iteration. The reasons for their popularity are many. Some of them are: (i) EOs do not require any derivative information, (ii) EOs are relatively simple to implement, and (iii) EOs are flexible and have a widespread applicability. For solving single-objective optimization problems or in other tasks focusing on finding a single optimal solution, the use of a population of solutions in each iteration may at first seem like an overkill but they help provide an implicit parallel search ability, thereby making EOs computationally efficient [48, 53], in solving multi-objective optimization problems an EO procedure is a perfect match [19].

Multi-objective optimization problems, by nature, give rise to a set of Pareto-optimal solutions which need further processing to arrive at a single preferred solution. To achieve the first task, it becomes quite a natural proposition to use an EO, because the use of a population in an iteration helps an EO to simultaneously find multiple nondominated solutions, which portrays a trade-off among objectives, in a single run of the algorithm.

In this chapter, we begin with a brief description of the principles of an EMO in solving multi-objective optimization problems and then illustrate its working through a specific EMO procedure, which has been popularly and extensively used over the past 5–6 years. Besides this specific algorithm, there exist a number of other equally efficient EMO algorithms which we do not describe here for brevity. Instead, in this chapter, we discuss a number of recent advancements of EMO research and application which are driving the researchers and practitioners ahead. Fortunately, researchers have utilized the EMO's principle of solving multi-objective optimization problems in handling various other problem-solving tasks. The diversity of EMO's research is bringing researchers and practitioners together with different backgrounds including computer scientists, mathematicians, economists, and engineers. The topics we discuss here amply demonstrate why and how EMO researchers from different backgrounds must and should collaborate in solving complex problem-solving tasks which have become the need of the hour in most branches of science, engineering, and commerce.

12.2 Evolutionary Multi-objective Optimization (EMO)

A multi-objective optimization problem involves a number of objective functions which are to be either minimized or maximized subject to a number of constraints and variable bounds:

$$\left.\begin{array}{lll} \text{Minimize/Maximize} & f_m(\mathbf{x}), & m = 1, 2, \ldots, M; \\ \text{subject to} & g_j(\mathbf{x}) \geq 0, & j = 1, 2, \ldots, J; \\ & h_k(\mathbf{x}) = 0, & k = 1, 2, \ldots, K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \ldots, n. \end{array}\right\} \quad (12.1)$$

A solution $\mathbf{x} \in \mathbf{R}^n$ is a vector of n decision variables: $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. The solutions satisfying the constraints and variable bounds constitute a S in the decision variable space $\mathbf{R}^n$. One of the striking differences between single-objective and multi-objective optimization is that in multi-objective optimization the objective function vectors belong to a multidimensional objective space $\mathbf{R}^M$. The objective function vectors constitute a feasible set Z in the objective space. For each solution $\mathbf{x}$ in S, there exists a point $\mathbf{z} \in Z$, denoted by $\mathbf{f}(\mathbf{x}) = \mathbf{z} = (z_1, z_2, \ldots, z_M)^T$. To make the descriptions clear, we refer a decision variable vector as a solution and the corresponding objective vector as a point.

The optimal solutions in multi-objective optimization can be defined from a mathematical concept of *partial ordering*. In the parlance of multi-objective optimization, the term *domination* is used for this purpose. In this section, we restrict ourselves to discuss unconstrained (without any equality, inequality, or bound constraints) optimization problems. The domination between two solutions is defined as follows [19, 72]:

Definition 12.1. A solution $\mathbf{x}^{(1)}$ is said to dominate the another solution $\mathbf{x}^{(2)}$, if both the following conditions are true:

1. The solution $\mathbf{x}^{(1)}$ is no worse than $\mathbf{x}^{(2)}$ in all objectives. Thus, the solutions are compared based on their objective function values (or location of the corresponding points ($\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$) in the objective function set Z).
2. The solution $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective.

For a given set of solutions (or corresponding points in the objective function set Z, for example, those shown in Fig. 12.1a), a pair-wise comparison can be made using the above definition and whether one point dominates another point can also be established.

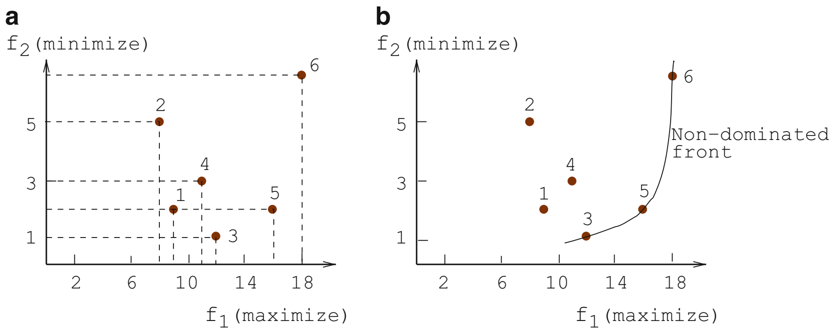


Fig. 12.1 A set of points and the first non-dominated front are shown

All points which are not dominated by any other member of the set are called the nondominated points of class one, or simply the nondominated points. For the set of six points shown in the figure, they are points 3, 5, and 6. One property of any two such points is that a gain in an objective from one point to the other happens only due to a sacrifice in at least one other objective. This *trade-off* property between the non-dominated points makes the practitioners interested in finding a wide variety of them before making a final choice. These points make up a front when viewed together on the objective space; hence the non-dominated points are often visualized to represent a *non-dominated front*. The theoretical computational effort needed to select the points of the non-dominated front from a set of N points is $O(N \log N)$ for two and three objectives, and $O(N \log^{M-2} N)$ for $M > 3$ objectives [65], but for a moderate number of objectives, the procedure need not be particularly computationally effective in practice.

With the above concept, it is now easier to define the *Pareto-optimal solutions* in a multi-objective optimization problem. If the given set of points for the above task contain all points in the decision variable space, the points lying on the non-domination front, by definition, do not get dominated by any other point in the objective space; hence are Pareto-optimal points (together they constitute the Pareto-optimal front) and the corresponding pre-images (decision variable vectors) are called Pareto-optimal solutions. However, more mathematically elegant definitions of Pareto-optimality (including the ones for continuous search space problems) exist in the multi-objective optimization literature [55, 72].

12.2.1 EMO Principles

In the context of multi-objective optimization, the extremist principle of finding the optimum solution cannot be applied to one objective alone, when the rest of the objectives are also important. This clearly suggests two ideal goals of multi-objective optimization:

Convergence: Find a (finite) set of solutions which lie on the Pareto-optimal front, and

Diversity: Find a set of solutions which are diverse enough to represent the entire range of the Pareto-optimal front.

EMO algorithms attempt to follow both the above principles, similar to a posteriori MCDM method. Figure 12.2 shows schematically the principles followed in an EMO procedure.

Since EMO procedures are heuristic based, they may not guarantee finding the exact Pareto-optimal points, as a theoretically provable optimization method would do for tractable (e.g., linear or convex) problems. But EMO procedures have essential operators to constantly improve the evolving nondominated points (from the point of view of convergence and diversity mentioned above) similar to the way most natural and artificial evolving systems continuously improve their solutions.

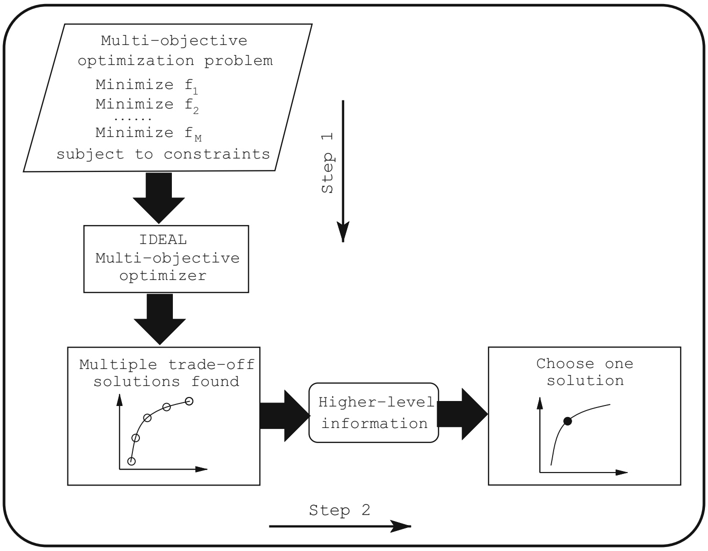


Fig. 12.2 Schematic of a two-step multi-criteria optimization and decision-making procedure

To this effect, a recent study [32] has demonstrated that a particular EMO procedure, starting from random non-optimal solutions, can progress towards the theoretical Karush-Kuhn-Tucker (KKT) points with iterations in real-valued multi-objective optimization problems. The main difference and advantage of using an EMO compared to a posteriori MCDM procedures is that multiple trade-off solutions can be found in a single run of an EMO algorithm, whereas most a posteriori MCDM methodologies would require multiple independent runs.

In Step 1 of the EMO-based multi-objective optimization and decision-making procedure (the task shown vertically downwards in Fig. 12.2), multiple trade-off, nondominated points are found. Thereafter, in Step 2 (the task shown horizontally, towards the right), higher-level information is used to choose one of the obtained trade-off points.

12.2.2 A Posteriori MCDM Methods and EMO

In the "a posteriori" MCDM approaches (also known as "generating MCDM methods"), the task of finding multiple Pareto-optimal solutions is achieved by executing many independent single-objective optimizations, each time finding a single Pareto-optimal solution [72]. A parametric scalarizing approach (such as the weighted-sum approach, ϵ-constraint approach, and others) can be used to convert multiple objectives into a parametric single-objective function. By simply varying the parameters (weight vector or ϵ-vector) and optimizing the scalarized function,

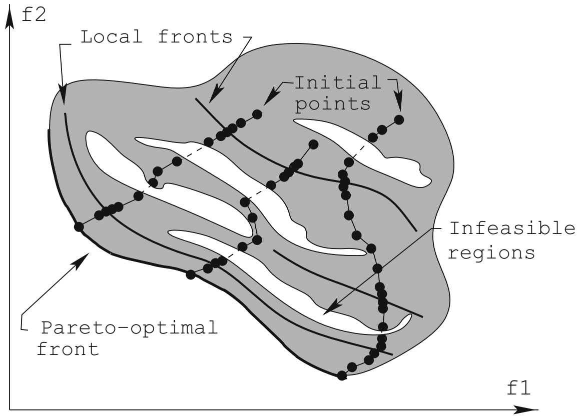


Fig. 12.3 A posteriori MCDM methodology employing independent single-objective optimizations

different Pareto-optimal solutions can be found. In contrast, in an EMO, multiple Pareto-optimal solutions are attempted to be found in a single run of the algorithm by emphasizing multiple non-dominated and isolated solutions in each iteration of the algorithm without the use of any scalarization of objectives.

Consider Fig. 12.3, in which we sketch how multiple independent parametric single-objective optimizations (through a posteriori MCDM method) may find different Pareto-optimal solutions.

It is worth highlighting here that the Pareto-optimal front corresponds to global optimal solutions of several problems each formed with a different scalarization of objectives. During the course of an optimization task, algorithms must overcome a number of difficulties, such as infeasible regions, local optimal solutions, flat or non-improving regions of objective landscapes, isolation of optimum, etc., to finally converge to the global optimal solution. Moreover, due to practical limitations, an optimization task must also be completed in a reasonable computational time. All these difficulties in a problem require that an optimization algorithm strikes a good balance between exploring new search directions and exploiting the extent of search in currently-best search direction. When multiple runs of an algorithm need to be performed independently to find a set of Pareto-optimal solutions, the above balancing act must be performed in every single run. Since runs are performed independently from one another, no information about the success or failure of previous runs is utilized to speed up the overall process. In difficult multi-objective optimization problems, such memory-less, a posteriori methods may demand a large overall computational overhead to find a set of Pareto-optimal solutions [85]. Moreover, despite the issue of global convergence, independent runs may not guarantee achieving a good distribution among obtained points by an easy variation of scalarization parameters.

EMO, as mentioned earlier, constitutes an inherent parallel search. When a particular population member overcomes certain difficulties and makes a progress towards the Pareto-optimal front, its variable values and their combination must reflect this fact. When a recombination takes place between this solution and another population member, such valuable information of variable value combinations gets shared through variable exchanges and blending, thereby making the overall task of finding multiple trade-off solutions a parallelly processed task.

12.3 A Brief History of EMO Methodologies

During the early years, EA researchers realized the need of solving multi-objective optimization problems in practice and mainly resorted to using weighted-sum approaches to convert multiple objectives into a single goal [40, 78].

However, the first implementation of a real multi-objective evolutionary algorithm (vector-evaluated GA or VEGA) was suggested by David Schaffer in the year 1984 [84]. Schaffer modified the simple three-operator genetic algorithm [53] (with selection, crossover, and mutation) by performing independent selection cycles according to each objective. The selection method is repeated for each individual objective to fill up a portion of the mating pool. Then the entire population is thoroughly shuffled to apply crossover and mutation operators. This is performed to achieve the mating of individuals of different subpopulation groups. The algorithm worked efficiently for some generations but in some cases suffered from its bias towards some individuals or regions (mostly individual objective champions). This does not fulfil the second goal of EMO, discussed earlier.

Ironically, no significant study was performed for almost a decade after the pioneering work of Schaffer, until a revolutionary 10-line sketch of a new non-dominated sorting procedure suggested by David E. Goldberg in his seminal book on GAs [48]. Since an EA needs a fitness function for reproduction, the trick was to find a single metric from a number of objective functions. Goldberg's suggestion was to use the concept of *domination* to assign more copies to non-dominated individuals in a population. Since diversity is the other concern, he also suggested the use of a *niching* strategy [49] among solutions of a non-dominated class. Getting this clue, at least three independent groups of researchers developed different versions of multi-objective evolutionary algorithms during 1993–1994 [43, 54, 87]. These algorithms differ in the way a fitness assignment scheme is introduced to each individual.

These EMO methodologies gave a good head-start to the research and application of EMO, but suffered from the fact that they did not use an elite-preservation mechanism in their procedures. Inclusion of elitism in an EO provides a monotonically non-degrading performance [79]. The second generation EMO algorithms implemented an elite-preserving operator in different ways and gave birth to elitist EMO procedures, such as NSGA-II [21], Strength Pareto EA (SPEA) [94], Pareto-archived ES (PAES) [60], and others. Since these EMO algorithms are state-of-the-art and commonly used procedures, we describe one of these algorithms in detail.

12.4 Elitist EMO: NSGA-II

The NSGA-II procedure [21] is one of the popularly used EMO procedures which attempt to find multiple Pareto-optimal solutions in a multi-objective optimization problem and has the following three features:

1. It uses an elitist principle
2. It uses an explicit diversity-preserving mechanism and
3. It emphasizes non-dominated solutions

At any generation t, the offspring population (say, Q_t) is first created by using the parent population (say, P_t) and the usual genetic operators. Thereafter, the two populations are combined together to form a new population (say, R_t) of size $2N$. Then, the population R_t is classified into different non-dominated classes. Thereafter, the new population is filled by points of different non-dominated fronts, one at a time. The filling starts with the first non-dominated front (of class one) and continues with points of the second non-dominated front, and so on. Since the overall population size of R_t is $2N$, not all fronts can be accommodated in N slots available for the new population. All fronts which could not be accommodated are deleted. When the last allowed front is being considered, there may exist more points in the front than the remaining slots in the new population. This scenario is illustrated in Fig. 12.4. Instead of arbitrarily discarding some members from the last front, the points which will make the diversity of the selected points the highest are chosen.

The crowded-sorting of the points of the last front which could not be accommodated fully is achieved in the descending order of their *crowding distance values* and

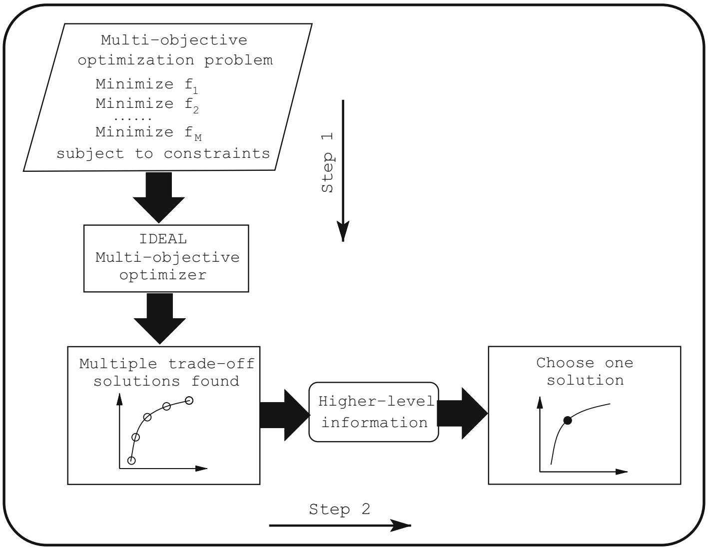


Fig. 12.4 Schematic of the NSGA-II procedure

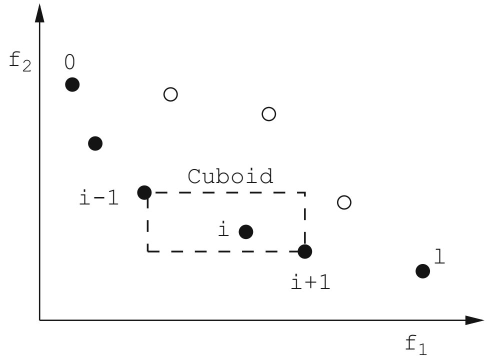


Fig. 12.5 The crowding distance calculation

points from the top of the ordered list are chosen. The crowding distance d_i of point i is a measure of the objective space around i which is not occupied by any other solution in the population. Here, we simply calculate this quantity d_i by estimating the perimeter of the cuboid (Fig. 12.5) formed by using the nearest neighbors in the objective space as the vertices (we call this the *crowding distance*).

12.4.1 Sample Results

Here, we show results from several runs of the NSGA-II algorithm on two test problems. The first problem (ZDT2) is two-objective, 30-variable problem with a concave Pareto-optimal front:

$$\text{ZDT2}: \begin{cases} \text{Minimize} & f_1(\mathbf{x}) = x_1, \\ \text{Minimize} & f_2(\mathbf{x}) = s(\mathbf{x})\left[1 - (f_1(\mathbf{x})/s(\mathbf{x}))^2\right], \\ \text{where} & s(\mathbf{x}) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i \\ & 0 \le x_1 \le 1, \\ & -1 \le x_i \le 1, \quad i = 2, 3, \ldots, 30. \end{cases} \tag{12.2}$$

The second problem (KUR), with three variables, has a disconnected Pareto-optimal front:

$$\text{KUR}: \begin{cases} \text{Minimize} & f_1(\mathbf{x}) = \sum_{i=1}^{2}\left[-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right], \\ \text{Minimize} & f_2(\mathbf{x}) = \sum_{i=1}^{3}\left[|x_i|^{0.8} + 5\sin\left(x_i^3\right)\right], \\ & -5 \le x_i \le 5, \quad i = 1, 2, 3. \end{cases} \tag{12.3}$$

NSGA-II is run with a population size of 100 for 250 generations. The variables are used as real numbers and an SBX recombination operator [20] with $p_c = 0.9$, distribution index of $\eta_c = 10$, a polynomial mutation operator [19] with $p_m = 1/n$ (n is the number of variables), and distribution index of $\eta_m = 20$ are used.

Fig. 12.6 NSGA-II on ZDT2

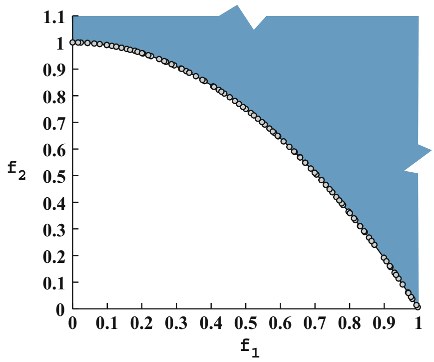
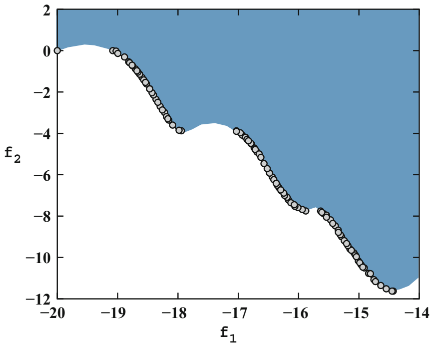


Fig. 12.7 NSGA-II on KUR



Figures 12.6 and 12.7 show that NSGA-II converges to the Pareto-optimal front and maintains a good spread of solutions on both test problems.

There also exist other competent EMOs, such as strength Pareto evolutionary algorithm (SPEA) and its improved version SPEA2 [93], Pareto-archived evolution strategy (PAES) and its improved versions PESA and PESA2 [16], multi-objective messy GA (MOMGA) [89], multi-objective-GA [12], neighbourhood constraint GA [69], ARMOGA [80], and others. Besides, there exists other EA-based methodologies, such as particle swarm EMO [19,73], ant-based EMO [50,71], and differential evolution-based EMO [1].

12.4.2 Constraint Handling in EMO

The constraint handling method modifies the binary tournament selection, where two solutions are picked from the population and the better solution is chosen. In the presence of constraints, each solution can be either feasible or infeasible. Thus, there may be at most three situations: (i) both solutions are feasible, (ii) one is feasible and other is not, and (iii) both are infeasible. We consider each case by simply redefining the domination principle as follows (we call it the *constrained-domination* condition for any two solutions $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$):

Definition 12.2. A solution $\mathbf{x}^{(i)}$ is said to 'constrained-dominate' a solution $\mathbf{x}^{(j)}$ (or $\mathbf{x}^{(i)} \preceq_c \mathbf{x}^{(j)}$), if any of the following conditions are true:

1. Solution $\mathbf{x}^{(i)}$ is feasible and solution $\mathbf{x}^{(j)}$ is not.
2. Solutions $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are both infeasible, but solution $\mathbf{x}^{(i)}$ has a smaller constraint violation, which can be computed by adding the normalized violation of all constraints:

$$\mathrm{CV}(\mathbf{x}) = \sum_{j=1}^{J} \max\left(0, -\bar{g}_j(\mathbf{x})\right) + \sum_{k=1}^{K} \mathrm{abs}(\bar{h}_k(\mathbf{x})).$$

   The normalization of a constraint $g_j(\mathbf{x}) \geq g_{j,r}$ can be achieved as $\bar{g}_j(\mathbf{x}) \geq 0$, where $\bar{g}_j(\mathbf{x}) = g_j(\mathbf{x})/g_{j,r} - 1$.
3. Solutions $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are feasible and solution $\mathbf{x}^{(i)}$ dominates solution $\mathbf{x}^{(j)}$ in the usual sense (Definition 12.1).

The above change in the definition requires a minimal change in the NSGA-II procedure described earlier. Figure 12.8 shows the nondominated fronts on a six-member population due to the introduction of two constraints (the minimization problem is described as CONSTR elsewhere [19]). In the absence of the constraints, the nondominated fronts (shown by dashed lines) would have been `((1,3,5), (2,6), (4))`, but in their presence, the new fronts are `((4,5), (6), (2), (1), (3))`.

The first nondominated front consists of the "best" (i.e., nondominated and feasible) points from the population and any feasible point lies on a better nondominated front than an infeasible point.

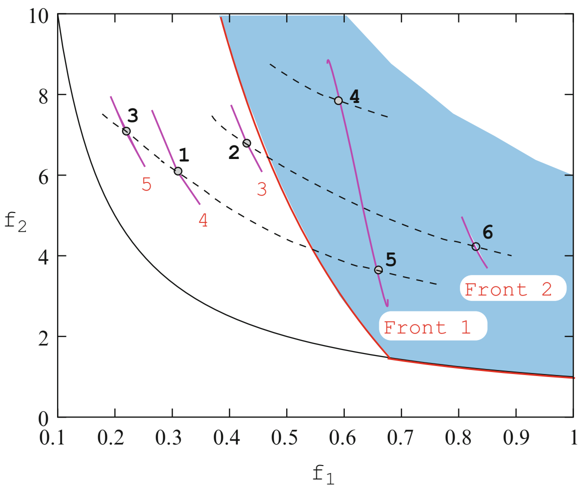


Fig. 12.8 Non-constrained-domination fronts

12.5 Applications of EMO

Since the early development of EMO algorithms in 1993, they have been applied to many challenging real-world optimization problems. Descriptions of some of these studies can be found in books [13, 19, 77], dedicated conference proceedings [15, 41, 76, 91], and domain-specific books, journals and proceedings. In this section, we describe one case study which clearly demonstrates the EMO philosophy which we described in Section 12.2.1.

12.5.1 Spacecraft Trajectory Design

Coverstone-Carroll et al. [17] proposed a multi-objective optimization technique using the original non-dominated sorting algorithm (NSGA) [87] to find multiple trade-off solutions in a spacecraft trajectory optimization problem. To evaluate a solution (trajectory), the SEPTOP (Solar Electric Propulsion Trajectory Optimization) software [81] is called, and the delivered payload mass and the total time of flight are calculated. The multi-objective optimization problem has eight decision variables controlling the trajectory, three objective functions: (i) maximize the delivered payload at destination, (ii) maximize the negative of the time of flight, and (iii) maximize the total number of heliocentric revolutions in the trajectory, and three constraints limiting the SEPTOP convergence error and minimum and maximum bounds on heliocentric revolutions.

On the Earth–Mars rendezvous mission, the study found interesting trade-off solutions [17]. Using a population of size 150, the NSGA was run for 30 generations. The obtained nondominated solutions are shown in Fig. 12.9 for two of the three objectives and some selected solutions are shown in Fig. 12.10.

It is clear that there exist short-time flights with smaller delivered payloads (solution marked 44 with 1.12 years of flight and delivering 685.28 kg load) and long-time flights with larger delivered payloads (solution marked 36 with close to 3.5 years of flight and delivering about 900 kg load).

While solution 44 can deliver a mass of 685.28 kg and requires about 1.12 years, solution 72 can deliver almost 862 kg with a travel time of about 3 years. In these figures, each continuous part of a trajectory represents a *thrusting* arc and each dashed part of a trajectory represents a *coasting* arc. It is interesting to note that only a small improvement in delivered mass occurs in the solutions between 73 and 72 with a sacrifice in flight time of about 1 year.

The multiplicity in trade-off solutions, as depicted in Fig. 12.10, is what we envisaged in discovering in a multi-objective optimization problem by using a posteriori procedure, such as a generating method or using an EMO procedure vis-a-vis a priori approach in which a single scalarized problem is solved with a single preferred parameter setting to find a single Pareto-optimal solution. This aspect was also discussed in Fig. 12.2. Once a set of solutions with a good trade-off among objectives is obtained, one can analyze them for choosing a particular solution. For example,

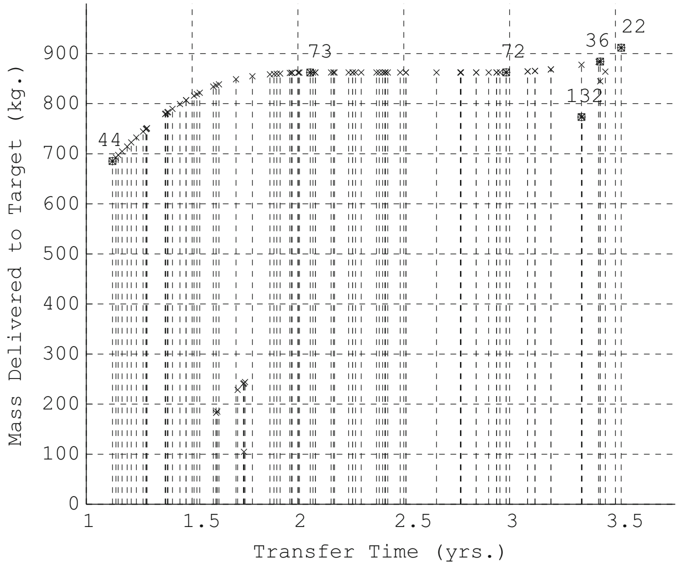


Fig. 12.9 Obtained nondominated solutions using NSGA

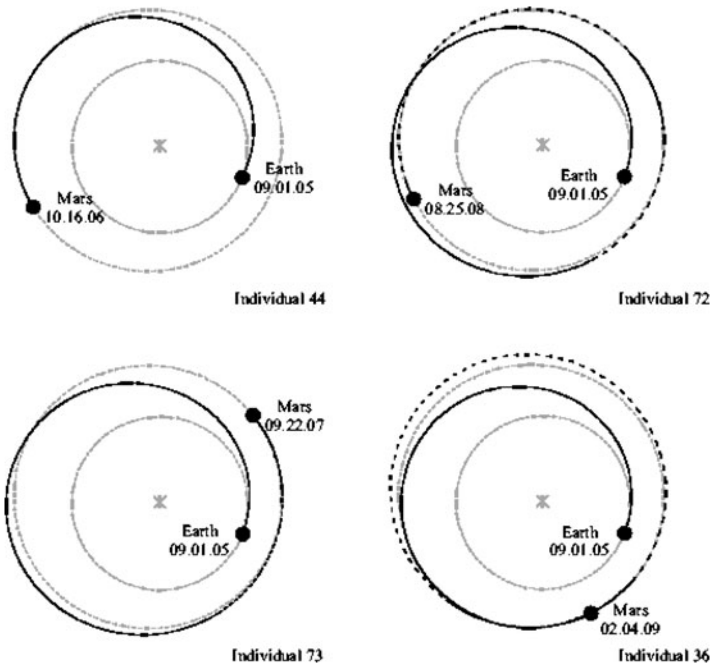


Fig. 12.10 Four trade-off trajectories

in this problem context, it makes sense to not choose a solution between points 73 and 72 due to poor trade-off between the objectives in this range, a matter which is only revealed after a representative set of trade-off solutions are found. On the other hand, choosing a solution within points 44 and 73 is worthwhile, but which particular solution to choose depends on other mission-related issues. But by first finding a wide range of possible solutions and revealing the shape of front in a computationally quicker manner, EMO can help a decision maker in narrowing down the choices and in allowing to make a better decision (e.g., in the above example, focussing to choose a solution with a transfer time less than 2 years). Without the knowledge of such a wide variety of trade-off solutions, proper decision making may be a difficult task. The use of a priori approach to find a single solution using for example, the ϵ-constraint method with a particular ϵ vector, the decision maker will always wonder what solution would have been derived if a different ϵ vector was chosen. For example, if $\epsilon_1 = 2.5$ years is chosen and mass delivered to the target is maximized, a solution in between points 73 and 72 will be found. As discussed earlier, this part of the Pareto-optimal front does not provide the best trade-offs between objectives that this problem can offer. A lack of knowledge of good trade-off regions before a decision is made may allow the decision maker to settle for a solution which, although optimal, may not be a good compromised solution. The EMO procedure allows a flexible and a pragmatic procedure for finding a well-diversified set of solutions simultaneously so as to enable picking a particular region for further analysis or a particular solution for implementation.

12.6 Salient Recent Developments of EMO

An interesting aspect regarding research and application of EMO is that soon after a number of efficient EMO methodologies had been suggested and applied in various interesting problem areas, researchers did not waste any time to look for opportunities to make the field broader and more useful by diversifying EMO applications to various other problem-solving tasks. In this section, we describe a number of such salient recent developments of EMO.

12.6.1 Hybrid EMO Algorithms

Search operators used in EMO are heuristic-based. Thus, these methodologies are not guaranteed to find Pareto-optimal solutions with a finite number of solution evaluations in an arbitrary problem. In single-objective EA research, hybridization of EAs is common for ensuring convergence to an optimal solution, it is not surprising that studies on developing hybrid EMOs are now being pursued to ensure finding of true Pareto-optimal solutions by hybridizing them with mathematically convergent ideas.

EMO methodologies provide adequate emphasis to currently non-dominated and isolated solutions so that population members progress towards the Pareto-optimal front iteratively. To make the overall procedure faster and to perform the task with a more theoretical emphasis, EMO methodologies are combined with mathematical optimization techniques having local convergence properties. A simple-minded approach would be to start the process with an EMO and the solutions obtained from EMO can be improved by optimizing a composite objective derived from multiple objectives to ensure a good spread by using a local search technique [22]. Another approach would be to use a local search technique as a mutation-like operator in an EMO so that all population members are at least guaranteed to be local optimal solutions [22, 86]. To save computational time, instead of performing the local search for every solution in a generation, a mutation can be performed only after a few generations. Some recent studies [56, 82, 86] have demonstrated the usefulness of such hybrid EMOs for a guaranteed convergence.

Although these studies have concentrated on ensuring convergence to the Pareto-optimal front, some emphasis should now be placed in providing an adequate diversity among obtained solutions, particularly when a continuous Pareto-optimal front is represented by a finite set of points. Some ideas of maximizing hypervolume measure [39] or maintenance of uniform distance between points are proposed for this purpose, but how such diversity-maintenance techniques would be integrated with convergence-ensuring principles in a synergistic way would be interesting and useful future research. Some relevant studies in this direction exist [4, 56, 66].

12.6.2 Multi-objectivization

Interestingly, the act of finding multiple trade-off solutions using an EMO procedure has found its application outside the realm of solving multi-objective optimization problems. The concept of finding near-optimal trade-off solutions is applied to solve other kinds of optimization problems as well. For example, the EMO concept is used to solve constrained single-objective optimization problems by converting the task into a two-objective optimization task of additionally minimizing an aggregate constraint violation [14]. This eliminates the need to specify a penalty parameter while using a penalty based constraint handling procedure. If viewed this way, the usual penalty function approach used in classical optimization studies is a special weighted-sum approach to the bi-objective optimization problem of minimizing the objective function and minimizing the constraint violation, for which the weight vector is a function of the penalty parameter. A well-known difficulty in genetic programming studies, called *bloating*, arises due to the continual increase in the size of evolved "genetic programs" with iteration. The reduction of bloating by minimizing the size of a program as an additional objective helped find high-performing solutions with a smaller size of the code [3, 57]. In clustering algorithms, minimizing the intra-cluster distance and maximizing inter-cluster distance simultaneously in a bi-objective formulation of a is found to yield better solutions than the usual single-objective minimization of the ratio of the intra-cluster distance to the inter-cluster distance [51]. An EMO is used to solve minimum spanning tree problem better than a single-objective EA [75]. A recent edited book [62] describes many such interesting applications in which EMO methodologies have helped solve problems which are otherwise (or traditionally) not treated as multi-objective optimization problems.

12.6.3 Uncertainty-based EMO

A major surge in EMO research has taken place in handling uncertainties among decision variables and problem parameters in multi-objective optimization. Practice is full of uncertainties and almost no parameter, dimension, or property can be guaranteed to be fixed at a value it is aimed at. In such scenarios, evaluation of a solution is not precise, and the resulting objective and constraint function values becomes probabilistic quantities. Optimization algorithms are usually designed to handle such stochasticities by using crude methods, such as the Monte Carlo simulation of stochasticities in uncertain variables and parameters and by sophisticated stochastic programming methods involving nested optimization techniques [24]. When these effects are taken care of during the optimization process, the resulting solution is usually different from the optimum solution of the problem and is known as a "robust" solution. Such an optimization procedure will then find a solution which may not be the true global optimum solution, but one which is less sensitive to uncertainties in decision variables and problem parameters. In the context of multi-objective optimization, a consideration of uncertainties for multiple objective functions will result in a robust frontier which may be different from the globally Pareto-optimal front. Each and every point on the robust frontier is then guaranteed to be less sensitive to uncertainties in decision variables and problem parameters. Some such studies in EMO are [2, 23].

When the evaluation of constraints under uncertainties in decision variables and problem parameters are considered, deterministic constraints become stochastic (they are also known as "chance constraints") and involves a *reliability index* (R) to handle the constraints. A constraint $g(\mathbf{x}) \geq 0$ then becomes $\mathrm{Prob}(g(\mathbf{x}) \geq 0) \geq R$. In order to find left side of the above chance constraint, a separate optimization methodology [18], is needed, thereby making the overall algorithm a bi-level optimization procedure. Approximate single-loop algorithms exist [34] and recently one such methodology has been integrated with an EMO [24] and shown to find a "reliable" frontier corresponding a specified reliability index, instead of the Pareto-optimal frontier, in problems having uncertainty in decision variables and problem parameters. More such methodologies are needed, as uncertainties is an integral part of practical problem-solving and multi-objective optimization researchers must look for better and faster algorithms to handle them.

12.6.4 EMO and Decision Making

Searching for a set of Pareto-optimal solutions by using an EMO fulfils only one aspect of multi-objective optimization, as choosing a particular solution for an implementation is the remaining decision-making task which is equally important. For many years, EMO researchers have postponed the decision-making aspect and concentrated on developing efficient algorithms for finding multiple trade-off solutions. Having pursued that part somewhat, now for the past couple of years or so, EMO researchers are putting efforts to design combined algorithms for optimization and decision making. In the view of the author, the decision-making task can be considered from two main considerations in an EMO framework:

1. **Generic consideration:** There are some aspects which most practical users would like to use in narrowing down their choice. We have discussed above the importance of finding robust and reliable solutions in the presence of uncertainties in decision variables and/or problem parameters. In such scenarios, an EMO methodology can straightway find a robust or a reliable frontier [23, 24] and no subjective preference from any decision maker may be necessary. Similarly, if a problem resorts to a Pareto-optimal front having *knee* points, such points are often the choice of decision makers. Knee points demand a large sacrifice in at least one objective to achieve a small gain in another thereby making it discouraging to move out from a knee point [7]. Other such generic choices are related to Pareto-optimal points depicting certain pre-specified relationship between objectives, Pareto-optimal points having multiplicity (say, at least two or more solutions in the decision variable space mapping to identical objective values), Pareto-optimal solutions which do not lie close to variable boundaries, Pareto-optimal points having certain mathematical properties, such as all Lagrange multipliers having more or less identical magnitude – a condition often desired to make an equal importance to all constraints, and others. These considerations are motivated from the fundamental and practical aspects of optimization and may be applied to most multi-objective problem-solving tasks, without any consent of a decision maker. These considerations may narrow down the set of non-dominated points. A further subjective consideration (discussed below) may then be used to pick a preferred solution.
2. **Subjective consideration:** In this category, any problem-specific information can be used to narrow down the choices and the process may even lead to a single preferred solution at the end. Most decision-making procedures use some preference information (utility functions, reference points [90], reference directions [63], marginal rate of return, and a host of other considerations [72]) to select a subset of Pareto-optimal solutions. A recent book [8] is dedicated to the discussion of many such multi-criteria decision analysis (MCDA) tools and collaborative suggestions of using EMO with such MCDA tools. Some hybrid EMO and MCDA algorithms are suggested in the recent past [25, 26, 31, 70, 88].

Many other generic and subjective considerations are needed and it is interesting that EMO and MCDM researchers are collaborating on developing such complete algorithms for multi-objective optimization [8].

12.6.5 EMO for Handling a Large Number of Objectives

Soon after the development of efficient EMO methodologies, researchers were interested in exploring whether existing EMO methodologies are adequate to handle a large number of objectives (say, ten or more). An earlier study [58] with eight objectives revealed somewhat negative results. But the author in his book [19] and recent other studies [59] have clearly explained the reason for this behavior of EMO algorithms. EMO methodologies work by emphasizing non-dominated solutions in a population. Unfortunately, as the number of objectives increase, most population members in a randomly created population tend to become non-dominated to each other. For example, in a three-objective scenario, about 10% members in a population of size 200 are nondominated, whereas in a 10-objective problem scenario, as high as 90% members in a population of size 200 are nondominated. Thus, in a large-objective problem, an EMO algorithm runs out of room to introduce new population members into a generation, thereby causing a stagnation in the performance of an EMO algorithm. It has been argued that to make EMO procedures efficient, an exponentially large population size (with respect to number of objectives) is needed. This makes an EMO procedure slow and computationally less attractive.

However, practically speaking, even if an algorithm can find tens of thousands of Pareto-optimal solutions for a multi-objective optimization problem, besides simply getting an idea of the nature and shape of the front, they are simply too many to be useful for any decision-making purposes. Keeping these views in mind, EMO researchers have taken two different approaches in dealing with large-objective problems.

12.6.5.1 Finding a Partial Set

Instead of finding the complete Pareto-optimal front in a problem having a large number of objectives, EMO procedures can be used to find only a part of the Pareto-optimal front. This can be achieved by indicating preference information by various means. Ideas, such as reference point-based EMO [31,70], "light beam search" [26], biased sharing approaches [6], cone dominance [33], etc. are suggested for this purpose. Each of these studies have shown that up to 10, and 20-objective problems, although finding the complete frontier is a difficulty, finding a partial frontier corresponding to certain preference information is not that difficult a proposition. Despite the dimension of the partial frontier being identical to that of the complete Pareto-optimal frontier, the closeness of target points in representing the desired partial frontier helps make only a small fraction of an EMO population to be nondominated, thereby making rooms for new and hopefully better solutions to be found and stored.

The computational efficiency and accuracy observed in some EMO implementations have led a distributed EMO study [33] in which each processor in a distributed computing environment receives a unique cone for defining domination. The cones are designed carefully so that at the end of such a distributed computing EMO procedure, solutions are found to exist in various parts of the complete Pareto-optimal front. A collection of these solutions together is then able to provide a good representation of the entire original Pareto-optimal front.

12.6.5.2 Identifying and Eliminating Redundant Objectives

Many practical optimization problems can easily list a large of number of objectives (often more than ten), as many different criteria or goals are often of interest to practitioners. In most instances, it is not entirely sure whether the chosen objectives are all in conflict to each other or not. For example, minimization of weight and minimization of cost of a component or a system are often mistaken to have an identical optimal solution, but may lead to a range of trade-off optimal solutions. Practitioners do not take any chance and tend to include all (or as many as possible) objectives into the optimization problem formulation. There is another fact which is more worrisome. Two apparently conflicting objectives may show a good trade-off when evaluated with respect to some randomly created solutions. But if these two objectives are evaluated for solutions close to their optima, they tend to show a good correlation. That is, although objectives can exhibit conflicting behavior for random solutions, near their Pareto-optimal front, the conflict vanishes and optimum of one becomes close to the optimum of the other.

Thinking of the existence of such problems in practice, recent studies [29, 83] have performed linear and non-linear principal component analysis (PCA) to a set of EMO-produced solutions. Objectives causing positively correlated relationship between each other on the obtained NSGA-II solutions are identified and are declared as redundant. The EMO procedure is then restarted with non-redundant objectives. This combined EMO-PCA procedure is continued until no further reduction in the number of objectives is possible. The procedure has handled practical problems involving five and more objectives and has shown to reduce the choice of real conflicting objectives to a few. On test problems, the proposed approach has shown to reduce an initial 50-objective problem to the correct three-objective Pareto-optimal front by eliminating 47 redundant objectives. Another study [9] used an exact and a heuristic-based conflict identification approach on a given set of Pareto-optimal solutions. For a given error measure, an effort is made to identify a minimal subset of objectives which do not alter the original dominance structure on a set of Pareto-optimal solutions. This idea has recently been introduced within an EMO [10], but a continual reduction of objectives through a successive application of the above procedure would be interesting.

This is a promising area of EMO research and definitely more computationally faster objective-reduction techniques are needed for the purpose. In this direction, the use of alternative definitions of domination is important. One such idea redefined the definition of domination: a solution is said to dominate another solution, if the former solution is better than latter in more objectives. This certainly excludes finding the entire Pareto-optimal front and helps an EMO to converge near the intermediate and central part of the Pareto-optimal front. Another EMO study used a fuzzy dominance [38] relation (instead of Pareto-dominance), in which superiority of one solution over another in any objective is defined in a fuzzy manner. Many other such definitions are possible and can be implemented based on the problem context.

12.6.6 Knowledge Extraction Through EMO

One striking difference between a single-objective optimization and multi-objective optimization is the cardinality of the solution set. In the latter, multiple solutions are the outcome and each solution is theoretically an optimal solution corresponding to a particular trade-off among the objectives. Thus, if an EMO procedure can find solutions close to the true Pareto-optimal set, what we have in our hand are a number of high-performing solutions trading-off the conflicting objectives considered in the study. Since they are all near-optimal, these solutions can be analyzed for finding properties which are common to them. Such a procedure can then become a systematic approach in deciphering important and hidden properties which optimal and high-performing solutions must have for that problem. In a number of practical problem-solving tasks, the so-called *innovization* procedure is shown to find important knowledge about high-performing solutions [30]. Such useful properties are expected to exist in practical problems, as they follow certain scientific and engineering principles at the core, but finding them through a systematic scientific procedure had not been paid much attention in the past. The principle of first searching for multiple trade-off and high-performing solutions using a multi-objective optimization procedure and then analyzing them to discover useful knowledge certainly remains a viable way forward. The current efforts to automate the knowledge extraction procedure through a sophisticated data-mining task should make the overall approach more appealing and useful in practice.

12.6.7 Dynamic EMO

Dynamic optimization involves objectives, constraints, or problem parameters which change over time. This means that as an algorithm is approaching the optimum of the current problem, the problem definition has changed and now the algorithm must solve a new problem. This is not equivalent to another optimization task in which a new and different optimization problem must be solved afresh.

Often, in such dynamic optimization problems, an algorithm is usually not expected to find the optimum, instead it is best expected to track the changing optimum with iteration. The performance of a dynamic optimizer then depends on how close it is able to track the true optimum (which is changing with iteration or time). Thus, practically speaking, optimization algorithms may hope to handle problems which do not change significantly with time. From the algorithm's point of view, since in these problems the problem is not expected to change too much from one time instance to another and some good solutions to the current problem are already at hand in a population, researchers fancied solving such dynamic optimization problems using evolutionary algorithms [5].

A recent study [28] proposed the following procedure for dynamic optimization involving single or multiple objectives. Let $\mathcal{P}(t)$ be a problem which changes with time t (from $t = 0$ to $t = T$). Despite the continual change in the problem, we assume that the problem is fixed for a time period τ, which is not known a priori and the aim of the (offline) dynamic optimization study is to identify a suitable value of τ for an accurate as well computationally faster approach. For this purpose, an optimization algorithm with τ as a fixed time period is run from $t = 0$ to $t = T$ with the problem assumed fixed for every τ time period. A measure $\Gamma(\tau)$ determines the performance of the algorithm and is compared with a pre-specified and expected value Γ_L. If $\Gamma(\tau) \geq \Gamma_L$, for the entire time domain of the execution of the procedure, we declare τ to be a permissible length of stasis. Then, we try with a reduced value of τ and check if a smaller length of statis is also acceptable. If not, we increase τ to allow the optimization problem to remain static for a longer time so that the chosen algorithm can now have more iterations (time) to perform better. Such a procedure will eventually come up with a time period τ^* which would be the smallest time of statis allowed for the optimization algorithm to work based on chosen performance requirement. Based on this study, a number of test problems and a hydrothermal power dispatch problem have been recently tackled [28].

In the case of dynamic multi-objective problem-solving tasks, there is an additional difficulty which is worth mentioning here. Not only does an EMO algorithm needs to find or track the changing Pareto-optimal fronts, in a real-world implementation, it must also make an immediate decision about which solution to implement from the current front before the problem changes to a new one. Decision-making analysis is considered to be time-consuming involving execution of analysis tools, higher-level considerations, and sometimes group discussions. If dynamic EMO is to be applied in practice, *automated* procedures for making decisions must be developed. Although it is not clear how to generalize such an automated decision-making procedure in different problems, problem-specific tools are certainly possible and certainly a worthwhile and fertile area for research.

12.6.8 Quality Estimates for EMO

When algorithms are developed and test problems with known Pareto-optimal fronts are available, an important task is to have performance measures with which the

EMO algorithms can be evaluated. Thus, a major focus of EMO research has been spent to develop different performance measures. Since the focus in an EMO task is multifaceted – convergence to the Pareto-optimal front and diversity of solutions along the entire front – it is also expected that one performance measure to evaluate EMO algorithms will be unsatisfactory. In the early years of EMO research, three different sets of performance measures were used:

1. Metrics evaluating convergence to the known Pareto-optimal front (such as error ratio, distance from reference set, etc.)
2. Metrics evaluating spread of solutions on the known Pareto-optimal front (such as spread, spacing, etc.) and
3. Metrics evaluating certain combinations of convergence and spread of solutions (such as hypervolume, coverage, R-metric, etc.)

Some of these metrics are described in texts [13, 19]. A detailed study [61] comparing most existing performance metrics based on out-performance relations has recommended the use of the S-metric (or the hypervolume metric) and R-metric suggested by [52]. A recent study has argued that a single unary performance measure or any finite combination of them (e.g., any of the first two metrics described above in the enumerated list or both together) cannot adequately determine whether one set is better than another [95]. That study also concluded that binary performance metrics (indicating usually two different values when a set of solutions A is compared against B and B is compared against A), such as epsilon indicator, binary hypervolume indicator, utility indicators R1 to R3, etc., are better measures for multi-objective optimization. The flip side is that the chosen binary metric must be computed $K(K-1)$ times when comparing K different sets to make a fair comparison, thereby making the use of binary metrics computationally expensive in practice. Importantly, these performance measures have allowed researchers to use them directly as fitness measures within indicator-based EAs (IBEAs) [92]. In addition, of [42, 44] provide further information about location and inter-dependencies among obtained solutions.

12.6.9 Exact EMO with Run-time Analysis

Since the suggestion of efficient EMO algorithms, they have been increasingly applied in a wide variety of problem domains to obtain trade-off frontiers. Simultaneously, some researchers have also devoted their efforts in developing exact EMO algorithms with a theoretical complexity estimate in solving certain discrete multi-objective optimization problems. The first such study [68] suggested a pseudo-Boolean multi-objective optimization problem – a two-objective LOTZ (Leading Ones Trailing Zeroes) – and a couple of EMO methodologies – a simple evolutionary multi-objective optimizer (SEMO) and an improved version fair evolutionary multi-objective optimizer (FEMO). The study then estimated the worst-case computational effort needed to find all Pareto-optimal solutions of the problem LOTZ. This study spurred a number of improved EMO algorithms with run-time estimates and resulted in many other interesting test problems [46, 47, 64, 67]. Although these test problems may not resemble common practical problems, the working principles of suggested EMO algorithms to handle specific problem structures bring in a plethora of insights about the working of multi-objective optimization, particularly in comprehensively finding all (not just one, or a few) Pareto-optimal solutions.

12.6.10 EMO with Meta-models

The practice of optimization algorithms is often limited by the computational overheads associated with evaluating solutions. Certain problems involving expensive computations, such as numerical solution of partial differential equations describing the physics of the problem, finite difference computations involving an analysis of a solution, computational fluid dynamics simulation to study the performance of a solution over a changing environment, etc. In some such problems, evaluation of each solution to compute constraints and objective functions may take a few hours to a complete day or two. In such scenarios, even if an optimization algorithm needs 100 solutions to get anywhere close to a good and feasible solution, the application needs an easy 3–6 months of continuous computational time. In most practical purposes, this is considered a "luxury" in an industrial set-up. Optimization researchers are constantly at their toes in coming up with approximate yet faster algorithms.

A little thought brings out an interesting fact about how optimization algorithms work. The initial iterations deal with solutions which may not be close to optimal solutions. Therefore, these solutions need not be evaluated with high precision. Meta-models for objective functions and constraints have been developed for this purpose. Two different approaches are mostly followed. In one approach, a sample of solutions are used to generate a meta-model (approximate model of the original objectives and constraints) and then efforts have been made to find the optimum of the meta-model, assuming that the optimal solutions of both the meta-model and the original problem are similar to each other [35, 45]. In the other approach, a successive meta-modelling approach is used in which the algorithm starts to solve the first meta-model obtained from a sample of the entire search space [27, 37, 74]. As the solutions start to focus near the optimum region of the meta-model, a new and more accurate meta-model is generated in the region dictated by the solutions of the previous optimization. A coarse-to-fine-grained meta-modelling technique based on artificial neural networks is shown to reduce the computational effort by about 30–80% on different problems [74]. Other successful meta-modeling implementations for multi-objective optimization based on Kriging and response surface methodologies exist [36, 37].

12.7 Conclusions

The research and application in evolutionary multi-objective optimization (EMO) is now at least over 15 years old and has resulted in a number of efficient algorithms for finding a set of well-diversified, near Pareto-optimal solutions. EMO algorithms are now regularly being applied to different problems involving most branches of science, engineering, and commerce.

This chapter started with discussing principles of EMO and illustrated the principle by depicting one efficient and popularly used EMO algorithm. Results from an interplanetary spacecraft trajectory optimization problem reveal the importance of principles followed in EMO algorithms. Thereafter, we made a brief description of a specific constraint handling procedure used in EMO studies.

However, the highlight of this chapter is the description of some of the current research and application activities involving EMO. One critical area of current research lies in collaborative EMO-MCDM algorithms for achieving a complete multi-objective optimization task of finding a set of trade-off solutions and finally arriving at a single preferred solution. Another direction taken by the researchers is to address guaranteed convergence and diversity of EMO algorithms through hybridizing them with mathematical and numerical optimization techniques as local search algorithms. Interestingly, EMO researchers have discovered its potential in solving traditionally hard optimization problems, but not necessarily multi-objective in nature, in a convenient manner using EMO algorithms. The so-called multi-objectivization studies are attracting researchers from various fields to develop and apply EMO algorithms in many innovative ways. A considerable research and application interest has also been put in addressing practical aspects into existing EMO algorithms. Towards this direction, handling uncertainty in decision variables and parameters, meeting an overall desired system reliability in obtained solutions, handling dynamically changing problems (on-line optimization), and handling a large number of objectives have been discussed in this paper. Besides the practical aspects, EMO has also attracted mathematically oriented theoreticians to develop EMO algorithms and design suitable problems for coming up with a computational complexity analysis. There are many other research directions which could not even mention due to space restrictions.

It is clear that the field of EMO research and application, in a short span of about 15 years, now has efficient algorithms and numerous interesting and useful applications, and has been able to attract theoretically and practically oriented researchers to come together and make collaborative activities. The practical importance of EMO's working principle, the flexibility of evolutionary optimization which lies at the core of EMO algorithms, and demonstrated diversification of EMO's principle to a wide variety of different problem-solving tasks are the main cornerstones for their success so far. The scope of research and application in EMO and using EMO are enormous and open-ended. This chapter remains an open invitation to everyone who is interested in any type of problem-solving tasks to take a look at what has been done in EMO and to explore how one can contribute in collaborating with EMO to address problem-solving tasks which are still in need of a better solution procedure.

Acknowledgments The author acknowledges the support of the Academy of Finland (Grant # 118319) and Foundation of Helsinki School of Economics. The support from India Science Laborarory, Bangalore is also appreciated.

References

1. B.V. Babu and M. L. Jehan. Differential evolution for multi-objective optimization. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2696–2703. IEEE Press, Piscataway NJ, 2003.
2. M. Basseur and E. Zitzler. Handling uncertainty in indicator-based multiobjective optimization. *International Journal of Computational Intelligence Research*, 2(3):255–272, 2006.
3. S. Bleuler, M. Brack, and E. Zitzler. Multiobjective genetic programming: Reducing bloat using SPEA2. In *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 536–543. IEEE Press, Piscataway NJ, 2001.
4. P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2), 2003.
5. J. Branke. *Evolutionary Optimization in Dynamic Environments*. Springer-Verlag, Heidelberg, Germany, 2001.
6. J. Branke and K. Deb. Integrating user preferences into evolutionary multi-objective optimization. In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 461–477. Springer-Verlag, Heidelberg, Germany, 2004.
7. J. Branke, K. Deb, H. Dierolf, and M. Osswald. Finding knees in multi-objective optimization. In *Parallel Problem Solving from Nature (PPSN-VIII)*, volume 3242 of *Lecture Notes in computer Science*, pages 722–731. Springer-Verlag, Heidelberg, Germany, 2004.
8. J. Branke, K. Deb, K. Miettinen, and R. Slowinski. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Germany, 2008.
9. D. Brockhoff and E. Zitzler. Dimensionality reduction in multiobjective optimization: The minimum objective subset problem. In K. H. Waldmann and U. M. Stocker, editors, *Operations Research Proceedings 2006*, pages 423–429. Springer-Verlag, Berlin, 2007.
10. D. Brockhoff and E. Zitzler. Offline and online objective reduction in evolutionary multiobjective optimization based on objective conflicts. TIK Report 269, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, 2007.
11. C. A. C. Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1051–1056. IEEE Service Center, Piscataway NJ, 2002.
12. C. A. C. Coello and G. Toscano. A micro-genetic algorithm for multi-objective optimization. Technical Report Lania-RI-2000-06, Laboratoria Nacional de Informatica Avanzada, Xalapa, Veracruz, Mexico, 2000.
13. C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, Boston, MA, 2002.
14. C. A. Coello Coello. Treating objectives as constraints for single objective optimization. *Engineering Optimization*, 32(3):275–308, 2000.
15. C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, editors. *Evolutionary Multi-Criterion Optimization: Third International Conference*, volume 3410 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 2005.
16. D. W. Corne, J. D. Knowles, and M. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature VI (PPSN-VI)*, volume 1917 of *Lecture Notes in Computer Science*, pages 839–848. Springer-Verlag, Berlin, Germany, 2000.
17. V. Coverstone-Carroll, J. W. Hartmann, and W. J. Mason. Optimal multi-objective low-thurst spacecraft trajectories. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4): 387–402, 2000.

18. T. R. Cruse. *Reliability-based Mechanical Design*. Marcel Dekker, New York, 1997.
19. K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK, 2001.
20. K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.
21. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
22. K. Deb and T. Goel. A hybrid multi-objective evolutionary approach to engineering shape design. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, volume 1993 of *Lecture Notes in Computer Science*, pages 385–399. Springer-Verlag, Heidelberg, Germany, 2001.
23. K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation Journal*, 14(4):463–494, 2006.
24. K. Deb, S. Gupta, D. Daum, J. Branke, A. Mall, and D. Padmanabhan. Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, doi 10.1109/TEVC.2009.2014361, 2009.
25. K. Deb and A. Kumar. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, pages 781–788. The Association of Computing Machinery, New York, 2007.
26. K. Deb and A. Kumar. Light beam search based multi-objective optimization using evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation (CEC-07)*, pages 2125–2132. IEEE Press, Piscataway NJ, 2007.
27. K. Deb and P. K. S. Nain. An evolutionary multi-objective adaptive meta-modeling procedure using artificial neural networks. In *Evolutionary Computation in Dynamic and Uncertain Environments*, pages 297–322. Springer-Verlag, Berlin, Germany, 2007.
28. K. Deb, U. B. Rao, and S. Karthik. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling bi-objective optimization problems. In *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2007)*, volume 4403 of *Lecture Notes in Computer Science*, 2007.
29. K. Deb and D. Saxena. Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, pages 3352–3360. IEEE Press, Piscataway NJ, 2006.
30. K. Deb and A. Srinivasan. Innovization: Innovating design principles through optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, pages 1629–1636, The Association of Computing Machinery, New York, 2006.
31. K. Deb, J. Sundar, N. Uday, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(6):273–286, 2006.
32. K. Deb, R. Tiwari, M. Dixit, and J. Dutta. Finding trade-off solutions close to KKT points using evolutionary multi-objective optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC-2007)*, pages 2109–2116. IEEE Press, Piscataway NJ, 2007.
33. K. Deb, P. Zope, and A. Jain. Distributed computing of pareto-optimal solutions using multi-objective evolutionary algorithms. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference*, volume 2632 of *Lecture Notes in Computer Science*, pages 535–549. Spriinger Verlag, Berlin, Germany, 2003.
34. X. Du and W. Chen. Sequential optimization and reliability assessment method for efficient probabilistic design. *ASME Journal of Mechanical Design*, 126(2):225–233, 2004.
35. M. A. El-Beltagy, P. B. Nair, and A. J. Keane. Metamodelling techniques for evolutionary optimization of computationally expensive problems: Promises and limitations. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, pages 196–203. Morgan Kaufmmann, San Mateo, CA, 1999.

36. M. Emmerich and B. Naujoks. Metamodel-assisted multiobjective optimisation strategies and their application in airfoil design. In *Adaptive Computing in Design and Manufacture VI*, pages 249–260. Springer, London, UK, 2004.
37. M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.
38. M. Farina and P. Amato. A fuzzy definition of optimality for many criteria optimization problems. *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*, 34(3):315–326, 2004.
39. M. Fleischer. The measure of Pareto optima: Applications to multi-objective optimization. In *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO-2003)*, volume 1993 of *Lecture Notes in Computer Science*, pages 519–533. Springer-Verlag, Berlin, Germany, 2003.
40. L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley, New York, 1966.
41. C. Fonseca, P. Fleming, E. Zitzler, K. Deb, and L. Thiele. *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference*, volume 2632 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, Germany, 2003.
42. C. M. Fonseca, V. Grunert da Fonseca, and L. Paquete. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Third International Conference on Evolutionary Multi-Criterion Optimization, EMO-2005*, volume 3410 of *Lecture Notes in Computer Science*, pages 250–264. Springer-Verlag, Berlin, Germany, 2005.
43. C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993.
44. C. M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature (PPSN IV)*, volume 1141 of *Lecture Notes in Computer Science*, pages 584–593. Springer-Verlag, Berlin, Germany, 1996.
45. K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Science*, 38(1):43–76, 2002.
46. O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, pages 1918–1925. IEEE Press, Piscataway NJ, 2003.
47. O. Giel and P. K. Lehre. On the effect of populations in evolutionary multi-objective optimization. In *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference (GECCO 2006)*, pages 651–658. ACM Press, New York, 2006.
48. D. E. Goldberg. *Genetic Algorithms for Search, Optimization, and Machine Learning*. Addison-Wesley, Reading MA, 1989.
49. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 41–49, 1987.
50. M. Gravel, W. L. Price, and C. Gagné. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, 143(1):218–229, 2002.
51. J. Handl and J. D. Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76, 2007.
52. M. P. Hansen and A. Jaskiewicz. Evaluating the quality of approximations to the nondominated set. IMM Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, 1998.
53. J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Ann Arbor, MI, 1975.
54. J. Horn, N. Nafploitis, and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In D. Fogel, editor, *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87. IEEE Press, Piscataway, NJ, 1994.

55. J. Jahn. *Vector Optimization*. Springer-Verlag, Berlin, Germany, 2004.
56. H. Jin and M.-L. Wong. Adaptive diversity maintenance and convergence guarantee in multiobjective evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation (CEC-2003)*, pages 2498–2505. IEEE Press, Piscataway, NJ, 2003.
57. E. D. De Jong, R. A. Watson, and J. B. Pollack. Reducing bloat and promoting diversity using multi-objective methods. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 11–18. Morgan Kaufmann, San Mateo, CA, 2001.
58. V. Khare, X. Yao, and K. Deb. Performance scaling of multi-objective evolutionary algorithms. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference*, volume 2632 of *Lecture Notes in Computer Science*, pages 376–390, 2003.
59. J. Knowles and D. Corne. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2007)*, volume 4403 of *Lecture Notes in Computer Science*, pages 757–771. Springer-Verlag, Berlin, Germany, 2007.
60. J. D. Knowles and D. W. Corne. Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation Journal*, 8(2):149–172, 2000.
61. J. D. Knowles and D. W. Corne. On metrics for comparing nondominated sets. In *Congress on Evolutionary Computation (CEC-2002)*, pages 711–716. IEEE Press, Piscataway, NJ, 2002.
62. J. D. Knowles, D. W. Corne, and K. Deb, editors. *Multiobjective Problem Solving from Nature*. Springer Natural Computing Series. Springer-Verlag, Berlin, 2008.
63. P. Korhonen and J. Laakso. A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Reseaech*, 24:277–287, 1986.
64. R. Kumar and N. Banerjee. Analysis of a multiobjective evolutionary algorithm on the 0-1 knapsack problem. *Theoretical Computer Science*, 358(1):104–120, 2006.
65. H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *Journal of the Association for Computing Machinery*, 22(4):469–476, 1975.
66. M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.
67. M. Laumanns, L. Thiele, and E. Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean Functions. *IEEE Transactions on Evolutionary Computation*, 8(2):170–182, 2004.
68. M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running time analysis of multiobjective evolutionary algorithms on a simple discrete optimization problem. In M. Guervós, P. Adamidis, H.-G. Beyer, J. L. Fernández-Villacañas Martín, and H.-P. Schwefel, editors, *Proceedings of the Seventh Conference on Parallel Problem Solving from Nature (PPSN-VII)*, volume 2439 of *Lecture Notes in Computer Science*, pages 44–53. Springer-Verlag, Berlin, Germany, 2002.
69. D. H. Loughlin and S. Ranjithan. The neighborhood constraint method: A multiobjective optimization technique. In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 666–673. Morgan Kaufmann, San Francisco, CA, 1997.
70. M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz. Three different ways for incorporating preference information in interactive reference point based methods. Technical Report W-410, Helsinki School of Economics, Helsinki, Finland, 2006.
71. P. R. McMullen. An ant colony optimization approach to addessing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering*, 15:309–317, 2001.
72. K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.
73. S. Mostaghim and J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *2003 IEEE Swarm Intelligence Symposium Proceedings*, pages 26–33. IEEE Service Center, Piscataway, NJ, 2003.
74. P. K. S. Nain and K. Deb. Computationally effective search and optimization procedure using coarse to fine approximations. In *Proceedings of the Congress on Evolutionary Computation (CEC-2003)*, pages 2081–2088. IEEE Press, Piscataway, NJ, 2003.
75. F. Neumann and I. Wegener. Minimum spanning trees made easier via multi-objective optimization. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 763–769. ACM, New York, 2005.

76. S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors. *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007, Proceedings*, volume 4403 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 2007.
77. A. Osyczka. *Evolutionary algorithms for single and multicriteria design optimization*. Physica-Verlag, Heidelberg, Germany, 2002.
78. R. S. Rosenberg. *Simulation of Genetic Populations with Biochemical Properties*. Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1967.
79. G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Network*, 5(1):96–101, 1994.
80. D. Sasaki, M. Morikawa, S. Obayashi, and K. Nakahashi. Aerodynamic shape optimization of supersonic wings by adaptive range multiobjective genetic algorithms. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, pages 639–652. Springer-Verlag, Berlin, Germany, 2001.
81. C. G. Sauer. Optimization of multiple target electric propulsion trajectories. In *AIAA 11th Aerospace Science Meeting*, 1973. Paper Number 73-205.
82. Z. M. Saul and C. A. C. Coello. A proposal to hybridize multi-objective evolutionary algorithms with non-gradient mathematical programming techniques. In *Proceedings of the Parallel Problem Solving from Nature (PPSN-2008)*, volume 5199 of *Lecture Notes in Computer Science*, pages 837–846. Springer-Verlag, Berlin, Germany, 2008.
83. D. K. Saxena and K. Deb. Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2007)*, volume 4403 of *Lecture Notes in Computer Science*, pages 772–787. Springer-Verlag, Berlin, Germany, 2007.
84. J. D. Schaffer. *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*. Ph.D. thesis, Vanderbilt University, Nashville, TN, 1984.
85. P. Shukla and K. Deb. On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. *European Journal of Operational Research*, 181(3):1630–1652, 2007.
86. K. Sindhya, K. Deb, and K. Miettinen. A local search based evolutionary multi-objective optimization technique for fast and accurate convergence. In G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, editors, *Proceedings of the Paralle Problem Solving From Nature (PPSN-2008)*, Lecture Notes in Computer Science, pages 815–824. Springer-Verlag, Berlin, Germany, 2008.
87. N. Srinivas and K. Deb. Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal*, 2(3):221–248, 1994.
88. L. Thiele, K. Miettinen, P. Korhonen, and J. Molina. A preference-based interactive evolutionary algorithm for multiobjective optimization. Technical Report W-412, Helsingki School of Economics, Finland, 2007.
89. D. Van Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation Journal*, 8(2):125–148, 2000.
90. A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Applications*, pages 468–486. Springer-Verlag, Berlin, Germany, 1980.
91. E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. W. Corne. *Proceedings of the First Evolutionary Multi-Criterion Optimization (EMO-01) Conference*, volume 1993 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, Germany, 2001.
92. E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer-Verlag, Berlin, Germany, 2004.
93. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux,

K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2001.

94. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
95. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.