Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems

Kalyanmoy Deb
Kanpur Genetic Algorithms Laboratory (KanGAL)
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
deb@iitk.ac.in

Abstract

In this paper, we study the problem features that may cause a multi-objective genetic algorithm (GA) difficulty in converging to the true Pareto-optimal front. Identification of such features helps us develop difficult test problems for multi-objective optimization. Multi-objective test problems are constructed from single-objective optimization problems, thereby allowing known difficult features of single-objective problems (such as multi-modality, isolation, or deception) to be directly transferred to the corresponding multi-objective problem. In addition, test problems having features specific to multi-objective optimization are also constructed. More importantly, these difficult test problems will enable researchers to test their algorithms for specific aspects of multi-objective optimization.

Keywords

Genetic algorithms, multi-objective optimization, niching, pareto-optimality, problem difficulties, test problems.

1 Introduction

After a decade since the pioneering work by Schaffer (1984), a number of studies on multi-objective genetic algorithms (GAs) have emerged. Most of these studies were motivated by a suggestion of a non-dominated GA outlined in Goldberg (1989). The primary reason for these studies is a unique feature of GAs—a population approach—that is highly suitable for use in multi-objective optimization. Since GAs work with a population of solutions, multiple Pareto-optimal solutions can be found in a GA population in a single simulation run. During the years 1993-95, a number of independent GA implementations (Fonseca and Fleming, 1993; Horn et al., 1994; Srinivas and Deb, 1995) emerged. Later, other researchers successfully used these implementations in various multi-objective optimization applications (Cunha et al., 1997; Eheart et al., 1993; Mitra et al., 1998; Parks and Miller, 1998; Weile et al., 1996). A number of studies have also concentrated on developing new GA implementations (Kursawe, 1990; Laumanns et al., 1998; Zitzler and Thiele, 1998). Fonseca and Fleming (1995) and Horn (1997) presented overviews of different multi-objective GA implementations, and Van Veldhuizen and Lamont (1998) made a survey of test problems that exist in the literature.

Despite these interests, there seems to be a lack of studies discussing problem features that may cause difficulty for multi-objective GAs. The literature also lacks a set of

©1999 by the Massachusetts Institute of Technology

test problems with known and controlled difficulty measure for systematically testing the performance of an optimization algorithm. Studies seeking problem features that cause difficulty for an algorithm may seem a pessimist's job, but we feel that the true efficiency of an algorithm is revealed when it is applied to challenging test problems, not easy ones. Such studies in single-objective GAs (studies on deceptive test problems, NK 'rugged' landscapes, and others) have all enabled researchers to better understand the working of GAs.

In this paper, we attempt to highlight a number of problem features that may cause a difficulty for a multi-objective GA . Keeping these properties in mind, we show procedures for constructing multi-objective test problems with controlled difficulty. Specifically, there exist some features shared by a multi-objective GA and a single-objective GA. Our construction of multi-objective problems from single-objective problems allow such difficulties to be directly transferred to an equivalent multi-objective GA. Some specific difficulties of multi-objective GAs are also discussed.

We also discuss and define local and global Pareto-optimal solutions. We show the construction of a simple two-variable, two-objective problem from single-variable, single-objective problems and show how multi-modal and deceptive multi-objective problems may cause difficulty for a multi-objective GA. We present a tunable two-objective problem of varying complexity constructed from three functionals. Specifically, a systematic construction of multi-objective problems having convex, non-convex, and discontinuous Pareto-optimal fronts is demonstrated. We then discuss the use of parameter-space versus function-space based niching and suggest which one to use when. Finally, future challenges in the area of multi-objective optimization are discussed.

2 Pareto-optimal Solutions

As the name suggests, Pareto-optimal solutions are optimal in some sense. Therefore, like single-objective optimization problems, there exist possibilities of having both *local* and *global* Pareto-optimal solutions. Before we define both these types of solutions, we discuss *dominated* and *non-dominated* solutions.

For a problem having more than one objective function (say, f_j, $j = 1, \dots, M$ and $M > 1$), a solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$ if both the following conditions are true (Steuer, 1986):

1. The solution $x^{(1)}$ is no worse (say the operator $\prec$ denotes worse and $\succ$ denotes better) than $x^{(2)}$ in all objectives, or $f_j(x^{(1)}) \not\prec f_j(x^{(2)})$ for all $j = 1, 2, \dots, M$ objectives.
2. The solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective, or $f_{\bar{j}}(x^{(1)}) \succ f_{\bar{j}}(x^{(2)})$ for at least one $\bar{j} \in \{1, 2, \dots, M\}$.

If any of the above conditions is violated, the solution $x^{(1)}$ does not dominate the solution $x^{(2)}$. If $x^{(1)}$ dominates the solution $x^{(2)}$, it is also customary to write $x^{(2)}$ is dominated by $x^{(1)}$, or $x^{(1)}$ is non-dominated by $x^{(2)}$.

The above concept can also be extended to find a non-dominated set of solutions in a population of solutions. Consider a set of N solutions, each having M (> 1) objective function values. The following procedure can be used to find the non-dominated set of solutions:

Step 0: Begin with $i = 1$.

Step 1: For all $j \neq i$, compare solutions $x^{(i)}$ and $x^{(j)}$ for domination using the above two conditions for all M objectives.

Step 2: If for any j, $x^{(i)}$ is dominated by $x^{(j)}$, mark $x^{(i)}$ as 'dominated'. Increment i by one and Go to Step 1.

Step 3: If all solutions (that is, when $i = N$ is reached) in the set are considered, Go to Step 4, else increment i by one and Go to Step 1.

Step 4: All solutions that are not marked 'dominated' are non-dominated solutions.

A population of solutions can be classified into groups of different non-domination levels (Goldberg, 1989). When the above procedure is applied for the first time in a population, the resulting set is the non-dominated set of first (or best) level. In order to have further classifications, these non-dominated solutions can be temporarily omitted from the original set and the above procedure can be applied again. What results is a set of non-dominated solutions of second (or next-best) level. This new set of non-dominated solutions can be omitted and the procedure applied again to find the third-level non-dominated solutions. This procedure can be continued until all population members are classified into a non-dominated level. It is important to realize that the number of non-domination levels in a set of N solutions is bound to lie within $[1, N]$. The minimum case of one non-domination level occurs when no solution dominates any other solution in the set, thereby classifying all solutions of the original population into one non-dominated level. The maximum case of N non-domination levels occurs when there is a hierarchy of domination of each solution and no two solutions are non-dominated by each other.

In a set of N arbitrary solutions, the first-level non-dominated solutions are candidates for possible Pareto-optimal solutions. The following definitions determine whether they are local or global Pareto-optimal solutions:

Local Pareto-optimal Set: If for every member x in a set $\underline{P}$ there exists no solution y satisfying $\|y - x\|_\infty \leq \epsilon$, where ϵ is a small positive number (in principle, y is obtained by perturbing x in a small neighborhood) dominating any member in the set $\underline{P}$, then the solutions belonging to the set $\underline{P}$ constitute a local Pareto-optimal set.

Global Pareto-optimal Set: If there exists no solution in the search space that dominates any member in the set $\bar{P}$, then the solutions belonging to the set $\bar{P}$ constitute a global Pareto-optimal set.

We describe the concept of local Pareto-optimal solutions in Figure 1, where both objectives f_1 and f_2 are minimized. By perturbing any solution in the local Pareto-optimal set (solutions marked by 'x') in a small neighborhood in the parameter space, it is not possible to obtain any solution that would dominate any member of the set.

The size and shape of Pareto-optimal fronts usually depend on the number of objective functions and interactions among the individual objective functions. If the objectives are 'conflicting' to each other, the resulting Pareto-optimal front may have a larger span than if the objectives are more 'cooperating'[1]. However, in most interesting multi-objective

[1]The terms 'conflicting' and 'cooperating' are used loosely here. If two objectives have similar individual optimum solutions and similar individual function values, they are 'cooperating', as opposed to a 'conflicting' situation where both objectives have drastically different individual optimum solutions and function values.

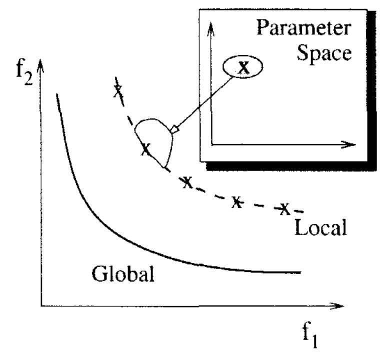


Figure 1: The illustrated concept of local and global Pareto-optimal sets.

optimization problems, the objectives are 'conflicting' to each other and usually the resulting Pareto-optimal front (local or global) contains many solutions.

3 Principles of Multi-objective Optimization

It is clear from the above discussion that a multi-objective optimization problem usually has a set of Pareto-optimal solutions, instead of one single optimal solution[2]. Thus, the objective in a multi-objective optimization is different from that in a single-objective optimization. In multi-objective optimization the goal is to find as many different Pareto-optimal (or near Pareto-optimal) solutions as possible. Since classical optimization methods work with a single solution in each iteration (Deb, 1995), in order to find multiple Pareto-optimal solutions they are required to be applied more than once, hopefully finding one distinct Pareto-optimal solution each time. Since GAs work with a population of solutions, a number of Pareto-optimal solutions can be captured in one single run of a multi-objective GA with appropriate adjustments to its operators. This aspect of GAs makes them naturally suited to solving multi-objective optimization problems for finding multiple Pareto-optimal solutions. Thus, it is no surprise that a number of different multi-objective GA implementations exist in the literature (Fonseca and Fleming, 1995; Horn et al., 1994; Srinivas and Deb, 1995; Zitzler and Thiele, 1998).

Before we discuss the problem features that may cause multi-objective GAs difficulty, let us mention a couple of matters[3] that are not addressed in the paper. First, we consider all objectives to be of minimization type. It is worth mentioning that identical properties as discussed here may also exist in problems with mixed optimization types (some are minimization and some are maximization). The concept of non-domination among solutions addresses only one type of problem. The meaning of 'worse' or 'better', discussed in Section 2, takes care of other cases. Second, although we refer to multi-objective optimization throughout the paper, we restrict ourselves to two objectives. This is because we believe that the two-objective optimization brings out the essential features of multi-objective optimization.

There are two tasks that a multi-objective GA should accomplish in solving multi-

[2] In multi-modal function optimization, there may exist more than one optimal solution, but usually the interest is to find global optimal solutions having identical objective function value.

[3] A number of other matters which need immediate attention are also outlined in Section 7.

objective optimization problems:

1. Guide the search towards the global Pareto-optimal region, and
2. Maintain population diversity (in the function space, parameter space, or both) in the current non-dominated front.

We discuss the above two tasks in the following subsections and highlight when a GA would have difficulty in achieving each task.

3.1 Difficulties in Converging to Pareto-optimal Front

Convergence to the true (or global) Pareto-optimal front may not occur because of various features that may be present in a problem:

1. Multi-modality,
2. Deception,
3. Isolated optimum, and
4. Collateral noise.

All the above features are known to cause difficulty in single-objective GAs (Deb et al., 1993) and, when present in a multi-objective problem, may also cause difficulty for a multi-objective GA.

In tackling a multi-objective problem having multiple Pareto-optimal fronts, a GA, like many other search and optimization methods, may converge to a local Pareto-optimal front. Later, we create a multi-modal multi-objective problem and show that a multi-objective GA can get stuck at a local Pareto-optimal front if appropriate GA parameters are not used.

Despite some criticism (Grefenstette, 1993), deception, if present in a problem, has been shown to cause GAs to be misled towards deceptive attractors (Goldberg et al., 1989). There is a difference between the difficulties caused by multi-modality and by deception. For deception to take place, it is necessary to have at least two optima in the search space (a true attractor and a deceptive attractor), but almost the entire search space *favors* the deceptive (non-global) optimum. Multi-modality may cause difficulty for a GA merely because of the sheer number of different optima where a GA can stick. We shall show how the concept of single-objective deceptive functions can be used to create multi-objective deceptive problems, which may cause difficulty for a multi-objective GA.

There may exist some problems where the optimum is surrounded by a fairly flat search space. Since there is no useful information provided by most of the search space, no optimization algorithm will perform better than an exhaustive search method to find the optimum in these problems. Multi-objective optimization methods also face difficulty in solving such a problem.

Collateral noise comes from the improper evaluation of low-order building blocks (partial solutions which may lead towards the true optimum) due to the excessive noise coming from other parts of the solution vector. These problems are usually 'rugged' with relatively large variation in the function landscape. Multi-objective problems having such 'rugged' functions may also cause difficulties for multi-objective GAs if adequate population size (adequate to discover signal from the noise) is not used.

3.2 Difficulties in Maintaining Diverse Pareto-optimal Solutions

As it is important for a multi-objective GA to find solutions near or on the true Pareto-optimal front, it is also necessary to find solutions as diverse as possible in the Pareto-optimal front. If most solutions found are confined in a small region near or on the true Pareto-optimal front, the purpose of multi-objective optimization is not served. This is because, in such cases, many interesting solutions with large trade-offs among the objectives and parameter values may have been undiscovered.

In most multi-objective GA implementations, a specific diversity-maintaining operator, such as a niching technique (Deb and Goldberg, 1989) or a clustering technique (Zitzler and Thiele, 1998) is used to find diverse Pareto-optimal solutions. However, the following features might be likely to cause a multi-objective GA difficulty in maintaining diverse Pareto-optimal solutions:

1. Convexity or non-convexity in the Pareto-optimal front,
2. Discontinuity in the Pareto-optimal front, and
3. Non-uniform distribution of solutions in the Pareto-optimal front.

There exist multi-objective problems where the resulting Pareto-optimal front is non-convex. Although it may not be apparent, a GA's success in maintaining diverse Pareto-optimal solutions largely depends on the fitness assignment procedure. In some GA implementations, the fitness of a solution is assigned proportionally to the number of solutions it dominates (Fonseca and Fleming, 1993; Zitzler and Thiele, 1998). Figure 2 shows how such a fitness assignment favors intermediate solutions, in the case of problems with convex Pareto-optimal front (the left figure). With respect to an individual champion[4] solution

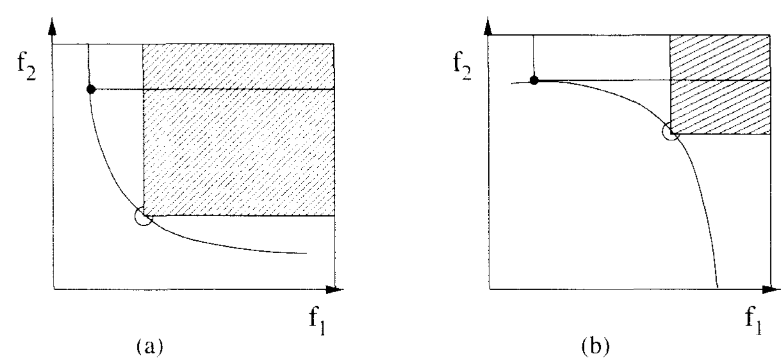


Figure 2: The fitness assignment proportional to the number of dominated solutions (the shaded area) favors intermediate solutions in convex Pareto-optimal front (a), compared to that in non-convex Pareto-optimal front (b).

(marked with a solid dot in the figures), the proportion of dominated region covered by an intermediate solution is more in Figure 2(a) than in Figure 2(b). Using such a GA (with GAs favoring solutions having more dominated solutions), there is a natural tendency to

[4]Optimum solution corresponding to an individual objective function.

find more intermediate solutions than solutions near individual champions, thereby causing an artificial bias towards some portion of the Pareto-optimal region.

In some multi-objective optimization problems, the Pareto-optimal front may not be continuous, instead it may be a collection of discretely spaced continuous sub-regions (Poloni et al., in press; Schaffer, 1984). In such problems, although solutions within each sub-region may be found, competition among these solutions may lead to extinction of some sub-regions.

It is also likely that the Pareto-optimal front is not uniformly represented by feasible solutions. Some regions in the front may be represented by a higher *density*[5] of solutions than other regions. In such cases, there may be a natural tendency for GAs to find a biased distribution in the Pareto-optimal region.

3.3 Constraints

In addition to the above, the presence of 'hard' constraints in a multi-objective problem may cause further difficulties. Constraints may hinder GAs from converging to the true Pareto-optimal region and they may also cause difficulty in maintaining a diverse set of Pareto-optimal solutions. It is intuitive that the success of a multi-objective GA in tackling both these problems will largely depend on the constraint-handling technique used. Traditionally, a simple penalty-function based method has been used to penalize each objective function (Deb and Kumar, 1995; Srinivas and Deb, 1995; Weile et al., 1996). Although successful applications are reported, penalty function methods demand an appropriate choice of a penalty parameter for each constraint. Recent suggestions of penalty parameter-less techniques (Deb, in press; Koziel and Michalewicz, 1998) may be worth investigating in the context of multi-objective constrained optimization.

4 A Special Two-Objective Optimization Problem

Let us begin our discussion with a simple two-objective optimization problem having two variables x_1 (> 0) and x_2:

$$\text{Minimize} \quad f_1(x_1, x_2) = x_1, \tag{1}$$

$$\text{Minimize} \quad f_2(x_1, x_2) = \frac{g(x_2)}{x_1}, \tag{2}$$

where $g(x_2)$ (> 0) is a function of x_2 only. Thus, the first objective function f_1 is a function of x_1 only[6] and the function f_2 is a function of both x_1 and x_2. In the function space (a space with (f_1, f_2) values), the above two functions obey the following relationship:

$$f_1(x_1, x_2) \cdot f_2(x_1, x_2) = g(x_2) \tag{3}$$

For a fixed value of $g(x_2) = c$, a f_1-f_2 plot becomes a hyperbola ($f_1 f_2 = c$). There exists a number of intuitive yet interesting properties of the above two-objective problem:

LEMMA 1: *If for any two solutions, the second variables x_2 (or more specifically $g(x_2)$) are the same, both solutions are not dominated by each other.*

[5]Density can be measured as the hyper-volume of a sub-region in the parameter space representing a unit hypercube in the fitness space.

[6]With this function, it is necessary that f_1 and g function values be strictly positive.

The proof follows from $f_1 f_2 = c$ property.

LEMMA 2: *If for any two solutions, the first variables x_1 are the same, the solution corresponding to the minimum $g(x_2)$ dominates the other solution.*

PROOF: Since $x_1^{(1)} = x_1^{(2)}$, the first objective function values are the same. So, the solution having smaller $g(x_2)$ (meaning better f_2) dominates the other solution.

LEMMA 3: *For any two arbitrary solutions $x^{(1)}$ and $x^{(2)}$, where $x_i^{(1)} \neq x_i^{(2)}$ for $i = 1, 2$, and $g(x_2^{(1)}) < g(x_2^{(2)})$, there exists a solution $x^{(3)} = (x_1^{(2)}, x_2^{(1)})$ which dominates the solution $x^{(2)}$.*

PROOF: Since the solutions $x^{(3)}$ and $x^{(2)}$ have the same x_1 value and since $g(x^{(1)}) < g(x^{(2)})$, $x^{(3)}$ dominates $x^{(2)}$, according to Lemma 2.

COROLLARY 1: *The solutions $x^{(1)}$ and $x^{(3)}$ have the same x_2 values and hence they are non-dominated to each other according to Lemma 1.*

Based on the above discussions, we can present the following theorem:

THEOREM 1: *The two-objective problem described in equations (1) and (2) has local or global Pareto-optimal solutions (x_1, x_2^*), where x_2^* is the locally or globally minimum solution of $g(x_2)$, respectively, and x_1 can take any value.*

PROOF: Since solutions with a minimum $g(x_2)$ have the smallest possible $g(x_2)$ (in the neighborhood sense, in the case of local minimum, and in the whole search space in the case of global minimum), according to Lemma 2, all such solutions dominate any other solution in the respective context. Since these solutions are also non-dominated to each other, they are Pareto-optimal solutions, in the respective sense.

Although obvious, we shall present a final lemma about the relationship between a non-dominated set of solutions and Pareto-optimal solutions.

LEMMA 4: *Although some members in a non-dominated set are members of the Pareto-optimal front, not all members are necessarily members of the Pareto-optimal front.*

PROOF: Say, there are only two distinct members in a set of which $x^{(1)}$ is a member of Pareto-optimal front and $x^{(2)}$ is not. We shall show that both these solutions still can be non-dominated to each other. The solution $x^{(2)}$ can be chosen in such a way that $x_1^{(2)} < x_1^{(1)}$. This makes $f_1(x^{(2)}) < f_1(x^{(1)})$. Since $g(x_2^{(2)}) > g(x_2^{(1)})$, it follows that $f_2(x^{(2)}) > f_2(x^{(1)})$. Thus, $x^{(1)}$ and $x^{(2)}$ are non-dominated solutions.

This lemma establishes a negative argument about multi-objective optimization methods which work with the concept of non-domination. Since these methods seek to find the Pareto-optimal front by finding the *best* non-dominated set of solutions, it is important to realize that all solutions in the best non-dominated set obtained by an optimizer may not necessarily be the members of the Pareto-optimal set. However, in the absence of any better approach, a method for seeking the best set of non-dominated solutions is a reasonable approach. Post-optimal testing (by locally perturbing each member of obtained non-dominated set) may be performed to establish Pareto-optimality of members in an experimentally obtained non-dominated set.

The above two-objective problem and the associated lemmas allow us to construct different types of multi-objective problems from single-objective optimization problems (defined by the function g). The optimality and complexity of function g is then directly transferred into the corresponding multi-objective problem. In the following subsections, we construct a multi-modal and a deceptive multi-objective problem.

4.1 Multi-modal Multi-objective Problem

According to Theorem 1, if the function $g(x_2)$ is multi-modal with local ($\underline{x}_2$) and global ($\bar{x}_2$) minimum solutions, the corresponding two-objective problem also has local and global Pareto-optimal solutions corresponding to solutions $(x_1, \underline{x}_2)$ and $(x_1, \bar{x}_2)$, respectively. The Pareto-optimal solutions vary in x_1 values.

We create a bimodal, two-objective optimization problem by choosing a bimodal $g(x_2)$ function:

$$g(x_2) = 2.0 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\} \qquad (4)$$

Figure 3 shows the above function for $0 \leq x_2 \leq 1$ with $x_2 \approx 0.2$ as the global minimum and $x_2 \approx 0.6$ as the local minimum solutions. Figure 4 shows the f_1-f_2 plot with local and global Pareto-optimal solutions corresponding to the two-objective optimization problem.

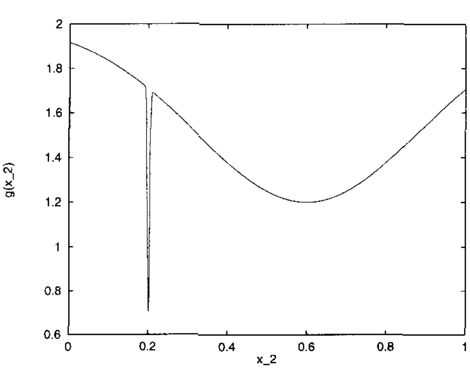


Figure 3: The function $g(x_2)$ has a global and a local minimum solution.

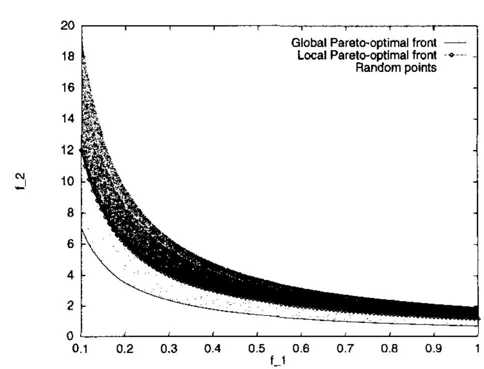


Figure 4: A random set of 50,000 solutions shown on a f_1-f_2 plot.

The local Pareto-optimal solutions occur at $x_2 \approx 0.6$ and the global Pareto-optimal solutions occur at $x_2 \approx 0.2$. The corresponding values for g function values are $g(0.6) = 1.2$ and $g(0.2) = 0.7057$, respectively. The density of the random solutions marked on the plot shows that most solutions lead towards the local Pareto-optimal front and only a few solutions lead towards the global Pareto-optimal front.

To investigate how a multi-objective GA would perform in this problem, the non-dominated sorting GA (NSGA) (Srinivas and Deb, 1995) is used. Variables are coded in 20-bit binary strings each, in the ranges $0.1 \leq x_1 \leq 1.0$ and $0 \leq x_2 \leq 1.0$. A population of size 60 is used[7]. Single-point crossover with $p_c = 1$ is chosen. No mutation is used. The niching parameter $\sigma_{\text{share}} = 0.158$ is calculated based on normalized parameter values and assumed to form about 10 niches in the Pareto-optimal front (Deb and Goldberg, 1989). Figure 5 shows a run of NSGA which, even at generation 100, gets trapped at the local Pareto-optimal solutions (marked with a '+'). When NSGA is tried with 100 different

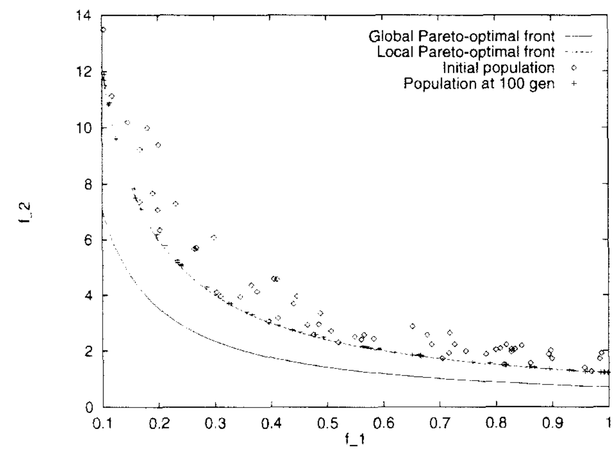


Figure 5: A NSGA run gets trapped at the local Pareto-optimal solution.

initial populations, it gets trapped into the local Pareto-optimal front in 59 out of 100 runs, whereas in the other 41 runs NSGA can find the global Pareto-optimal front. We also observe that in 25 runs there exists at least one solution in the global basin of function g in the initial population and still NSGAs cannot converge to the global Pareto-optimal front. Instead, they get attracted to the local Pareto-optimal front. These results show that a multi-objective GA can have difficulty even with a simple bimodal problem. A more difficult test problem can be constructed by using a standard single-objective multi-modal test problem, such as Rastrigin's function, Schwefel's function, or by using a higher-dimensional, multi-modal g function.

4.2 Deceptive Multi-objective Optimization Problem

Next, we shall create a deceptive multi-objective optimization problem from a deceptive g function. This function is defined over binary alphabets. Let us say that the following multi-objective function is defined over ℓ bits, which is a concatenation of N substrings of variable size ℓ_i such that $\sum_{i=1}^{N} \ell_i = \ell$:

$$\begin{aligned} \text{Minimize} \quad f_1 &= 1 + u(\ell_1), \\ \text{Minimize} \quad f_2 &= \frac{\sum_{i=2}^{N} g(u(\ell_i))}{1+u(\ell_1)}, \end{aligned} \tag{5}$$

[7]This population size is determined to have, on an average, one solution in the global basin of function g in a random initial population.

where $u(\ell_1)$ is the unitation[8] of the first substring of length ℓ_1. To keep matters simple, we have used a tight encoding of bits representing each substring. The first function f_1 is a simple one-min problem, where the optimal solution has all 0s. A one is added to make all function values strictly positive. The function g is defined[9] in the following:

$$g(u(\ell_i)) = \begin{cases} 2 + u(\ell_i), & \text{if } u(\ell_i) < \ell_i, \\ 1, & \text{if } u(\ell_i) = \ell_i \end{cases} \tag{6}$$

This makes the true attractor (with all 1s in the substring) have the worst neighbors with a function value $g(\ell_i) = 1$ and the deceptive attractor (with all 0s in the substring) have the good neighbors with a function value $g(0) = 2$. Since most of the substrings lead toward the deceptive attractor, GAs may find difficulty converging to the true attractor (all 1s).

The global Pareto-optimal front corresponds to the solution for which the summation of g function values is absolutely minimum. Since at each minimum, g has a value one, the global Pareto-optimal solutions have a summation of g equal to $(N-1)$. Since each g function has two minima (one true and another deceptive), there are a total of 2^{N-1} local minima, of which one is global. Corresponding to each of these local minima, there exists a local Pareto-optimal front (some of them are identical since the functions are defined over unitation), to which a multi-objective GA may be attracted.

In the experimental set up, we used $\ell_1 = 10$, $\ell_2 = 5$, $\ell_3 = 5$, $\ell_4 = 5$, such that $\ell = 25$. Since the functions are defined with unitation values, we have used genotypic niching with Hamming distance as the distance measure between two solutions (Deb and Goldberg, 1989). Since we expect 11 different function values in f_1 (all integers from 1 to 11), we use guidelines suggested in that study and calculate $\sigma_{\text{share}} = 9$. Figure 6 shows that when a population size of 80 is used, an NSGA is able to find the global Pareto-optimal front from the initial population shown (solutions marked with a '+').

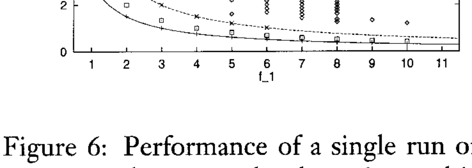


Figure 6: Performance of a single run of NSGA is shown on the deceptive multi-objective function.

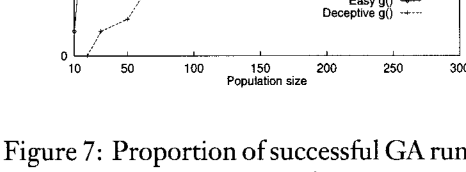


Figure 7: Proportion of successful GA runs (out of 50 runs) versus population size with easy and deceptive multi-objective problems.

When a smaller population size ($n = 60$) is used, the NSGA cannot find the true substring in all three deceptive subproblems. Instead, it converges to the deceptive substring

[8]Unitation is the number of 1s in the substring. Note that minimum and maximum values of unitation of a substring of length ℓ_i is zero and ℓ_i, respectively.

[9]It can be shown that an equivalent dual maximization function $G = \ell_i + 1 - g(u(\ell_i))$ is deceptive according to conditions outlined elsewhere (Deb and Goldberg, 1994). Thus, the above minimization problem is also deceptive.

in one subproblem and to the true substring in the two other subproblems. When a sufficiently small population ($n = 16$) is used, the NSGA converges to the deceptive attractor in all three subproblems. The corresponding local Pareto-optimal front is shown in Figure 6 with a dashed line.

In order to further investigate the difficulties that a deceptive multi-objective function may cause to a multi-objective GA, we construct a 30-bit function with $\ell_1 = 10$ and $\ell_i = 5$ for $i = 2, \ldots, 5$ and use $\sigma_{\text{share}} = 11$. For each population size, 50 GA runs are started from different initial populations and the proportion of *successful* runs is plotted in Figure 7. A run is considered successful if all four deceptive subproblems are solved correctly. The figure shows that NSGAs with small population sizes could not be successful in many runs. Moreover, the performance improves as the population size is increased. To show that this difficulty is due to deception in subproblems alone, we use a linear function for $g(u) = u + 1$, instead of the deceptive function used earlier. Figure 7 shows that multi-objective GAs with a reasonable population size worked more frequently with this easy problem than with the deceptive problem.

The above two problems show that by using a simple construction methodology (by choosing a suitable g function), any problem feature that may cause single-objective GAs difficulty can also be introduced in a multi-objective GA. Based on the above construction methodology, we now present a tunable two-objective optimization problem which may have additional difficulties pertaining to multi-objective optimization.

5 Tunable Two-Objective Optimization Problems

Let us consider the following N-variable two-objective problem:

$$\begin{array}{ll} \text{Minimize} & f_1(\vec{x}) = f_1(x_1, x_2, \ldots, x_m), \\ \text{Minimize} & f_2(\vec{x}) = g(x_{m+1}, \ldots, x_N) \times h(f_1, g) \end{array} \tag{7}$$

The function f_1 is a function of m ($< N$) variables ($\vec{x}_I = (x_1, \ldots, x_m)$), and the function f_2 is a function of all N variables. The function g is a function of $(N - m)$ variables ($\vec{x}_{II} = (x_{m+1}, \ldots, x_N)$) which do not appear in the function f_1. The function h is a function of f_1 and g function values directly. We avoid complications by choosing f_1 and g functions that take only positive values (or $f_1 > 0$ and $g > 0$) in the search space. By choosing appropriate functions for f_1, g, and h, multi-objective problems having specific features can be created:

1. Convexity or discontinuity in the Pareto-optimal front can be affected by choosing an appropriate h function.
2. Convergence to the true Pareto-optimal front can be affected by using a difficult g function (multi-modal, deceptive, or others) as demonstrated in the previous section.
3. Diversity in the Pareto-optimal front can be affected by choosing an appropriate (non-linear or multi-dimensional) f_1 function.

We describe each of the above issues in the following subsections.

5.1 Convexity or Discontinuity in Pareto-optimal Front

By choosing an appropriate h function, multi-objective optimization problems with convex, non-convex, or discontinuous Pareto-optimal fronts can be created. Specifically, if the following two properties of h are satisfied, the global Pareto-optimal set will correspond to the global minimum of the function g and to all values of the function f_1[10]:

1. The function h is a monotonically non-decreasing function in g for a fixed value of f_1.
2. The function h is a monotonically decreasing function of f_1 for a fixed value of g.

The first condition ensures that the global Pareto-optimal front occurs for the global minimum value for g function. The second condition ensures that there is a continuous 'conflicting' Pareto-front. However, we realize that when we violate the second condition, we shall no longer create problems having continuous Pareto-optimal front. However, if the first condition is met alone, for every local minimum of g there will exist one local Pareto-optimal set (corresponding value of g and all possible values of f_1).

Although many different functions may exist, we present two such functions—one leading to a convex Pareto-optimal front and the other leading to a more generic problem having a control parameter which decides the convexity or non-convexity of the Pareto-optimal fronts.

5.1.1 Convex Pareto-optimal Front

For the following function

$$h(f_1, g) = \frac{1}{f_1}, \tag{8}$$

we only allow $f_1 > 0$. The resulting Pareto-optimal set is $(\vec{x}_I^*, \vec{x}_{II}) = \{(\vec{x}_I, \vec{x}_{II}) : \nabla g(\vec{x}_I) = 0\}$. In Section 4, we have seen that the resulting Pareto-optimal set is convex. In the following, we present another function which can be used to create convex and non-convex Pareto-optimal sets by simply tuning a parameter.

5.1.2 Non-convex Pareto-optimal Front

We choose the following function for h:

$$h(f_1, g) = \begin{cases} 1 - \left(\frac{f_1}{\beta g}\right)^{\alpha}, & \text{if } f_1 \leq \beta g, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

With this function, we may allow $f_1 \geq 0$, but $g > 0$. The global Pareto-optimal set corresponds to the global minimum of g function. The parameter β is a normalization factor to adjust the range of values of functions f_1 and g. To have a significant Pareto-optimal region, β may be chosen as $\beta \geq f_{1,\max}/g_{\min}$, where $f_{1,\max}$ and $g_{\min}$ are the maximum value of the function f_1 and the minimum (or global optimal) value of the function g, respectively. It is interesting to note that when $\alpha > 1$, the resulting Pareto-optimal front is non-convex. In tackling these problems, the classical weighted-sum method cannot find any intermediate Pareto-optimal solution by using a weight vector. The above function can also be used to create multi-objective problems having convex Pareto-optimal sets by setting $\alpha \leq 1$. Other interesting functions for the function h may also be chosen with properties mentioned in Section 5.1.

[10]Although the condition for Pareto-optimality of multi-objective problems can be established for other h functions, here, we state the sufficient conditions for the functional relationships of h with g and f_1. Note that this allows us to directly relate the optimality of g function with the Pareto-optimality of the resulting multi-objective problem.

Test problems having local and global Pareto-optimal fronts being of mixed type (some are convex and some are non-convex shape) can also be created by making the parameter α a function of g. These problems may cause difficulty to algorithms that work by exploiting the shape of the Pareto-optimal front simply because the search algorithm needs to adapt while moving from a local to global Pareto-optimal front. Here, we illustrate one such problem, where the local Pareto-optimal front is non-convex, and the global Pareto-optimal front is convex. Consider the following functions ($x_1, x_2 \in [0,1]$) along with function h defined in Equation 9:

$$g(x_2) = \begin{cases} 4 - 3\exp\left(-\frac{x_2-0.2}{0.02}\right)^2, & \text{if } 0 \le x_2 \le 0.4, \\ 4 - 2\exp\left(-\frac{x_2-0.7}{0.2}\right)^2, & \text{if } 0.4 < x_2 \le 1, \end{cases} \tag{10}$$

$$f_1(x_1) = 4x_1, \tag{11}$$

$$\alpha = 0.25 + 3.75\frac{g(x_2) - g^{**}}{g^* - g^{**}}, \tag{12}$$

where g^* and g^{**} are the local and the global optimal function value of g, respectively. Equation 12 is set to have a non-convex local Pareto-optimal front at $\alpha = 4.0$ and a convex global Pareto-optimal front at $\alpha = 0.25$. The function h is given in Equation 9 with $\beta = 1$. A random set of 40,000 solutions ($x_1, x_2 \in [0.0, 1.0]$) is generated and the corresponding

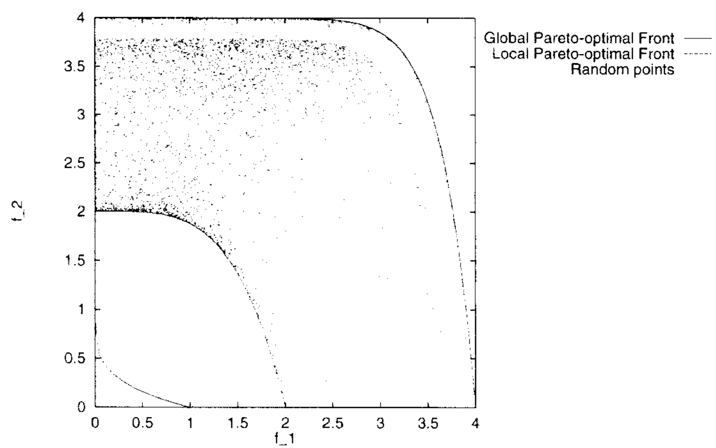


Figure 8: A two-objective function with a non-convex local Pareto-optimal front and a convex global Pareto-optimal front. 40,000 random solutions are shown.

solutions in the f_1-f_2 space are shown in Figure 8. The figure clearly shows the nature of the convex global and non-convex local Pareto-optimal fronts (solid and dashed lines, respectively). Notice that only a small portion of the search space leads to the global Pareto-optimal front. An apparent front at the top of the figure is due to the discontinuity in the $g(x_2)$ function at $x_2 = 0.4$.

Another simple way to create a non-convex Pareto-optimal front is to use Equation 8 but maximize both functions f_1 and f_2. The Pareto-optimal front corresponds to the maximum value of g function and the resulting Pareto-optimal front is non-convex.

5.1.3 Discontinuous Pareto-optimal Front

As mentioned earlier, we have to relax the condition for h being a monotonically decreasing function of f_1 to construct multi-objective problems with a discontinuous Pareto-optimal front. In the following, we show one such construction where the function h is a periodic function of f_1:

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{\alpha} - \frac{f_1}{g}\sin(2\pi q f_1) \tag{13}$$

The parameter q is the number of discontinuous regions in a unit interval of f_1. By choosing the following functions:

$$f_1(x_1) = x_1, \quad g(x_2) = 1 + 10x_2,$$

and allowing variables x_1 and x_2 to lie in the interval [0,1], we have a two-objective optimization problem which has a discontinuous Pareto-optimal front. Since the h (and hence f_2) function is periodic to x_1 (and hence to f_1), we generate discontinuous Pareto-optimal regions.

Figure 9 shows the 50,000 random solutions in f_1-f_2 space. Here, we use $q = 4$ and $\alpha = 2$. When NSGAs (population size of 200, σ_{share} of 0.1, crossover probability of 1, and no mutation) are applied to this problem, the resulting population at generation 300 is shown in Figure 10. The plot shows that if reasonable GA parameter values are

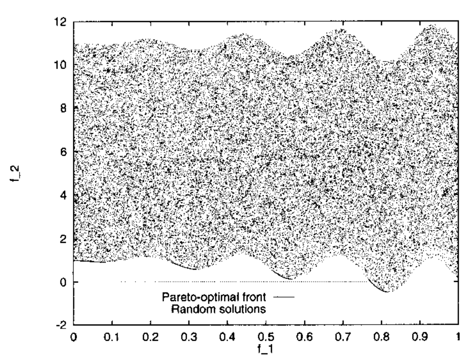


Figure 9: 50,000 random solutions are shown on a f_1-f_2 plot of a multi-objective problem having discrete Pareto-optimal front.

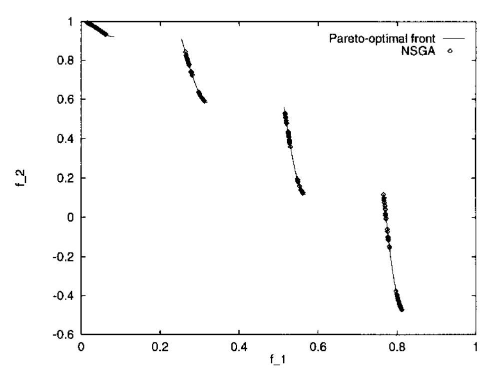


Figure 10: The population at generation 300 for a NSGA run is shown to have found solutions in all four discontinuous Pareto-optimal regions.

chosen, NSGAs can find solutions in all four discontinuous Pareto-optimal regions. In general, discontinuity in the Pareto-optimal front may cause difficulty to multi-objective GAs which do not have an efficient way of implementing diversity among discontinuous regions. Function-space niching may have difficulty in these problems because of the discontinuities in the Pareto-optimal front.

5.2 Hindrance to Reach True Pareto-optimal Front

It is shown earlier that by choosing a difficult function for g alone, a difficult multi-objective optimization problem can be created. Some instances of multi-modal and deceptive multi-objective optimization have been created earlier. Test problems with standard multi-modal functions used in single-objective GA studies, such as Rastrigin's functions, NK landscapes, and others can all be chosen for the g function.

5.2.1 Biased Search Space

The function g plays a major role in introducing difficulty to a multi-objective problem. Even though the function g is not chosen to be a multi-modal function nor to be a deceptive function, with a simple monotonic g function the search space can have adverse density of solutions toward the Pareto-optimal region. Consider the following function for g:

$$g(x_{m+1},\ldots,x_N) = g_{\min} + (g_{\max} - g_{\min}) \left(\frac{\sum_{i=m+1}^{N} x_i - \sum_{i=m+1}^{N} x_i^{\min}}{\sum_{i=m+1}^{N} x_i^{\max} - \sum_{i=m+1}^{N} x_i^{\min}} \right)^{\gamma}, \quad (14)$$

where $g_{\min}$ and $g_{\max}$ are the minimum and maximum function values that the function g can take. The values $x_i^{\min}$ and $x_i^{\max}$ are minimum and maximum values of the variable x_i. It is important to note that the Pareto-optimal region occurs when g takes the value $g_{\min}$. The parameter γ controls the bias in the search space. If $\gamma < 1$, the density of solutions away from the Pareto-optimal front is large. We show this on a simple problem with $m = 1$, $N = 2$, and with the following functions:

$$f_1(x_1) = x_1, \quad h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2$$

We also use $g_{\min} = 1$ and $g_{\max} = 2$. Figures 11 and 12 show 50,000 random solutions each with γ equal to 1.0 and 0.25, respectively. It is clear that for $\gamma = 0.25$, no solution

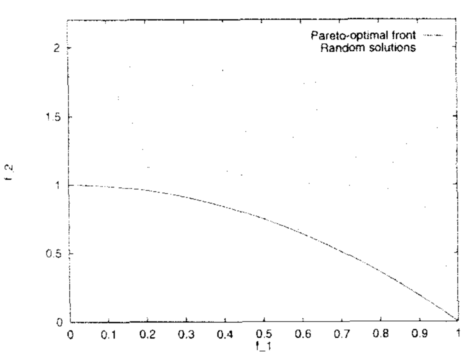


Figure 11: 50,000 random solutions are shown for $\gamma = 1.0$.

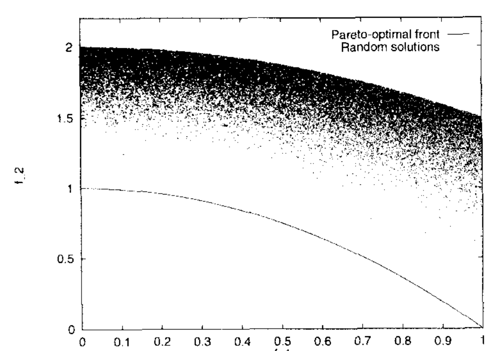


Figure 12: 50,000 random solutions are shown for $\gamma = 0.25$.

is found in the Pareto-optimal front, whereas for $\gamma = 1.0$, many Pareto-optimal solutions exist in the set of 50,000 random solutions. Random-like search methods are likely to face difficulty in finding the Pareto-optimal front in the case with γ close to zero, mainly due to the low density of solutions towards the Pareto-optimal region.

5.2.2 Parameter Interactions

The difficulty in converging to the true Pareto-optimal front may also arise because of parameter interactions. It was discussed before that the Pareto-optimal set in the two-objective optimization problem described in Equation 7 corresponds to all solutions of different f_1 values. Since the purpose of a multi-objective GA is to find as many Pareto-optimal solutions as possible and, since in Equation 7 the variables defining f_1 are different from variables defining g, a GA may work in two stages. In one stage, all variables $\vec{x}_I$ may be found and in the other, optimal $\vec{x}_{II}$ may be found. This rather simple mode of a GA working in two stages can face difficulty if the above variables are mapped to another set of variables. If M is a random orthonormal matrix of size $N \times N$, the true variables $\vec{y}$ can first be mapped to derived variables $\vec{x}$ using

$$\vec{x} = M\vec{y} \tag{15}$$

Thereafter, objective functions defined in Equation 7 can be computed using the variable vector $\vec{x}$. Since the components of $\vec{x}$ can now be negative, care must be taken in defining f_1 and g functions so as to satisfy restrictions suggested on them in previous subsections. A translation of these functions by adding a suitable large positive value may have to be used to force these functions to take non-negative values. Since the GA will be operating on the variable vector $\vec{y}$, and the function values depend on the interaction among variables of $\vec{y}$, any change in one variable must be accompanied by related changes in other variables in order to remain on the Pareto-optimal front. This makes this mapped version of the problem difficult to solve. We discuss more about mapped functions near the end of the following section.

5.3 Non-uniformly Represented Pareto-optimal Front

In all the test functions constructed above (except the deceptive problem), we have used a linear, single-variable function for f_1. This helped us create a problem with a uniform distribution of solutions in f_1. Unless the underlying problem has discretely spaced Pareto-optimal regions (as in Section 5.1.3), there is no bias for the Pareto-optimal solutions to be spread over the entire range of f_1 values. However, a bias for some portions of range of values for f_1 may also be created by choosing any of the following f_1 functions:

1. The function f_1 is non-linear, or
2. The function f_1 is a function of more than one variable.

It is clear that if a non-linear f_1 function (whether single or multi-variable) is chosen, the resulting Pareto-optimal region (or, for that matter, the entire search region) will have bias towards some values of f_1. The non-uniformity in distribution of the Pareto-optimal region can also be created by simply choosing a multi-variable function (whether linear or non-linear). Multi-objective optimization algorithms, which are poor at maintaining diversity among solutions (or function values), will produce a biased Pareto-optimal front in such problems. Thus, the non-linearity in function f_1 or dimension of f_1 measures how well an algorithm is able to maintain distributed non-dominated solutions in a population. Consider the single-variable, multi-modal function f_1:

$$f_1(x_1) = 1 - \exp(-4x_1)\sin^4(5\pi x_1), \quad 0 \le x_1 \le 1 \tag{16}$$

The above function has five minima for different values of x_1 as shown in Figure 13. The figure also shows the corresponding non-convex Pareto-optimal front in a f_1-f_2 plot with h function defined in Equation 9 having $\beta = 1$ and $\alpha = 4$ (since $\alpha > 1$, the Pareto-optimal front is non-convex) . The right figure is generated from 500 uniformly-spaced solutions in x_1. The value of x_2 is fixed so that the minimum value of $g^*(x_2)$ is equal to 1. The figure shows that the Pareto-optimal region is biased for solutions for which f_1 is near one.

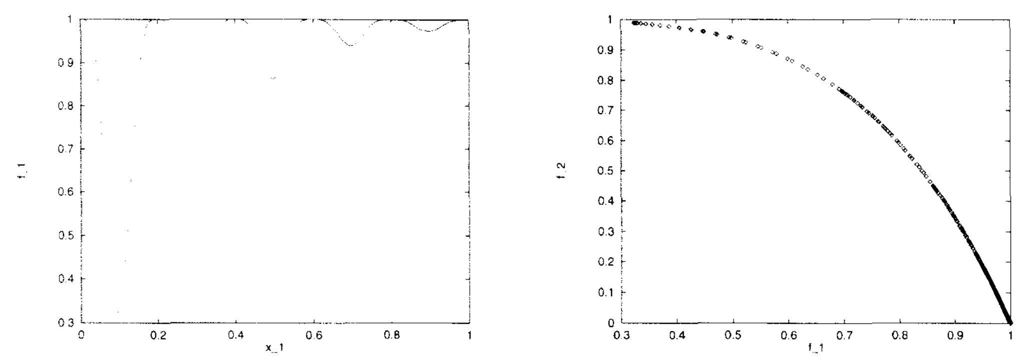


Figure 13: A multi-modal f_1 function and corresponding non-uniformly distributed non-convex Pareto-optimal region. In the right plot, Pareto-optimal solutions derived from 500 uniformly-spaced x_1 solutions are shown.

5.3.1 Function-Space and Parameter-Space Niching

The working of a multi-objective GA on the above function provides interesting insights about function-space niching (Fonseca and Fleming, 1993) and parameter-space niching (Srinivas and Deb, 1995). It is clear that when function-space niching is performed, diversity in the context of objective function values is anticipated, whereas when parameter space niching is performed, diversity in the phenotype (or genotype) of solutions is expected. We illustrate the difference by comparing the performance of NSGAs with both niching methods on the above problem. NSGAs with a reasonable parameter setting (population size of 100, 15-bit coding for each variable, σ_{share} of 0.2236 (assuming 5 niches), crossover probability of 1, and no mutation) are run for 500 generations. A typical run for both niching methods are shown in Figure 14. Although it seems that both niching methods are

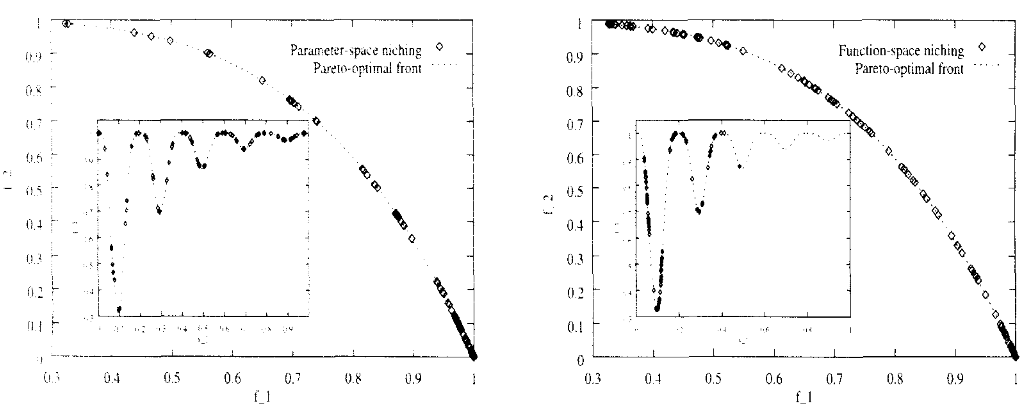
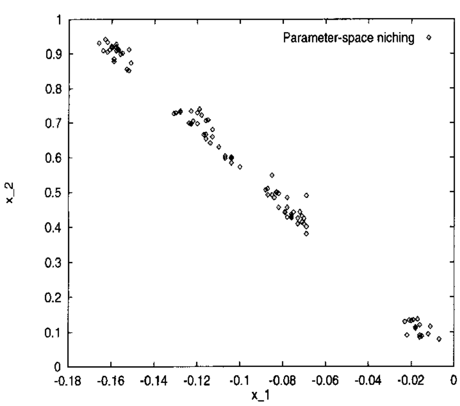


Figure 14: The left plot is with parameter-space niching and the right is with function-space niching. The figures show that both methods find solutions with diversity in the f_1-f_2 space.

able to maintain diversity in function space (with a better distribution in f_1-f_2 space with function-space niching), the left plot (inside figure) shows that the NSGA with parameter-space niching has truly found diverse solutions, whereas the NSGA with function-space niching (right plot) converges to about 50% of the entire region of the Pareto-optimal solutions. Since the first minimum and its basin of attraction spans the complete space for the function f_1, the function-space niching does not have the motivation to find other important solutions. Thus, in problems like this, function-space niching may hide information about important Pareto-optimal solutions in the search space.

It is important to understand that the choice between parameter-space or function-space niching depends entirely on what is desired in a set of Pareto-optimal solutions in the underlying problem. In some problems, it may be important to have solutions with trade-off in objective function values without concern for the similarity or diversity of the actual solutions (x vectors or strings). In such cases, function-space niching will, in general, provide solutions with better trade-off in objective function values. Since there is no induced pressure for the solutions to differ from each other, the Pareto-optimal solutions may not be very different, unless the underlying objective functions demand them to be so. On the other hand, in some problems the emphasis could be on finding more diverse solutions and with a trade-off among objective functions. Parameter-space niching would be better in such cases. This is because, in some sense, categorizing a population using non-domination helps to preserve some diversity among objective functions and an explicit parameter-space niching helps to maintain diversity in the solution vector.

To show the effect of parameter interactions (Section 5.2.2), we map the solution vector $\vec{x}$ into another vector $\vec{y}$ (obtained by rotation and translation). Now the distinction between parameter-space and function-space niching is even more clear (see Figure 15). GA parameter values identical to those in the unmapped case above are used here. Clearly, parameter-space niching is able to find more diverse solutions than function-space niching. However, an usual f_1-f_2 plot would reveal that the function-space niching is also able to



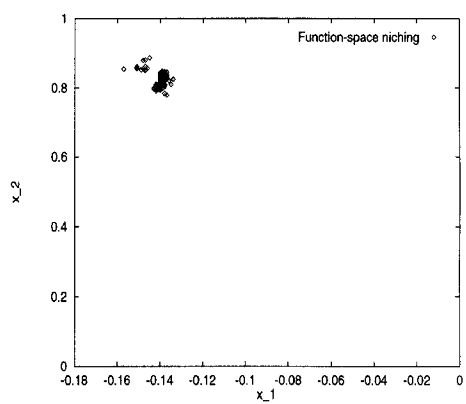


Figure 15: Solutions for a mapped problem are shown. The plots are made with all 100 solutions at generation 500.

find diverse solutions. A plot as in Figure 15 truly reveals the diversity achieved in the solutions.

6 Summary of Test Problems

The two-objective optimization problem discussed above requires three functions—f_1, g, and h—which can be set to various complexity levels to create complex two-objective optimization test problems. In the following, we summarize the properties of a two-objective optimization problem due to each of above functions:

1. The function f_1 tests a multi-objective GA's ability to find diverse Pareto-optimal solutions. Thus, this function tests an algorithm's ability to handle difficulties *along* the Pareto-optimal front.

2. The function g tests a multi-objective GA's ability to converge to the true (or global) Pareto-optimal front. Thus, this function tests an algorithm's ability to handle difficulties *lateral* to the Pareto-optimal front.

3. The function h tests a multi-objective GA's ability to tackle multi-objective problems having convex, non-convex, or discontinuous Pareto-optimal fronts. Thus, this function tests an algorithm's ability to handle different *shapes* of the Pareto-optimal front.

In the light of the above discussion, we summarize and suggest in Tables 1, 2, and 3 a few test functions for the above three functionals, which may be used in combination with each other. Unless specified, all variables x_i mentioned in the tables take real values in the range [0,1]. The functions mentioned in the third column in each table are representative

Table 1: Effect of function f_1 on the test problem.

Function $f_1(x_1, \ldots, x_m)$ (> 0) Controls search space along the Pareto-optimal front		
	Type	Example and Effect
F1-I	Single-variable ($m = 1$) and linear	**Example:** $\delta_f + c_1 x_1$ $(\delta_f, c_1 > 0)$ **Effect:** Uniform representation of solutions in the Pareto-optimal front. Most of the Pareto-optimal region is likely to be found.
F1-II	Multi-variable ($m > 1$) and linear	**Example:** $\delta_f + \sum_{i=1}^{m} c_i x_i$ $(\delta_f, c_i > 0)$ **Effect:** Non-uniform representation of Pareto-optimal front. Some Pareto-optimal regions are not likely to be found.
F1-III	Non-linear (any m)	**Example:** Eqn (16) for $m = 1$ or, $1 - \exp(-4r)\sin^4(5\pi r)$ where $r = \sqrt{\sum_{i=1}^{m} x_i^2}$ **Effect:** Same as above.
F1-IV	Multi-modal	**Example:** Eqn (4) with $g(x_2)$ replaced by $f_1(x_1)$ or other standard multi-modal test problems (such as Rastrigin's function, see Table 2) **Effect:** Same as above. Solutions at global optimum of f_1 and corresponding function values are difficult to find.
F1-V	Deceptive	**Example:** $f_1 = \sum_{i=1}^{m} f(\ell_i)$, where f is same as g defined in Eqn (6) **Effect:** Same as above. Solutions at true optimum of f_1 are difficult to find.

functions which will produce the desired effect mentioned in the respective fourth column. While testing an algorithm for its ability to overcome a particular feature of a test problem, we suggest varying the complexity of the corresponding function (f_1, g, or h) and fixing the other two functions at their easiest complexity level. For example, while testing an algorithm for its ability to find the global Pareto-optimal front in a multi-modal, multi-objective problem, we suggest choosing a multi-modal g function (G-III) and fixing f_1 as in F1-I and h as in H-I. Similarly, using g function as G-I, h function as H-I, and by first choosing f_1 function as F1-I, test a multi-objective optimizer's capability to distribute solutions along the Pareto-optimal front. By only changing the f_1 function to F1-III (even

Table 2: Effect of function g on the test problem.

Function $g(x_{m+1},\ldots,x_N)$ (>0), say $n=N-m$ Controls search space lateral to the Pareto-optimal front		
	Type	Example and Effect
G-I	Uni-modal, single-variable ($n=1$), and linear	**Example:** $\delta_g + c_2x_2$ $(\delta_g, c_2 > 0)$, or Eqn (14) with $\gamma = 1$ **Effect:** No bias for any region in the search space.
G-II	Uni-modal and non-linear	**Example:** Eqn (14) with $\gamma \neq 1$ **Effect:** With $\gamma > 1$, bias towards the Pareto-optimal region and with $\gamma < 1$, bias against the Pareto-optimal region.
G-III	Multi-modal	Rastrigin: **Example:** $1 + 10n + \sum_{i=m+1}^{N} x_i^2 - 10\cos(2\pi x_i)$ $x_i \in [-30, 30]$ **Effect:** Many $(61^n - 1)$ local and one global Pareto-optimal fronts Schwefel: **Example:** $1 + (6.5\pi)^2 n - \sum_{i=m+1}^{N} x_i \sin(\sqrt{\lvert x_i \rvert})$ $x_i \in [-512, 511]$ **Effect:** Many $(8^n - 1)$ local and one global Pareto-optimal fronts Griewangk: **Example:** $2 + \sum_{i=m+1}^{N} x_i^2/4000 - \Pi_{i=m+1}^{N} \cos(x_i/\sqrt{i})$ $x_i \in [-512, 511]$ **Effect:** Many $(163^n - 1)$ local and one global Pareto-optimal fronts
G-IV	Deceptive	**Example:** Eqn (6) **Effect:** Many $(2^n - 1)$ deceptive attractors and one global attractor
G-V	Multi-modal, deceptive	**Example:** $g(u(\ell_i)) = \begin{cases} 2+e, & \text{if } e < \ell_i/2, \\ 1, & \text{if } e = \ell_i/2. \end{cases}$ where $e = \lvert u(\ell_i) - \ell_i/2 \rvert$ **Effect:** Many $(\Pi_{i=m+1}^{N} \left[\binom{\ell_i}{\ell_i/2} + 2 \right] - 2^n)$ deceptive attractors and 2^n global attractors

Table 3: Effect of function h on the test problem.

Function $h(f_1, g)$ (>0) Controls shape of the Pareto-optimal front		
	Type	Example and Effect
H-I	Monotonically non-decreasing in g and convex on f_1	**Example:** Eqn (8) or Eqn (9) with $\alpha \leq 1$ **Effect:** Convex Pareto-optimal front
H-II	Monotonically non-decreasing in g and non-convex on f_1	**Example:** Eqn (9) with $\alpha > 1$ **Effect:** Non-convex Pareto-optimal front
H-III	Convexity in f_1 as a function of g	**Example:** Eqn (9) along with Eqn (12) **Effect:** Mixed convex and non-convex shapes for local and global Pareto-optimal fronts
H-IV	Non-monotonic periodic in f_1	**Example:** Eqn (13) **Effect:** Discontinuous Pareto-optimal front

with $m = 1$), the same optimizer can be tested for its ability to find distributed solutions in the Pareto-optimal front.

Along with any such combination of three functionals, parameter interactions can be introduced to create even more difficult problems. Using a transformation of the coordinate system as suggested in section 5.2.2, all the above-mentioned properties can be tested in a space where simultaneous adjustment of all parameter values is desired for finding an improved solution.

7 Future Directions for Research

This study suggests a number of immediate areas of research for developing better multi-objective GAs. A list of them are outlined and discussed in the following:

1. Compare existing multi-objective GA implementations
2. Understand dynamics of GA populations with generations
3. Investigate scalability issue of multi-objective GAs with number of objectives
4. Develop constrained test problems for multi-objective optimization
5. Study convergence properties to the true Pareto-optimal front
6. Introduce elitism in multi-objective GAs
7. Develop metrics for comparing two populations
8. Apply multi-objective GAs to more complex real-world problems
9. Develop multi-objective GAs for scheduling and other kinds of optimization problems

As mentioned earlier, there exists a number of different multi-objective GA implementations primarily varying in the way non-dominated solutions are emphasized and in the way the diversity in solutions are maintained. Although some studies have compared different GA implementations (Zitzler and Thiele, 1998), they all have been done on a specific problem without much knowledge about the complexity of the test problems. With the ability to construct test functions having controlled complexity, as illustrated in this paper, an immediate task would be to compare the existing multi-objective GAs and to establish the power of each algorithm in tackling different types of multi-objective optimization problems.

The test functions suggested here provide various degrees of complexity. The construction of all these test problems has been done without much knowledge of how multi-objective GAs work. If we know more about how such GAs work based on a non-domination principle, problems can be created to test more specific aspects of multi-objective GAs. In this regard, an interesting study would be to investigate how an initial random population of solutions moves from one generation to the next. An initial random population is expected to have solutions belonging to many non-domination levels. One hypothesis about the working of a multi-objective GA would be that most population members soon collapse to a single non-dominated front and each generation thereafter proceeds by improving this large non-dominated front. On the other hand, it may also be conjectured that GAs work by maintaining a number of non-domination levels at each generation. Both these modes of working should provide enough diversity for the GAs to find new and improved solutions and are likely candidates, although the actual mode of working may depend on the problem at hand. Thus, it will be worthwhile to investigate how existing multi-objective GA implementations work in the context of different test problems.

In this paper, we have not considered more than two objectives, although extensions of the concept to problems having more than two objectives can also be done. It is intuitive that as the number of objectives increases, the Pareto-optimal region is represented by multi-dimensional surfaces. With more objectives, multi-objective GAs must have to maintain more diverse solutions in the non-dominated front in each iteration. Whether GAs are able to find and maintain diverse solutions (as demanded by the search space of the problem) with many objectives would be an interesting study. Whether population size alone can solve this scalability issue or a major structural change (implementing a better niching method) is required would be the outcome of such a study.

We also have not considered constraints in this paper. Constraints can introduce additional complexity in the search space by inducing infeasible regions in the search space, thereby obstructing the progress of an algorithm towards the global Pareto-optimal front. Thus, creation of constrained test problems is an interesting area which should be emphasized in the future. With the development of such complex test problems, there is also a need to develop efficient constraint handling techniques that would be able to help GAs to overcome hurdles caused by constraints.

Most multi-objective GAs that exist to date work with the non-domination ranking of population members. Ironically, we have shown in Section 4 that all solutions in a non-dominated set need not be members of the true Pareto-optimal front, although some of them could be. In this regard, it would be interesting to introduce special features (such as elitism, mutation, or other diversity-preserving operators), the presence of which may help us to prove convergence of a GA population to the global Pareto-optimal front. Several attempts have been made to achieve such proofs for single-objective GAs (Suzuki, 1993; Rudolph, 1994) and similar attempts may also be made for multi-objective GAs.

Elitism is a useful and popular mechanism used in single-objective GAs. Elitism ensures that the best solutions in each generation will not be lost. What is more important is that these good solutions get a chance to participate in recombination with other solutions in the hope of creating better solutions. In multi-objective optimization, all non-dominated solutions of the first level are the best solutions in the population. Copying all such solutions to subsequent generations may make GAs stagnate. Thus, strategies for copying only a subset of non-dominated solutions must be developed.

Comparison of two populations in the context of multi-objective GAs also raises some interesting questions. As mentioned earlier, there are two goals in a multi-objective optimization—convergence to the true Pareto-optimal front and maintenance of diversity among Pareto-optimal solutions. A multi-objective GA may have found a population which has many Pareto-optimal solutions but with less diversity among them. How would such a population be compared with respect to another which has a fewer number of Pareto-optimal solutions but with wider diversity? Although there exists a suggestion of using a statistical metric (Fonseca and Fleming, 1996), most researchers use visual means of comparison which causes difficulty in problems having many objectives. The practitioners of multi-objective GAs must address this issue before they would be able to compare different GA implementations in a reasonable manner.

Test functions test an algorithm's ability to overcome a specific aspect of a real-world problem. In this respect, an algorithm which can overcome more aspects of problem difficulty is naturally a better algorithm. This is precisely the reason why so much effort is spent on doing research in test function development. As it is important to develop better algorithms by applying them on test problems with known complexity, it is also equally important that the algorithms are tested in real-world problems with unknown complexity. As mentioned earlier, the advantages of using a multi-objective GA in real-world problems are many and there is a need for interesting application case studies which would clearly show the advantages and flexibilities in using a multi-objective GA, as opposed to a single-objective GA.

With the advent of efficient multi-objective GAs for function optimization, the concept of multi-objective optimization can also be applied to other search and optimization problems such as multi-objective scheduling and other multi-objective combinatorial optimization problems. Since in tackling these problems using permutation GAs, the main differences from binary GAs are in the way the solutions are represented and in the construction of GA operators, an identical non-domination principle along with a similar niching concept can still be used in solving such problems having multiple objectives. In this context, similar concepts can also be implemented in developing other population-based, multi-objective EAs. Multi-objective evolution strategies, multi-objective genetic programming, or multi-objective evolutionary programming may better solve specific multi-objective problems which are ideally suited for the respective evolutionary method.

8 Conclusions

For the past few years, there has been a growing interest in the studies of multi-objective optimization using genetic algorithms (GAs). Although there exists a number of multi-objective GA implementations and applications to interesting multi-objective optimization problems, there is no systematic study to speculate what problem features may cause a multi-objective GA to face difficulties. In this paper, a number of such features are identified and a simple methodology is suggested to construct test problems from single-objective optimization problems. The construction method requires the choice of three functions, each of which controls a particular aspect of difficulty for a multi-objective GA. One function, (f_1), tests an algorithm's ability to handle difficulties along the Pareto-optimal region; function (g) tests an algorithm's ability to handle difficulties lateral to the Pareto-optimal region; and function (h) tests an algorithm's ability to handle difficulties arising because of different shapes of the Pareto-optimal region. This allows a multi-objective GA to be tested in a controlled manner on various aspects of problem difficulties. Since test problems are constructed from single-objective optimization problems, most theoretical or experimental studies on problem difficulties or on test function development in single-objective GAs are of direct importance to multi-objective optimization.

This paper has made a modest attempt to reveal and test some interesting aspects of multi-objective optimization. A number of other salient and related studies are suggested for future research. We believe that more studies are needed to better understand the working principles of a multi-objective GA. An obvious outcome of such studies would be the development of new and improved multi-objective GAs.

Acknowledgments

The author acknowledges support from the Alexander von Humboldt Foundation, Germany, and The Department of Science and Technology, India, during the course of this study. The comments of Günter Rudolph, Eckert Zitzler, and anonymous reviewers have improved the readability of the paper.

References

Cunha, A. G., Oliveira, P. and Covas, J. A. (1997). Use of genetic algorithms in multicriteria optimization to solve industrial problems. In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 682–688, Morgan Kauffman, San Mateo, California.

Deb, K. (1995). *Optimization for engineering design: Algorithms and examples.* Prentice-Hall, New Delhi, India.

Deb, K. (in press). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, North-Holland, Amsterdam, Netherlands.

Deb, K. and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In Schaffer, D., editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, Morgan Kauffman, San Mateo, California.

Deb, K. and Goldberg, D. E. (1994). Sufficient conditions for arbitrary binary functions. *Annals of Mathematics and Artificial Intelligence*, 10:385–408.

Deb, K., Horn, J. and Goldberg, D. E. (1993). Multi-Modal deceptive functions. *Complex Systems*, 7:131–153.

Deb, K. and Kumar, A. (1995). Real-coded genetic algorithms with simulated binary crossover: Studies on multi-modal and multi-objective problems. *Complex Systems*, 9:431–454.

Eheart, J. W., Cieniawski, S. E. and Ranjithan, S. (1993). Genetic-algorithm-based design of groundwater quality monitoring system. WRC Research Report No. 218, Department of Civil Engineering, The University of Illinois at Urbana-Champaign, Urbana, Illinois.

Fonseca, C. M. and Fleming, P. J. (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, Morgan Kauffman, San Mateo, California.

Fonseca, C. M. and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3:1–16.

Fonseca, C. M. and Fleming, P. J. (1996). On the performance assessment and comparison of stochastic multi-objective optimizers. In Voigt, H.-M., Ebeling, W., Rechenberg, I. and Schwefel, H.-P., editors, *Proceedings of the Parallel Problem Solving from Nature IV*, pages 584–593, Springer, Berlin, Germany.

Goldberg, D. E. (1989). *Genetic algorithms for search, optimization, and machine learning.* Addison-Wesley, Reading, Massachusetts.

Goldberg, D. E., Korb, B. and Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3:93–530.

Grefenstette, J. J. (1993). Deception considered harmful. In Whitley, D., editor, *Foundations of Genetic Algorithms, II*, pages 75–91, Morgan Kauffman, San Mateo, California.

Horn, J. (1997). Multicriterion decision making. In Bäck, T., Fogel, D. B. and Michalewicz, Z., editors, *Handbook of Evolutionary Computation*, pages F1.9:1–15, Oxford University Press, New York, New York.

Horn, J., Nafploitis, N. and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multi-objective optimization. In Michalewicz, Z., editor, *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, IEEE Service Center, Piscataway, New Jersey.

Koziel, S. and Michalewicz, Z. (1998). A decoder-based evolutionary algorithm for constrained parameter optimization problems. In Eiben, A. E., Bäck, T., Schoenauer, M. and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature, V*, pages 231–240, Springer, Berlin, Germany.

Kurusawe, F. (1990). A variant of evolution strategies for vector optimization. In Schwefel, H.-P. and Männer, R., editors, *Parallel Problem Solving from Nature, I*, pages 193–197, Springer-Verlag, Berlin, Germany.

Laumanns, M., Rudolph, G. and Schwefel, H.-P. (1998). A spatial predator-prey approach to multi-objective optimization: A preliminary study. In Eiben, A. E., Bäck, T., Schoenauer, M. and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature, V*, pages 241–249, Springer, Berlin, Germany.

Mitra, K., Deb, K. and Gupta, S. K. (1998). Multiobjective dynamic optimization of an industrial Nylon 6 semibatch reactor using genetic algorithms. *Journal of Applied Polymer Science*, 69:69–87.

Parks, G. T. and Miller, I. (1998). Selective breeding in a multi-objective genetic algorithm. In Eiben, A. E., Bäck, T., Schoenauer, M. and Schwefel, H.-P., editors, *Proceedings of the Parallel Problem Solving from Nature, V*, pages 250–259, Springer, Berlin, Germany.

Poloni, C., Giurgevich, A., Onesti, L. and Pediroda, V. (in press). Hybridisation of multiobjective genetic algorithm, neural networks and classical optimiser for complex design problems in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, North Holland, Amsterdam, Netherlands.

Rudolph, G. (1994). Convergence properties of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, NN-5:96–101.

Schaffer, J. D. (1984). Some experiments in machine learning using vector evaluated genetic algorithms. Doctoral Dissertation, Vanderbilt University, Nashville, Tennessee.

Srinivas, N. and Deb, K. (1995). Multi-Objective function optimization using non-dominated sorting genetic algorithms, *Evolutionary Computation*, 2(3):221–248.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application*. John Wiley, New York, New York.

Suzuki, J. (1993). A markov chain analysis on a genetic algorithm. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 146–153, Morgan Kauffman, San Mateo, California.

Van Veldhuizen, D. and Lamont, G. B. (1998). Multiobjective evolutionary algorithm research: A history and analysis. Technical Report Number TR-98-03. Wright-Patterson AFB, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Ohio.

Weile, D. S., Michielssen, E. and Goldberg, D. E. (1996). Genetic algorithm design of Pareto-optimal broad band microwave absorbers. *IEEE Transactions on Electromagnetic Compatibility*, 38:518–525.

Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—A comparative case study. In Eiben, A. E., Bäck, T., Schoenauer, M. and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature, V*, pages 292–301, Springer, Berlin, Germany.