Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction

Kalyanmoy Deb
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, PIN 208016, India
deb@iitk.ac.in
http://www.iitk.ac.in/kangal/deb.htm

February 10, 2011
KanGAL Report Number 2011003

Abstract

As the name suggests, multi-objective optimization involves optimizing a number of objectives simultaneously. The problem becomes challenging when the objectives are of conflict to each other, that is, the optimal solution of an objective function is different from that of the other. In solving such problems, with or without the presence of constraints, these problems give rise to a set of trade-off optimal solutions, popularly known as Pareto-optimal solutions. Due to the multiplicity in solutions, these problems were proposed to be solved suitably using evolutionary algorithms which use a population approach in its search procedure. Starting with parameterized procedures in early nineties, the so-called evolutionary multi-objective optimization (EMO) algorithms is now an established field of research and application with many dedicated texts and edited books, commercial softwares and numerous 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 provide a brief introduction to its operating principles and outline the current research and application studies of EMO.

1 Introduction

In the past 15 years, evolutionary multi-objective optimization (EMO) has become a popular and useful field of research and application. 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: (i) EOs do not require any derivative information (ii) EOs are relatively simple to implement and (iii) EOs are flexible and have a wide-spread applicability. For solving single-objective optimization problems, particularly in finding a single optimal solution, the use of a population of solutions may sound redundant, in solving multi-objective optimization problems an EO procedure is a perfect choice [1]. The multi-objective optimization problems, by nature, give rise to a set of Pareto-optimal solutions which need a 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 population in an iteration helps an EO to simultaneously find multiple non-dominated solutions, which portrays a trade-off among objectives, in a single simulation run.

In this chapter, we present a brief description of an evolutionary optimization procedure for single-objective optimization. Thereafter, we describe the principles of evolutionary multi-objective optimization. Then, we discuss some salient developments in EMO research. It is clear from these discussions that EMO is not only being found to be useful in solving multi-objective optimization problems, it is also helping to solve other kinds of optimization problems in a better manner than they are traditionally solved. As a by-product, EMO-based solutions are helping to reveal important hidden knowledge about a problem – a matter which is difficult to achieve otherwise. EMO procedures with a decision making concept are

discussed as well. Some of these ideas require further detailed studies and this chapter mentions some such current and future topics in this direction.

2 Evolutionary Optimization (EO) for Single-Objective Optimization

Evolutionary optimization principles are different from classical optimization methodologies in the following main ways [2]:

- An EO procedure does not usually use gradient information in its search process. Thus, EO methodologies are direct search procedures, allowing them to be applied to a wide variety of optimization problems.
- An EO procedure uses more than one solution (a *population* approach) in an iteration, unlike in most classical optimization algorithms which updates one solution in each iteration (a *point* approach). The use of a population has a number of advantages: (i) it provides an EO with a parallel processing power achieving a computationally quick overall search, (ii) it allows an EO to find multiple optimal solutions, thereby facilitating the solution of multi-modal and multi-objective optimization problems, and (iii) it provides an EO with the ability to normalize decision variables (as well as objective and constraint functions) within an evolving population using the population-best minimum and maximum values.
- An EO procedure uses stochastic operators, unlike deterministic operators used in most classical optimization methods. The operators tend to achieve a desired effect by using higher probabilities towards desirable outcomes, as opposed to using predetermined and fixed transition rules. This allows an EO algorithm to negotiate multiple optima and other complexities better and provide them with a global perspective in their search.

An EO begins its search with a population of solutions usually created at random within a specified lower and upper bound on each variable. Thereafter, the EO procedure enters into an iterative operation of updating the current population to create a new population by the use of four main operators: selection, crossover, mutation and elite-preservation. The operation stops when one or more pre-specified termination criteria are met.

The initialization procedure usually involve a random creation of solutions. If in a problem the knowledge of some good solutions is available, it is better to use such information in creating the initial population. Elsewhere [3], it is highlighted that for solving complex real-world optimization problems, such a customized initialization is useful and also helpful in achieving a faster search. After the population members are evaluated, the selection operator chooses above-average (in other words, better) solutions with a larger probability to fill an intermediate mating pool. For this purpose, several stochastic selection operators exist in the EO literature. In its simplest form (called the *tournament* selection [4]), two solutions can be picked at random from the evaluated population and the better of the two (in terms of its evaluated order) can be picked.

The 'variation' operator is a collection of a number of operators (such as crossover, mutation etc.) which are used to generate a modified population. The purpose of the crossover operator is to pick two or more solutions (parents) randomly from the mating pool and create one or more solutions by exchanging information among the parent solutions. The crossover operator is applied with a crossover probability ($p_c \in [0, 1]$), indicating the proportion of population members participating in the crossover operation. The remaining $(1 - p_c)$ proportion of the population is simply copied to the modified (child) population. In the context of real-parameter optimization having n real-valued variables and involving a crossover with two parent solutions, each variable may be crossed at a time. A probability distribution which depends on the difference between the two parent variable values is often used to create two new numbers as child values around the two parent values [5]. Besides the variable-wise recombination operators, vector-wise recombination operators also suggested to propagate the correlation among variables of parent solutions to the created child solutions [6, 7].

Each child solution, created by the crossover operator, is then perturbed in its vicinity by a mutation operator [2]. Every variable is mutated with a mutation probability p_m, usually set as $1/n$ (n is the number

of variables), so that on an average one variable gets mutated per solution. In the context of real-parameter optimization, a simple Gaussian probability distribution with a predefined variance can be used with its mean at the child variable value [1]. This operator allows an EO to search locally around a solution and is independent on the location of other solutions in the population.

The elitism operator combines the old population with the newly created population and chooses to keep better solutions from the combined population. Such an operation makes sure that an algorithm has a monotonically non-degrading performance. [8] proved an asymptotic convergence of a specific EO but having elitism and mutation as two essential operators.

Finally, the user of an EO needs to choose termination criteria. Often, a predetermined number of generations is used as a termination criterion. For goal attainment problems, an EO can be terminated as soon as a solution with a predefined goal or a target solution is found. In many studies [2, 9, 10, 11], a termination criterion based on the statistics of the current population vis-a-vis that of the previous population to determine the rate of convergence is used. In other more recent studies, theoretical optimality conditions (such as the extent of satisfaction of Karush-Kuhn-Tucker (KKT) conditions) are used to determine the termination of a real-parameter EO algorithm [12]. Although EOs are heuristic based, the use of such theoretical optimality concepts in an EO can also be used to test their converging abilities towards local optimal solutions.

To demonstrate the working of the above-mentioned GA, we show four snapshots of a typical simulation run on the following constrained optimization problem:

$$\begin{array}{rl} \text{Minimize} & f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \\ \text{subject to} & g_1(x) \equiv 26 - (x_1 - 5)^2 - x_2^2 \geq 0, \\ & g_2(x) \equiv 20 - 4x_1 - x_2 \geq 0, \\ & 0 \leq (x_1, x_2) \leq 6. \end{array} \tag{1}$$

10 points are used and the GA is run for 100 generations. The SBX recombination operator is used with probability of $p_c = 0.9$ and index $\eta_c = 10$. The polynomial mutation operator is used with a probability of $p_m = 0.5$ with an index of $\eta_m = 50$. Figures 1 to 4 show the populations at generation zero, 5, 40 and 100, respectively. It can be observed that in only five generations, all 10 population members become feasible. Thereafter, the points come close to each other and creep towards the constrained minimum point.

The EA procedure is a population-based stochastic search procedure which iteratively emphasizes its better population members, uses them to recombine and perturb locally in the hope of creating new and better populations until a predefined termination criterion is met. The use of a population helps to achieve an *implicit parallelism* [2, 13, 14] in an EO's search mechanism (causing an inherent parallel search in different regions of the search space), a matter which makes an EO computationally attractive for solving difficult problems. In the context of certain Boolean functions, a computational time saving to find the optimum varying polynomial to the population size is proven [15]. On one hand, the EO procedure is flexible, thereby allowing a user to choose suitable operators and problem-specific information to suit a specific problem. On the other hand, the flexibility comes with an onus on the part of a user to choose appropriate and tangible operators so as to create an efficient and consistent search [16]. However, the benefits of having a flexible optimization procedure, over their more rigid and specific optimization algorithms, provide hope in solving difficult real-world optimization problems involving non-differentiable objectives and constraints, non-linearities, discreteness, multiple optima, large problem sizes, uncertainties in computation of objectives and constraints, uncertainties in decision variables, mixed type of variables, and others.

A wiser approach to solving optimization problems of the real world would be to first understand the niche of both EO and classical methodologies and then adopt hybrid procedures employing the better of the two as the search progresses over varying degrees of search-space complexity from start to finish. As demonstrated in the above typical GA simulation, there are two phases in the search of a GA. First, the GA exhibits a more *global* search by maintaining a diverse population, thereby discovering potentially good regions of interest. Second, a more *local* search takes place by bringing the population members closer together. Although the above GA degenerates to both these search phases automatically without any external intervention, a more efficient search can be achieved if the later local search phase can be identified and executed with a more specialized local search algorithm.

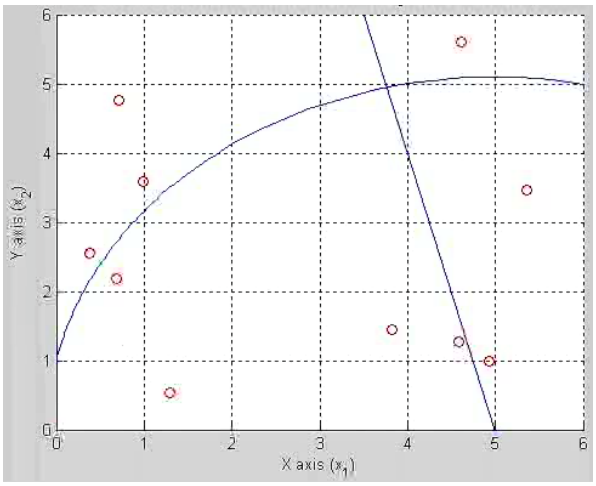

Figure 1: Initial population.

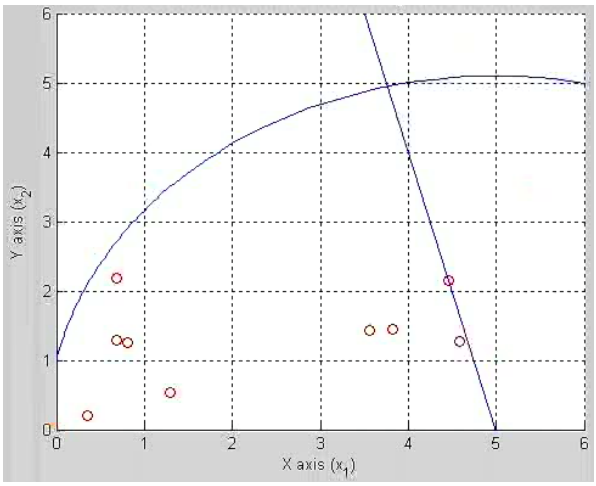

Figure 2: Population at generation 5.

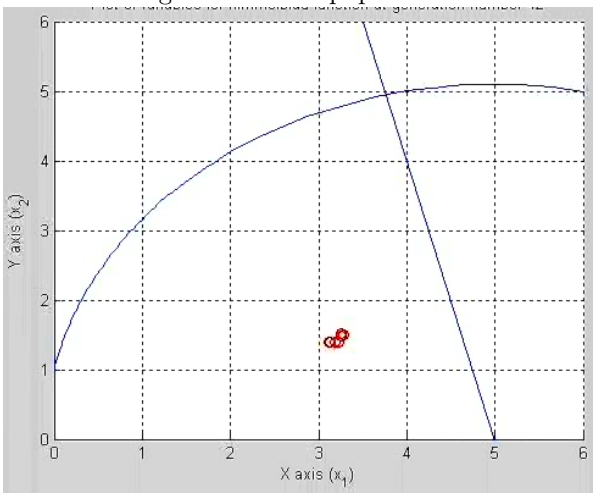

Figure 3: Population at generation 40.

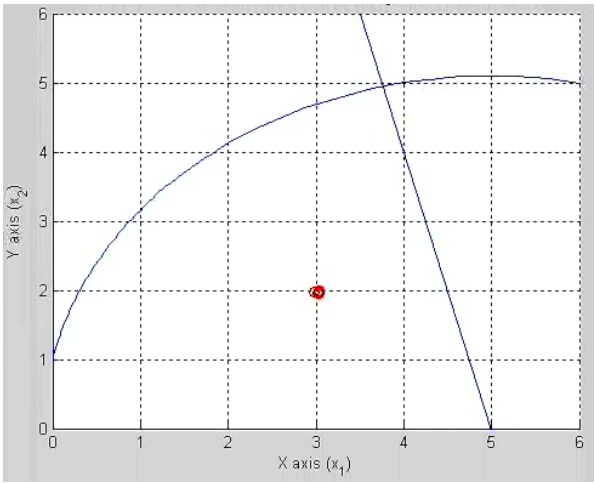

Figure 4: Population at generation 100.

3 Evolutionary Multi-objective Optimization (EMO)

A multi-objective optimization problem involves a number of objective functions which are to be either minimized or maximized. As in a single-objective optimization problem, the multi-objective optimization problem may contain a number of constraints which any feasible solution (including all optimal solutions) must satisfy. Since objectives can be either minimized or maximized, we state the multi-objective optimization problem in its general form:

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

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 *feasible decision variable space* $S \subset \mathbf{R}^n$. One of the striking differences between single-objective and multi-objective optimization is that in multi-objective optimization the objective functions constitute a multi-dimensional space, in addition to the usual decision variable space. This additional M-dimensional space is called the *objective space*, $Z \subset \mathbf{R}^M$. For each solution $\mathbf{x}$ in the decision variable space, there exists a point $\mathbf{z} \in \mathbf{R}^M$) in the objective space, denoted by $\mathbf{f}(\mathbf{x}) = \mathbf{z} = (z_1, z_2, \ldots, z_M)^T$. To make the descriptions clear, we refer a 'solution' as a variable vector and a 'point' as the corresponding objective vector.

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 [1, 17]:

Definition 3.1 *A solution $\mathbf{x}^{(1)}$ is said to dominate the other 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 $\mathbf{z}^{(2)}$) on the objective space).*
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 on the objective space, for example, those shown in Figure 5(a)), a pair-wise comparison can be made using the above definition and whether one point dominates the other can be established. All points which are not dominated by any other member of the

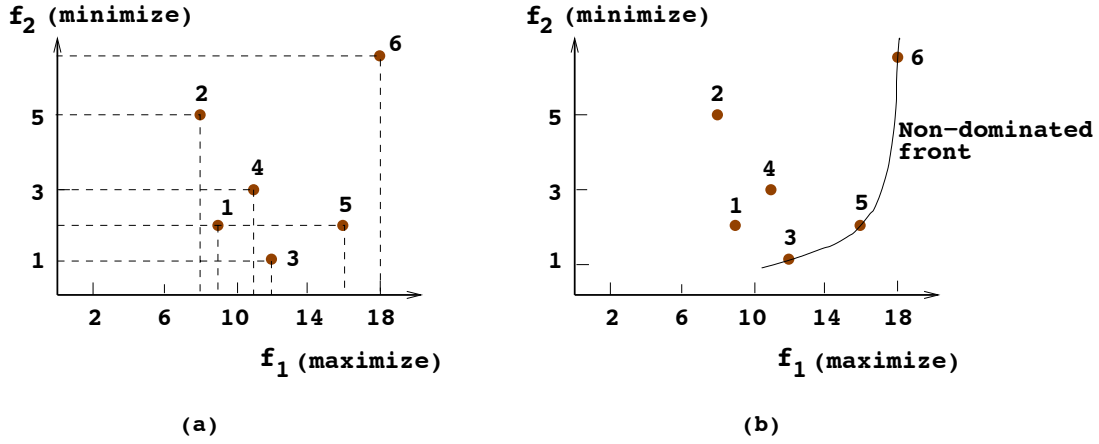

Figure 5: A set of points and the first non-domination front are shown.

set are called the non-dominated points of class one, or simply the non-dominated points. For the set of six solutions 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 them together on the objective space; hence the non-dominated points are often visualized to represent a *non-domination front*. The computational effort needed to select the points of the non-domination front from a set of N points is $O(N \log N)$ for 2 and 3 objectives, and $O(N \log^{M-2} N)$ for $M > 3$ objectives [18].

With the above concept, now it is 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 search space (assuming a countable number), 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 literature [17, 19].

3.1 Principle of EMO's Search

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. Different solutions may produce trade-offs (conflicting outcomes among objectives) among different objectives. A solution that is extreme (in a better sense) with respect to one objective requires a compromise in other objectives. This prohibits one to choose a solution which is optimal with respect to only one objective. This clearly suggests two ideal goals of multi-objective optimization:

1. Find a set of solutions which lie on the Pareto-optimal front, and
2. Find a set of solutions which are diverse enough to represent the entire range of the Pareto-optimal front.

Evolutionary multi-objective optimization (EMO) algorithms attempt to follow both the above principles similar to the other a posteriori MCDM methods (refer to Chapter).

Although one fundamental difference between single and multiple objective optimization lies in the cardinality in the optimal set, from a practical standpoint a user needs only one solution, no matter whether the associated optimization problem is single or multi-objective. The user is now in a dilemma. Since a number of solutions are optimal, the obvious question arises: Which of these optimal solutions must one choose? This is not an easy question to answer. It involves higher-level information which is often non-technical, qualitative and experience-driven. However, if a set of many trade-off solutions are already worked out or available, one can evaluate the pros and cons of each of these solutions based on all such non-technical and qualitative, yet still important, considerations and compare them to make a choice. Thus, in a multi-objective optimization, ideally the effort must be made in finding the set of trade-off optimal solutions by considering all objectives to be important. After a set of such trade-off solutions are found, a user can then use higher-level qualitative considerations to make a choice. Since an EMO procedure deals with a population of solutions in every iteration, it makes them intuitive to be applied in multi-objective optimization to find a set of non-dominated solutions. Like other a posteriori MCDM methodologies, an EMO based procedure works with the following principle in handling multi-objective optimization problems:

Step 1 Find multiple non-dominated points as close to the Pareto-optimal front as possible, with a wide trade-off among objectives.

Step 2 Choose one of the obtained points using higher-level information.

Figure 6 shows schematically the principles, followed in an EMO procedure. Since EMO procedures are heuristic based, they may not guarantee in finding Pareto-optimal points, as a theoretically provable optimization method would do for tractable (for example, linear or convex) problems. But EMO procedures have essential operators to constantly improve the evolving non-dominated points (from the point of view of convergence and diversity discussed above) similar to the way most natural and artificial evolving systems

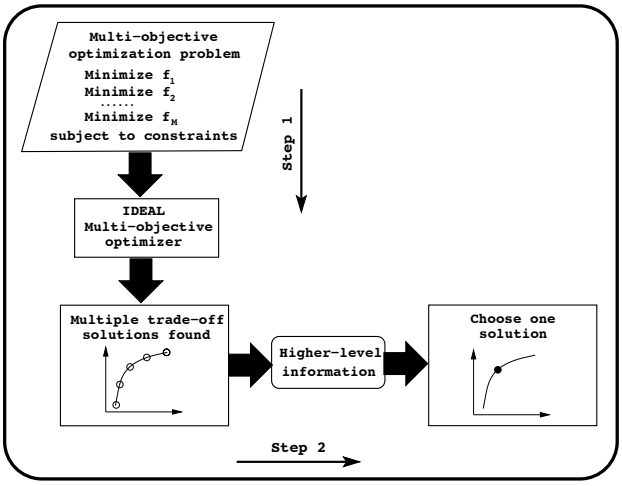

Figure 6: Schematic of a two-step multi-objective optimization procedure.

continuously improve their solutions. To this effect, a recent simulation study [12] has demonstrated that a particular EMO procedure, starting from random non-optimal solutions, can progress towards 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 simulation run, as most a posteriori MCDM methodologies would require multiple applications.

In Step 1 of the EMO-based multi-objective optimization (the task shown vertically downwards in Figure 6), multiple trade-off, non-dominated 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. This dual task allows an interesting feature, if applied for solving single-objective optimization problems. It is easy to realize that a single-objective optimization is a degenerate case of multi-objective optimization, as shown in details in another study [20]. In the case of single-objective optimization having only one globally optimal solution, Step 1 will ideally find only one solution, thereby not requiring us to proceed to Step 2. However, in the case of single-objective optimization having multiple global optima, both steps are necessary to first find all or multiple global optima, and then to choose one solution from them by using a higher-level information about the problem. Thus, although seems ideal for multi-objective optimization, the framework suggested in Figure 6 can be ideally thought as a generic principle for both single and multiple objective optimization.

3.2 Generating Classical Methods and EMO

In the generating MCDM approach, 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. 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 objective function. By simply varying the parameters (weight vector or ϵ-vector) and optimizing the scalarized function, different Pareto-optimal solutions can be found. In contrast, in an EMO, multiple Pareto-optimal solutions are attempted to be found in a single simulation by emphasizing multiple non-dominated and isolated solutions. We discuss a little later some EMO algorithms describing how such dual emphasis is provided, but now discuss qualitatively the difference between a posteriori MCDM and EMO approaches.

Consider Figure 7, in which we sketch how multiple independent parametric single-objective optimizations may find different Pareto-optimal solutions. The Pareto-optimal front corresponds to global optimal solutions of several scalarized objectives. However, during the course of an optimization task, algorithms must overcome a number of difficulties, such as infeasible regions, local optimal solutions, flat regions of objective functions, isolation of optimum, etc., to converge to the global optimal solution. Moreover, due to practical limitations, an optimization task must also be completed in a reasonable computational time.

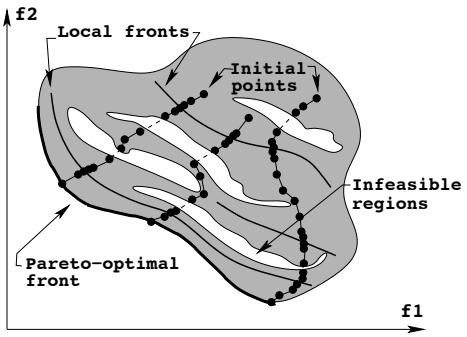

Figure 7: Generative MCDM methodology employs multiple, independent single-objective optimizations.

This requires an algorithm to strike a good balance between the extent of these tasks its search operators must do to overcome the above-mentioned difficulties reliably and quickly. When multiple simulations are to performed to find a set of Pareto-optimal solutions, the above balancing act must have to performed in every single simulation. Since simulations are performed independently, no information about the success or failure of previous simulations is used to speed up the process. In difficult multi-objective optimization problems, such memory-less a posteriori methods may demand a large overall computational overhead to get a set of Pareto-optimal solutions. Moreover, even though the convergence can be achieved in some problems, independent simulations can never guarantee finding a good distribution among obtained points.

EMO, as mentioned earlier, constitutes an inherent parallel search. When a population member overcomes certain difficulties and make a progress towards the Pareto-optimal front, its variable values and their combination reflect this fact. When a recombination takes place between this solution and other population members, 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.

3.3 Elitist Non-dominated Sorting GA or 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 classified into different non-domination classes. Thereafter, the new population is filled by points of different non-domination fronts, one at a time. The filling starts with the first non-domination front (of class one) and continues with points of the second non-domination 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 Figure 8. 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 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

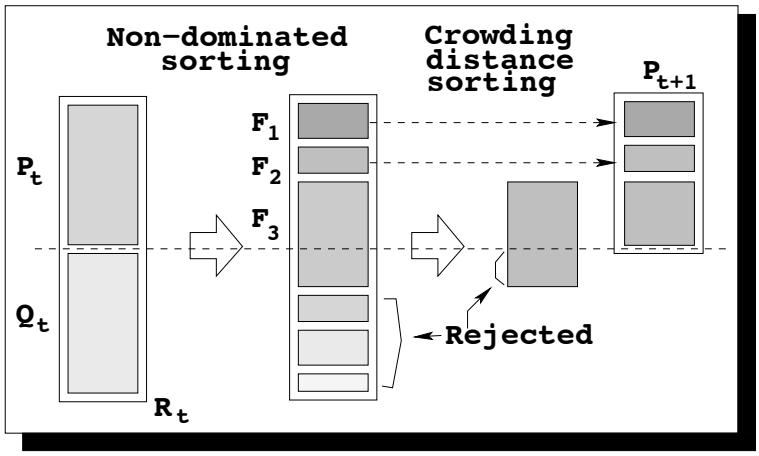

Figure 8: Schematic of the NSGA-II procedure.

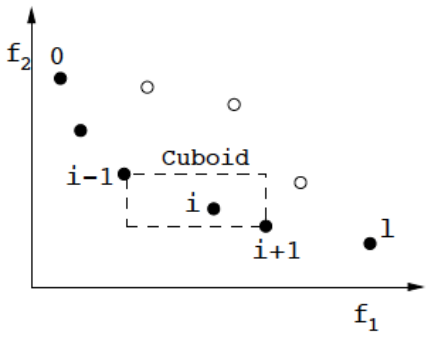

Figure 9: The crowding distance calculation.

the perimeter of the cuboid (Figure 9) formed by using the nearest neighbors in the objective space as the vertices (we call this the *crowding distance*).

Next, we show snapshots of a typical NSGA-II simulation on a two-objective test problem:

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

NSGA-II is run with a population size of 100 and for 100 generations. The variables are used as real numbers and an SBX recombination operator with $p_c = 0.9$ and distribution index of $\eta_c = 10$ and a polynomial mutation operator [1] with $p_m = 1/n$ (n is the number of variables) and distribution index of $\eta_m = 20$ are used. Figure 10 is the initial population shown on the objective space. Figures 11, 12, and 13 show populations at generations 10, 30 and 100, respectively. The figures illustrates how the operators of NSGA-II cause the population to move towards the Pareto-optimal front with generations. At generation 100, the population comes very close to the true Pareto-optimal front.

4 Applications of EMO

Since the early development of EMO algorithms in 1993, they have been applied to many real-world and interesting optimization problems. Descriptions of some of these studies can be found in books [1, 22, 23], dedicated conference proceedings [24, 25, 26, 27], 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 3.1.

4.1 Spacecraft Trajectory Design

[28] proposed a multi-objective optimization technique using the original non-dominated sorting algorithm (NSGA) [29] 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 [30] is called for, 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.

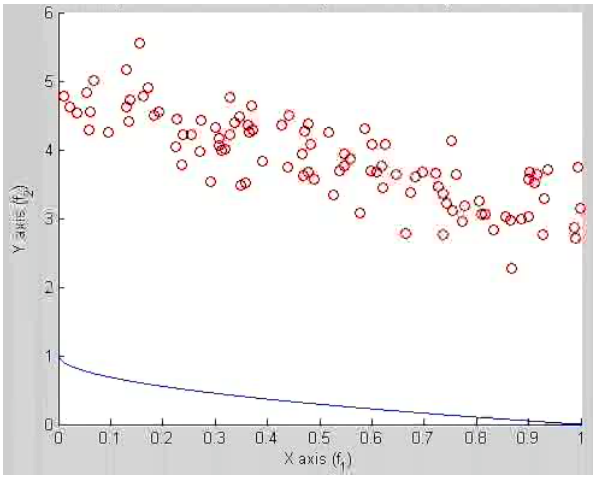

Figure 10: Initial population.

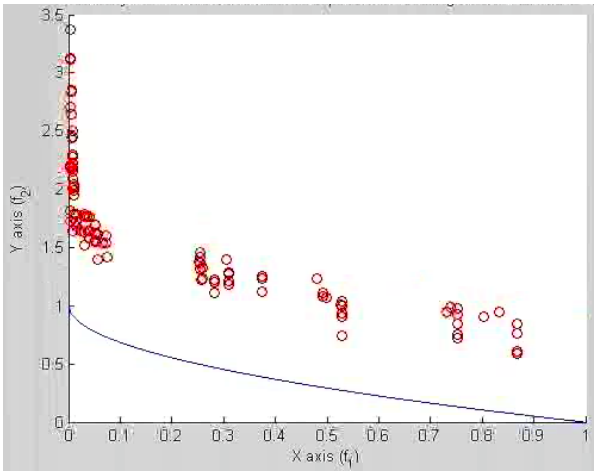

Figure 11: Population at generation 10.

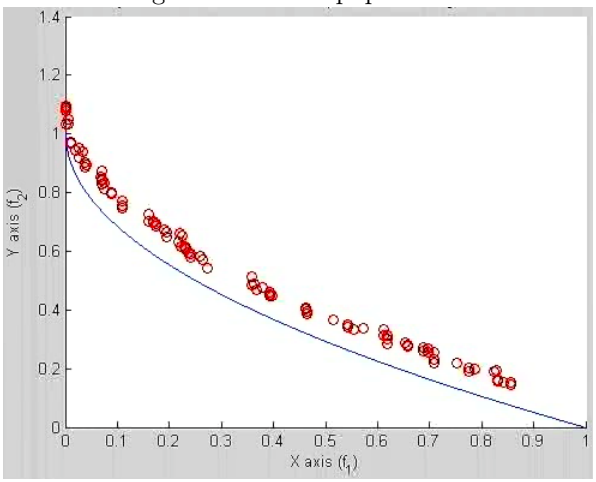

Figure 12: Population at generation 30.

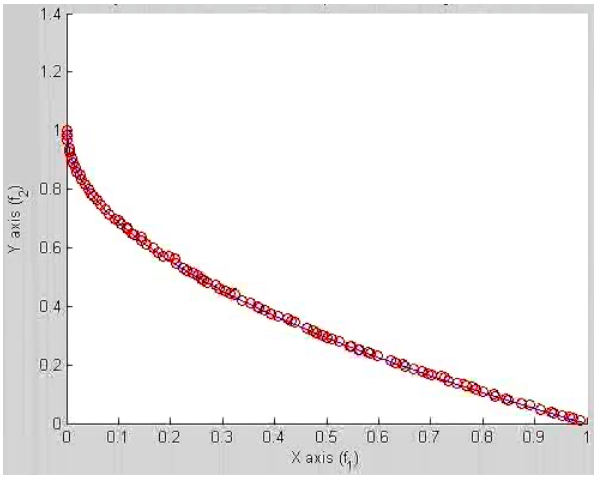

Figure 13: Population at generation 100.

On the Earth–Mars rendezvous mission, the study found interesting trade-off solutions [28]. Using a population of size 150, the NSGA was run for 30 generations. The obtained non-dominated solutions are shown in Figure 14 for two of the three objectives and some selected solutions are shown in Figure 15. It is

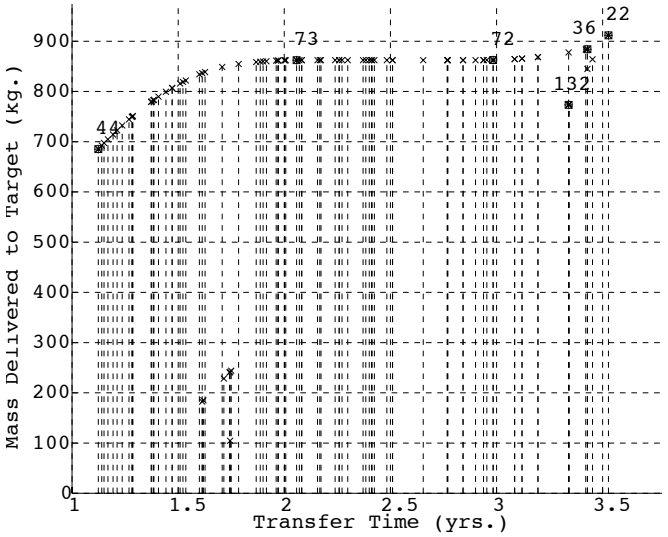

Figure 14: Obtained non-dominated solutions using NSGA.

clear that there exist short-time flights with smaller delivered payloads (solution marked 44) and long-time flights with larger delivered payloads (solution marked 36). Solution 44 can deliver a mass of 685.28 kg and requires about 1.12 years. On other hand, an intermediate 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 an year.

The multiplicity in trade-off solutions, as depicted in Figure 15, is what we envisaged in discovering in a multi-objective optimization problem by using a posteriori procedure, such as an EMO algorithm. This aspect was also discussed in Figure 6. Once such a set of solutions with a good trade-off among objectives is obtained, one can analyze them for choosing a particular solution. For example, 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. 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, EMO can help narrow down the choices and allow a decision maker to make a better decision. Without the knowledge of such a wide variety of trade-off solutions, a proper decision-making may be a difficult task. Although one can choose a scalarized objective (such as the ϵ-constraint method with a particular ϵ vector) and find the resulting optimal solution, 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.

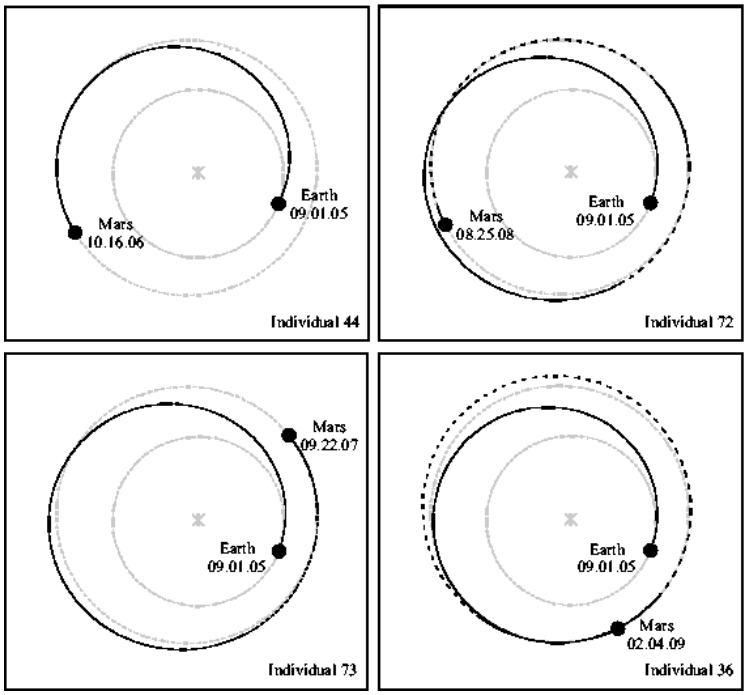

Figure 15: Four trade-off trajectories.

5 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 5.1 *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:*

$$\mathrm{CV}(\mathbf{x}) = \sum_{j=1}^{J} \langle \bar{g}_j(\mathbf{x}) \rangle + \sum_{k=1}^{K} \mathrm{abs}(\bar{h}_k(\mathbf{x})),$$

where $\langle\alpha\rangle$ is $-\alpha$, if $\alpha < 0$ and is zero, otherwise. The normalization is achieved with the population minimum ($\langle g_j\rangle_{\min}$) and maximum ($\langle g_j\rangle_{\max}$) constraint violations: $\bar{g}_j(\mathbf{x}) = (\langle g_j(\mathbf{x})\rangle - \langle g_j\rangle_{\min})/(\langle g_j\rangle_{\max} - \langle g_j\rangle_{\min})$.

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 3.1).*

The above change in the definition requires a minimal change in the NSGA-II procedure described earlier. Figure 16 shows the non-domination fronts on a six-membered population due to the introduction of two constraints (the minimization problem is described as CONSTR elsewhere [1]). In the absence of the constraints, the non-domination 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 non-domination front

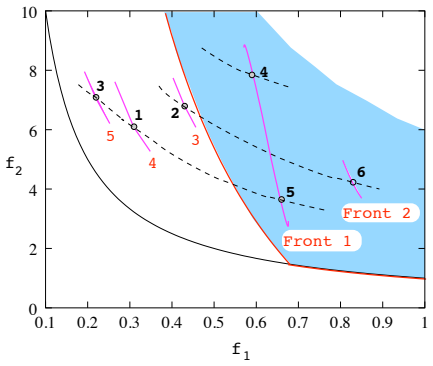

Figure 16: Non-constrained-domination fronts.

consists of the "best" (that is, non-dominated and feasible) points from the population and any feasible point lies on a better non-domination front than an infeasible point.

6 Performance Measures Used in EMO

There are two goals of an EMO procedure: (i) a good convergence to the Pareto-optimal front and (ii) a good diversity in obtained solutions. Since both are conflicting in nature, comparing two sets of trade-off solutions also require different performance measures. 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-metrics, etc.).

A detailed study [31] comparing most existing performance metrics based on out-performance relations has concluded that R-metrics suggested by [32] are the best. However, a study has argued that a single unary performance measure (any of the first two metrics described above in the enumerated list) cannot adequately determine a true winner, as both aspects of convergence and diversity cannot be measured by a single performance metric [33]. 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 binary metrics computes $M(M-1)$ performance values for two algorithms in an M-objective optimization problem, by analyzing all pair-wise

performance comparisons, thereby making them difficult to use in practice. In addition, unary and binary attainment indicators of [34, 35] are of great importance. Figures 17 and 18 illustrate the hypervolume and attainment indicators. Attainment surface is useful to determine a representative front obtained from

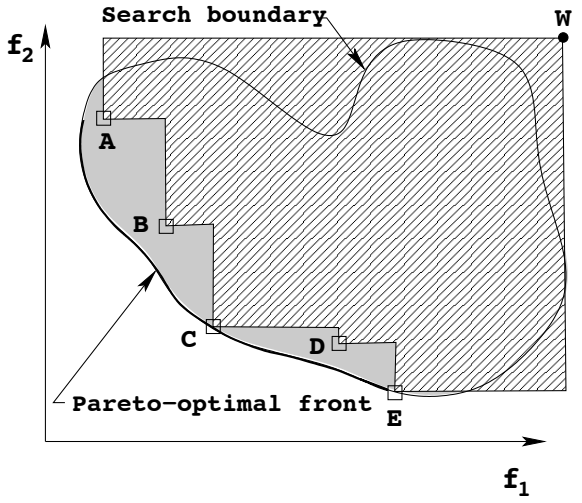

Figure 17: The hypervolume enclosed by the non-dominated solutions.

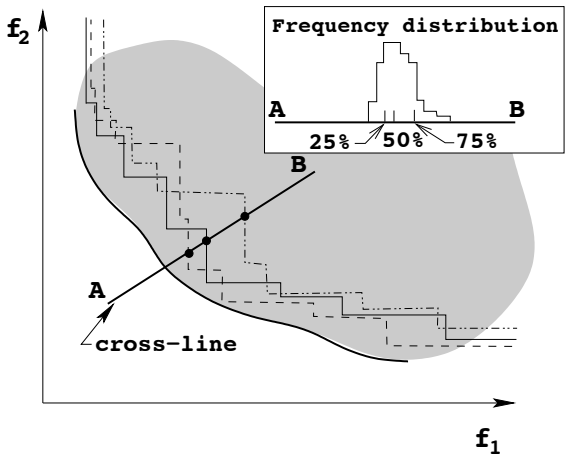

Figure 18: The attainment surface is created for a number of non-dominated solutions.

multiple runs of an EMO algorithm. In general, 50% surface can be used to indicate the front that is dominated by 50% of all obtained non-dominated points.

7 EMO and Decision-making

Finding a set of representative Pareto-optimal solutions using an EMO procedure is half the task; choosing a single preferred solution from the obtained set is also an equally important task. There are three main directions of developments in this direction.

In the a-priori approach, preference information of a decision-maker (DM) is used to focus the search effort into a part of the Pareto-optimal front, instead of the entire frontier. For this purpose, a reference point approach [36], a reference direction approach [37], "light beam" approach [38] etc. have been incorporated in a NSGA-II procedure to find a preferred part of the Pareto-optimal frontier.

In the a-posteriori approach, preference information is used after a set of representative Pareto-optimal solutions are found by an EMO procedure. The multiple criteria decision making (MCDM) approaches including reference point method, Tchebyshev metric method etc. [17] can be used. This approach is now believed to be applicable only to two, three or at most four-objective problems. As the number of objectives increase, EMO methodologies exhibit difficulties in converging close to the Pareto-optimal front and the a-posteriori approaches become a difficult proposition.

In the interactive approach, DM's preference information is integrated to an EMO algorithm during the optimization run. In the progressively interactive EMO approach [39], the DM is called after every τ generations and is presented with a few well-diversified solutions chosen from the current non-dominated front. The DM is then asked to rank the solutions according to preference. The information is then processed through an optimization task to capture DM's preference using an utility function. This utility function is then used to drive NSGA-II's search till the procedure is repeated in the next DM call.

The decision-making procedure integrated with an EMO procedure makes the multi-objective optimization procedure complete. More such studies must now be executed to make EMO more usable in practice.

8 Multiobjectivization

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 per se. The concept of finding multiple trade-off solutions using an EMO procedure is applied to solve other kinds of optimization problems that are otherwise not multi-objective in nature. 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 [40]. This eliminates the need to specify a penalty parameter while using a penalty based constraint handling procedure. A recent study [41] utilizes a bi-objective NSGA-II to find a Pareto-optimal frontier corresponding to minimizations of the objective function and constraint violation. The frontier is then used to estimate an appropriate penalty parameter, which is then used to formulate a penalty based local search problem and is solved using a classical optimization method. The approach is shown to require an order or two magnitude less function evaluations than the existing constraint handling methods on a number of standard test problems.

A well-known difficulty in genetic programming studies, called the 'bloating', arises due to the continual increase in size of genetic programs with iteration. The reduction of bloating by minimizing the size of programs as an additional objective helped find high-performing solutions with a smaller size of the code [42]. Minimizing the intra-cluster distance and maximizing inter-cluster distance simultaneously in a bi-objective formulation of a clustering problem 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 [43]. A recent edited book [44] 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.

8.1 Knowledge Discovery Through EMO

One striking difference between a single-objective optimization and multi-objective optimization is the cardinality of the solution set. In 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 [45]. Figure 19 shows that of the five decision variables involved in an electric motor design problem involving minimum cost and maximum peak-torque, four variables have identical values for all Pareto-optimal solutions [46]. Of the two allowable electric connections, the 'Y'-type connection; of three laminations, 'Y'-type lamination; of 10 to 80 different turns, 18 turns, and of 16 different wire sizes, 16-gauge wire remain common to all Pareto-optimal solutions. The only way the solutions vary is having different number of laminations. In fact, for a motor having more peak-torque, a linearly increasing number of laminations becomes a recipe for optimal more design. 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 [47] to automate the knowledge extraction procedure through a sophisticated data-mining task is promising and should make the overall approach more appealing to the practitioners.

9 Hybrid EMO procedures

The search operators used in EMO are generic. There is no guarantee that an EMO will find any Pareto-optimal solution in a finite number of solution evaluations for an arbitrary problem. However, as discussed above, 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 guaranteed manner, EMO methodologies must

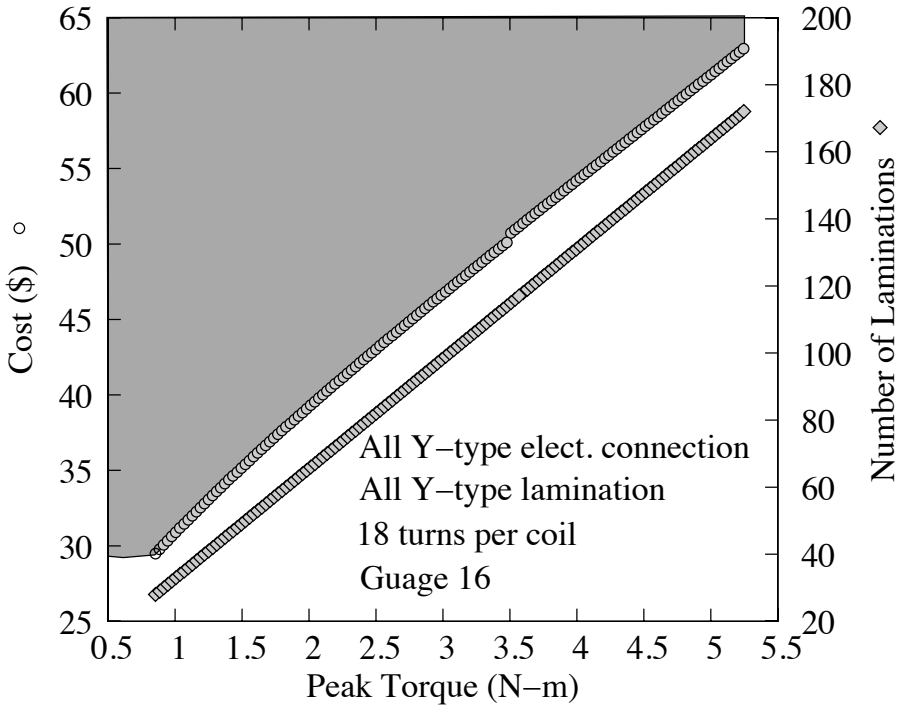

Figure 19: Innovization study of an electric motor design problem.

be combined with mathematical optimization techniques having local convergence properties. A simple-minded approach would be to start the optimization task 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. 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 local optimal solutions. A study [48] has demonstrated that the the latter approach is an overall better approach from a computational point of view.

However, the use of a local search technique within an EMO has another advantage. Since, a local search can find a weak or a near Pareto-optimal point, the presence of such super-individual in a population can cause other near Pareto-optimal solutions to be found as a outcome of recombination of the super-individual with other population members. A recent study has demonstrated this aspect [49].

10 Practical EMOs

Here, we describe some recent advances of EMO in which different practicalities are considered.

10.1 EMO for Many Objectives

With the success of EMO in two and three objective problems, it has become an obvious quest to investigate if an EMO procedure can also be used to solve four or more objective problems. An earlier study [50] with eight objectives revealed somewhat negative results. 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 non-dominated, whereas in a 10-objective problem scenario, as high as 90% members in a population of size 200 are non-dominated. 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. Moreover, an exponentially large population size is needed to represent a large-dimensional Pareto-optimal front. 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.

10.1.1 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 [36, 51], 'light beam search' [38], biased sharing approaches [52], cone dominance [53] etc. are suggested for this purpose. Each of these studies have shown that up to 10 and 20-objective problems, although finding the complete fortier 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 non-dominated, 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 [53] 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.

10.1.2 Identifying and Eliminating Redundant Objectives

Many practical optimization problems can easily list a large of number of objectives (often more than 10), as many different criterion 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 [54, 55] 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 [56] 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 [57], 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 [58] 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.

10.2 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. 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 [59].

A recent study [60] 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 hydro-thermal power dispatch problem have been recently tackled [60].

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.

10.3 Uncertainty Handling Using 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 stochastiticies 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 [61]. 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 [62, 63].

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 $\text{Prob}(g(\mathbf{x}) \geq 0) \geq R$. In order to find left side of the above chance constraint, a separate optimization methodology [64], is needed, thereby making the overall algorithm a bi-level optimization procedure. Approximate single-loop algorithms exist [65] and recently one such methodology has been integrated with an EMO [61] 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.

10.4 Meta-model Assisted EMO

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 day or two. In such scenarios, even if an optimization algorithm needs one hundred solutions to get anywhere close to a good and feasible solution, the application needs an easy three to six 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.

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 [66, 67]. In the other approach, a successive meta-modeling approach is used in which the algorithm starts to solve the first meta-model obtained from a sample of the entire search space [68, 69, 70]. 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-modeling technique based on artificial neural networks is shown to reduce the computational effort by about 30 to 80% on different problems [68]. Other successful meta-modeling implementations for multi-objective optimization based on Kriging and response surface methodologies exist [70, 71].

11 Conclusions

This chapter has introduced the fast-growing field of multi-objective optimization based on evolutionary algorithms. First, the principles of single-objective evolutionary optimization (EO) techniques have been discussed so that readers can visualize the differences between evolutionary optimization and classical optimization methods. The EMO principle of handling multi-objective optimization problems is to find a representative set of Pareto-optimal solutions. Since an EO uses a population of solutions in each iteration, EO procedures are potentially viable techniques to capture a number of trade-off near-optimal solutions in a single simulation run. This chapter has described a number of popular EMO methodologies, presented some simulation studies on test problems, and discussed how EMO principles can be useful in solving real-world multi-objective optimization problems through a case study of spacecraft trajectory optimization.

Finally, this chapter has discussed the potential of EMO and its current research activities. The principle of EMO has been utilized to solve other optimization problems that are otherwise not multi-objective in nature. The diverse set of EMO solutions have been analyzed to find hidden common properties that can act as valuable knowledge to a user. EMO procedures have been extended to enable them to handle various

practicalities. Finally, the EMO task is now being suitably combined with decision-making activities in order to make the overall approach more useful in practice.

EMO addresses an important and inevitable fact of problem-solving tasks. EMO has enjoyed a steady rise of popularity in a short time. EMO methodologies are being extended to address practicalities. In the area of evolutionary computing and optimization, EMO research and application currently stands as one of the fastest growing fields. EMO methodologies are still to be applied to many areas of science and engineering. With such applications, the true value and importance of EMO will become evident.

Acknowledgments

This chapter contains some excerpts from previous publications by the same author entitled 'Introduction to Evolutionary Multi-Objective Optimization', In J. Branke, K. Deb, K. Miettinen and R. Slowinski (Eds.) *Multiobjective Optimization: Interactive and Evolutionary Approaches* (LNCS 5252), 2008, Berlin: Springer-Verlag., (pp. 59–96) and 'Recent Developments in Evolutionary Multi-Objective Optimization' in M. Ehrgott et al. (Eds.) *Trends in Multiple Criteria Decision Analysis*, 2010, Berlin: Springer-Verlag, (pp. 339-368). This paper will appear as a chapter in a Springer book entitled 'Multi-objective Evolutionary Optimisation for Product Design and Manufacturing' edited by Lihui Wang, Amos Ng, and Kalyanmoy Deb in 2011.

References

[1] Deb K. Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley; 2001.

[2] Goldberg DE. Genetic Algorithms for Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley; 1989.

[3] Deb K, Reddy AR, Singh G. Optimal scheduling of casting sequence using genetic algorithms. Journal of Materials and Manufacturing Processes. 2003;18(3):409–432.

[4] Deb K. An introduction to genetic algorithms. Sādhanā. 1999;24(4):293–315.

[5] Deb K, Agrawal RB. Simulated binary crossover for continuous search space. Complex Systems. 1995;9(2):115–148.

[6] Deb K, Anand A, Joshi D. A computationally efficient evolutionary algorithm for real-parameter optimization. Evolutionary Computation Journal. 2002;10(4):371–395.

[7] Storn R, Price K. Differential evolution – A fast and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 1997;11:341–359.

[8] Rudolph G. Convergence analysis of canonical genetic algorithms. IEEE Transactions on Neural Network. 1994;5(1):96–101.

[9] Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer-Verlag; 1992.

[10] Gen M, Cheng R. Genetic Algorithms and Engineering Design. New York: Wiley; 1997.

[11] Bäck T, Fogel D, Michalewicz Z, editors. Handbook of Evolutionary Computation. Bristol: Institute of Physics Publishing and New York: Oxford University Press; 1997.

[12] Deb K, Tiwari R, Dixit M, Dutta J. Finding trade-off solutions close to KKT points using evolutionary multi-objective optimization. In: Proceedings of the Congress on Evolutionary Computation (CEC-2007); 2007. p. 2109–2116.

[13] Holland JH. Adaptation in Natural and Artificial Systems. Ann Arbor, MI: MIT Press; 1975.

[14] Vose MD, Wright AH, Rowe JE. Implicit parallelism. In: Proceedings of GECCO 2003 (Lecture Notes in Computer Science. vol. 2723–2724. Springer-Verlag; 2003. .

[15] Jansen T, Wegener I. On the utility of populations. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001). San Mateo, CA: Morgan Kaufmann; 2001. p. 375–382.

[16] Radcliffe NJ. Forma analysis and random respectful recombination. In: Proceedings of the Fourth International Conference on Genetic Algorithms; 1991. p. 222–229.

[17] Miettinen K. Nonlinear Multiobjective Optimization. Boston: Kluwer; 1999.

[18] Kung HT, Luccio F, Preparata FP. On finding the maxima of a set of vectors. Journal of the Association for Computing Machinery. 1975;22(4):469–476.

[19] Ehrgott M. Multicriteria Optimization. Berlin: Springer; 2000.

[20] Deb K, Tiwari S. Omni-optimizer: A generic evolutionary algorithm for global optimization. European Journal of Operations Research (EJOR). 2008;185(3):1062–1087.

[21] Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and Elitist multi-objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002;6(2):182–197.

[22] Coello CAC, VanVeldhuizen DA, Lamont G. Evolutionary Algorithms for Solving Multi-Objective Problems. Boston, MA: Kluwer; 2002.

[23] Osyczka A. Evolutionary algorithms for single and multicriteria design optimization. Heidelberg: Physica-Verlag; 2002.

[24] Zitzler E, Deb K, Thiele L, Coello CAC, Corne DW. Proceedings of the First Evolutionary Multi-Criterion Optimization (EMO-01) Conference (Lecture Notes in Computer Science (LNCS) 1993). Heidelberg: Springer; 2001.

[25] Fonseca C, Fleming P, Zitzler E, Deb K, Thiele L. Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (Lecture Notes in Computer Science (LNCS) 2632). Heidelberg: Springer; 2003.

[26] Coello CAC, Aguirre AH, Zitzler E, editors. Evolutionary Multi-Criterion Optimization: Third International Conference. Berlin, Germany: Springer; 2005. LNCS 3410.

[27] Obayashi S, Deb K, Poloni C, Hiroyasu T, Murata T, editors. Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007, Proceedings. vol. 4403 of Lecture Notes in Computer Science. Springer; 2007.

[28] Coverstone-Carroll V, Hartmann JW, Mason WJ. Optimal multi-objective low-thurst spacecraft trajectories. Computer Methods in Applied Mechanics and Engineering. 2000;186(2–4):387–402.

[29] Srinivas N, Deb K. Multi-Objective function optimization using non-dominated sorting genetic algorithms. Evolutionary Computation Journal. 1994;2(3):221–248.

[30] Sauer CG. Optimization of multiple target electric propulsion trajectories. In: AIAA 11th Aerospace Science Meeting; 1973. Paper Number 73-205.

[31] Knowles JD, Corne DW. On metrics for comparing nondominated sets. In: Congress on Evolutionary Computation (CEC-2002). Piscataway, NJ: IEEE Press; 2002. p. 711–716.

[32] Hansen MP, Jaskiewicz A. Evaluating the quality of approximations to the non-dominated set. Lyngby: Institute of Mathematical Modelling, Technical University of Denmark; 1998. IMM-REP-1998-7.

[33] Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VG. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation. 2003;7(2):117–132.

[34] Fonseca CM, Fleming PJ. On the performance assessment and comparison of stochastic multiobjective optimizers. In: Voigt HM, Ebeling W, Rechenberg I, Schwefel HP, editors. Parallel Problem Solving from Nature (PPSN IV). Berlin: Springer; 1996. p. 584–593. Also available as Lecture Notes in Computer Science 1141.

[35] Fonseca CM, da Fonseca VG, Paquete L. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Third International Conference on Evolutionary Multi-Criterion Optimization, EMO-2005. Berlin: Springer; 2005. p. 250–264.

[36] Deb K, Sundar J, Uday N, Chaudhuri S. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. International Journal of Computational Intelligence Research (IJCIR). 2006;2(6):273–286.

[37] Deb K, Kumar A. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007). New York: The Association of Computing Machinery (ACM); 2007. p. 781–788.

[38] Deb K, Kumar A. Light Beam Search Based Multi-objective Optimization using Evolutionary Algorithms. In: Proceedings of the Congress on Evolutionary Computation (CEC-07); 2007. p. 2125–2132.

[39] Deb K, Sinha A, Kukkonen S. Multi-objective test problems, linkages and evolutionary methodologies. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006). New York: The Association of Computing Machinery (ACM); 2006. p. 1141–1148.

[40] Coello CAC. Treating objectives as constraints for single objective optimization. Engineering Optimization. 2000;32(3):275–308.

[41] Deb K, Datta R. A Fast and Accurate Solution of Constrained Optimization Problems Using a Hybrid Bi-Objective and Penalty Function Approach. In: Proceedings of the IEEE World Congress on Computational Intelligence (WCCI-2010); 2010. .

[42] Bleuler S, Brack M, Zitzler E. Multiobjective genetic programming: Reducing bloat using SPEA2. In: Proceedings of the 2001 Congress on Evolutionary Computation; 2001. p. 536–543.

[43] Handl J, Knowles JD. An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation. 2007;11(1):56–76.

[44] Knowles JD, Corne DW, Deb K. Multiobjective problem solving from nature. Springer Natural Computing Series, Springer-Verlag; 2008.

[45] Deb K, Srinivasan A. Innovization: Innovating design principles through optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006). New York: ACM; 2006. p. 1629–1636.

[46] Deb K, Sindhya K. Deciphering innovative principles for optimal electric brushless D.C. permanent magnet motor design. In: Proceedings of the World Congress on Computational Intelligence (WCCI-2008). Piscatway, NY: IEEE Press; 2008. p. 2283–2290.

[47] Bandaru S, Deb K. Towards automating the discovery of certain innovative design principles through a clustering based optimization technique. Engineering Optimization. in press;.

[48] Deb K, Goel T. A hybrid multi-objective evolutionary approach to engineering shape design. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01); 2001. p. 385–399.

[49] Sindhya K, Deb K, Miettinen K. A local search based evolutionary multi-objective optimization technique for fast and accurate convergence. In: Proceedings of the Parallel Problem Solving From Nature (PPSN-2008). Berlin, Germany: Springer-Verlag; 2008. .

[50] Khare V, Yao X, Deb K. Performance Scaling of Multi-objective Evolutionary Algorithms. In: Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632); 2003. p. 376–390.

[51] Luque M, Miettinen K, Eskelinen P, Ruiz F. Incorporating preference information in interactive reference point based methods for multiobjective optimization. Omega. 2009;37(2):450–462.

[52] Branke J, Deb K. Integrating user preferences into evolutionary multi-objective optimization. In: Jin Y, editor. Knowledge Incorporation in Evolutionary Computation. Heidelberg, Germany: Springer; 2004. p. 461–477.

[53] Deb K, Zope P, Jain A. Distributed Computing of Pareto-Optimal Solutions Using Multi-Objective Evolutionary Algorithms. In: Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632); 2003. p. 535–549.

[54] Deb K, Saxena D. 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); 2006. p. 3352–3360.

[55] Saxena DK, Deb K. Non-linear Dimensionality Reduction Procedures for Certain Large-Dimensional Multi-Objective Optimization Problems: Employing Correntropy and a Novel Maximum Variance Unfolding. In: Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2007); 2007. p. 772–787.

[56] Brockhoff D, Zitzler E. Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In: Waldmann KH, Stocker UM, editors. Operations Research Proceedings 2006. Springer; 2007. p. 423–429.

[57] Brockhoff D, Zitzler E. Offline and Online Objective Reduction in Evolutionary Multiobjective Optimization Based on Objective Conflicts. Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich; 2007. 269.

[58] Farina M, Amato P. A Fuzzy Definition of Optimality for Many Criteria Optimization Problems. IEEE Trans on Systems, Man and Cybernetics Part A: Systems and Humans. 2004;34(3):315–326.

[59] Branke J. Evolutionary Optimization in Dynamic Environments. Heidelberg, Germany: Springer; 2001.

[60] Deb K, Rao UB, Karthik S. 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 Evol. Multi-Criterion Optimization (EMO-2007); 2007. .

[61] Deb K, Gupta S, Daum D, Branke J, Mall A, Padmanabhan D. Reliability-based optimization using evolutionary algorithms. IEEE Trans on Evolutionary Computation. in press;.

[62] Deb K, Gupta H. Introducing robustness in multi-objective optimization. Evolutionary Computation Journal. 2006;14(4):463–494.

[63] Basseur M, Zitzler E. Handling Uncertainty in Indicator-Based Multiobjective Optimization. International Journal of Computational Intelligence Research. 2006;2(3):255–272.

[64] Cruse TR. Reliability-based mechanical design. New York: Marcel Dekker; 1997.

[65] Du X, Chen W. Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design. ASME Transactions on Journal of Mechanical Design. 2004;126(2):225–233.

[66] El-Beltagy MA, Nair PB, Keane AJ. Metamodelling techniques for evolutionary optimization of computationally expensive problems: promises and limitations. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999). San Mateo, CA: Morgan Kaufman; 1999. p. 196–203.

[67] Giannakoglou KC. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. Progress in Aerospace Science. 2002;38(1):43–76.

[68] Nain PKS, Deb K. Computationally effective search and optimization procedure using coarse to fine approximations. In: Proceedings of the Congress on Evolutionary Computation (CEC-2003); 2003. p. 2081–2088.

[69] Deb K, Nain PKS. In: An Evolutionary Multi-objective Adaptive Meta-modeling Procedure Using Artificial Neural Networks. Berlin, Germany: Springer; 2007. p. 297–322.

[70] Emmerich MTM, Giannakoglou KC, Naujoks B. Single and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. IEEE Transactions on Evolutionary Computation. 2006;10(4):421–439.

[71] Emmerich M, Naujoks B. Metamodel-Assisted Multiobjective Optimisation Strategies and Their Application in Airfoil Design. In: Adaptive Computing in Design and Manufacture VI. London, UK: Springer; 2004. p. 249–260.