EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION IN UNCERTAIN ENVIRONMENTS

GOH CHI KEONG
(B.Eng (Hons.), NUS)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2007

Summary

Many real-world problems involve the simultaneous optimization of several competing objectives and constraints that are difficult, if not impossible, to solve without the aid of powerful optimization algorithms. What makes multi-objective optimization so challenging is that, in the presence of conflicting specifications, no one solution is optimal to all objectives and optimization algorithms must be capable of finding a number of alternative solutions representing the tradeoffs. However, multi-objectivity is just one facet of real-world applications. Most optimization problems are also characterized by various forms of uncertainties stemming from factors such as data incompleteness and uncertainties, environmental conditions uncertainties, and solutions that cannot be implemented exactly.

Evolutionary algorithms are a class of stochastic search methods that have been found to be very efficient and effective in solving sophisticated multi-objective problems where conventional optimization tools fail to work well. Evolutionary algorithms' advantage can be attributed to it's capability of sampling multiple candidate solutions simultaneously, a task that most classical multi-objective optimization techniques are found to be wanting. Much work has been done to the development of these algorithms in the past decade and it is finding increasingly application to the fields of bioinformatics, logical circuit design, control engineering and resource allocation. Interestingly, many researchers in the field of evolutionary multi-objective optimization assume that the optimization problems are deterministic, and uncertainties are rarely examined. While multi-objective evolutionary algorithms draw its inspiration from nature where uncertainty is a common phenomenon, it cannot be taken for granted that these algorithms will hence be inherently robust to uncertainties without any further investigation.

The primary motivation of this work is to provide a comprehensive treatment on the design and application of multi-objective evolutionary algorithms for multi-objective optimization in the presence of uncertainties. This work is divided into three parts, which each part considering a different form of uncertainties: 1) noisy fitness functions, 2) dynamic fitness functions, and 3) robust optimization. The first part addresses the issues of noisy fitness functions. In particular, three noise-handling mechanisms are developed to improve algorithmic performance. Subsequently, a basic multi-objective evolutionary algorithm incorporating these three mechanisms are validated against existing techniques under different noise levels. As a specific instance of a noisy MO problem, a hybrid multi-objective evolutionary algorithm is also presented for the evolution of artificial neural network classifiers. Noise is introduced as a consequence of synaptic weights that are not well trained for a particular network structure. Therefore, a local search procedure consisting of a micro-hybrid genetic algorithm and pseudo-inverse operator is applied to adapt the weights to reduce the influence of noise.

Part II is concerned with dynamic multi-objective optimization and extends the notion of coevolution to track the Pareto front in a dynamic environment. Since problem characteristics may change with time, it is not possible to determine one best approach to problem decomposition. Therefore, this chapter introduces a new coevolutionary paradigm that incorporates both competitive and cooperative mechanisms observed in nature to facilitate the adaptation and emergence of the decomposition process with time.

The final part of this work addresses the issues of robust multi-objective optimization where the optimality of the solutions is sensitive to parameter variations. Analyzing the existing benchmarks applied in the literature reveals that the current corpus has severe limitations. Therefore, a robust multi-objective test suite with noise-induced solution space, fitness landscape and decision space variation is presented. In addition, the vehicle routing problem with stochastic demand (VRPSD) is presented a practical example of robust combinatorial multi-objective optimization problems.

Acknowledgements

During the entire course of completing my doctoral dissertation, I have gained no less than three inches of fat. Remarkably, my weight stays down which definitely says that a hair loss programe is definitely better than any weight-loss regime you can find out there. Conclusions: A thoroughly enjoyable experience.

First and foremost, I like to thank my thesis supervisor, Associate Professor Dr. Tan Kay Chen for introducing me to the wonderful field of computational intelligence and giving me the opportunity to pursue research. His advice have kept my work on course during the past three years.

I am also grateful to the rowdy bunch at the Control and Simulation laboratory: Yang Yinjie for the numerous discussions, Teoh Eujin for sharing the same many interests, Chiam Swee Chiang for convincing me that I am the one taking “kiasuism” to the extreme, Brian for infecting the lab with “bang effect” , Cheong Chun Yew for each and every little lab entertainment (with his partner in crime), Liu Dasheng for his invaluable services to the research group, Tan Chin Hiong who has not been too seriously affected by the “bang effect” yet, and Quek Han Yang who takes perverse pleasure in reminding what a bunch of slackers we are.

Last but not least, I want to thank my family for all their love and support: My parents for their patience, my brother for his propaganda that I am kept in school because I am a threat to the society and my sister who loves reminding me of my age. Those little rascals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optimization may be considered as a decision-making process to get the most out of avaliable resources for the best attainable results. Simple examples include everyday decisions, such as the type of transport to take, which clothes to wear and what groceries to buy. For these routine tasks, the decision to be made for, say, cheapest transport can be exceedingly clear. Consider now, the situation where we are running late for a meeting due to some unforseen circumstances. Since the need for expedition is conflicting to the first consideration of minimizing cost, the selection of the right form of transportation is no longer as straightforward as before and the final solution will represent a compromise between the different objectives. This type of problems which involves the simultaneous consideration of multiple objectives are commonly termed as multi-objective (MO) problems.

Many real-world problems naturally involve the simultaneous optimization of several competing objectives. Unfortunately, these problems are characterized by objectives that are much more complex as compared to routine tasks and the decision space are often so large that it is often difficult, if not impossible, to be solved without advanced and efficient optimization techniques. In addition, as reflected by the element of uncertainty in the example given above, the magnitude of this task is exacerbated by uncertainties such as the presence of noise and time-varying components that are inherent to real-world problems. MO optimization in the presence of uncertainties are of great importance in practice, where the slight difference in environmental conditions or implementation variations can be crucial to overall operational success or failure.

1.1 MO optimization

Real-world optimization tasks are typically represented by its mathematical model and the specification of MO criteria captures more information about the modeled problem as several problem characteristics are taken into consideration. For instance, consider the design of a system controller that can be found in process plants, automated vehicles and in household appliances. Apart from obvious tradeoffs between cost and performance, the performance criteria required by some applications such as fast response time, small overshoot and good robustness, are also conflicting in nature [34, 62, 138, 205].

Without any loss of generality, a minimization problem is considered here and the MO problem can be formally defined as

$$\min_{\vec{x} \in \vec{X}^{n_x}} \vec{f}(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), ..., f_M(\vec{x})\} \tag{1.1}$$

$$\text{s.t. } \vec{g}(\vec{x}) > 0, \vec{h}(\vec{x}) = 0$$

where $\vec{x}$ is the vector of decision variables bounded by the decision space, $\vec{X}^{n_x}$ and $\vec{f}$ is the set of objectives to be minimized. The terms "solution space" and "search space" are often used to denote the decision space and will be used interchangeably throughout this work. The functions $\vec{g}$ and $\vec{h}$ represents the set of inequality and equality constraints that defines the feasible region of the n_x-dimensional continuous or discrete feasible solution space. The relationship between the decision variables and the objectives are governed by the objective function $\vec{f} : \vec{X}^{n_x} \longmapsto \vec{F}^M$. Figure. 1.1 illustrates the mapping between the two spaces. Depending on the actual objective function and constraints of the particular MO problem, this mapping is not unique and may be one-to-many or many-to-one.

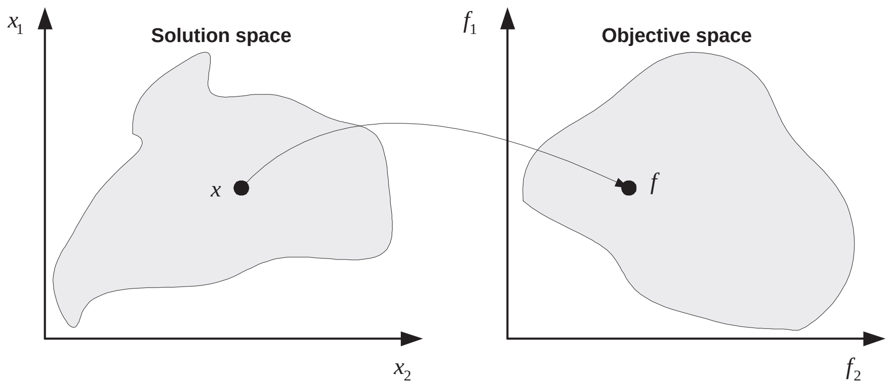


Figure 1.1: Illustration of the mapping between the solution space and the objective space.

1.1.1 Totally conflicting, nonconflicting, and partially conflicting MO problems

One of the key differences between SO and MO optimization is that MO problems constitute a multi-dimensional objective space, $\vec{F}^M$. This leads to three possible instances of MO problem, depending on whether the objectives are totally conflicting, nonconflicting, or partially conflicting. For MO problems of the first category, the conflicting nature of the objectives are such that no improvements can be made without violating any constraints. This result in an interesting situation where all feasible solutions are also optimal. Therefore, totally conflicting MO problems are perhaps the simplest of the three since no optimization is required. On the other extreme, a MO problem is nonconflicting if the various objectives are correlated and the optimization of any arbitrary objective leads to the subsequent improvement of the other objectives. This class of MO problem can be treated as a SO problem by optimizing the problem along an arbitrarily selected objective or by aggregating the different objectives into a scalar function. Intuitively, a single optimal solution exist for such a MO problem.

More often than not, real-world problems are instantiations of the third type of MO problems and this is the class of MO problems that we are interested in. One serious implication is that a set of solutions representing the tradeoffs between the different objectives is now sought rather than an unique optimal solution. Consider again the example of cost vs performance of a controller. Assuming that the two objectives are indeed conflicting, this present a least two possible extreme solutions, each representing the best achievable situation for one objective at the expense of the other. The other solutions, if any, making up this optimal set of solutions represent the varying degree of optimality with respect to the two different objectives. Certainly, our conventional notion of optimality gets thrown out of the window and a new definition of optimality is required for MO problems.

1.1.2 Pareto Dominance and Optimality

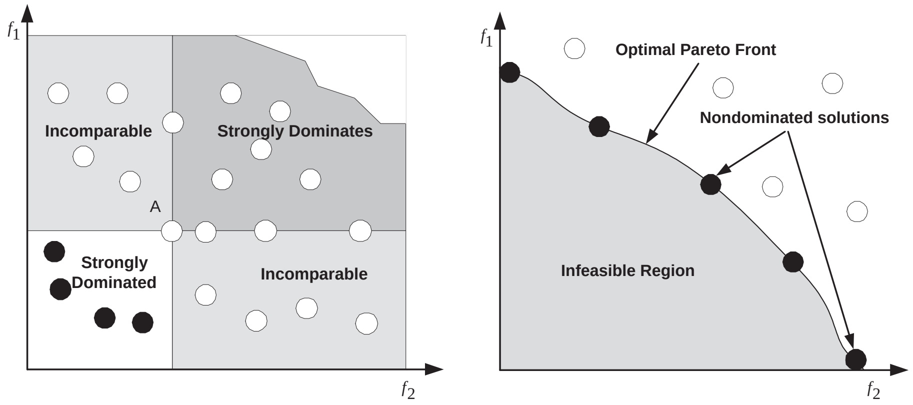
The concepts of Pareto dominance and Pareto optimality are fundamental in MO optimization, with Pareto dominance forming the basis of solution quality. Unlike SO optimization where there is a complete order exist (i.e, $f_1 \leq f_2$ or $f_1 \geq f_2$), $\vec{X}^{n_x}$ is partially-ordered when multiple objectives are involved. In fact, there are three possible relationship between the solutions that is defined by Pareto dominance.

Definition 1.1: Weak Dominance:$\vec{f_1} \in \vec{F}^M$ weakly dominates $\vec{f_2} \in \vec{F}^M$, denoted by $\vec{f_1} \preceq \vec{f_2}$ iff $f_{1,i} \leq f_{2,i}$ $\forall i \in \{1, 2, ..., M\}$ and $f_{1,j} < f_{2,j}$ $\exists j \in \{1, 2, ..., M\}$

Definition 1.2: Strong Dominance: $\vec{f_1} \in \vec{F}^M$ strongly dominates $\vec{f_2} \in \vec{F}^M$, denoted by $\vec{f_1} \prec \vec{f_2}$ iff $f_{1,i} < f_{2,i}$ $\forall i \in \{1, 2, ..., M\}$

Definition 1.3: Incomparable: $\vec{f_1} \in \vec{F}^M$ is incomparable with $\vec{f_2} \in \vec{F}^M$, denoted by $\vec{f_1} \sim \vec{f_2}$ iff $f_{1,i} > f_{2,i}$ $\exists i \in \{1, 2, ..., M\}$ and $f_{1,j} < x_{2,j}$ $\exists j \in \{1, 2, ..., M\}$

With solution A as our point of reference, the regions highlighted in different shades of grey in Figure 1.2(a) illustrates the three different dominance relations. Solutions located in the dark grey regions are dominated by solution A because A is better in *both* objectives. For the same reason, solutions located in the white region dominates solution A. Although A has a smaller objective value as compared to the solutions located at the boundaries between the dark and light grey regions, it only weakly dominates these solutions by virtue



(a) (b)

Figure 1.2: Illustration of the (a) Pareto Dominance relationship between candidate solutions relative to solution A and (b) the relationship between the Approximation Set, PF^A and the true Pareto front, PF^*.

of the fact that they share a similar objective value along either one dimension. Solutions located in the light grey regions are incomparable to solution A because it is not possible to establish any superiority of one solution over the other: solutions in the left light grey region are better only in the second objective while solutions in the right grey region are better only in the first objective. It can be easily noted that there is a natural ordering of these relations: $\vec{f_1} \prec \vec{f_1} \;\Rightarrow\; \vec{f_1} \preceq \vec{f_1} \;\Rightarrow\; \vec{f_1} \sim \vec{f_2}$.

With the definition of Pareto dominance, we are now in the position to consider the set of solutions desirable for MO optimization.

Definition 1.4: Pareto Optimal Front: The Pareto optimal front, denoted as PF^*, is the set of nondominated solutions with respect to the objective space such that $\text{PF}^* = \{\vec{f}_i^* | \nexists \vec{f}_j \prec \vec{f}_i^*, \vec{f}_j \in \vec{F}^M\}$

Definition 1.5: Pareto Optimal Set: The Pareto optimal set, denoted as PS^*, is the set of solutions that are nondominated in the objective space such that $\text{PS}^* = \{\vec{x}_i^* | \nexists \vec{F}(\vec{x}_j) \prec \vec{F}(\vec{x}_i^*), \vec{F}(\vec{x}_j) \in \vec{F}^M\}$

The set of tradeoff solutions is known as the Pareto optimal set and these solutions are also termed “noninferior”, “admissible” or “efficient” solutions. The corresponding objective vectors of these solutions are termed “non-dominated” and each objective component of any non-dominated solution in the Pareto optimal set can only be improved by degrading at least one of its other objective components [188].

1.1.3 MO Optimization Goals

An example of the PF* is illustrated in Figure 1.2(b). Most often, information regarding the PF* and its tradeoffs are either limited or not known *a priori*. It is also not easy to find a nice closed analytic expression for the tradeoff surface because real-world MO problems usually have complex objective functions and constraints. Therefore, in the absence of any clear preference on the part of the decision-maker, the ultimate goal of MOO is to discover the entire tradeoff. However, by definition, this set of objective vectors is possibly an infinite set as in the case of numerical optimization and it is simply not achievable.

On a more practical note, the presence of too many alternatives could very well overwhelm the decision-making capabilities of the decision-maker. In this light, it would be more practical to settle for the discovery of as many nondominated solutions possible as our limited computational resources permits. More precisely, we are interested in finding a good approximation of the PF* and this approximate set, PF^A should satisfy the following optimization goals.

- Minimize the distance between the PF^A and PF*.
- Obtain a good distribution of generated solutions along the PF^A.
- Maximize the spread of the discovered solutions.

An example of such an approximation is illustrated by the set of nondominated solutions denoted by the filled circles residing along the PF* in Figure 1.2(b). While the first optimization goal of convergence is the first and foremost consideration of all optimization problems, the second and third optimization goal of maximizing diversity are entirely unique to MO optimization. The rationale of finding a diverse and uniformly distributed PF^A is to provide the decision maker with sufficient information about the tradeoffs between the different solutions before the final decision is made. It should also be noted that the optimization goals of convergence and diversity are somewhat conflicting in nature, which explains why MO optimization is much more difficult than SO optimization.

1.2 MO Optimization in The Presence of Uncertainties

The MO problem formulated in the previous section reflects the conventional methodology adopted in the vast majority of the optimization literature which assumes that the MO problem is deterministic and the core optimization concern is the maximization of solution set quality. However, Pareto optimality of the PF^A does not necessarily mean that any of the solutions along the tradeoff is desirable or even implementable in practice. This is primarily because such a deterministic approach neglects the fact that real-world problems are characterized by uncertainty.

Jin and Branke [107] identified four general forms of uncertainty that are encountered in evolutionary optimization: 1) noisy fitness functions [72], 2) dynamic fitness functions, 3) uncertainty of design variables or environmental parameters [40, 73], and 4) approximation errors. The first three types of uncertainties are inherent to the environment and are due to factors such as data incompleteness and uncertainties, environmental conditions uncertainties, and solutions that cannot be implemented exactly. On the other hand, the fourth type of uncertainty is introduced as a consequence of the use of approximated fitness function to reduce computational cost.

Uncertainties due to noise in the objective functions may arise from different sources such as sensor measurement errors, incomplete simulations of computational models and stochastic simulations. Apart from these external sources, noise can also be intrinsic to the problem. A good example is the evolution of neural networks where the same network structure can give rise to different fitness values due to different weight instantiations [107]. A distinctive feature of noisy fitness function is that each evaluation of the same solution may result in different fitness values. Mathematically, for *noisy* MO optimization, (1.1) can be rewritten as

$$\min_{\vec{x}\in\vec{X}^{n_x}} \vec{F}(\vec{x}) = \{f_1(\vec{x}) + \delta_1, f_2(\vec{x}) + \delta_2, ..., f_M(\vec{x}) + \delta_M\} \tag{1.2}$$

where δ_i is a scalar noise parameter added to the original objective function of f_i and $\vec{F}$ is the resultant objective vector.

In contrast to noisy fitness functions, the fitness topology of *dynamic* MO problems may change but the objective values is deterministic at any one time. In this context, the term *static* is more appropriate than deterministic for denoting MO problems without explicit consideration of its dynamism. For such problems, the PF* and the PS* is unlikely to remain invariant and the optimization algorithm must be capable of tracking the PS* over time. In a certain sense, the dynamic MO problem can considered as the consecutive optimization of different time-constrained MO problems with varying complexities. However, information from the previous environment may be exploited to improve convergence speed. The dynamic MO problem can be described as

$$\min_{\vec{x}\in\vec{X}^{n_x}} \vec{F}(\vec{x}, t) = \{f_1(\vec{x}, t), f_2(\vec{x}, t), ..., f_M(\vec{x}, t)\} \tag{1.3}$$

where t is typically measured in terms of solution evaluations.

The third class of uncertainty arises because small deviations from the design during the manufacturing process and fluctuations in the operating environment is inevitable in the real-world. Designs that are optimized without taking robustness into account are susceptible to large or unacceptable performance variation due to decision or environmental parameter variation. Therefore, uncertainties arise in the design space rather than the objective space in robust optimization. In order to reduce the consequences of uncertainty on optimality and practicality of the solution set, factors such as decision variable variation and environmental variation have to be considered explicitly. Therefore, the *robust* MO problem can be given as

$$\min_{\vec{x}\in\vec{X}^{nx}} \vec{F}(\vec{x},\vec{\sigma_x},\vec{\sigma_e}) = \{f_1(\vec{x},\vec{\sigma_x},\vec{\sigma_e}), f_2(\vec{x},\vec{\sigma_x},\vec{\sigma_e}), ..., f_M(\vec{x},\vec{\sigma_x},\vec{\sigma_e})\} \tag{1.4}$$

where σ_x and σ_e represent the uncertainty associated with $\vec{x}$ and environmental conditions. Both forms of uncertainties may be treated equivalently. In this context, the PF^A and PS^{A}that is evolved based on (1.1) can be denoted as the *efficient* front and *efficient* solution set respectively. A major distinction between noisy and robust optimization is that noise is introduced deliberately into the robust optimization problem to simulate the effects of parametric variation.

The fourth class of uncertainty is a consequence of the use of meta-models in place of the original fitness functions, and often represents a tradeoff between model fidelity and computational cost. One distinct feature of this form of uncertainty is that it introduces a bias into the problem. The MO problem with approximated fitness can be given as

$$\min_{\vec{x}\in\vec{X}^{nx}} \vec{F}(\vec{x}) = \vec{F}(\vec{x}) + \vec{E}(\vec{x}) \tag{1.5}$$

where E is the approximation error of the meta-model.

1.3 Evolutionary Multi-objective Optimization

Traditional operational research approaches to MO optimization typically entails the transformation of the original problem into a SO problem and employs point-by-point algorithms such as branch-and-bound to iteratively obtain a better solution. Such approaches have several limitations including the requirement of the MO problem to be well-behaved, i.e. differentiability or satisfying the Kuhn-Tucker conditions, sensitivity to the shape of the Pareto-front and the generation of only one solution for each simulation run. On the other hand, metaheuristical approaches that are inspired by biological or physics phenomena such as evolutionary algorithms and simulated annealing have been gaining increasing acceptance as a much more flexible and effective alternative to complex optimization problems in the recent years. This is certainly a stark contrast to just two decades ago, as Reeves remarked in [169] that an eminent person in operational research circles suggested that using a heuristic was an admission of defeat!

Among these metaheuristics, MOEA is one of the more popular stochastic search methodology to solve MO problems. Emulating the DarwinianWallace principle of "survival-of-the-fittest" in natural selection and adaptation, MOEAs have the distinct advantage of being able to sample multiple solutions simultaneously. Such a feature provides the MOEA with a global perspective of the MO problem as well as the capability to find a set of Pareto-optimal solutions in a single run. Applying genetic operators such as the selection process and crossover operator allows the MOEA to intelligently sieve through the large amount of information embedded within each individual representing a candidate solution and exchange information between them to increase the overall quality of the individuals in the population. In this section, state-of-the-arts MOEAs, MO test problems and performance indicators that are used for algorithmic performance evaluation in this work are discussed.

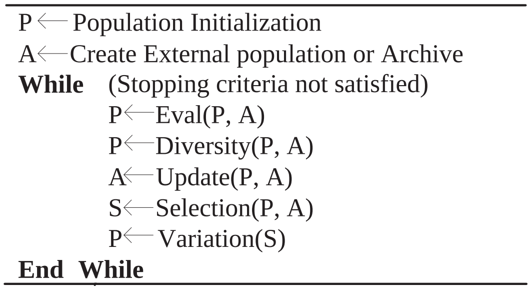
1.3.1 MOEA Framework

Many different evolutionary techniques for MO optimization have been proposed since the pioneering effort of Schaffer in [179], with the aim of fulfilling the three optimization goals described previously. Most of these MOEAs are largely based on the computational models of genetic algorithms (GAs) [88], evolutionary programming (EP) [59] and evolutionary strategies (ES) [168]. Interestingly, ES is the only paradigm developed for the purpose of optimization; GAs are designed as a general adaptive system while ES are developed as a learning process to create artificial intelligence.

Recent years have also seen the emergence of other biologically inspired models such as particle swarm optimization (PSO), differential evolution (DE), cultural algorithms (CA), and artificial immune systems (AIS) for MO optimization. While all these algorithms are different in methodology, particularly in the generation of new candidate solutions, the distinctions between them have become increasingly vague as researchers sought to exploit the advantages offered by the different algorithms in a common platform. Moreover, MO optimization requires researchers to address many similar issues that are unique to MO problems, regardless of the computational model applied. Therefore, no distinction will be made between the different evolutionary computation models and all these techniques developed for MO optimization are referred as MOEA.

One distinct feature that characterizes state-of-the-art MOEAs such as nondominated sorting genetic algorithm II (NSGAII) [43], Pareto archived evolution strategy (PAES) [127], Pareto envelope based selection algorithm (PESA) [32], incrementing multi-objective evolutionary algorithm (IMOEA) [199] and strength Pareto evolutionary algorithm 2 (SPEA2) [228] from early research efforts is the incorporation of elitism. Elitism involves two closely related process, 1) the preservation of good solutions and 2) the reinsertion of these solutions into the evolving population. While the general motivations may be similar, these algorithms can be distinguished by the way in which the mechanisms of elitism and diversity preservation are implemented.

The general MOEA framework can be represented in the pseudocode shown in Fig. 1.3 and it can be shown that most MOEAs fit into this framework. There seem to be many similarities between SO evolutionary algorithms (SOEAs) and MOEAs with both techniques involving an iterative adaptation of a set of solutions until a pre-specified optimization goal/stopping criterion is met. What sets these two techniques apart is the manner in which solution assessment and elitism are performed. This is actually a consequence of the



```
P ⟵ Population Initialization
A⟵Create External population or Archive
While   (Stopping criteria not satisfied)
        P⟵Eval(P, A)
        P⟵Diversity(P, A)
        A⟵Update(P, A)
        S⟵Selection(P, A)
        P⟵ Variation(S)
End  While
```

Figure 1.3: Framework of MOEA

three optimization goals described in Section 1.1.3. In particular, solution assessment must exert a pressure to drive the solutions toward the global tradeoffs as well as to diversify the individuals uniformly along the discovered PF^{A}. The archive updating and selection process must also take diversity into consideration to encourage and maintain a diverse solution set.

The optimization process starts with the initialization of the population. This is followed by evaluation (Eval) and density assessment (Diversity) of candidate solutions. After which, good solutions are updated into an external population or archive (Update). MOEAs perform the archiving process differently, some of which maintains a fixed sized archive while others store only nondominated solutions. Nonetheless, in most cases, a truncation process will be conducted based on some density assessment to restrict the number of archived solutions. Both NSGAII and SPEA2 maintains a fixed sized archive which includes both dominated and nondominated solutions while PAES and PESA stores only nondominated solutions. For the truncation process, PAES and PESA employ a hyper-grid measure while SPEA, NSGAII and IMOEA employ Euclidean-based measures.

The selection process typically involves the set of nondominated solutions updated in the previous stage. For NSGAII, SPEA2 and PESA, tournament selection is conducted directly on the archive. In [196], the archive of nondominated solutions and evolving population is combined before tournament selection is performed. Bosman and Thierens [15] noted that diversity usually serves only as a secondary selection criteria to the optimization goal of convergence. As a specific instance, NSGAII applies the crowded comparison operator only to break any tie in rank occurred during the tournament selection. On the other hand, the selection process in PESA is based on the degree of crowding or the squeeze factor only.

After the selection process, variation operators are applied to explore and exploit the selected individuals to generate a new population of solutions. Different methods of generating individuals can be found in the literature. Uniform crossover and bit-flip mutation have been used for NSGAII and SPEA2. In AIS-inspired MOEAs [29, 141], cloning and hypermutation are applied while EDA-based MOEAs [16, 152] enforce sampling from learnt probabilistic models. Variation operators associated with the various paradigms have been applied across the different computational model resulting in very similar implementations, a point mentioned earlier. Some recent examples include the introduction of recombination into the AIS-inspired MOEAs in [192, 106] and the hybridization of clonal selection and hypermutation with PSO-inspired MOEAs [223].

1.3.2 Basic MOEA Components

The framework presented in the previous section serves to highlight the primary components of the MOEA, elements without which the algorithm is unable to fulfill its basic function of finding PF*satisfactorily. More elaborate frameworks with different concerns exist in the literature. For instance, Bosman and Thierens [15] presented a framework that considers how MOEAs can be constructed to control the emphasis on the exploration and exploitation of diversity or proximity. In another work, Laumanns *et al* [134] focused on the design and incorporation of elitism into MOEAs.

Fitness Assignment

As illustrated in Figure 1.4, solution assessment in MOEA should be designed in such a way that a pressure $\overleftarrow{P}_n$ is exerted to promote the solutions in a direction normal to the

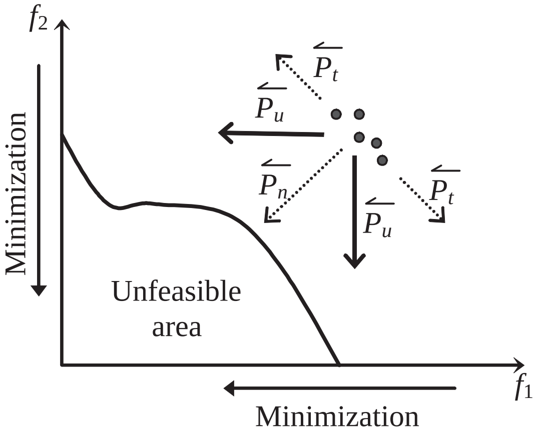


Figure 1.4: Illustration of Selection Pressure Required to Drive Evolved Solutions Towards PF*

tradeoffs region and at the same time, another pressure $\overleftarrow{P}_t$ to promote the solutions in a direction tangentially to that region. These two orthogonal pressures result in the unified pressure $\overleftarrow{P}_u$ to direct the evolutionary search in the MO optimization context. Based on the literature, it is possible to identify three different classes of fitness assignment: 1) Pareto-based assignment, 2) aggregation-based assignment and 3) indicator based assignment.

Pareto-based Fitness Assignment: Pareto-based MOEAs have emerged as the most popular approach [198] since Fonseca and Fleming [63] put into practice the notion of dominance suggested in [76]. On its own, Pareto dominance is unable to induce $\overleftarrow{P}_t$ and the solutions will converge to arbitrary portions of the PF^A, instead of covering the whole surface. Thus Pareto-based fitness assignments are usually applied in conjuction with density measures, which can be incorporated in two ways. The first approach, commonly known as fitness sharing, aggregates the Pareto-based fitness and some form of density measure to form a scalar fitness. In this case, the aggregation function must be carefully constructed to maintain a balance between $\overleftarrow{P}_t$ and $\overleftarrow{P}_n$. This approach has been applied by successfully in works such as [61,140,228]. The second approach adopts a two stage process where comparison between solutions is conducted based on Pareto-fitness before density measure is used. Note that this indirectly assigns higher priority levels to proximity. Another interesting consequence is that $\overleftarrow{P}_n$ will be higher in the initial stages of the evolution. On the other hand, when the solutions begin to converge to the PF*, the influence of $\overleftarrow{P}_t$ becomes more dominant as most of the solutions are equally fit. This approach is used in algorithms such as PAES, NSGAII and IMOEA.

However, Fonseca and Fleming [63] highlighted that Pareto-based assignment may not be able to produce sufficient selection pressure in high-dimensional problems and it has been shown empirically that Pareto-based MOEAs do not scale well with respect to the number of objectives in [92, 116]. To understand this phenomenon, let us consider a M-objective problem where M>>2. Under the definition of Pareto dominance, as long as a solution has one objective value that is better than another solution, never mind the degree of superiority, it is still considered to be nondominated even if it grossly inferior in the other M-1 objectives. Intuitively, the number of nondominated solutions in the evolving population grows with the number of objective resulting in the lost of selection pressure.

To this end, some researchers have sought to relax the definition of Pareto-optimality. Ikeda *et al* [97] proposed the α-dominance scheme which considers the contribution of all the weighted difference between the respective objectives of any two solutions under comparison to prevent the above situation from occuring. Laumanns *et al* [132] suggested an ϵ-dominance scheme which has the interesting property of ensuring convergence and diversity. In this scheme, an individual dominates another individual only if it offers an improvement in all aspects of the problem by a pre-defined factor of ϵ. A significant difference between α-dominance and ϵ-dominance is that a solution that strongly dominates another solution also α-dominates that solution while this relationship is not always valid for the latter scheme. Another interesting alternative in the form of fuzzy Pareto-optimality is presented by Farina and Amato [53] to take into account the number and size of improved objective values.

Aggregation-based Fitness Assignment: Aggregation of the objectives into a single scalar is perhaps the simplest approach to generate PF^A. Interestingly, unlike the Pareto-based approach, aggregation-based fitness induces $\overleftarrow{P}_u$ directly. However, aggregation is usually associated with several limitations such as its sensitivity to PF^* shape and the lack of control on the direction of $\overleftarrow{P}_u$. This results in the contrasting lack of interest paid by evolutionary MO optimization (EMOO) researchers as compared to Pareto-based techniques. Ironically, the failure of Pareto-based MOEAs in high-dimensional objective space may well turn the attention towards the use of aggregation-based fitness assignment in MOEAs.

The multi-objective genetic local search (MOGLS) [102–105] is a well-known instance of aggregation-based MOEA that has been demonstrated to be capable of evolving uniformly distributed and diverse PF^A. Different search trajectories are generated during the evolution through the use of random weights in [102, 103] while Jaszkiewicz [104, 105] applied different instances of predefined utility functions. Jin *et al* investigated two different approaches in [110]. In the first method, each individual is assigned its own weights that will be regenerated every generation while the second method periodically change the weights along the evolutionary process. The most significant result of this work is that both methods are able to converge on concave PF^* empirically, which is against conventional wisdom on the limitations of aggregation. According to [111], this is because the aggregation-based MOEA will transverse the entire Pareto front regardless of PF^* shape and the archive plays a significant role in retaining the nondominated solutions found.

Instead of performing the aggregation of objective values, Hughes [93, 94] suggested the aggregation of individual performance with respect to a set of predetermined target vectors. In this approach, individuals are ranked according to their relative performance in an ascending order for each target. These ranks are then sorted and stored in a matrix such that is may be used to rank the population, with the most fit being the solution that achieves the best scores most often. It has been shown to outperform nondominated sorting applied in NSGAII for high-dimensional MO problems [93].

At this point of time, it seems that Pareto-based fitness are more effective in low-dimensional MO problems while aggregation-based fitness has an edge with increasing number of objectives. Naturally, some researchers have attempted to marry both methods together. For example, Turkcan and Akturk [209] proposed an hybrid MO fitness assignment method which assigns a nondominated rank that is normalized by niche count and an aggregation of weighted objective values. On the other hand, Pareto-based fitness and aggregation-based fitness are used independently in various stages of the evolutionary process in [54, 149].

Indicator-based Fitness Assignment: Since the performance of MOEAs are assessed and compared using performance indicators, it is therefore desirable to maximize algorithmic performance according to these measures. Fleischer [58] is probably the first to suggest that MO performance indicators can be used to guide the evolutionary process and recasted the MO problem as a SO problem that maximizes the hypervolume metric of the discovered PF^A. In [50], hypervolume is used as the selection criteria to remove the worst individuals in the worst-ranked PF^* after nondominated sorting to maintain a fixed population size. Zitzler and Kunzli [226] took a step further and applied binary indicators directly to determine the relative fitness of the evolving individuals. The utility of indicator-based fitness has also been investigated in [11]. While no clear guidelines on the choice of metrics exist at this time, it is clear that the selected measure must be able to provide an indication of solution quality in the aspects of diversity and convergence in order to exert the $\overleftarrow{\overline{P}}_u$.

Diversity Preservation

Density Assessment: A basic component of diversity preservation strategies is density assessment. Density assessment evaluates the density at different sub-divisions in a feature space, which may be in the parameter or objective domain, before any action is taken to influence the survival rate of the solution points [117]. Depending on the manner in which solution density is measured, the different density assessment techniques can be broadly categorized under 1) Distance-based, 2) Grid-based, and 3) Distribution-based. One of the basic issues to be examined is whether density assessment should be computed in the decision space or objective space. Horn and Nafpliotis [90] stated that density assessment should be conducted in the feature space where the decision-maker is most concerned about its distribution. Since we are interested in obtaining a well-distributed and diverse PF^{A}, most works reported in the EMOO literature applied density assessment in the objective space. There are also researchers who performed density assessment in the decision space [188] as well as in both objective and decision spaces simultaneously [46, 87, 171]. In fact, there may be little correlation between diversity in the two feature spaces. Tan *et al* [196] pointed out that it essentially depends on what is desired for the underlying problem.

Distance-based assessments is based on the relative distance between individuals in the feature space. Examples include niche sharing [63, 72, 90, 188], crowding [43], clustering [36, 230], lateral interference [118], Pareto potential regions [83] and *k-th* nearest neighbor [1, 228]. Niche sharing or niching is by far the most popular distance-based approach.

Niching is originally proposed by Goldberg [77] to promote population distribution to prevent genetic drift as well as to search for possible multiple peaks in SO optimization. The main limitation of this method is that its performance is sensitive to the setting of niche radius. Fonseca and Fleming [63] gave some bounding guidelines of appropriate niche radius values for MO problems when the number of individuals in the population and the minimum/maximum values in each objective dimension are given. However, such information are often not known *a prior* in many real-world problems. Tan *et al* [197] presented a dynamic sharing scheme where the niche radius is computed online based on the evolved tradeoffs.

The *k-th* nearest neighbor is another approach which requires the specification of an external parameter. Zitzler *et al* [228] adopted k as the square root of the total population size based on some rule-of-the-thumb used in statistical density estimation. In [1, 176], average Euclidean distance between the two nearest solutions is used as the measure of density. Like niching, this approach is sensitive to the setting of the external parameter, which in this case is k. The *k-th* nearest neighbor can also be misleading in situations where all the nearest neighbors are located in a similar region of the feature space. In certain sense, the nearest neighbor is similar to the method of crowding. However, crowding do not have such bias since it is based on the average distance of the two points on either side of current point along each dimension of the feature space.

Crowding, clustering and lateral interference are instances of distance-based assessments that are not influenced any external parameters. Nonetheless, distance-based assessment schemes are susceptible to scaling issues and their effectiveness are limited by the presence of noncommensurable objectives.

Grid-based assessment is probably the most popular approach after niching and it can be found in [29,30,32,127,140]. In this approach, the feature space is divided into a predetermined number of cells along each dimension and distribution density within a particular cell has direct relation to the number of individuals residing within that cell location. Contrary to distance-based assessments methods which include methods that are very different, both conceptually and in implementation, the main difference among the various implementation of this approach, if any, lies in the way in which the cells and individuals are located and referenced. For example, the cell location of an individual in PAES and PESA is found using recursive subdivision. However, in [140], the location of each cell center is stored and the cell associated with an individual is found by searching for the nearest cell center. The primary advantage of grid-based assessment is that it is not affected by the presence of noncommensurable objectives. However, this technique depends heavily on the number of cells in the feature space containing the population and it works well if knowledge of the geometry of the PF* is known. Furthermore, it's computational requirements are considerably more than distance-based assessments.

Distribution-based assessment is rather different from distance-based and grid-based methods because distribution density is based on the probability density of the individuals.

The probability density is used directly in [16] to identify least crowded regions of the PF^A. It has also been used to compute the entropy as a means to quantify the information contributed by each individual to the PF^A in [35,120,192]. Like grid-based methods, it is not affected by noncommensurable objectives. The tradeoff is that it can be computationally intensive because it involves the estimation of probability density of the individuals. On the other hand, the computational effort is a linear function of population size which is advantageous for large population sizes. While some distribution-based methods require external parameter setting such as the window width in Parzen window estimation [72], there exist an abundance of guidelines in the literature.

Finally, an empirical investigation is conducted in [117] on the effectiveness of the various density assessment methods in dealing with convex, nonconvex and line distributions. In general, the study shows that all techniques under investigation are able to improve distribution quality in the sense of uniformity. But the findings also suggest that it is not possible to ascertain which method is better for which type of problem distribution because of the interactions between density assessment and genetic selection.

Encouraging Density Growth: Apart from inducing appropriate $\overleftarrow{P}_t$ and $\overleftarrow{P}_u$ to generate new and diverse solutions, other means of encouraging diversity growth can also be found in the literature. For instance, in [206], Toffolo and Benini applied diversity as an objective to be optimized. Specifically, the MO problem is transformed into a two-objective problem with genetic diversity as one of the objectives and the other objective being the ranks with respect to the objectives of the original MO problem.

Mating restriction is another alternative approach and it is extended from SOEA where it is originally intended to promote diversity in the population. Mating restriction has been applied in [63, 82, 101] and it works by preventing similar Parents from participating in the recombination process together in order to avoid the formation of lethal individuals. However, contrary results on the effectiveness of mating restriction in promoting diversity has been reported in [100]. In particular, Ishibuchi and Shibata [100] noted that mating restriction improves convergence at the expense of solution set diversity.

Diversity can also be encouraged through the simultaneous evolution of multiple isolation subpopulations. In [42,149,178], each subpopulation is guided towards a particular region of PF*. Okuda *et al* [153] assigned one subpopulation for each objective and used an additional subpopulation as a normal MOEA solving for the MO problem. The best individuals from the SOEA subpopulations are migrated to the MOEA subpopulation.

Elitism

The use of the elitist strategy is conceptualized by De Jong in [44] to preserve the best individuals found to prevent the lost of good individuals due to the stochastic nature of the evolutionary process in SOEA. When appropriate individuals are reinserted or retained in the evolving population, elitism can improve convergence greatly, although it maybe achieved at the risk of premature convergence. Zitzler [231] is probably the first to introduce elitism into MOEAs, sparking off the design trend of a new generation of MOEAs [28]. Elitism can be considered as an indispensable component of MOEA, having being shown to be a theoretical necessity for MOEA convergence [123, 173, 174].

Archiving: The first issue to be considered in the incorporation of elitism is the storage or archiving of elitist solutions. Archiving usually involves an external population or archive as the repository and this process is much more complex than in SOEAs since we are now contenting with a set of Pareto-optimal solutions instead of a single solution. However, the PF* is an infinite set which raises the natural question of *what should be maintained?*. Without any restriction on the archive size, the number of nondominated solutions can grow exceedingly large. Therefore, in the face of limited computing and memory resources in implementation, it is sometimes unwise to store all the nondominated or elitist solutions found.

Most works enforce a bounded set of elitist solutions which requires a truncation process when the size of the elitist solutions exceeds a predetermined bound. This leads to the interesting question of *which* solution should be kept? Some works [43, 193, 228] maintains a fixed sized archive which updates dominated solutions as long as space is available while others store strictly nondominated solutions only [32, 127, 195, 196]. In either case, it is only natural to truncate the archive based on some form of density assessment discussed earlier when the number of elitist solutions exceeds the upper bound. However, other measures such as hypervolume [122] and relaxed forms of Pareto dominance have been applied as well [45, 156].

For bounded archiving, two implementations of truncation can be found in the literature, i.e., batch and recurrence mode. The truncation criteria will be based on the density assessment process described earlier. In the batch-mode, all solutions in the archive will undergo density assessment and the worst individuals are removed in a batch. On the other hand, in the recurrence mode, an iterative process of density assessment and truncation is repeated to the least promising solution from the archive until the desired size is achieved. While the recurrence-mode of truncation has higher capability to avoid the extinction of local individuals, which somehow leads to the discontinuity of the discovered Pareto front, compared to the batch-mode truncation, the recurrence-mode truncation often requires more computational effort.

The restriction on the number of archive solutions leads to two phenomena [55] which have a detrimental effect on the search process. The first is the shrinking PF^A phenomenon which results from the removal of extremal solutions and the subsequent failure to rediscover them. In the second phenomenon, nondominated solutions in the archive are replaced by least crowded individuals. In the subsequent generations, new individuals that would have been dominated by the removed solutions are updated into the archive only to be replaced solutions dominating them. Repeated cycles of this process is known as the oscillating PF^A. The alternative and simplest approach is, of course, to store all the nondominated solutions found [51, 56, 150, 159]. One potential problem is the computational cost involved with the pairwise comparison between a new individual and archived solution. To this end, more efficient data-structures have been proposed in [55].

Reinsertion: The next issue to be considered is the introduction of elitist solution into the evolving population. Empirical investigations are also conducted in [133, 159] and the results demonstrate the advantages of elitism in improving convergence.

One problem faced is the "exploration-exploitation" dilemma; a higher degree of exploitation attained through the reintroduction of elitist solutions leads to the lost of diversity while too much exploration leads to slow convergence rates. The consequences of the lack of necessary diversity to fuel the evolutionary process is a PF* that fails to span the entire PF* uniformly and, in the worst case, premature convergence to local optimal solutions.

Elitist schemes that sought to balance the tradeoffs between exploration and exploitation have been proposed recently. Bosman and Thierens [15] highlighted the importance of improving diversity through elitism and presented a general framework for MOEAs which allows designers to control the balancing diversity and proximity exploration. Designing an elitist scheme along the same lines, Tan *et al* [196] proposed an enhanced exploration strategy in which the ratio of solutions selected based on ranking and diversity is adapted based on an online performance measure. Solutions selected on the basis on rank are subjected to normal genetic operators while those selected based on niche count undergo local search to improve solution distribution. In [41], controlled elitism is explored in NSGAII where the number of individuals from each nondominated front is fixed by a user-defined parameter. Furthermore, each front is allowed to have an exponentially reducing number of solutions.

1.3.3 Benchmark Problems

Benchmark problems are used to reveal the capabilities, important characteristics and possible pitfalls of the algorithm under validation. In the context of MO optimization, these test functions must pose sufficient difficulty to impede MOEAs search for Pareto optimal solutions. Deb [37] has identified several characteristics that may challenge MOEAs ability

Table 1.1: Definition of ZDT Test Functions

Problem	Definition
ZDT1	$f_1(x_1) = x_1,$ $\frac{g(x_2,...x_m)=1+9\sum_{i=2}^{m}\big(x_i}{(m-1)\big)},$ $h(f_1,g) = 1 - \sqrt{\frac{f_1}{g}}$ where $m = 30,\ x_i \in [0,1]$
ZDT4	$f_1(x_1) = x_1,$ $g(x_2,...x_m) = 1 + 10(m-1) + \sum_{i=2}^{m}(x_i^2 - 10\cos(4\pi x_i)),$ $h(f_1,g) = 1 - \sqrt{\frac{f_1}{g}}$ where $m = 10,\ x_1 \in [0,1],\ -1 \leq x_i < 1,\ \forall i = 2,...,10$
ZDT6	$f_1(x_1) = 1 - exp(-4x_1) \cdot \sin^6(6\pi x_1),$ $g(x_2,...x_m) = 1 + 9 \cdot (\frac{\sum_{i=2}^{m} x_i}{m-1})^{0.25},$ $h(f_1,g) = 1 - (\frac{f_1}{g})^2$ where $m = 10,\ \ x_i \in [0,1]$

to converge and maintain population diversity. Multi-modality is one of the characteristics that hinder convergence. Convexity, discontinuity and non-uniformity of the PF may prevent the MOEA from finding a diverse set of solution.

The problems of ZDT1, ZDT4, ZDT6, DTLZ3, FON and KUR are selected to validate the effectiveness of multi-objective optimization techniques in converging and maintaining a diverse Pareto solution set in this work. This set of test problems are characterized by the different features mentioned above and should be a good test suite for a fair comparison of different multi-objective algorithms. Many researchers, such as [32, 43, 199, 215, 229], have used these problems in the validations of their algorithms.

The test problems of ZDT1 through ZDT6 are constructed by Zitzler *et al* [229] based on the guideline described by Deb [37]. Formally, the ZDT test problems have the following functional structure.

$$\begin{aligned} \min f_1(\vec{x_{d1}}) &= x_1 \\ \min f_2(\vec{x_{d2}}) &= g(\vec{x_{d2}}) \cdot h(f_1, g) \end{aligned} \tag{1.6}$$

where $\vec{x_{d1}}, \vec{x_{d2}} \in \vec{x}$, and the g and h functions control the problem difficulty and the shape of the Pareto front respectively. By having independent functions relating to convergence and diversity, this framework facilitates the incorporation of various problem features to construct different test problems. Table. 1.1 summaries the definition and features of the various ZDT test functions.

DTLZ3 belongs to the DTLZ test suite proposed by Deb *et al* in [43] which is different from most existing MO test problems in the sense these test problems are scalable in the number of objectives. In the light of recent studies [92, 116] reporting on MOEA's apparent inability to scale up its performance with high dimensional space, DTLZ3 will undoubtably be useful in the investigation of MOEA capability to handle high dimensional objective spaces. DTLZ3 is also characterized by the presence of multiple local fronts. The definitions of DTLZ3 are given below,

$$\begin{aligned} \min f_1(\vec{x}) &= (1 + g(\vec{x}_M)) \cdot \cos(0.5\pi x_1) \cdots \cos(0.5\pi x_{M-1}) \\ \min f_2(\vec{x}) &= (1 + g(\vec{x}_M)) \cdot \cos(0.5\pi x_1) \cdots \sin(0.5\pi x_{M-1}) \\ &\vdots \\ \min f_M(\vec{x}) &= (1 + g(\vec{x}_M)) \cdot \sin(0.5\pi x_1) \\ \min g(\vec{x}_M) &= 100\Big\{|\vec{x}_M + \textstyle\sum_{x_i \in \vec{x}_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\Big\} \end{aligned} \tag{1.7}$$

where $M = 5$, $\vec{x}_M = \{x_M, ..., x_{M+9}\}$, $x_i \in [0, 1]$

FON [61] is a two-objective minimization test problem that has been widely used in the literature. Besides having a nonconvex Pareto front, there are high interactions between decision variables and this problem has a large and nonlinear tradeoff curve that is suitable for challenging an algorithms ability to find and maintain the entire Pareto front uniformly.

$$\begin{aligned} f_1(x_1,...,x_8) &= 1-\exp[-\textstyle\sum_{i=1}^{8}(x_i-\frac{1}{\sqrt{8}})^2], \\ f_2(x_1,...,x_8) &= 1+\exp[-\textstyle\sum_{i=1}^{8}(x_i-\frac{1}{\sqrt{8}})^2], \end{aligned} \tag{1.8}$$

where $-2 \leq x_i < 2, \forall i = 1,2,...,8$

KUR [126] is characterized by an optimal Pareto front that is non-convex and disconnected, i.e., it contains three distinct disconnected regions on the final tradeoff. The decision variables correspond to the global tradeoff for KUR are difficult to be discovered, since they are disconnected in the decision variable space. Like FON, there are high interactions between the decision variables which will pose problems to the MOEAs.

$$\begin{aligned} f_1(x_1,x_2) &= \textstyle\sum_{i=1}^{2}[-10\exp(-0.2\sqrt{x_i^2+x_{i+1}^2})], \\ f_2(x_1,x_3) &= \textstyle\sum_{i=1}^{3}[|x_i|^{0.8}+5\cdot\sin(x_i^3)], \end{aligned} \tag{1.9}$$

where $x_i \in [-5,5]$.

1.3.4 Performance Metrics

Performance analysis of different MOEAs essentially boils down to the evaluation of the approximate Pareto front under the constraints of some computational budget. Then performance metrics or indicators play an important role as functions that return a scalar quantity, reflecting the quality of the scrutinized solution set with respect to some measure. In SO optimization, this quality comes in the form of the objective function. For MO optimization, however, quality can be defined in a variety of ways, for example, the uniformity of solutions, the dominance relationship between two solution sets and the closeness to the Pareto-optimal front.

There has been increasing concerns on the choice of performance metrics. To this end, Knowles and Corne [124] and Zitzler *et al* [227] have discussed at length, the suitability and limitations of various performance metrics. Comparative studies performed by researchers such as Jaszkiewicz [105], Deb *et al* [43], Tan *et al* [196], Veldhuizen and Lamont [215], and etc., made use of a suite of unary performance metrics pertinent to the optimization goals of proximity, diversity and distribution. The metrics used in this work are described below. Appropriate performance indicators for measuring uncertainties will be discussed in the relevant chapters.

Proximity Indicator: The metric of generational distance (GD) gives a good indication of the gap between the PF* and the evolved PF^A. Mathematically, the metric is a function of individual distance given as,

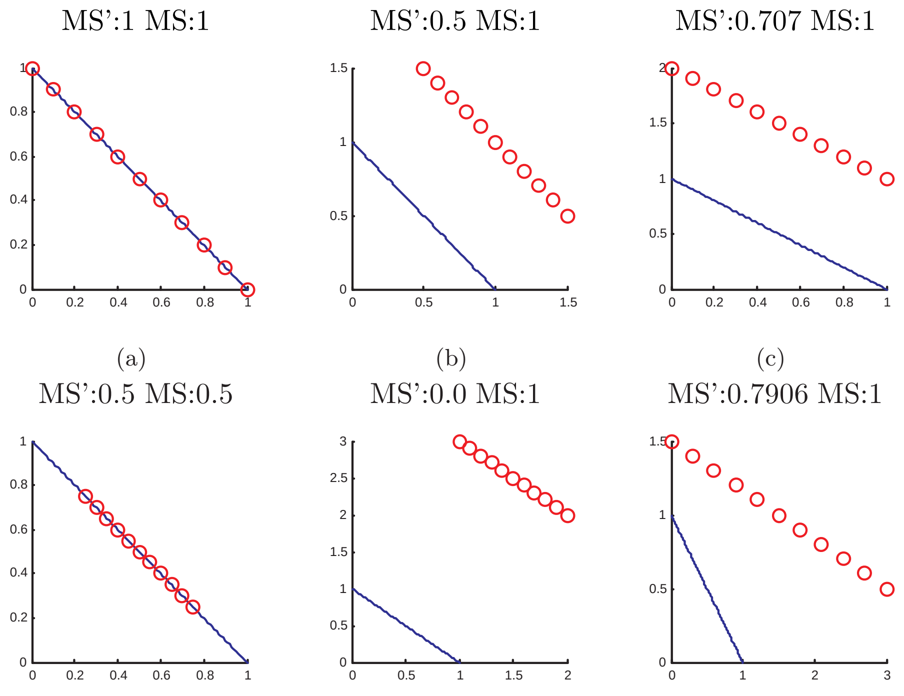
$$\mathrm{GD} = \frac{1}{n_{PF}} \cdot \Big(n_{PF} \sum_{i=1}^{n_{PF}} d_i^2 \Big)^{\frac{1}{2}} \tag{1.10}$$

where $n_{PF} = |\mathrm{PF}^A|$, d_i is the Euclidean distance (in objective space) between the i-th member of PF^A and the nearest member of PF*. Intuitively, a low value of GD is desirable, which reflects a small deviation between the evolved and the true Pareto front. However, this metric gives no indication of diversity achieved by the various algorithms. In order to evaluate the true ability of the algorithm, GD has to be complemented by diversity indicators

Diversity Indicator: One of the primary concerns regarding the use of unary diversity indicator is that a good measure of diversity is meaningless if the approximate Pareto front is far away from the ideal tradeoffs. Taking into account these concerns, we propose a simple modification of maximum spread (MS) to measure how well the true Pareto front is covered by the discovered Pareto front

$$\mathrm{MS'} = \Big\{ \frac{1}{M} \sum_{i=1}^{M} \Big(\frac{\min[\overline{f}_i^A, \overline{f}_i^*] - \max[\underline{f}_i^A, \underline{f}_i^*]}{\overline{f}_i^* - \underline{f}_i^*} \Big)^2 \Big\}^{\frac{1}{2}} \tag{1.11}$$

where $\overline{f}_i^A$ and $\underline{f}_i^A$ is the maximum and minimum of the i-th objective in PF^A respectively;



(a) (b) (c)

(d) (e) (f)

Figure 1.5: Different Characteristics exhibited by MS' and MS. MS' takes into account the proximity to the ideal front as well.

$\overline{f}_i^*$ and $\underline{f}_i^*$ is the maximum and minimum of the i-th objective in PF^* respectively. The greater the MS' is, the more area of PF* is covered by the PF^A. The modified metric is illustrated in Figure. 1.5.

Distribution Indicator: Further, the uniformity among the distributed points or individuals is also an important issue in order to ensure consistent transition among the solution points when searching for the most suitable solution from the best possible compromise. The metric of spacing [181] gives an indication of how evenly the solutions are distributed along the discovered front. It is defined as,

$$\mathrm{S} = \frac{1}{\bar{d}'} \cdot \left(\frac{1}{n_{PF}} \cdot \sum_{i=1}^{n_{PF}} (d_i' - \bar{d}')^2 \right)^{\frac{1}{2}} \tag{1.12}$$

$$\bar{d}' = \frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d'_i$$

where d'_i is the Euclidean distance (in the objective domain) between the i-th member and its nearest member in PF^A.

General Quality Indicator: By taking into account performance in diversity and proximity, the metric of hypervolume (HV) provides a general quality measure of the solution set. In order to calculate a normalized value and eliminate bias, Veldhuizen and Lamont [215] expressed the metric of HV as a ratio between the HV of PF^A and PF*,

$$\mathrm{HVR} = \frac{\mathrm{HVR}(\mathrm{PF}^A)}{\mathrm{HVR}(\mathrm{PF}^*)} \tag{1.13}$$

$$\mathrm{HV} = \mathrm{volume} \bigcup_{i=1}^{n_{PF}} v_i \tag{1.14}$$

. Mathematically, for each member $\vec{f}_i^A$ in the non-dominated set, a hypercube v_i is constructed with a reference point and the member $\vec{f}_i^A$ as the diagonal corners of the hypercube. The reference point can be found by constructing a vector of the worst objective function values.

Pareto Dominance Indicator: In [227], Zitzler *et al* showed that no combinations of unary performance metrics can provide a clear indication of whether an evolved set is better than another in the Pareto dominance sense. Therefore, an n-ary Pareto dominance indicator is proposed in this paper as a complement to the above metrics. Considering the different PF, $A_1, A_2, ..., A_n$ evolved by n algorithms, this metric measures the ratio of nondominated solutions that is contributed by a particular solution set A_i to the nondominated solution set provided by all solution sets. Mathematically, the nondominance ratio (NR) is given by,

$$\mathrm{NR}(A_1, A_2, ..., A_n) = \frac{|B \cap A_1|}{|B|}$$

$$B = \{b_i | \; \forall \; b_i \nexists a_j \in (A_1 \cup A_2 ... \cup A_n) \prec b_i\} \tag{1.15}$$

where A_1 is the solution set under evaluation.

1.4 Overview of This Work

MO optimization is a challenging research topic not only because it involves the simultaneous optimization of several complex objectives in the Pareto optimal sense, it also requires researchers to address many issues that are unique to MO problems. Advances made in the field of EMOO is the result of two decades worth of intense research examining topics such as fitness assignment [53, 140], diversity preservation [117], balance between exploration and exploitation [15], and elitism [134].

The primary motivation of this work is to provide a comprehensive treatment on the design and application of MOEAs for MO optimization in the presence of uncertainties. As mentioned right at the start of this chapter, the difficulty of multiple criteria decision making (MCDM) is exacerbated by the fact that real world problems are not deterministic in nature. While it has been shown that MOEAs are powerful and efficient optimizers of static MO problems, their performance are rarely examined in the presence of uncertainties and it is unlikely that the state-of-the-arts are capable of handling the demands that the task entails.

This work is organized into three parts, with each part addressing a different form of uncertainty. The first part comprising of Chapters 2-4 focuses on the optimization of noisy MO problems. Unlike existing studies of multi-objective evolutionary algorithms (MOEAs) [198, 229], Chapter 2 examines the performance of MOEAs in noisy environments. Based on the analysis of empirical results, three noise-handling features are then proposed in Chapter 3, including an experiential learning directed perturbation operator that adapts the magnitude and direction of variation according to past experiences for fast convergence, a gene adaptation selection strategy that helps the evolutionary search in escaping from local optima or premature convergence, and a possibilistic archiving model based on the concept of possibility and necessity measures to deal with problem of uncertainties. Chapter 4 considers the design of artificial neural networks as a specific instance of noisy problem. For this problem, the synaptic weights of the neural network must be optimized to reduce the effects of noise. Therefore, a new hybrid multi-objective evolutionary approach which includes the features of a variable length representation that allow for easy adaptation of neural networks structures, an architectural recombination procedure based on singular-value decomposition that adapts the number of necessary hidden neurons, and a micro-hybrid genetic algorithm with an adaptive local search intensity scheme is developed.

The second part starts with a survey of existing works for dynamic multi-objective optimization. A formal categorization of dynamic MO test functions and the requirements of performance indicators for assessment of dynamic MOEAs are also provided in Chapter 5. Chapter 6 introduces a new coevolutionary paradigm that incorporates both competitive and cooperative mechanisms observed in nature to solve MO optimization problems and to track the Pareto front in a dynamic environment. The main idea of competitive-cooperation coevolution is to allow the decomposition process of the optimization problem to adapt and emerge rather than being hand designed and fixed at the start of the evolutionary optimization process. In particular, each species subpopulation will compete to represent a particular subcomponent of the MO problem while the eventual winners will cooperate to evolve the better solutions. Through this iterative process of competition and cooperation, the various subcomponents are optimized by different species subpopulations based on the optimization requirements of that particular time instant, enabling the algorithm to handle both the static and dynamic MO problems.

The third and final part concentrates on the issues of robust MO optimization. In particular, the suitability of existing robust test problems for MO optimization is examined and a set of guidelines for the construction of robust MO test problems is presented. The fundamental component of the robust test problems is a Gaussian landscape generator that facilitates the specification of robust optimization-specific features such as noise-induced solution space, fitness landscape and decision space variation. This generator is developed with the purpose of generating noise-sensitive landscapes in conjuction with existing MO test problems, and due to its independent nature, it can be used to generate robust single objective test problems as well. Subsequently, a robust MO test suite is built upon the ZDT framework. In addition, the vehicle routing problem with stochastic demand (VRPSD) is presented a practical example of robust combinatorial MO optimization problems.

1.5 Conclusion

In this chapter, we have covered the necessary concepts and definitions of MO optimization and uncertainties to appreciate this work. This chapter also presented an introduction to MOEAs, with a general framework which illustrates the basic design issues of the state-of-the-arts. Subsequently, a survey of the state-of-the-arts based on the basic MOEA components of fitness assignment, diversity maintenance and elitism is presented to highlight the development trends of multi-objective evolutionary techniques. Finally, the overview of this work is presented.

Chapter 2

Noisy Evolutionary Multi-objective Optimization

Many real-world applications are characterized by the disruptive presence of noise which involves the consideration of different issues altogether. Detrimental impact of noise observed by Beyer [13] includes the reduction of the convergence rate and pre-mature convergence to sub-optimal solutions. To address the issue of noisy fitness problems, a number of studies concerning evolutionary SO optimization in noisy environments have been reported [6, 8, 12, 18, 75, 151, 166, 172, 175, 191]. In contrast, the issue of noise-handling in EMOO has not been studied in literature until recently [21, 94, 186]. This chapter examines the impact of noise on EMOO.

2.1 Noisy Optimization Problems

Noise occur naturally in real-world applications and it stems from several sources, resulting in different objective values for the same set of design parameters. At a very general level, noise can be classified into aleatory noise or epistemic noise. Aleatory noise are random noise such as sensor measurement errors which can be modeled by some random number with known probability distribution. Detrimental effects of aleatory noise can be limited by means of averaging. On the other hand, epistemic noise are due to the lack of sufficient knowledge about the problem and, in many cases, the consequence of the tradeoff between computational speed and accuracy. Examples of epistemic noise includes the incomplete simulations of computational models.

Noise changes the way in which the decision-making module, be it the selection mechanism of the MOEA or the human, perceives a solution. A bad solution may be "enhanced" by noise to appear good and, vice versa, a good solution may be perceived as bad due to the influence of noise. Noisy MO problems are certainly much more complex than SO problems due to the fact that MO optimization is partially-ordered. In fact, a strongly dominated solution only needs to be "improved" by noise along one aspect of the MO problem to become a nondominated solution!

Intuitively, the optimization process of noisy problems is be greatly influenced by the noise model adopted and the level of noise intensity. Several studies concerning evolutionary optimization in noisy environments, the vast majority of them conducted in the domain of SO optimization, have been reported [6,8,12,13,18,75,151,166,172,175]. Most of these investigations are done on the basis of Gaussian noise. Notable exceptions include the investigation conducted by Arnold and Beyer [7] which revealed significant differences between the influence of Gaussian, Cauchy and χ^2 distributed noise on the performance of $(\mu/\mu, \lambda)$ ES. In the context of MO optimization, Teich [203] considers a uniform noise model while Buche *et al* [21] incorporates the effects of outliers on the optimization process.

The common practice is to incorporate the selected noise model as an additive perturbation to the original test functions. Unlike the study of dynamic optimization problems, there is no specific test problems or test suites for the analysis of noise impact on evolutionary optimization. However, it should be noted that the different problem features will determine the extent and effect that noise has on the optimization process. For instance, we can expect problems with strong parameter dependencies and small isolated PF* to be more susceptible to noise as compared to those without these features. On the other hand, it has been reported that noise has a smoothing effect on the fitness landscape which allows

Table 2.1: Summary of MO test problems extended for noise analysis

Literature	Test Problems
Basseur and Zitzler [11]	ZDT1, ZDT6, DTLZ2, KUR, COMET, and QV
Buche *et al* [21]	BSDK1-BSDK6 [37]
Bui *et al* [22]	ZDT1-ZDT6
Fieldsend and Everson [57]	DTLZ2
Goh and Tan [72]	ZDT1, ZDT4, ZDT6, FON and KUR
Hughes [94]	MOP3 [216]
Singh [186]	S1 [37]

the EA to handle multi-modality with greater success [165]. The different test problems that have been extended for noise analysis are summarized in Table 2.1.

The same guidelines for the selection of test problems or the construction of test suites in deterministic MO optimization are applicable to noisy MO optimization as well. In fact, applying a suite of MO test problems with different features will allow us to examine the influence of noise on these features.

2.2 Performance Metrics for Noisy MO Optimization

Like deterministic MO optimization, the optimization goal of noisy MO optimization is to find a near-optimal, diverse and uniformly distributed PF^A. To be precise, we are concerned about how good the PF^A truly is, and not how it is perceived since it is the true objective values that matters during implementation.

Therefore, performance metrics proposed for deterministic MO optimization can be used directly to assess the performance of MOEAs in the presence of noise. The only difference between deterministic and noisy MO optimization is that the objectives are perturbed by noise and its' true values not be known. In this case, re-evaluation can be employed to compute the effective objective values before the results are assessed.

Visualization of the evolved PF^A is used in [22, 57, 94] to demonstrate the effectiveness of the proposed methods. Bui *et al* [22] employed GD and attainment surface to provide a quantitative measure of algorithmic performance. Teich [203] used coverage to compare the quality between different true PF^A while Buche *et al* [21] measure that distance between PF^A and PF^* with respect to ten predefined points in the decision space.

However, noise-specific performance measures can also be found. Fieldsend and Everson [57] measured the Euclidean distance between true and noisy PF^A. Such a measurement provides an indication of how well the re-sampling technique performs. Basseur and Zitzler [11] proposed a probabilistic extension of the attainment function which provides the visualization of the probabilistic $k\%$ approximation set. This probabilistic $k\%$ approximation set is defined as the set of evolved solutions which dominates objective vectors that have been attained with a probability up to $k\%$.

In cases where the true PF^A can be determined, deterministic performance metrics should be used because it will provide a more accurate assessment of algorithmic performance. On a more practical side, it should be highlighted that the selection of final solution for implementation will be based on the corrupted PF^A. Therefore, apart from expending a certain amount of computational resource to reduce uncertainty, we can also expect noise-specific metrics such as the probabilistic attainment function to play a dual role in the evaluation of algorithmic performance as well as the solution selection.

2.3 Noise Handling Techniques

Although the EA is known to be inherently robust to low-level of noise due to its distributed nature of individuals and its non-reliance on gradient information, such a property may not extend well into EMOO that requires the evolutionary search to maintain a diverse set of non-dominated solutions uniformly distributing along the tradeoff. A few existing noise-handling techniques in EMOO include the approaches of periodic re-evaluation of archived solutions [21], probabilistic Pareto ranking [94], and extended averaging scheme [186] etc.

According to Jin and Branke [107], the different approaches for handling noise can be classified as 1) explicit averaging, 2) implicit averaging, and 3) selection modification. In explicit averaging, the objective values are averaged over a number of samples, H to compute the expected values. Increasing the number of samples reduces the degree uncertainty by a factor of $\sqrt{\mathrm{H}}$ at the expense of increasing computational cost. In implicit averaging, a large population is used instead of re-evaluating and averaging the objective values over a number of samples. When population size is large, there are many similar solutions and the solutions are implicitly averaged as the MOEA revisit promising regions repeatedly. In selection modification, the ranking and selection procedures are modified such that a solution is judged better than another solution only if it satisfies certain conditions. However, the two noise-handling heuristics, namely the experiential learning directed perturbation operator and the gene adaptation selection strategy, that will be presented in Chapter 3 do not fall under any of the three categories. Therefore, it would be appropriate to define an additional class of "heuristical" techniques for improving MOEA performance in noisy environments.

Explicit averaging: Existing EMOO approaches that applies explicit averaging include [186] and [22]. Using NSGAII [43] as the baseline algorithm, Singh extended the re-sampling method and probabilistic selection scheme [94] to solve a noisy groundwater remediation design problem. In this work, the technique of extended averaging is proposed to reduce the bias introduced by small sample size used in simple averaging. The extended averaging approach performs the averaging over all samples of identical individuals, which can be easily extended over different generations.

Similar to Singh, Bui *et al* [22] applied NSGAII as the baseline algorithm as three different approaches are investigated: NSGAII with probabilistic Pareto ranking and NSGAII with two variants of explicit averaging based NSGAII. In order to reduce the number of evaluations required, the mechanisms of fitness inheritance is also extended from noisy SO optimization in this work. In particular, a threshold that is calculated from the offspring's objective values and estimated variance is used to determine if the offspring will undergo multiple re-evaluation or adopt the mean fitness of the parents. The investigation concludes that the probabilistic approach will yield better results initially but explicit averaging will provide better results eventually.

In [11], Basseur and Zitzler studied the impact of noise on indicator-based MOEAs. A significant difference between this work and the previous two approaches is that, instead of expected objectives values, the expected ϵ-indicator values are sought. As the computation of the expected ϵ-indicator values is very intensive, three different approaches of estimating the expected indicator values are compared and analyzed.

Implicit averaging: The periodic re-evaluation and reinsertion of archived solutions can be classified under implicit averaging. Adapting from SPEA [230], Buche *et al* [21] proposed the noise tolerant strength Pareto evolutionary algorithm (NTSPEA) with an improved robust performance against noise. In particular, the elite preservation scheme is modified to reduce the detrimental effect of outliers, and every solution is assigned a lifetime that is dependent on the fraction of the archive it dominates. Any archive solutions with expiring lifetime is re-evaluated and added to the evolving population. In the subsequent archive updating, expired archive solutions will not be considered. However, the re-evaluated solutions will participate in the archiving process.

Selection modification: Currently, noise-handling schemes of the third category, particularly the use of probabilistic Pareto ranking scheme, is the most popular approach. Hughes [94] introduced a probabilistic approach for Pareto ranking scheme to account for the presence of uncertainties, and demonstrated the possible deficiencies of layered ranking approach adopted in NSGA [188]. In a similar vein, Hughes [95] extended the probabilistic ranking to handle constraints in noisy environments. In the proposed multi-objective probabilistic selection evolutionary algorithm (MOPSEA), elitism is implemented by replacing

Table 2.2: Parameter settings of the simulation study

Parameter	Settings
Chromosome	Binary coding; 15 bits per decision variable.
Population	Population size 100; Archive (or secondary population) size 100.
Selection	Binary tournament selection
Crossover operator	Uniform crossover
Crossover rate	
	0.8
Mutation operator	Bit-flip mutation
Mutation rate	$\frac{1}{chromosome\_length}$ for ZDT1, ZDT4 and ZDT6; $\frac{1}{bit\_number\_per\_variable}$ for FON and KUR;
Ranking scheme	Pareto ranking.
Diversity operator	Niche count with radius 0.01 in the normalized objective space.
Evaluation number	50,000

part of the evolving population with the best individuals.

While Hughes assumes that noise is normal distributed, Teich [203] considers a uniform noise distribution. Teich extended the SPEA algorithm in two ways 1) a probabilistic strength fitness is used and 2) the update of the external set is based on the a percentage of the best solutions and solutions with a fitness that is above a certain threshold. Building upon these works, Fieldsend and Everson [57] considered the computation of Probabilistic ranking under different conditions such as unknown noise properties, independent noise for each objectives and etc. Based on preliminary theoretical analysis, an online variance learning scheme is presented and validated empirically.

2.4 Empirical Results of Noise Impact

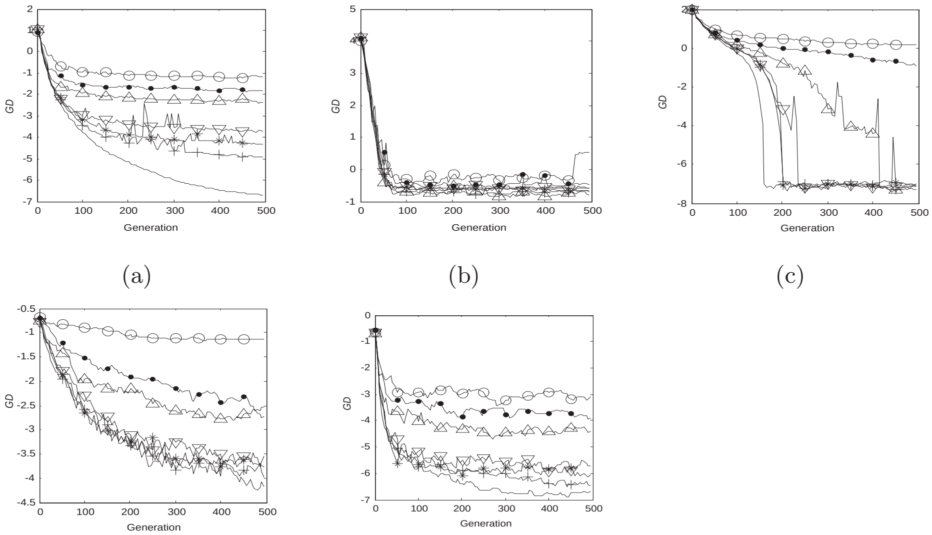
The evolutionary model adopted in this section is based on the conceptual framework in Chapter 1. The algorithm employs a fixed-size population and an archive to store nondominated solutions along the evolution. The archive is updated at each cycle, i.e., a candidate solution will be added to the archive if it is not dominated by any members in the archive. Likewise, any archive members dominated by this solution will be removed from the archive. When the predetermined archive size is reached, a recurrent truncation process based on niche count is used to eliminate the most crowded archive member. Although MOEAs have been implemented in different ways, most current state-of-the-art MOEAs include some form of elitism and diversity preservation mechanisms. In this paper, elitism is implemented by selecting individuals to a mating pool through a binary tournament selection of the combined archive and evolving population. Taking into account the study in [94], the selection criterion adopted in this paper is based on Pareto ranking scheme described in [63], and niche count [78] is used in the event of a tie. Note that the mechanism of niche sharing is used in the tournament selection and diversity maintenance in the archive.

Five benchmark problems, ZDT1, ZDT4, ZDT6, FON, and KUR are selected to examine the effectiveness of MOEAs in converging and maintaining a diverse set of non-dominated solutions under the influence of noise. In this study, noise is implemented as an additive normal distributed perturbation with zero mean. It is assumed that noise has a disruptive influence on the value of each individual in the objective space [8, 18, 94, 95, 175], i.e.,

$$\bar{f}(\vec{x}) = f(\vec{x}) + Normal(0, \sigma^2) \tag{2.1}$$

where σ^2 represents the level of noise present; $Normal$ denotes the normal distribution function; $\bar{f}$ and f denotes the objective function with and without the additive noise, respectively. Investigations of other noise models are left for future work.

Experiments are conducted at noise levels of $\sigma^2 = \{0.2\% \ , 0.5\%,\ 1\%, 5\% \ , 10\% \ , 15\%,$ $20\%\}$ in order to study the impact of noise on EMOO. Thirty independent simulation runs are performed for each of the test problems, and the values of the various parameter settings in the algorithm are tabulated in Table 2.2. The mutation rates adopted in this chapter are based on experimental studies that have been successfully applied in [195]. A random



(a) (b) (c)

(d) (e)

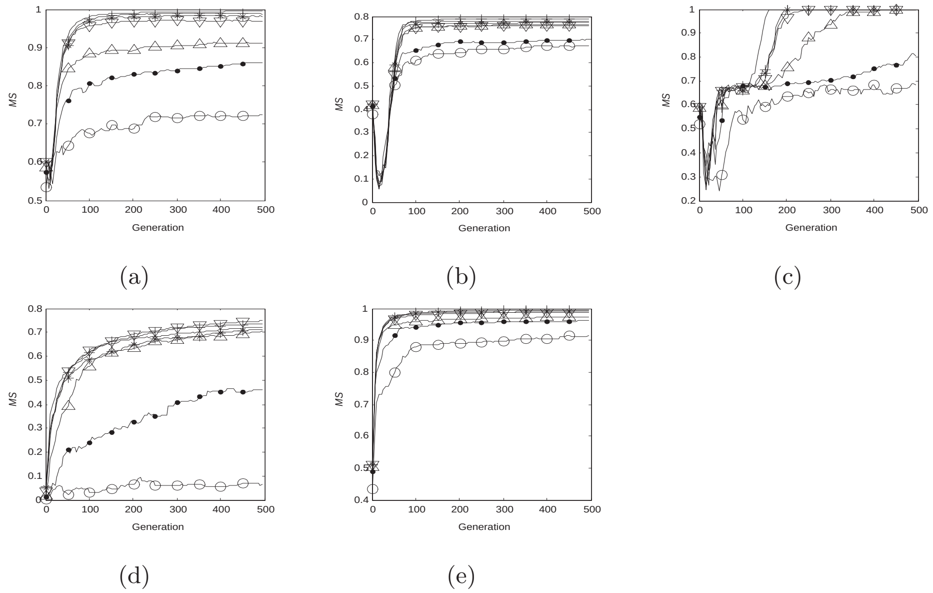
Figure 2.1: Performance trace of GD for (a) ZDT1, (b) ZDT4, (c) ZDT6, (d) FON, and (e) KUR under the influence of noise level at 0.0%, 0.2%, 0.5%, 1.0%, 5.0%, 10% and 20%

initial population is created for each of the simulation runs in every test problem. Unless otherwise specified, BMOEA refers to the baseline evolutionary algorithm.

2.4.1 General MOEA Behavior Under Different Noise Levels

The performance trace representing the mean of true values of GD and MS over 30 simulation runs for ZDT1, ZDT4, ZDT6, FON and KUR with different noise levels is showed in Figure 2.1 and Figure 2.2, respectively. The trace of GD and MS are sufficient to demonstrate the impact of noise on convergence and diversity.

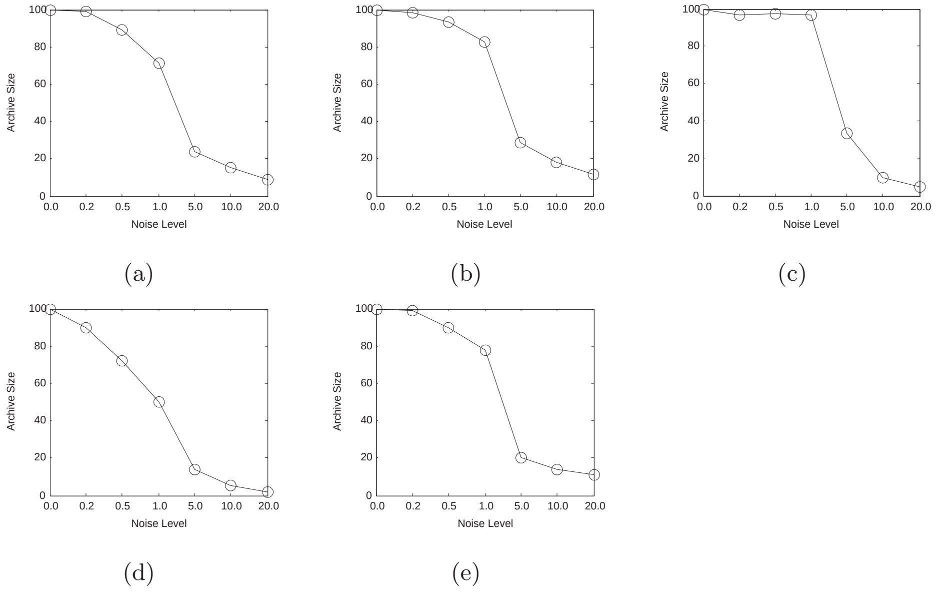
According to Nissen and Propach [151], population-based EAs are inherently robust in SO optimization under low level of noise. It can be seen from Figure 2.1 and Figure 2.2 that BMOEA is also capable of evolving satisfactory solutions in MO optimization under the influence of low noise level.



(a) (b) (c)

(d) (e)

Figure 2.2: Performance trace of MS for (a) ZDT1, (b) ZDT4, (c) ZDT6, (d) FON, and (e) KUR under the influence of noise level at 0.0%, 0.2%, 0.5%, 1.0%, 5.0%, 10% and 20%



(a) (b) (c)

(d) (e)

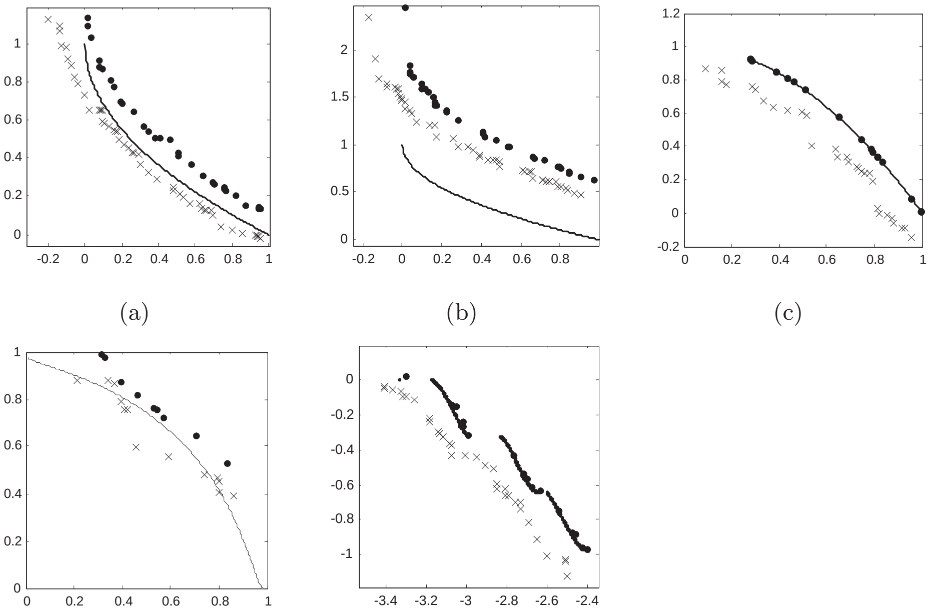
Figure 2.3: Number of non-dominated solutions found for (a) ZDT1, (b) ZDT4, (c) ZDT6, (d) FON, and (e) KUR under the influence of different noise levels.

In addition, the smoothing effect described by Rana *et al* [165] is also observable from the simulation results at low noise levels. An interesting finding is that the performance of BMOEA actually improves with the introduction of low noise levels. For instance, there is a high tendency to evolve better coverage of for FON that challenges the algorithm's ability to find and maintain the entire Pareto front uniformly. In the case of ZDT4 that challenges the algorithm's ability to deal with multi-modality, the presence of noise allows BMOEA to reduce the gap between PF* and PF^A. As in the study conducted in [165], smoothing is achieved without resampling, probably an indication of implicit averaging.

In contrast, it can be observed that BMOEA suffers from degenerate convergence properties and faces the problem of maintaining a diverse solution set under the influence of sufficiently high noise levels. Figure 2.3 shows that the archiving process deteriorates with increasing noise levels and fails to maintain a stable archive of non-dominated solutions. Further investigations revealed that good solutions are actually kept out of the archive by presence of noise enhanced solutions. The impact of noise is also observed to be more severe on problems such as FON, ZDT1, and ZDT6. In particular, the BMOEA is unable to improve in performance beyond the initial population for the problem of FON. Although the BMOEA fails to escape the local optima of ZDT4, its performance for ZDT4 appears to be insignificantly affected by noise.

2.4.2 MOEA Behavior in the Objective Space

It is important to analyze the behavior of MOEA in the objective space since how it performs depends on the degree at which noise affects the fitness landscape. A straight-forward approach to examine algorithmic behavior in the objective space is, of course, to look at how the MOEA will perceives the perturbed solutions. On a more critical note, if the significance of point on the erroneous selection of solution for implementation made earlier not been fully appreciated, a quick inspection on the relationship between the actual and perceived location of the final tradeoff illustrated in Figure 2.4 should do the trick. Notice



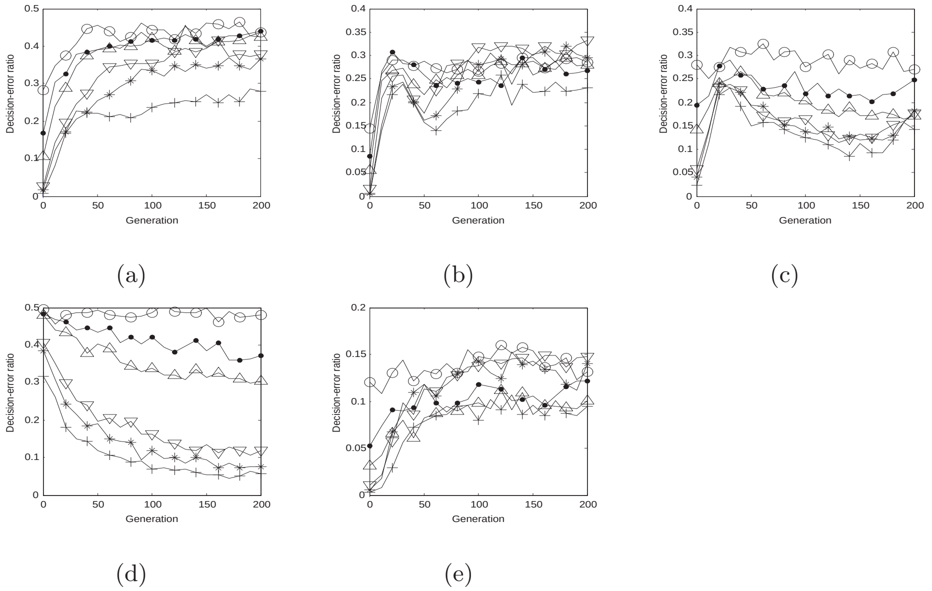
(a) (b) (c)

(d) (e)

Figure 2.4: The actual and corrupted location of the evolved tradeoff for (a) ZDT1, (b) ZDT4, (c) ZDT6, (d) FON, and (e) KUR under the influence of 5% noise. The solid line represents PF* while closed circles and crosses represent the actual and corrupted PF^A respectively.

how the perceived PF^A of FON in Figure 2.4(d) seems to imply the presence of a knee solution. The situation will actually get worse with increasing noise levels. Therefore, it is defintely worth the extra computational effort required to perform re-evaluation to obtain the expected objective values for the final PF* to get a better indication of solution quality.

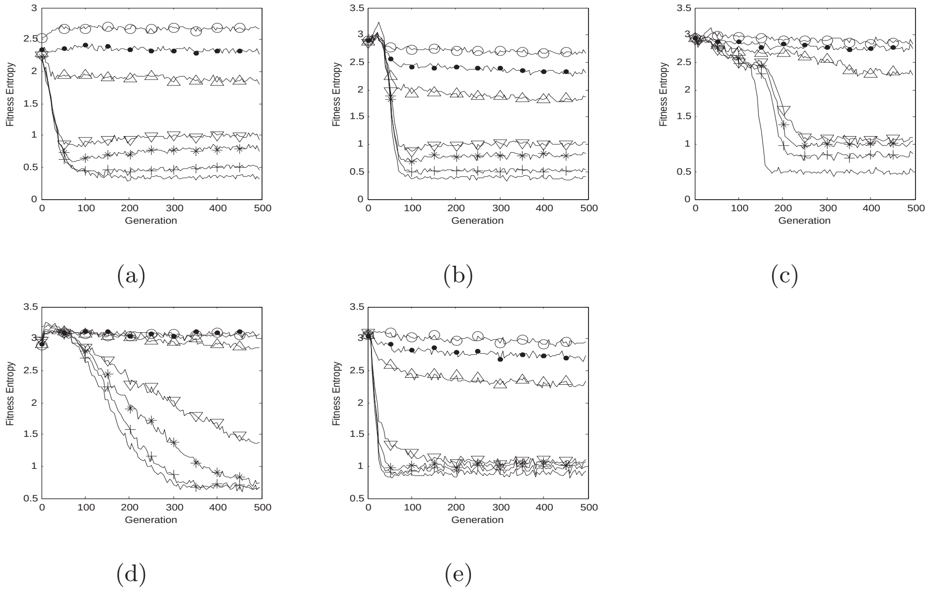
For a more in-depth analysis of how this affects the MOEA, we will first consider how the MOEA makes decisions based on the perturbed fitness values. Figure 2.5 shows the decision-error ratio against the number of generations for the five benchmark problems. The decision-error ratio is defined as the ratio between the number of incorrect decisions made for these operations and the total number of decisions made. From the Figure, we can observe two trends, the first of which, is the positive correlation between the decision-error



(a) (b) (c)

(d) (e)

Figure 2.5: Decision-error ratio for the various benchmark problems (a) ZDT1, (b) ZDT4, (c) ZDT6, (d) FON, and (e) KUR under the influence of different noise levels.



(a) (b) (c)

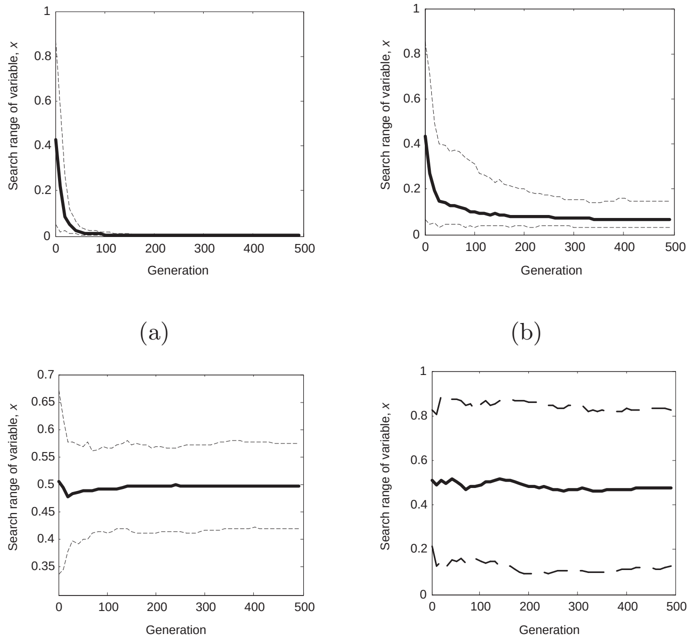
(d) (e)

Figure 2.6: The entropy value of individual fitness for (a) ZDT1, (b) ZDT4, (c) ZDT6, (d) FON, and (e) KUR under the influence of different noise levels.

ratio and noise. There is actually a special significance attached to the ratio at 0.5 because it provides an indication of the degree in which the evolutionary optimization process has degenerated into a random search process. Intuitively, the decision-error ratio should not exceed this 0.5 mark. In the event of such an interesting situation, then all we need to do is to "hard code" the MOEA to select the percieved inferior solution instead! True enough, with the exception of FON and ZDT1, the decision-error ratio did not exceed 0.4 even at $\sigma = 0.2$. This seems to imply that the algorithm should converge to PF* eventually. Unfortunately, this may not happen because a phenomenon which we term as the *curse of elitism* 1) the noise enhanced solutions in the archive are keeping out the good solutions (a point mentioned earlier) and 2) the optimization process is biased towards less promising areas due to the reinsertion of elitist solutions.

With the exception of FON, the second trend observed is that the decision-error ratio generally increases as PF^A approaches PF*, indicating a performance deterioration of optimization process along the evolution. For problems which demonstrates such characteristics, it is desirable to devise a mechanism that makes effective use of initial decisions to improve the convergence of EMOO. Comparing Figure 2.1, Figure 2.2 and Figure 2.5, it is apparent that the evolutionary optimization process stagnates as the error ratio saturates in the evolution. Such a degenerate convergence behavior of BMOEA is due to the unreliability of selection, ranking and archiving in the presence of noise. On the other hand, FON exhibits exactly the opposite behavior, with decision-error ratio improving with the number of generations. This is because the solutions of the initial population of BMOEA is always located around a small region about $f_1 = f_2 = 1$ due to the high parameter interaction between the decision variable, which amplifies the effects of noise.

Another way of analyzing the impact of noise on the objective space is to examine the distribution of Pareto ranks assigned to the noisy solutions. Shannon's entropy [184] is applied to quantify the amount of uncertainty in the ranking process and the entropy of solution Pareto rank in the evolving population for all noise levels is shown in Figure 2.6. It



(a)

(b)

(c)

(d)

Figure 2.7: Search range of an arbitrary decision variable for ZDT1 at (a) 0%, (b) 20% noise and FON at (c) 0% and (d) 20% noise. The thick line denotes the trace of the population mean along an arbitrary decision variable space, while the dashed line represents the bounds of the decision variable search range along the evolution.

can be seen that only simulation runs with no or low noise levels exhibit behavior of a stable optimization process with a converged fitness distribution. This is because the ranks of these individuals should also converged to better rank values as the evolving population converges to a better set of individuals in a low-noise environment. In contrast, simulation runs at high noise levels demonstrated high levels of uncertainty in the evolutionary optimization process.

2.4.3 MOEA Behavior in Decision Space

We have observed that the optimization process tends to converge to sub-optimal PF even though the decision error ratio is less than 0.4. So the natural question now is *what exactly* *is the effects of those right decisions?* Since the behavior of MOEA in the objective space seems to reveals little, we turn toward our attention towards how it behaves in the decision space. In order to examine algorithmic behavior in the decision space, Figure 2.7(a)-(d) illustrates trace of the search range for ZDT1 and FON along the evolution for an arbitrary selected decision variable at 0% and 20% noise levels.

By comparing the search range depicted in Figure 2.7(a)-(b) and Figure 2.7(c)-(d), it is clear that a disciplined evolutionary search is lacking at high noise level. Specifically, BMOEA is capable of narrowing down the search range for better evolutionary search optimization in a noise free environment. On the other hand, it can be observed that the mean location of individuals remains relatively the same despite a more diverse search space. This implies that the evolutionary process roughly knows where the promising regions are despite the presence of noise, most probably a consequence of the correct decision-making. More significant, it also means that the impact of keeping out the true nondominated solutions is greater than the reinsertion of inferior solutions.

2.5 Conclusion

In this chapter, extensive studies have been performed to examine the impact of noisy environments on EMOO, particularly for the population dynamics of fitness and diversity. It has been observed that the impact of noise on MOEA is different for the various benchmark problems, i.e., MOEA tends to evolve better solutions for some of the problems in the presence of low-level noise, while the evolutionary optimization process degenerates into a random search under increasing level of noise. Furthermore, it seems that the selection process is more reliable in the early stage of evolution and the statistical analysis of online optimization behavior in the decision space has revealed that the evolution defines a population distribution with a mean value that remains relatively invariant in the decision space despite the different environmental conditions.

Chapter 3

Noise Handling in Evolutionary Multi-objective Optimization

From the empirical study conducted in previous chapter, it is clear that the performance of MOEAs deteriorates sharply at high noise levels. One simple approach to improve algorithmic performance is to perform the re-sampling of individuals. However, such re-evaluation of individuals is often computationally expensive and it may be infeasible to perform a large number of observations. Based upon the analysis of noise impact on population dynamics, three noise-handling features including experiential learning directed perturbation, gene adaptation selection strategy, and possibilistic archiving model, are proposed in this chapter to improve the robustness of EMOO.

3.1 Design of Noise-Handling Techniques

For the ensuing discussing, an individual is represented as a vector, $\vec{X} = (\vec{g}, \vec{p}, \vec{f})$, where the vector $\vec{g}$ and $\vec{p}$ represents the decision vector in the genotype space $\vec{G} \in B^{chromosome\_length}$ and the phenotype space $\vec{X}^{n_x}$, respectively; $\vec{f}$ is the associated objective vector in the objective space, $\vec{F}^M$. The binary representation $\vec{g}$ of the decision variables is mapped by the function $f : \vec{G} \rightarrow \vec{X}$ from the genotype space to the phenotype space and there is a corresponding inverse function $f^{-1} : \vec{X} \rightarrow \vec{G}$.

3.1.1 Experiential Learning Directed Perturbation (ELDP)

It has been observed that the decision-error ratio for selection, ranking and archiving is lower at the early stage of evolution. Therefore, the proposed ELDP makes use of the better decisions at early generations to improve performance. The ELDP is a deviation from conventional mutation paradigm in two aspects: 1) the change in chromosome is ordered instead of being by chance, and 2) variation can be performed either in genotype or phenotype space. In particular, the actual adaptation in ELDP is based on posterior knowledge of favorable movements in the search space.

The experiential learning strategy adopted by ELDP for directed perturbation in the phenotype space is inspired by the role of momentum term in back-propagation for neural networks; accelerating movement in the direction of improvement while restricting movement otherwise. The variation for each decision variable x_j can be described as follows,

$$\bar{\triangle} x_j(t) = \triangle x_j(t) + \alpha \cdot \bar{\triangle} x_j(t-1) \tag{3.1}$$

where α represents the learning rate; $\triangle$ refers to changes acquired through prior genetic operations such as crossover, while $\bar{\triangle}$ corresponds to changes including the effect of momentum. According to (3.1), the posterior knowledge comes in the form of past movements made by the individual in concern. The ELDP defines a two-mode operation to impose the necessary control for directed variation in the phenotype space and to perform bit-flip mutation in the genotype space for genetic diversity. The ELDP operation is given as follows,

$$x_j(t+1) = \begin{cases} p_j(t) + \bar{\triangle} x_j(t), & \text{if } \triangle_{min} < |\alpha \cdot \bar{\triangle} x_j(t-1)| < \triangle_{max} \\ f\Big(g_{BF}(\vec{g}(t) + \triangle \vec{g}(t))\Big), & \text{otherwise.} \end{cases} \tag{3.2}$$

where $\bar{\triangle} x_j(t)$ refers to the variation described in (3.1) and $g_{BF}()$ denotes bit-flip mutation for the j-th decision variable. Note that the corresponding changes will also be updated in the genotype for any variation in the phenotype space. From (3.2), the magnitude of directed

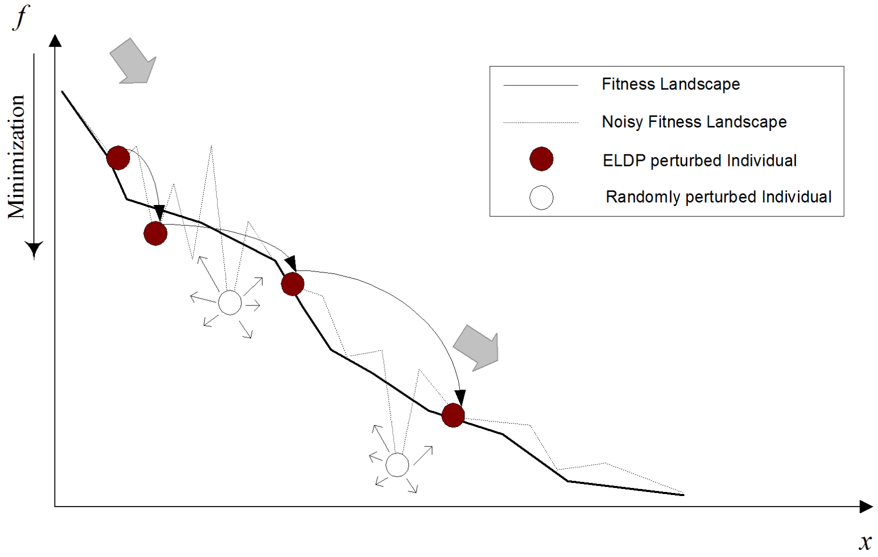


Figure 3.1: Operation of ELDP.

perturbation is bounded by $\triangle_{min}$ and $\triangle_{max}$, which can be set by the user. The limiting bounds on directed perturbation for x_j ensure that a new search direction is initiated through bit-flip mutation to reduce the impact of outliers or whenever the evolutionary search process has stalled. For simplicity, and $\triangle_{min}$ and $\triangle_{max}$ is set as 0.0 and 0.1, respectively, on the normalized decision space.

The operation of ELDP is illustrated in Figure 3.1. By considering each and every decision variable, the ELDP provides a simple and efficient way for adaptation of the required variation associated with each parameter. From (3.1) and (3.2), the variation increases in magnitude in the direction of change and thus accelerates convergence when $\triangle x_j(t)$ and $\bar{\triangle} x_j(t-1)$ have the same sign. Likewise, the variation is small if $\triangle x_j(t)$ and $\bar{\triangle} x_j(t-1)$ are different in sign, implying that the ELDP performs local fine-tuning in the later stage of evolution where movements tend to fluctuate. Moreover, such properties are desirable in the context of noisy objective function optimization where inferior solutions are likely to participate in the recombination process. In such cases, the ELDP helps to reduce the stochastic influence of noise and prevents the individuals from changing haphazardly.

3.1.2 Gene Adaptation Selection Strategy (GASS)

From Figure 2.7 in Section 2.4, we noted that the mean of the population distribution remains relatively invariant in the decision space despite the different environmental conditions. In addition, the search range of the different variables tends to converge to a smaller region within the search space in a noise free environment and remains relatively unchanged or even diverges in a noisy environment. It is thus useful to construct an approximate model of the ideal population behavior for guiding the evolutionary search process to 1) escape premature convergence, and 2) prevent it from becoming a random search in the presence of noise.

The proposed GASS attempts to manipulate population distribution so that the evolutionary algorithm exhibits certain desirable search characteristics. Specifically, it builds a posterior model of the desired population distribution and subsequently adapts part of the selected individuals chromosome. Mathematically, the adaptation of gene structure is given as

$$x'_j(t) = \begin{cases} \mathrm{U}(a_j, b_j), & \mathrm{U}(0,1) < \frac{1}{n_x} \\ x_j(t), & \text{otherwise.} \end{cases} \tag{3.3}$$

where n_x is the number of decision variables to be optimized. Here $\frac{1}{n_x}$ is the probability of decision variable j being selected for adaptation. The GASS defines an operation in the phenotype space which is characterized by a uniformly distributed number U on the interval $[a_j, b_j]$ for each decision variable. After which, the corresponding genotypic adaptation will be updated. It adopts two different models to control the evolution for a better convergence, i.e., the interval $[a_j, b_j]$ is dependent on the state of evolution and the archival population distribution in the decision space.

Convergence Model

The population distribution tends to converge as the evolving population approaches the final tradeoff. Since it is difficult to determine if $\vec{PF}^A$ corresponds to $\vec{PF}^*$, the adopted

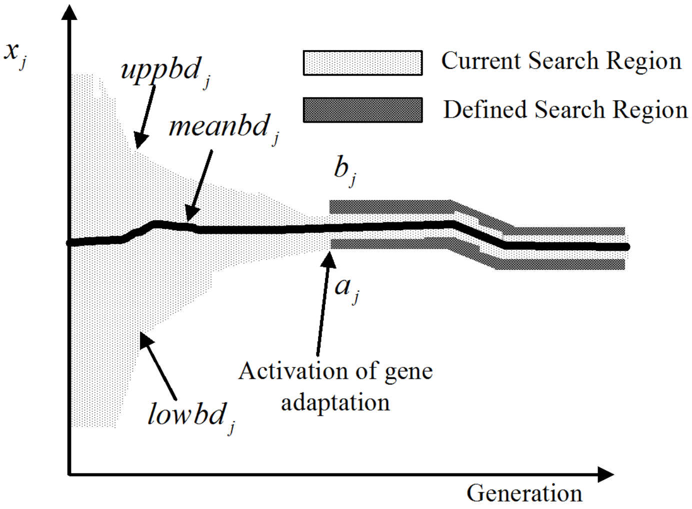


Figure 3.2: Search range defined by convergence model.

model needs to define a space that is larger than the current search range along the j-th dimension to prevent a premature convergence. The corresponding interval is given as

$$\begin{aligned} a_j &= lowbd_j - w \cdot meanbd_j \\ b_j &= uppbd_j + w \cdot meanbd_j \end{aligned} \tag{3.4}$$

where w is a fixed parameter that controls the step change in the search range, $lowbd_j$, $uppbd_j$ and $meanbd_j$ corresponds to the minimum, maximum and mean of x_j in the archive, respectively. The aim of the convergence model is to compel EA to look beyond the current search region as shown in Figure 3.2. In the case where an individual corresponds to the global optimum, the overall quality of the evolving population is not adversely affected. This is because similar individuals have similar genetic information, and the model creates individuals based on the converged search region.

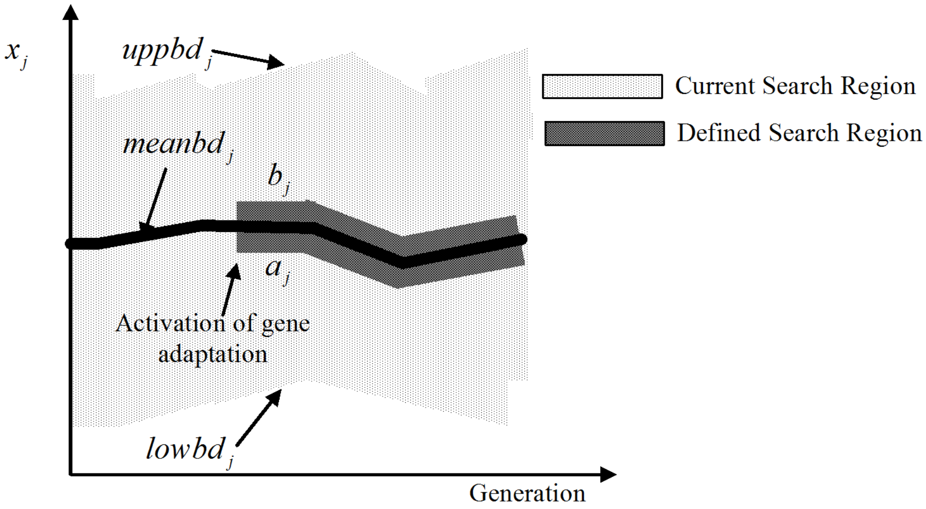


Figure 3.3: Search range defined by divergence model.

Divergence Model

A degenerate evolutionary search process is characterized by a non-convergent population distribution. In the situation where the evolutionary process degenerates into a random search, such as due to high level of uncertainty in the system, the interval for j-th decision variable defines a small search region about its mean given as

$$\begin{aligned} a_j &= meanbd_j - w \cdot meanbd_j \\ b_j &= meanbd_j + w \cdot meanbd_j \end{aligned} \tag{3.5}$$

The aim of the model is to reduce stochastic change in gene structure due to random selection of individuals by providing a stable search range as shown in Figure 3.3. Note that the interval specifying the location of new individuals is only a rough deduction of the search region based on the available information. Intuitively, the utilization of statistical model can improve robustness of existing selection strategies, where individuals selected based on fitness are included directly in the evolving population. The selection of appropriate model is performed autonomously based on the condition of the evolutionary process. As shown by the experiments in Section 2.4, the search process degenerated by noise can hardly fill up more than 30% of the archives capacity. Hence, the behaviors of convergence and random search can be determined based upon the growth rate of archives population, and the gene adaptation strategy can be activated when there is sufficient indication of convergence or random search behavior in the evolution. In this paper, a simple scheme is adopted; divergence model is activated when 60% of the archive capacity is reached while convergence model is activated when less than 60% of the archive is filled after 150 generations.

3.1.3 A Possibilistic Archiving Methodology

This section describes two archiving models, i.e., necessity-possible (NP-) archiving model and necessity (N-) archiving model, based on the concept of possibilistic Pareto dominance relation. Similar to [94] where probabilities are employed to model uncertainty as part of the Pareto ranking procedure, fuzzy numbers that are uniquely suitable for representing uncertain information are used here to represent the objective vectors. The proposed approach is based on the concept of possibility and necessity measures [49], which aims to rectify certain deficiencies present in the current Pareto-based updating strategy in handling noisy environments. Besides, a tagging system is proposed to allow both the models to co-exist in the situation where the uncertainty level is low.

The archive updating schemes adopted in existing MOEAs are largely based on the concept of Pareto optimality, and some form of truncation process is usually applied to limit the number of good individuals stored in the archive due to the limitation of memory resource. Although such an updating scheme is simple and effective, it is not competent in dealing with individuals containing uncertainty in the objective functions, since the dominance relationship for these individuals in the presence of noise is no longer deterministic. In the absence of a reliable decision maker, the standard archiving scheme can be easily deceived into removing non-dominated individuals from the archive or inserting dominated

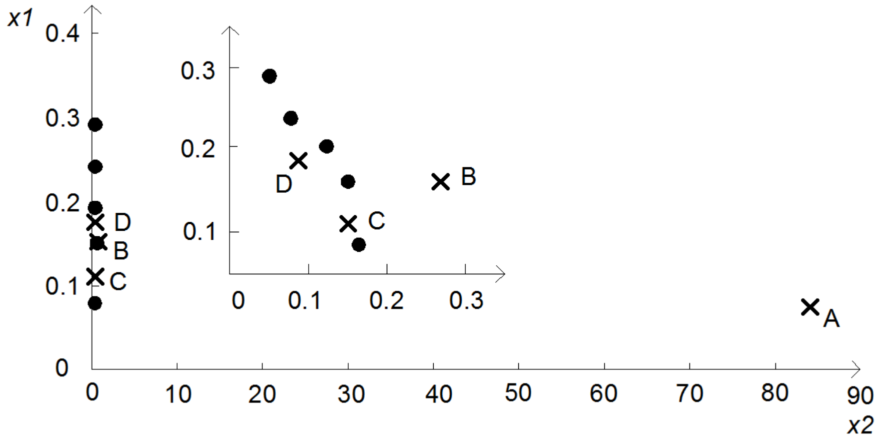


Figure 3.4: Distribution of archived individuals marked by closed circles and the newly evolved individuals marked by crosses in a two-dimensional objective space.

individuals to the archive, which could subsequently affect the performance of EMOO in noisy environments.

The instance in Figure 3.4 shows the distribution of archived individuals marked by closed circles and the newly evolved individuals marked by crosses in a two-dimensional objective space. From the definition of Pareto dominance, it is clear that A, C and D will be selected to fill the archive in the evolution. However, A provides only marginal improvement for x_1 at a great expense of x_2, which gives little contribution to the overall quality of the solution set. In the face of limited archive storage, non-contributing individuals occupying valuable space that are usually located in isolated regions in the objective space are less unlikely to be removed during the truncation process. It is thus desirable if the updating function is capable of rejecting such non-dominated individuals according to some *a-priori* knowledge or user preference. In addition, it is also desirable if the updating mechanism can minimize removal of non-dominated individuals and provide a chance for individuals degraded by noise to survive in the evolution.

To understand the proposed archiving models, a number of definitions are given as follows:

Definition 3.1 *Necessity Condition*: Given that f_1 and f_2 are fuzzy numbers with membership functions μ_{f_1} and μ_{f_2}, respectively. The necessity that the largest possible value of f_1 is smaller than the smallest value of f_2 is given by

$$Nec(\overline{z}_1 < \underline{z}_2) = \text{inf}_u \text{max}\Big[1 - \mu_{f_1}(u), \text{inf}_{v<u}(1 - \mu_{f_2}(v))\Big] \tag{3.6}$$

Definition 3.2 *Possibility condition*: Given that f_1 and f_2 are fuzzy numbers with membership functions μ_{f_1} and μ_{f_2}, respectively. The possibility that the smallest possible value of f_1 is smaller than the largest value of f_2 is given by

$$Pos(\underline{z}_1 < \overline{z}_2) = \text{sup}_u \text{min}[1 - \mu_{f_1}(u), \text{sup}_{u<v}\mu_{f_2}(v)] \tag{3.7}$$

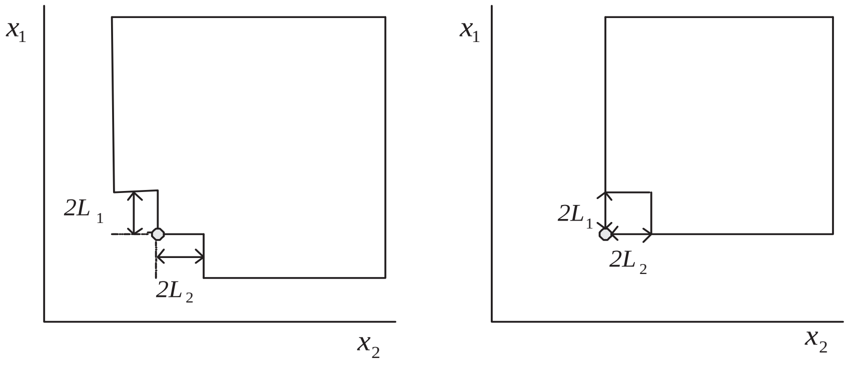
Definition 3.3 *NP-dominance*: Given that $\vec{f_1}$ and $\vec{f_2}$ are M-dimensional objective vectors of fuzzy numbers with membership functions $\vec{\mu}_{f_1}$ and $\vec{\mu}_{f_2}$, respectively. Then $\vec{X}_1$ NP-dominates $\vec{X}_2$, denoted by $\vec{X}_1 \prec_{NP} \vec{X}_2$, *iff*

$$\begin{array}{rcl} Pos(\underline{z}_{1,j} < \overline{z}_{2,j}) & \geq & Pos(\underline{z}_{1,i} < \overline{z}_{2,i}) \forall 1, 2, ..., M \\ & \text{or} & \\ Nec(\overline{z}_{1,i} < \underline{z}_{2,i}) & = & 1 \exists\ i \in 1, 2, ..., M \text{ and } Nec(\overline{z}_{1,j} < \underline{z}_{2,j}) < 1\ \forall j \in 1, 2, ..., M \end{array} \tag{3.8}$$

Definition 3.4 *N-dominance*: Given that $\vec{f_1}$ and $\vec{f_2}$ are M-dimensional objective vectors of fuzzy numbers with membership functions $\vec{\mu}_{f_1}$ and $\vec{\mu}_{f_2}$, respectively. Then $\vec{f_1}$ N-dominates $\vec{f_2}$, denoted by , *iff*

$$Nec(\overline{z}_{1,i} < \underline{z}_{2,i}) = 1\ \forall i \in 1, 2, ..., M \tag{3.9}$$

Figure 3.5 illustrates the different dominance relation for a minimization problem. The shaded region represents the area dominated by the individual marked by a circle. The NP-model behaves similarly to existing archiving models, but allows decision-maker to



(a) (b)

Figure 3.5: Region of dominance based on (a) NP-dominance relation, and (b) N-dominance relation.

reject certain non-dominated individuals in the evolution if necessary. This archiving model compares and updates individuals according to the NP-dominance relation. As shown in Figure 3.5(b), the width of the fuzzy membership function associated with the i-th objective is denoted by L_i, which represents the tolerance level of inferiority for each objective. As L_i tends to zero, the behavior of NP-dominance approaches that of Pareto-dominance relation. The pruning criterion is based upon some degree of crowding or niche count, which helps to maintain population diversity in the archive.

The N-archiving model updates individuals according to the N-dominance relation, which stores a set of possibly non-dominated individuals. The membership function is a reflection of the uncertainty level present in the system, and the width, L_i, represents the possible values of the i-th objective. In order to minimize deletion of non-dominated individuals, the N-archiving model removes an archived individual only if it is N-dominated by an individual in the archive. In this model, an individual is selected if there is no archived individual that necessarily dominates it. Intuitively, the size of archive will grow exceedingly large with the increase of noise, and any form of niche count or crowding comparison is of no practical meaning in the presence of noise. Therefore, the truncation criterion for the

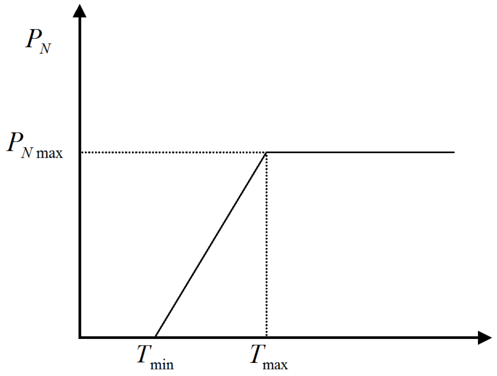


Figure 3.6: Decision process for tag assignment based on the level of noise present.

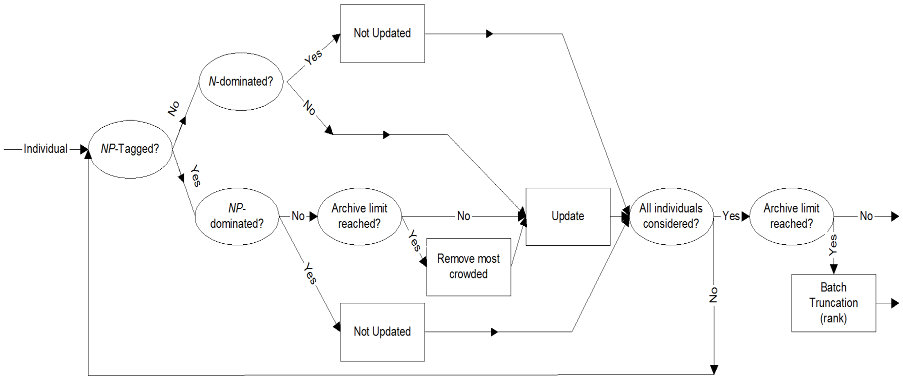


Figure 3.7: Possibilistic archiving model.

archive should be based upon the apparent ranking provided by the prior evaluation process.

It is clear that the proposed two archiving models operate at the two ends of the noise spectrum. A tagging system is thus proposed to provide a graceful integration of both models, since it is often more desirable to incorporate both the model properties in the presence of low noise level. Each individual is assigned either a NP-tag or N-tag that defines the behavior it will experience during the archiving process, e.g., an individual assigned with the NP-tag is regarded as if only the NP-model is implemented. The assignment of tags is based on a probability distribution as shown in Figure 3.6. If the noise level is below the minimum threshold of T_{min}, all individuals will be assigned the NP-tag. When the noise level is above the maximum threshold of T_{max}, all individuals will be assigned the N-tag with a probability of $P_N\text{max}$. If the noise level is between the two thresholds, the probability of P_N is a linear function of noise as depicted in Figure 3.6. The *Possibilistic* archiving model is shown in Figure 3.7.

3.1.4 Implementation

The proposed features are incorporated into BMOEA described in Chapter 2 and named as MOEA-RF, as shown in the program flowchart in Figure 3.8. The fitness assignment is based on Pareto ranking scheme [63] and the mechanism of niche sharing is used in tournament selection as well as diversity maintenance in the archive. The tournament selection of individuals for the mating pool is followed by gene adaptation if the criterion for convergence or random search is satisfied. The genetic operations of uniform crossover and ELDP are then applied to the mating pool. Both the step size, w, for GASS and the learning rate, α, for ELDP are set as 0.3 in the algorithm. The *Possibilistic* archiving approach as shown in Figure 3.6 is applied with triangular membership function for both the N- and NP-archiving models. Since the width of membership function for the N-archiving model represents the noise level, it can be estimated by re-sampling one individual at the beginning of the evolution. The parameters for tag assignment, such as T_{min}, T_{max} and $P_N\text{max}$, is set as 0.0, 0.1 and 1.0 respectively.

3.2 Comparative Study

In order to examine the effectiveness of MOEA-RF, a comparative study with NTSPEA, MOPSEA, SPEA2 [228], NSGAII and PAES [127] is carried out based upon the five benchmark problems. Since re-sampling is probably the simplest and most common noise compensation technique, the baseline algorithm with a re-sampling rate of 10 (named as RMOEA)

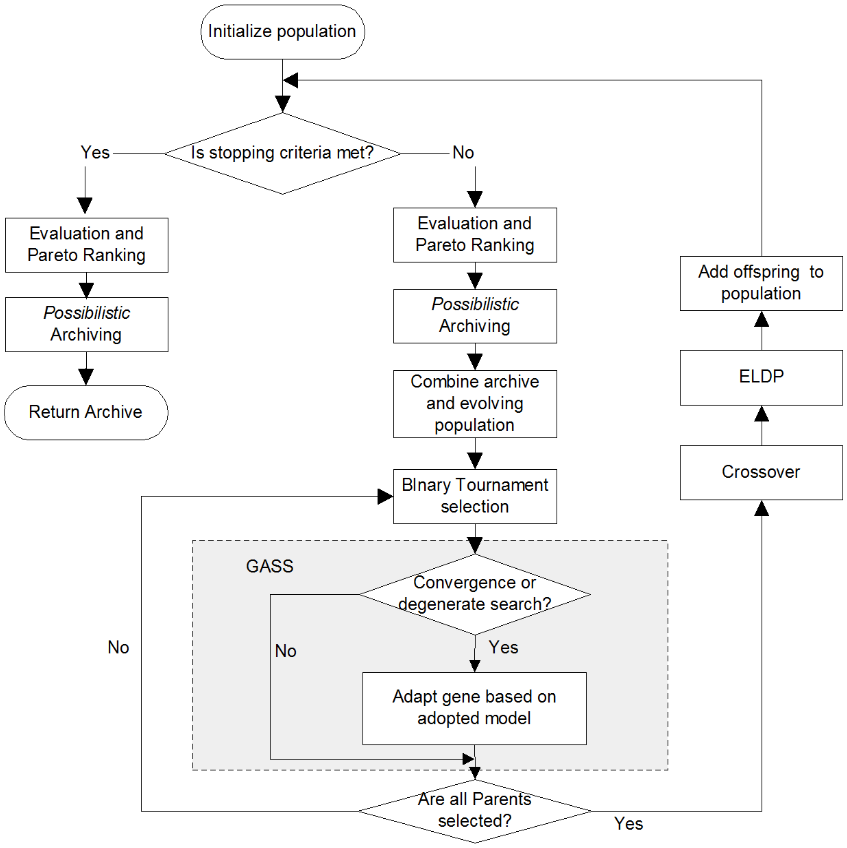

Initialize population
Is stopping criteria met?
Yes
No
Evaluation and Pareto Ranking
Possibilistic Archiving
Return Archive
Evaluation and Pareto Ranking
Possibilistic Archiving
Combine archive and evolving population
BInary Tournament selection
GASS
Convergence or degenerate search?
No
Yes
Adapt gene based on adopted model
No
Are all Parents selected?
Yes
Add offspring to population
ELDP
Crossover

Figure 3.8: Program flowchart of MOEA-RF.

Table 3.1: Indices of the different algorithms

Index	1	2	3	4	5	6	7
Algorithm	MOEA-RF	RMOEA	NTSPEA	MOSPEA	SPEA2	NSGAII	PAES

is also included in the study. The indices of the seven algorithms are listed in Table 3.1. In this study, different experimental setups with noise settings of $\sigma^2 = \{0\%, 5\% \ , 10\%, 20\%\}$ are applied to evaluate the performance of the algorithms.

The simulations are implemented in C++ on an Intel Pentium 4 2.8 GHz computer and

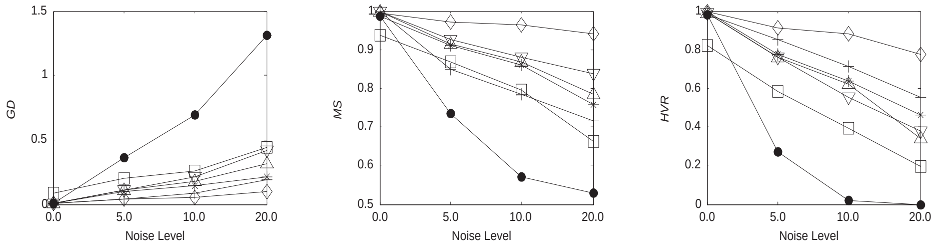
Table 3.2: Parameter setting for different algorithms

Parameter	Settings

Populations	Population size 100 in NSGAII, SPEA2, NTSPEA, MOPSEA, RMOEA and MOEA-EF;
	Population size 1 in PAES;
	Archive (or secondary population) size 100.

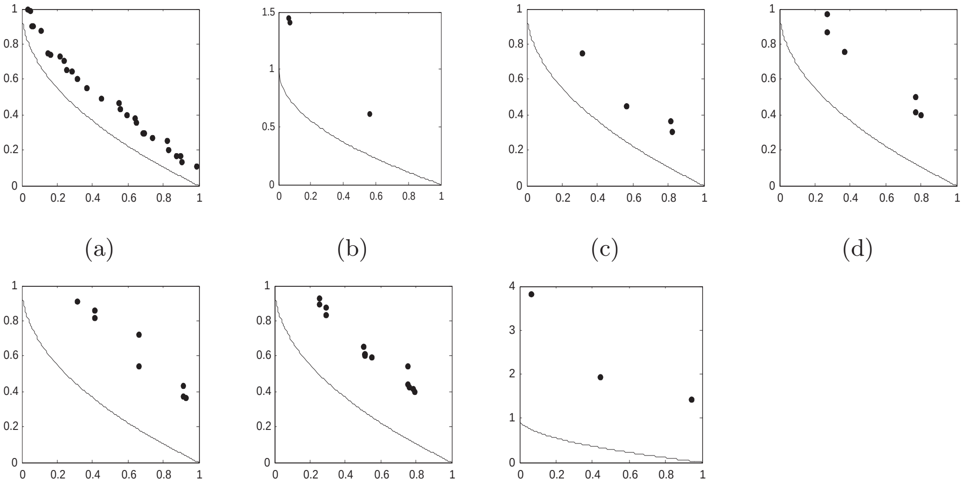
Chromosome	Binary coding; 15 bits per decision variable.
Selection	Binary tournament selection
Crossover operator	Uniform crossover
Crossover rate	
	0.8
Mutation operator	Bit-flip mutation in NSGAII, SPEA2, NTSPEA, RMOEA and ELDP in MOEA-RF
Mutation rate	$\frac{1}{chromosome\_length}$ for ZDT1, ZDT4 and ZDT6; $\frac{1}{bit\_number\_per\_variable}$ for FON and KUR;
Hyper-grid size	2^5 per dimension
Niche Radius	Dynamic for MOEA-RF.
Evaluation number	50,000

the results shown are based on the true objective function values. Thirty independent runs are performed for each of the test functions in order to obtain the statistical information, such as consistency and robustness of the algorithms. In order to assess statistical difference of the simulation results, Kolmogorov-Smirnov (KS) test is applied to the different performance metrics. The various parameter settings for each algorithm are listed in Table 3.2. All the algorithms are implemented using the same binary coding scheme, tournament selection, uniform crossover, and bit flip mutation. In accordance to the original paper [21], $k_{\max}$ is set as 4, while c_1 and c_2 is set as 10% and 30%, respectively, for NTSPEA. The value of s is calculated by re-sampling ten individuals immediately after the first evaluation for MOPSEA.



(a) (b) (c)

Figure 3.9: Performance metric of (a) GD, (b) MS, and (c) HVR for ZDT1 attained by MOEA-RF (◇), RMOEA (□), NTSPEA(|), MOPSEA (∗), SPEA2 (▽), NSGAII (△) and PAES (●) under the influence of different noise levels.



(a) (b) (c) (d)

(e) (f) (g)

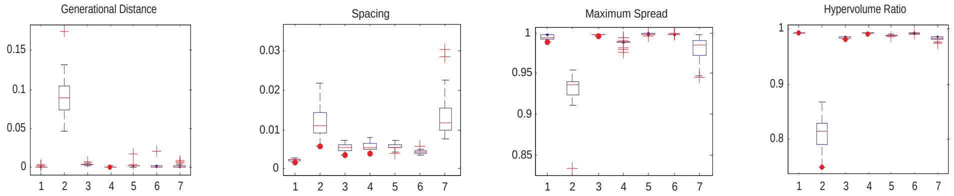
Figure 3.10: The PF^A from (a) MOEA-RF, (b) RMOEA, (c) NTSPEA, (d) MOPSEA, (e) SPEA2, (f) NSGAII, and (g) PAES for ZDT1 with 20% noise.

3.2.1 ZDT1

ZDT1 has a convex Pareto front with a large number of variables to be optimized. The performance of the different algorithms over noise levels of $\{0\%, 5\%, 10\%, 20\%\}$ is shown in Figure 3.9. The evolved tradeoff with noise level of 20% from the different algorithms using the same initial population is illustrated in Figure 3.10(a)-(g). The distribution of the different performance metrics is represented by box plots in Figure 3.11(a)-(d) and Figure 3.12(a)-(d) for 0% and 20% noise respectively.

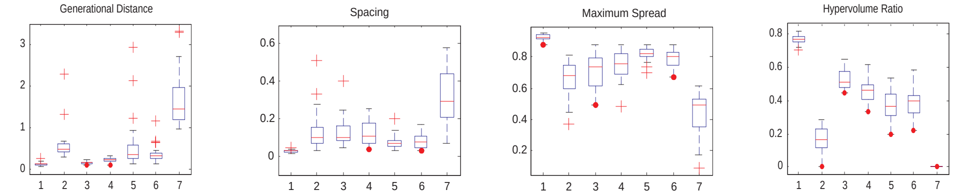
Figure 3.9 shows that the performance of the algorithms deteriorates with the increase of noise level; particularly there is a drastic performance change in PAES when the noise level is increased to 5%. Figure 3.9(c) shows that the MOEA-RF, NTSPEA and MOPSEA are capable of evolving better solutions in a noisy environment as compared to algorithms without any noise compensation techniques. With the exception of RMOEA, Figure 3.11 shows that most algorithms encountered no problem in converging and maintaining a diverse set of solutions for ZDT1 under noiseless environment. As shown by the evolutionary trace of GD in Figure 3.12, the poor performance of RMOEA can be attributed to the re-evaluation of candidate individuals. Although the performance of MOEA-RF for proximity is not the best, it has the fastest convergence for both GD and MS as can be seen from the evolutionary trace of GD and MS in Figure 3.13. The metric of HVR in Figure 3.11(d) indicates that the solutions evolved by MOEA-RF have the best overall quality. It also produces a more uniformly distributed Pareto front as shown by the low value of S. The KS test also revealed that MOEA-RF and other algorithms are statistically different in terms of S, MS, and HVR.

Among the conventional MOEAs, i.e., SPEA2, NSGAII and PAES, it is apparent that the PAES is worst affected by the noise. As can be seen from the distribution of GD in Figure 3.12(a), MOEA-RF, NTSPEA and MOPSEA produce competitive results since various features are included in these algorithms to deal with the noise. On the other hand, the performance of RMOEA is the worst among all algorithms except for PAES. As can be observed in Figure 3.9 and Figure 3.12(b)-(d), the MOEA-RF is capable of evolving a



(a) (b) (c) (d)

Figure 3.11: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT1 with 0% noise.



(a) (b) (c) (d)

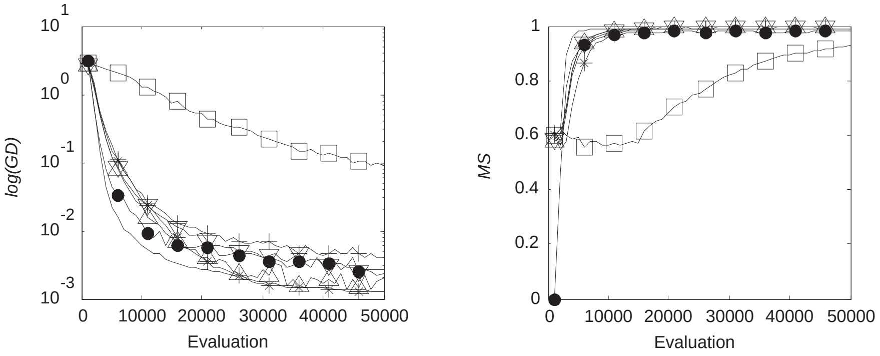
Figure 3.12: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT1 with 20% noise.

more diverse and uniformly distributed Pareto front for ZDT1 in the presence of noise as compared to other algorithms.

3.2.2 ZDT4

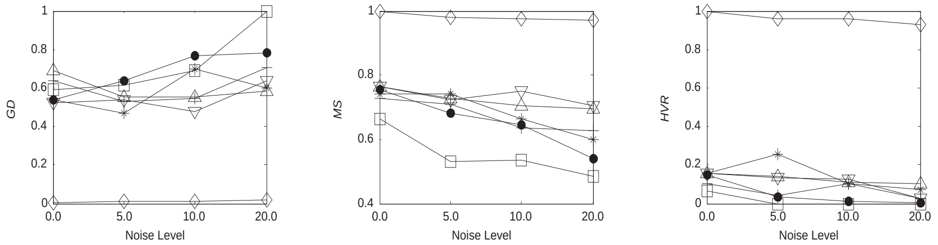
ZDT4 challenges the algorithms ability to deal with the problem of multi-modality. The performance of the different algorithms over the noise levels of {0%, 5%, 10%, 20%} is shown in Figure 3.14. The evolved tradeoff from the different algorithms using the same initial population is shown in Figure 3.15(a)-(g) and Figure 3.16(a)-(g) for noise level of 0% and 20%, respectively. The distribution of the different performance metrics is represented by box plots in Figure 3.17(a)-(d) and Figure 3.18(a)-(d) for 0% and 20% noise respectively.

From the trend of GD over the various noise levels in Figure 3.14(a), it is apparent that



(a) (b)

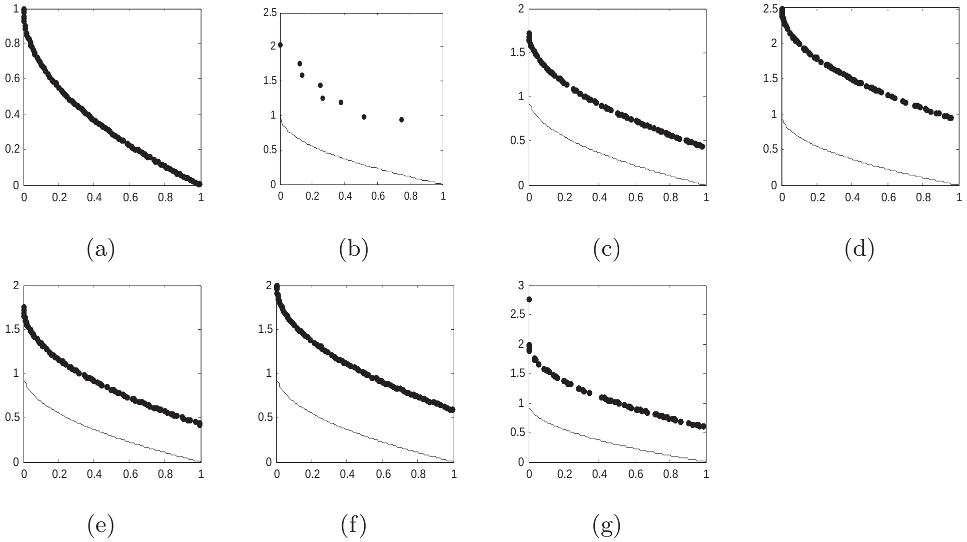
Figure 3.13: Evolutionary trace of (a) GD and (b) MS for ZDT1 with 0% noise.



(a) (b) (c)

Figure 3.14: Performance metric of (a) GD, (b) MS, and (c) HVR for ZDT4 attained by MOEA-RF (◇), RMOEA (□), NTSPEA(|), MOPSEA (∗), SPEA2 (▽), NSGAII (△) and PAES (●) under the influence of different noise levels.

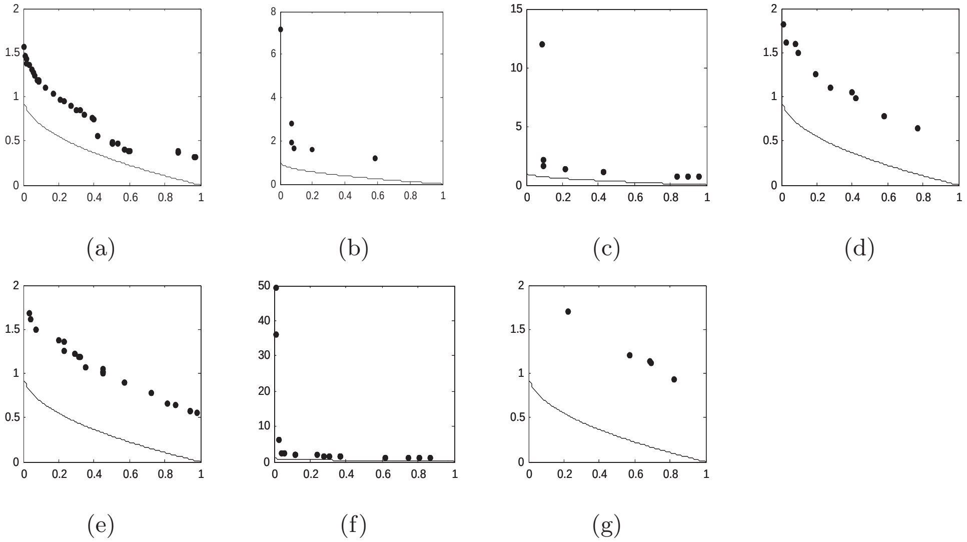
the smoothing effect of noise described in Section 2 is also present for the noise levels of 5% and 10%. In contrast to the algorithmic behaviors observed for ZDT1, this phenomenon enables some of the algorithms, such as SPEA2, MOPSEA to evolve better solutions as shown in Figure 3.14(a) and Figure 3.14(c). It can be observed from Figure. 3.15 and Figure 3.17 that the local optima imposed by this benchmark appear to be a formidable barrier against the global convergence. At the end of 50,000 evaluations, RMOEA, MOPSEA, SPEA2, NSGAII and PAES only managed to discover one of the local Pareto fronts. On the



(a) (b) (c) (d)

(e) (f) (g)

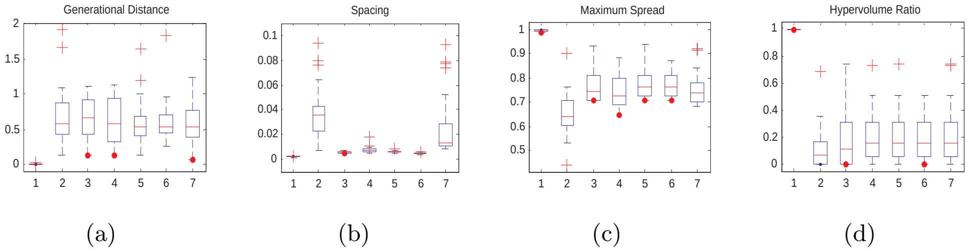
Figure 3.15: The PF^A from (a) MOEA-RF, (b) RMOEA, (c) NTSPEA, (d) MOPSEA, (e) SPEA2, (f) NSGAII, and (g) PAES for ZDT4 with 0% noise.



(a) (b) (c) (d)

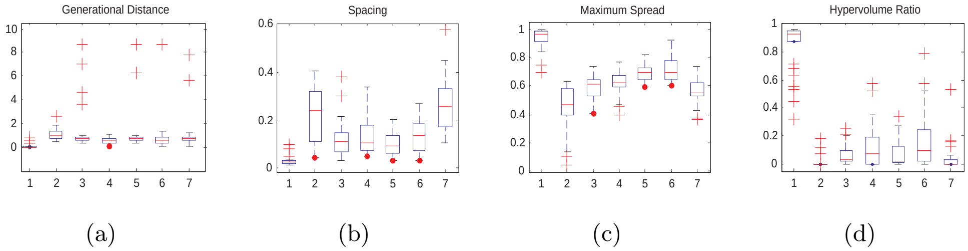
(e) (f) (g)

Figure 3.16: The PF^A from (a) MOEA-RF, (b) RMOEA, (c) NTSPEA, (d) MOPSEA, (e) SPEA2, (f) NSGAII, and (g) PAES for ZDT4 with 20% noise.



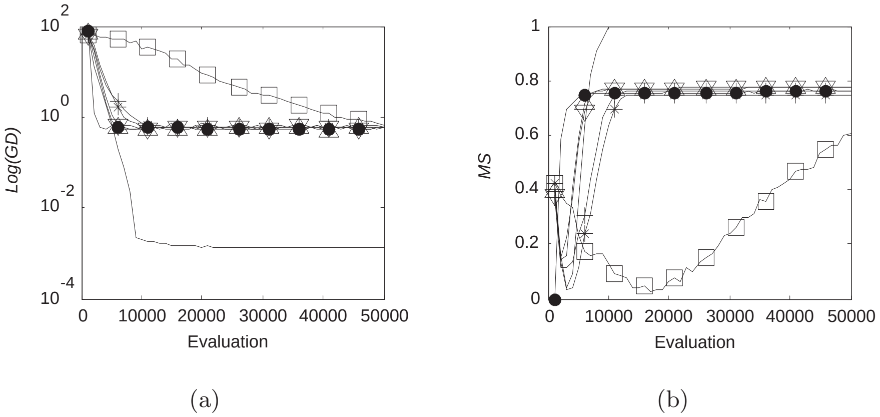
(a) (b) (c) (d)

Figure 3.17: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT4 with 0% noise.



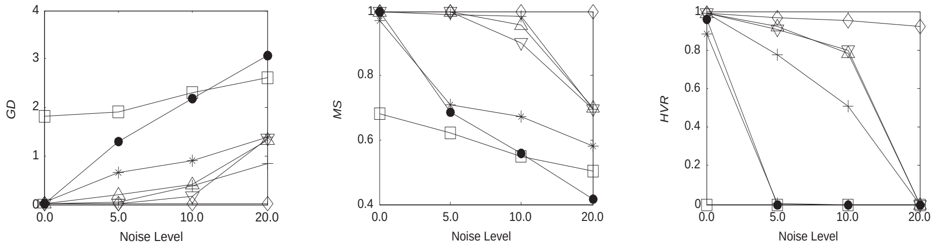
(a) (b) (c) (d)

Figure 3.18: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT4 with 20% noise.



(a) (b)

Figure 3.19: Evolutionary trace of (a) GD and (b) MS for ZDT4 with 0% noise.

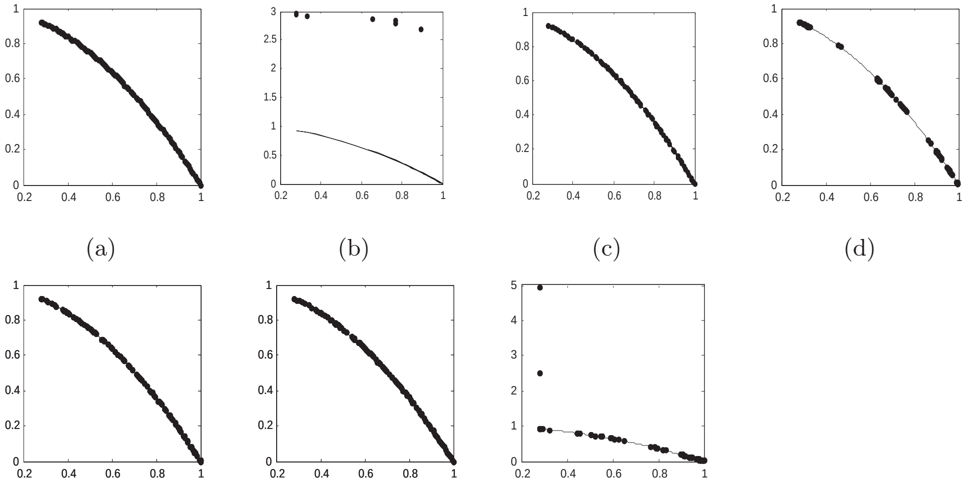


(a) (b) (c)

Figure 3.20: Performance metric of (a) GD, (b) MS, and (c) HVR for ZDT6 attained by MOEA-RF (◇), RMOEA (□), NTSPEA(|), MOPSEA (∗), SPEA2 (▽), NSGAII (△) and PAES (●) under the influence of different noise levels.

other hand, Fig.3.17(a) shows that MOEA-RF incorporated with ELDP and GASS is able to evolve individuals near to the global Pareto front consistently. From the convergence trace of GD in Figure 3.19(a), it is clear that ELDP plays an important role in the algorithm to escape from the local optima. Moreover GASS is activated whenever the criterion of convergence is satisfied, which diverts the evolutionary search and avoids the local optima. The dips on the metric of MS observed in Figure 3.19(b) correspond to the effect of jumping from one local Pareto front to another during the evolutionary search. In this intermediate state of jumping, there is a transition from one relatively diverse set of individuals along a local Pareto front to another, which results in the effect of sudden dips. As can be observed in Figure 3.17(b)-(d), MOEA-RF is capable of evolving a more diverse and uniformly distributed Pareto front under noiseless environment as compared to other algorithms.

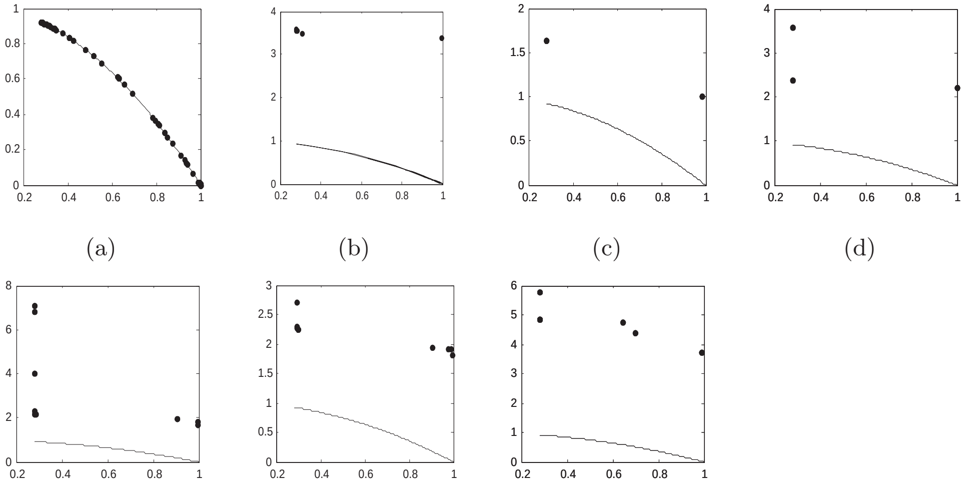
As can be seen in Figure 3.16 and Figure 3.18, the performance of NTSPEA, MOPSEA, SPEA2, NSGAII and PAES is poor under the influence of noise. Furthermore, the number of non-dominated individuals discovered by these algorithms is also greatly reduced as shown in Table 3.3. By comparing Figure 3.17(b)-(d) and Figure 3.18(b)-(d), it is apparent that MOEA-RF is able to evolve individuals that are on or close to the global tradeoff for ZDT4, although its performance is generally affected by the presence of noise.



(a) (b) (c) (d)

(e) (f) (g)

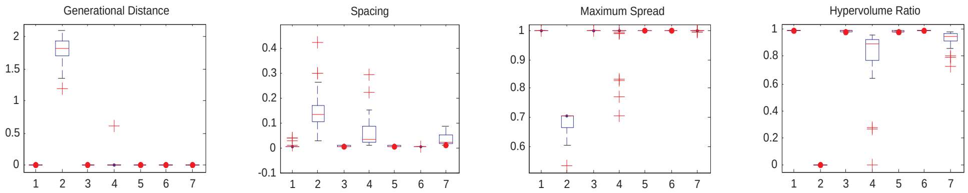
Figure 3.21: The PF^A from (a) MOEA-RF, (b) RMOEA, (c) NTSPEA, (d) MOPSEA, (e) SPEA2, (f) NSGAII, and (g) PAES for ZDT6 with 0% noise.



(a) (b) (c) (d)

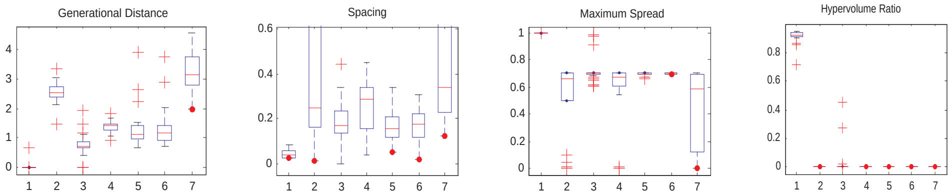
(e) (f) (g)

Figure 3.22: The PF^A from (a) MOEA-RF, (b) RMOEA, (c) NTSPEA, (d) MOPSEA, (e) SPEA2, (f) NSGAII, and (g) PAES for ZDT6 with 20% noise.



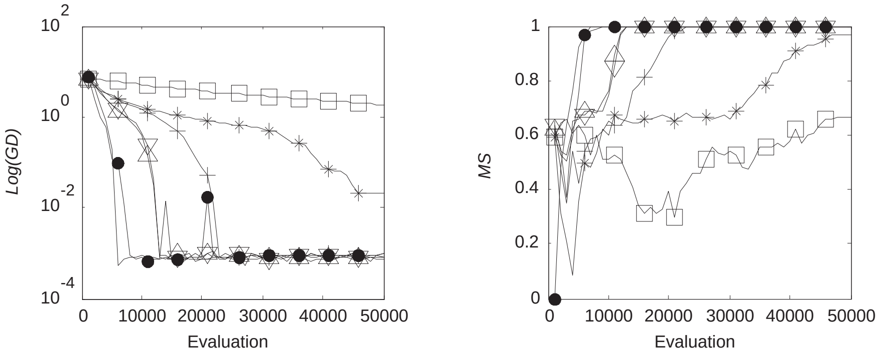
(a) (b) (c) (d)

Figure 3.23: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT6 with 0% noise.



(a) (b) (c) (d)

Figure 3.24: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT6 with 20% noise.



(a) (b)

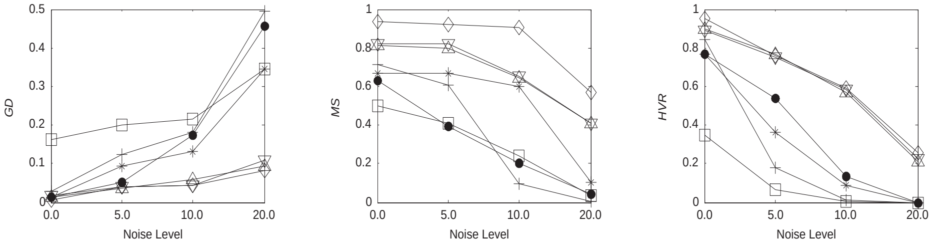
Figure 3.25: Evolutionary trace of (a) GD and (b) MS for ZDT6 with 0% noise.

3.2.3 ZDT6

ZDT6 has a biased search space and non-uniformly distributed solutions along the global tradeoff, which makes it difficult for algorithms to evolve a well-distributed Pareto front. The performance of the different algorithms over the noise levels of {0%, 5%, 10%, 20%} is shown in Figure 3.20. The evolved tradeoff from the different algorithms using the same initial population is shown in Figure 3.21(a)-(g) and Figure 3.22(a)-(g) for noise level of 0.0% and 20.0% respectively. The distribution of the different performance metrics is represented by box plots in Figure 3.23(a)-(d) and Figure 3.24(a)-(d) for 0% and 20% noise respectively.

It can be observed from Figure 3.20 that different algorithms behave differently although the performance is generally deteriorated with increasing noise levels. For instance, Figure 3.20(c) shows that there is a drastic drop in the performance of MOPSEA and PAES as reflected by the metric of HVR when the noise level is increased to 5%. On the other hand, the performance of NTSPEA, NSGAII and SPEA2 seems unaffected for MS and GD over the noise levels of 0%, 5% and 10%, but deteriorates sharply when the noise level is increased to 20%. It can also be observed that the noise-handling algorithms of RMOEA, NTSPEA, MOPSEA and MOEA-RF have different degree of success in the presence of noise. For example, the re-sampling mechanism employed by RMOEA has a slight edge over only PAES at noise level of 20% for GD and MS, while MOEA-RF outperforms other algorithms on the various metrics of GD, MS and HVR.

Although RMOEA, MOPSEA and PAES can identify some parts of the tradeoff for ZDT6, Figure 3.21(b), (d) and (g) shows that these algorithms are unable to evolve a well-distributed Pareto front. Figure 3.23(c) also shows that RMOEA is unable to find a diverse solution set consistently. On the other hand, NSGAII, SPEA2 and MOEA-RF provide competitive results in all aspects, particularly for the metric of GD as shown in Figure 3.23. In addition, the convergence trace of GD and MS in Figure 3.25 shows that MOEA-RF offers the fastest convergence among all algorithms due to the incorporation of ELDP.



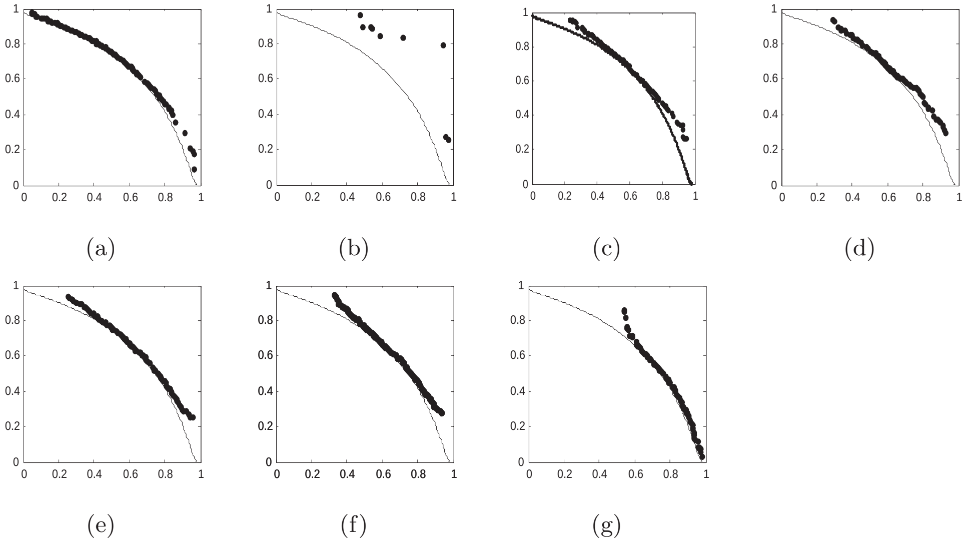
(a) (b) (c)

Figure 3.26: Performance metric of (a) GD, (b) MS, and (c) HVR for FON attained by the algorithms under the influence of different noise levels.

Figure 3.22 and Figure 3.24 show that NTSPEA, MOPSEA, NSGAII, SPEA2 and PAES are unable to find any individuals along the global tradeoff under the influence of noise. The simple archiving technique employed in NTSPEA allows it to cope with noise better than MOPSEA, SPEA2, NSGAII and PAES. On the other hand, MOEA-RF is able to find a set of solutions near the global tradeoff consistently. With the exception of GD at 0% noise, the KS test indicates that the performance between MOEA-RF and other algorithms is statistically different in all aspects of the MO optimization goals. The MOEA-RF also maintains a stable evolving environment through GASS that defines a concentrated search region. This results in a consistent algorithmic performance as reflected by the small variance of all metrics in Figure 3.24. Conversely, RMOEA shows a large variance for the metric of S, MS and HVR, despite the presence of re-sampling technique in the algorithm.

3.2.4 FON

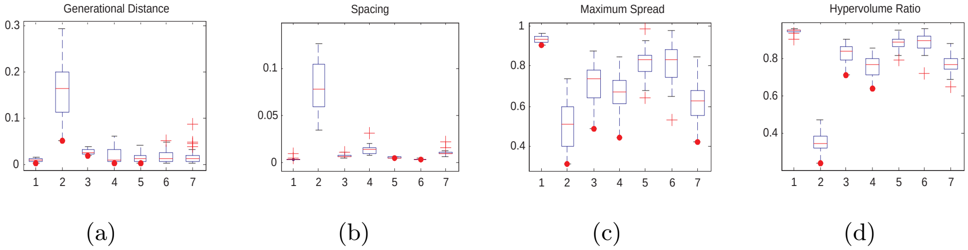
FON challenges the algorithms ability to find and maintain the entire tradeoff curve uniformly. Since the tradeoff curve is non-convex and nonlinear in nature, it is difficult for the algorithms to maintain a stable evolving population for FON especially in a noisy environment. The performance of the different algorithms over the noise levels of $\{0\%, 5\%, 10\%, 20\%\}$ is shown in figure 3.26. The evolved tradeoff from the different algorithms using the same



(a) (b) (c) (d)

(e) (f) (g)

Figure 3.27: The PF^A from (a) MOEA-RF, (b) RMOEA, (c) NTSPEA, (d) MOPSEA, (e) SPEA2, (f) NSGAII, and (g) PAES for FON with 20% noise.

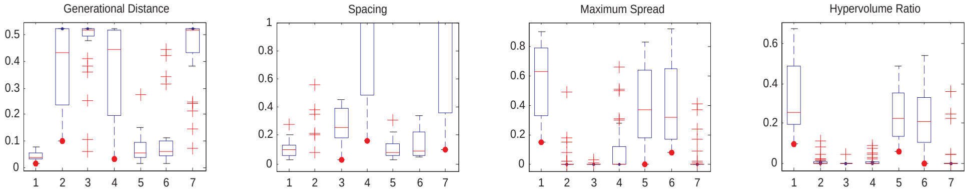


(a) (b) (c) (d)

Figure 3.28: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for FON with 0% noise.

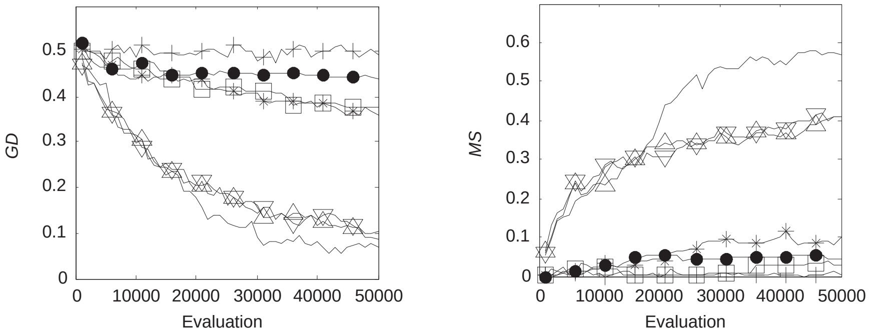
initial population is showed in Figure 3.27(a)-(g) for noise level of 0%. The distribution of the different performance metrics is represented by box plots in Figure 3.28(a)-(d) and Figure 3.29(a)-(d) for 0% and 20% noise respectively.

It can be observed from Figure 3.26 that none of the noise-handling MOEAs provides distinct advantage over NSGAII and SPEA2 for solving FON in noisy environments. In fact, only MOEA-RF is able to match the performance of NSGAII and SPEA2 in terms



(a) (b) (c) (d)

Figure 3.29: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for FON with 20% noise.

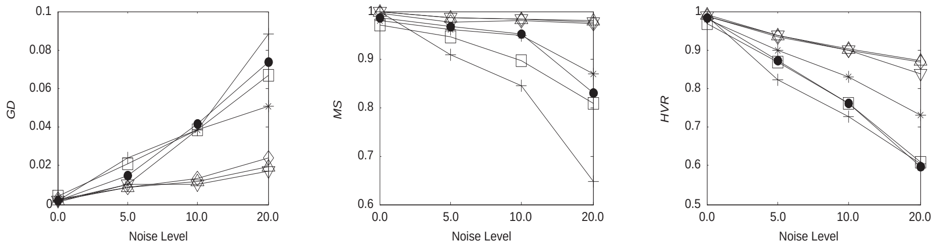


(a) (b)

Figure 3.30: Evolutionary trace of (a) GD and (b) MS for FON with 0% noise.

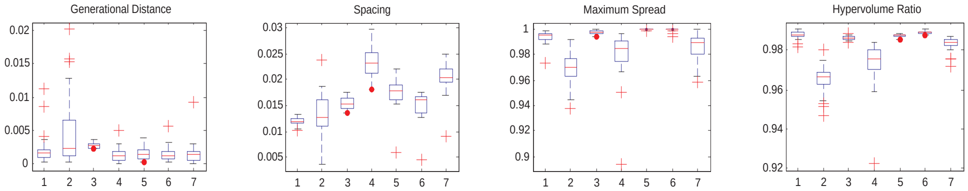
of convergence and diversity over the different noise levels. Conversely, the performance of other algorithms deteriorates drastically at noise levels of 10% and 20% as shown in Figure 32. It can be observed from Figure 3.27 and Figure 3.28 that RMOEA are unable to find the final tradeoff, while other algorithms are capable of finding at least some parts of the optimal Pareto front. The results also show that MOEA-RF offers the best performance in terms of spacing and spread, and the KS test reveals that the performance between MOEA-RF and other algorithms is statistically different in terms of MS, S, and HVR.

From the evolutionary trace in Figure 3.30, it is obvious that RMOEA, NTSPEA, MOPSEA and PAES are unable to improve beyond the initial candidate solutions at 20%



(a) (b) (c)

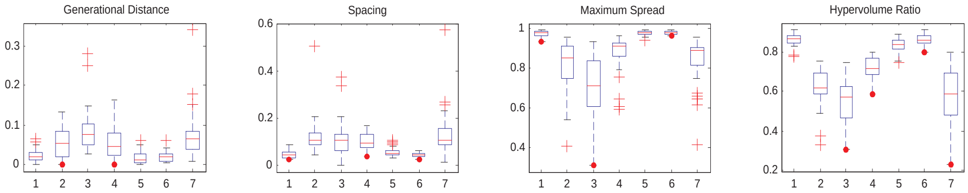
Figure 3.31: Performance metric of (a) GD, (b) MS, and (c) HVR for KUR attained by the algorithms under the influence of different noise levels.



(a) (b) (c) (d)

Figure 3.32: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for KUR with 0% noise.

noise. As a result, NTSPEA, MOPSEA and PAES only manage to find one or two solutions that are far away from the final tradeoff for most of the 30 simulation runs, leading to the high values of GD and S as shown in Figure 3.29(a)-(b). On the other hand, MOEA-RF, SPEA2 and NSGAII are able to discover some individuals that are near to the final tradeoff. The KS test indicates that the three algorithms are rather similar in performance for the various MO optimization metrics. Figure 3.29(d) and Figure 3.30 also show that MOEA-RF has a slight edge in producing better solutions as compared to other algorithms, due to the proposed GASS that concentrates the evolutionary search to reduce the stochastic influence of noise as shown in Figure 3.30(a), where the improvement of convergence for MOEA-RF coincides with the activation of GASS.



(a) (b) (c) (d)

Figure 3.33: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for KUR with 20% noise.

3.2.5 KUR

KUR is characterized by an optimal Pareto front that is non-convex and disconnected, i.e., it contains three distinct disconnected regions on the final tradeoff. The decision variables correspond to the global tradeoff for KUR are difficult to be discovered, since they are disconnected in the decision variable space. The performance of the different algorithms over the noise levels of $\{0\%, 5\%, 10\%, 20\%\}$ is shown in Figure 3.31. The distribution of the different performance metrics is represented by box plots in Figure 3.32(a)-(d) and Figure 3.33(a)-(d) for 0% and 20% noise respectively. Similar to FON, Figure 3.31 shows that MOEA-RF, NSGAII and SPEA2 are better for solving KUR in noisy environments as compared to other algorithms.

Figure 3.32 shows that the global search mechanism of MOEAs generally responds well to the challenges of discontinuity and non-convexity posted by noiseless KUR. Among these algorithms, MOEA-RF, NTSPEA, SPEA2, and NSGAII are capable of finding a diverse and uniformly distributed Pareto front for most of the 30 simulation runs. It can be observed from Figure 3.33(b)-(d) that RMOEA, NTSPEA, MOPSEA and PAES have difficulty in distributing individuals uniformly along the discovered Pareto front in noisy environments. On the other hand, MOEA-RF, SPEA2 and NSGAII give good performance in terms of distribution and diversity under the influence of noise. Besides having similar results for GD and MS, Table 3.3 depicts that MOEA-RF, SPEA2 and NSGAII are capable of archiving

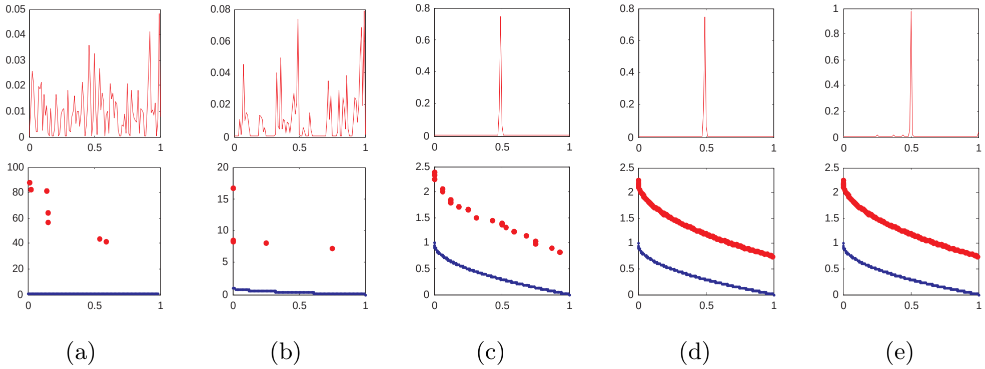
Table 3.3: Number of non-dominated individuals found for the various benchmark problems at 20% noise level

		MOEA-RF	RMOEA	NTSPEA	MOSPEA	SPEA2	NSGAII	PAES
ZDT1	1st quartile	28	6	5	7	18	17	4
	Median	31	7.5	6	9	21.5	19.5	4.5
	3rd quartile	32	10	8	10	27	23	5
ZDT4	1st quartile	29	5	6	9	12	20	5
	Median	33	7	8	11	26.5	15	6
	3rd quartile	41	8	10	13	29	21	9
ZDT6	1st quartile	82	2	4	3	8	8	2
	Median	85	3	5	4	9	9	4
	3rd quartile	88	5	6	5	11	11	6
FON	1st quartile	9	1	1	1	6	6	1
	Median	11	2	1.5	2	8.5	8.5	2
	3rd quartile	17	2	2	3	12	12	3
KUR	1st quartile	25	6	5	8	23	25	7
	Median	27	8	5.5	9	25	27	9
	3rd quartile	30	9	7	11	28	30	10

more non-dominated individuals as compared to other algorithms.

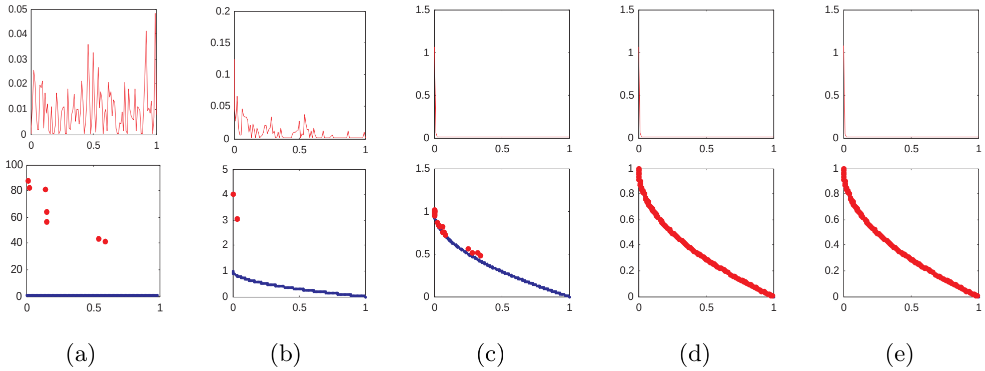
3.3 Effects of The Proposed Features

It can be observed from the comparative studies that MOEA-RF is capable of evolving a near-optimal, diverse and uniformly distributed Pareto front for the different benchmark problems. In this section, the dynamics and parameter settings of ELDP and GASS are examined in the presence of the possibilistic archiving model. Simulation results show that the proposed archiving model plays a complementary but crucial role in the preservation of



(a) (b) (c) (d) (e)

Figure 3.34: The first row represents the distribution of one decision variable and the second row shows the associated non-dominated individuals of baseline MOEA at generation (a) 0, (b) 10, (c) 60, (d) 200, and (e) 350 for ZDT4.



(a) (b) (c) (d) (e)

Figure 3.35: The first row represents distribution of one decision variable and the second row shows the associated non-dominated individuals of baseline MOEA with ELDP at generation (a) 0, (b) 10, (c) 60, (d) 200, and (e) 350 for ZDT4.

good individuals discovered by ELDP and GASS, without which the potential of ELDP and GASS may not be easily exploited. Note that ZDT4 and FON are used in the study here since it has been observed in previous section that most algorithms are unable to deal with these two benchmark problems effectively across the different noise conditions.

The Parzen window density estimation [160] is used to estimate the distribution of individuals in the decision space. Figure 3.34(a)-(e) shows the distribution of one decision

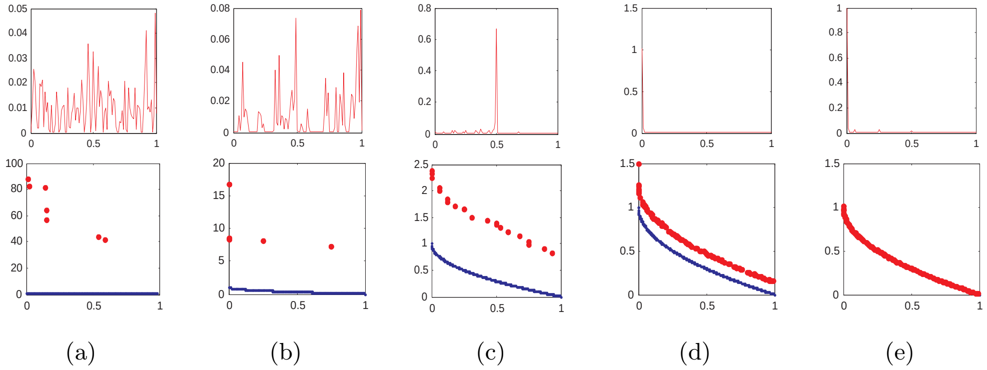


Figure 3.36: The first row represents distribution of one decision variable and the second row shows the associated non-dominated individuals of baseline MOEA with GASS at generation (a) 0, (b) 10, (c) 60, (d) 200, and (e) 350 for ZDT4.

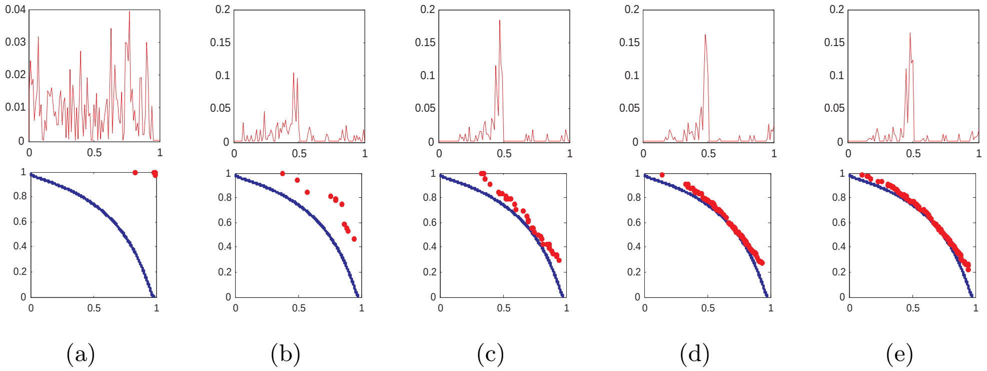
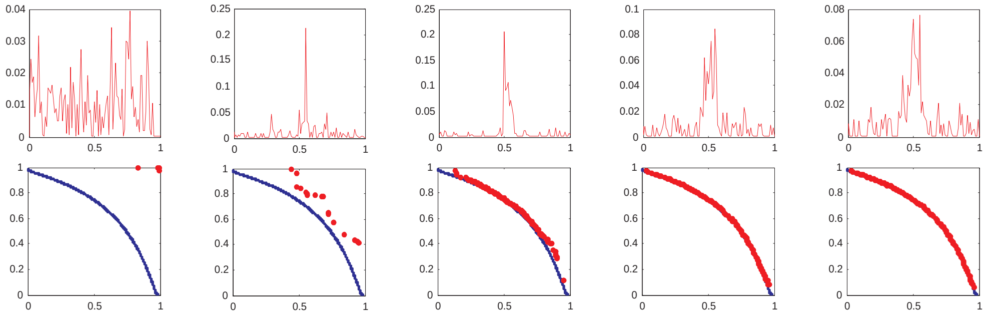


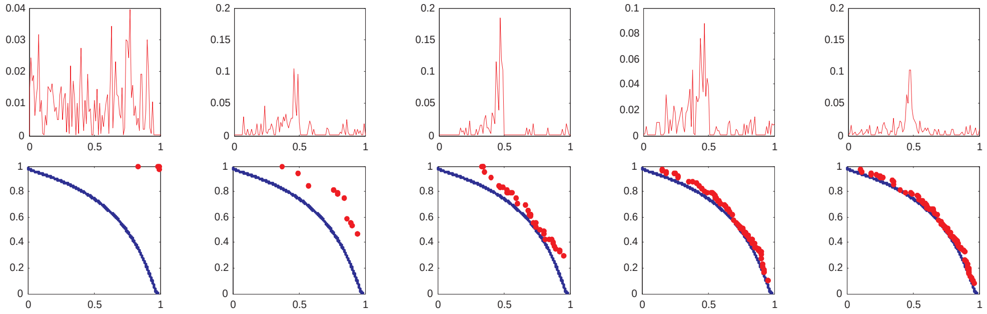
Figure 3.37: The first row represents the distribution of one decision variable and the second row shows the associated non-dominated individuals of baseline MOEA at generation (a) 0, (b) 50, (c) 150, (d) 350, and (e) 500 for FON.

variable and the associated non-dominated individuals of baseline MOEA at the generation of 0, 10, 60, 200, and 350 for ZDT4. Similarly, the effects of ELDP and GASS is shown in Figure 3.35 and Figure 3.36, respectively. To illustrate the working dynamic for the problem of FON, the distribution of one decision variable and the associated non-dominated individuals for baseline MOEA without and with the proposed features at the generation of 0, 50, 150, 350, and 500 are shown in Figure 3.37 to Figure 3.39. Note that the possibilistic



(a) (b) (c) (d) (e)

Figure 3.38: The first row represents the distribution of one decision variable and the second row shows the associated non-dominated individuals of baseline MOEA with ELDP at generation (a) 0, (b) 50, (c) 150, (d) 350, and (e) 500 for FON.



(a) (b) (c) (d) (e)

Figure 3.39: The first row represents the distribution of one decision variable and the second row shows the associated non-dominated individuals of baseline MOEA with GASS at generation (a) 0, (b) 50, (c) 150, (d) 350, and (e) 500 for FON.

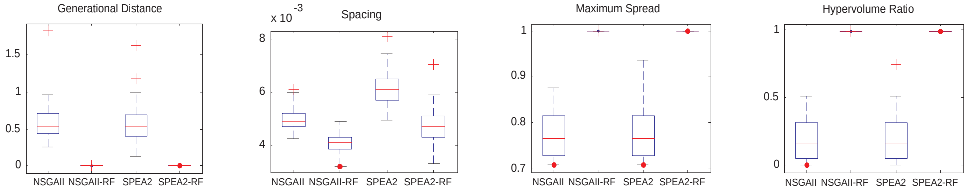
archiving model behaves like the standard archive in the absence of any preference or noise. The distribution and the associated non-dominated individuals demonstrate how the different features influence and improve the optimization process. In addition, it shows whether the proposed features are behaving in accordance to the design specifications.

It can be seen from the figures that ELDP and GASS have a distinct advantage in overcoming local optimality for ZDT4 as well as in finding a diverse tradeoff for FON. By comparing the decision variable distribution and the evolved non-dominated solutions across the different generations, it is evident from Figure 3.35(a)-(c) and Figure 3.38(a)-(c) that the population distribution converges faster when ELDP is incorporated. The slight divergence of the decision variable distribution about the main peak in Figure 3.38(d)-(e) illustrates the local fine-tuning capability of ELDP, which is important in leading the evolution towards the global tradeoff. By comparing the decision variable distribution between Figure 3.34(c) and Figure 3.36(c) as well as between Figure 3.37(c)-(e) and Figure 3.39(c)-(e), it can be seen that the incorporation of GASS results in a diverse distribution of individuals in the decision space. This shows that GASS is capable of diverting the evolution to other search regions upon the detection of a convergence, thus allowing the algorithm to discover the global tradeoff for ZDT4 as well as to achieve a good spread of non-dominated individuals for FON.

To examine the effect of parameter sensitivity for ELDP and GASS, a number of simulations are performed with different settings of α={0.0,0.05,0.1,0.3,0.5} for ELDP and w={0.05,0.1,0.3,0.5} for GASS at noise levels of 0% and 20%. The setting of α=0 for ELDP is equivalent to the operation of bit-flip mutation. Apart from demonstrating that ELDP provides better performances over the bit-flip mutation, it is observed that ELDP and GASS are capable of performing consistently and effectively within a large range of and w settings for ZDT4 and FON at different noise levels.

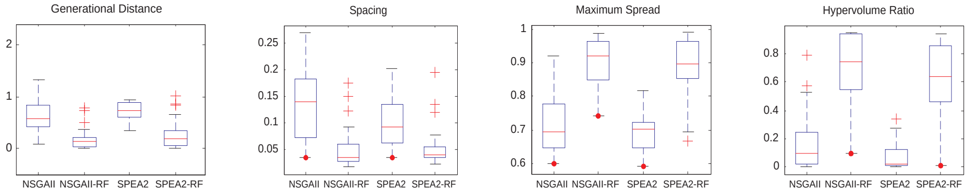
3.4 Further Examination

The results in Sections 3.2 and 3.3 reveal that the proposed features can improve the performance of MO optimization in terms of proximity, diversity and distribution under the influence of noise. In this section, the features of ELDP and GASS are applied to SPEA2 and NSGAII to examine if their effects can be reproduced in conventional MOEAs. The ELDP is used in place of the bit-flip mutation operator in SPEA2 and NSGAII, while the GASS



(a) (b) (c) (d)

Figure 3.40: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT4 with 0% noise.

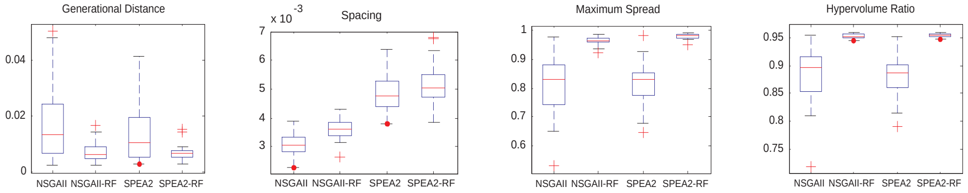


(a) (b) (c) (d)

Figure 3.41: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for ZDT4 with 20% noise.

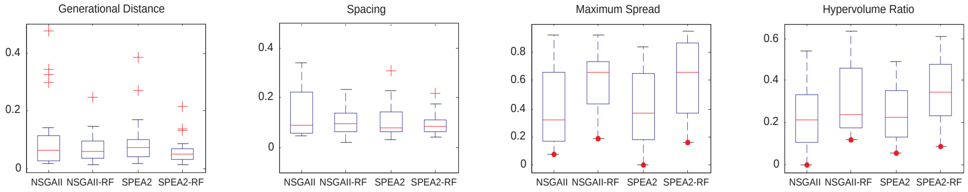
is implemented in conjunction with existing selection schemes. The possibilistic archiving model is not implemented here, since the archiving strategy of SPEA2 and NSGAII plays an important role in defining the behavior of the algorithms.

It has been observed in previous section that SPEA2 and NSGAII can neither discover the global tradeoff for ZDT4 nor maintain a well-distributed set of individuals for FON. The performance of these two algorithms is also largely affected by noise in ZDT4 and FON. Hence, these two benchmark problems are used in the study here. NSGAII-RF and SPEA2-RF denotes the algorithm incorporated with the proposed features. The metric distribution of the simulation results for noiseless and noisy ZDT4 is shown in Figure 3.40(a)-(d) and Figure 3.41(a)-(d), respectively. Similarly, the performance of the algorithms for noiseless and noisy FON is shown in Figure 3.42(a)-(d) and Figure 3.43(a)-(d), respectively.



(a) (b) (c) (d)

Figure 3.42: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for FON with 0% noise.



(a) (b) (c) (d)

Figure 3.43: Performance metric of (a) GD, (b) S, (c) MS, and (d) HVR for FON with 20% noise.

It can be observed from Figure 3.40 - Figure 3.43 that ELDP and GASS are capable of improving the performance of SPEA2 and NSGAII in terms of convergence and diversity of individuals along the tradeoff for ZDT4 and FON. In the case of ZDT4, the incorporation of the proposed features allows NSGAII-RF and SPEA2-RF to escape the local optima in ZDT4. In the case of FON, Figure 3.42 show that the incorporation of ELDP and GASS improves the performance in terms of GD, MS, and HVR. It can also be observed from Figure 3.43 that NSGAII-RF and SPEA2-RF have a slight edge over NSGAII and SPEA2 in almost all aspects of the MO optimization goals.

3.5 Conclusion

Based on the empirical results conducted, three noise-handling features have been proposed in this chapter, including an experiential learning directed perturbation operator that adapts the magnitude and direction of variation according to past experiences for fast convergence, a gene adaptation selection strategy that helps the evolutionary search in escaping from local optima, and a possibilistic archiving model based on the concept of possibility and necessity measures to deal with the problem of uncertainties. It has been shown in the comparative study that basic algorithm incorporating the proposed features exhibit competitive or superior performance in terms of proximity, diversity and distribution for both the noiseless and noisy benchmark problems. Besides, the working dynamics and parameter settings of ELDP and GASS with and without the presence of noise have been examined, which illustrate that the proposed features are robust to different parameter settings and the individual feature of ELDP and GASS plays an important role in the overall evolutionary optimization process. Furthermore, it has been depicted that existing MOEAs such as SPEA2 and NSGAII incorporated with the proposed features of GASS and ELDP are capable of giving better convergence and population diversity along the global tradeoff for the benchmark problems with and without the presence of noise.

Chapter 4

Hybrid Multi-objective Evolutionary Design for Neural Networks

In this chapter, we consider the design of artificial neural networks (ANNs) as an instance of noisy design problem. As mentioned by [107, 219], network architecture optimization is a noisy problem in which the same network structure can give rise to different fitness values due to different weight instantiations. Given that the intrinsic relationship between the architecture and the associated synaptic weights can be quite complex, the design methodology would be flawed if we were to decouple these two properties during the training phase of the network. Local search is applied to optimize the synaptic weights with respect to any new ANN structure introduced to reduce the effects of noise.

4.1 Evolutionary Artificial Neural Networks

EANNs provide a global approach for synaptic weight training, architectural design, rule extraction, etc., and it has been shown to possess several advantages over conventional methods of training. Thus, the field of EANNs is at present, receiving increasing attention from the research community. EANNs provide a platform for the simultaneous optimization of synaptic weights and network architecture. According to Stanley and Miikkulainen [189], it is necessary to evolve network architecture and weights simultaneously to achieve a desirable performance. EANNs have been demonstrated to be less sensitive to initial choice of weights and are also capable of dealing with the issue of premature convergence that is usually associated with traditional gradient-based approaches [60, 137, 220]. However, the simultaneous evolution of both architecture and network weights will inevitably result in a large increase in the size of the search space. While EAs are capable of exploring and identifying promising regions of the search space, they require a relatively longer time to locate the local optima.

With this in mind, many researchers have sought to complement the global exploration ability of EA by incorporating dedicated learning or local search (LS) heuristics. Support from experimental studies has shown that EA-LS hybrids or hybrid EAs are capable of more efficient search capabilities [145, 154]. The backpropagation (BP) algorithm using steepest-gradient is the dominant local search operator in EANN. According to Kinnebrock [119], the number of epochs for training an ANN can be significantly reduced by subjecting the weights of the network to mutation. Yao and Liu [220] applied an Evolutionary Programming (EP) based approach for the simultaneous optimization of the architecture of an ANN and its corresponding synaptic weights. Adaptation of connection weights is based on a hybrid training that comprises of a modified version of the BP algorithm together with simulated annealing (SA). In the event that no improvement in performance is made after this hybrid learning, modifications are made to the ANN structure by means of node deletion, connection deletion, as well as connection and node addition operators. In order to encourage the evolution of compact neural architectures, addition-based operators are employed only if deletion-based operators fail to improve network performance. Verma and Ghosh [217] utilized QR factorization for optimizing the weights in the least-squares sense. Specifically, an EA is used to evolve the hidden layer weights while the least-squares method is applied to optimize the output layer weights. More recently, the authors subsequently proposed a modification to the original algorithm in [70] to reduce the required computational complexity. Furthermore, two different heuristics are used to determine the optimal number of hidden neurons.

While many researchers have acknowledged the importance in considering both architecture and connection weights concurrently, [143, 221, 224, 225] the MO nature of the problem is rarely considered until recently [2, 3, 56, 66, 71]. It should be noted that the design of ANN involves the optimization of two competing objectives, namely the maximization of network capacity and the minimization of neural architecture complexity. Abbass [2] applied a MO approach in the optimization of an ANN, where the proposed memetic Pareto ANN (MPANN) is based on a differential evolution. As is typical with most local search based approaches to EANN, the BP algorithm is used as the local search operator. Similar to [220], the network weights and architecture are evolved simultaneously. The author further extends his original approach by suggesting an alternative self-adaptive algorithm that is claimed to be computationally less intensive. While Abbass [2,3] sought to achieve a good generalization by optimizing the objectives of training accuracy number of hidden neurons simultaneously, Giustolisi and Simeone [71] considers the additional objective of model input dimension. Garcia-Pedrajas *et al* [66] investigated the influence of 10 different objectives and results indicated the advantages of the MO approach over SO approach. MOEAs have also been sucessfully applied to the evolution of neural network ensembles recently [25, 26].

It can also be observed from the literature that any fine-tuning or adaptation of network architectures is mainly stochastic or performance-driven (by the classification accuracies, in our case), which will inevitably result in larger network complexities, as measure by the architectural size, or otherwise known as structural complexity. The primary difficulty in this area is the formulation of an appropriate measure to quantify the contribution of the “information” that emerges during the process of evolution while the learning, or training mechanism takes place.

4.2 Singular Value Decomposition for ANN Design

As mentioned in the previous section, it is difficult to quantify the contribution of additional neurons in the hidden layer without the use of an independent validation set of data. As such, we used a simple, yet robust information measure based on the SVD operator in the framework of EANNs, to achieve this purpose, in removing neurons in the hidden layer of the evolved single hidden layer feedforward neural network.

Computationally, the SVD is very robust and allows the discrimination against noise contamination. Typically, the SVD is utilized in computing the pseudoinverse (Moore-Penrose generalized inverse) of a rectangular, possibly singular, matrix. The SVD has also been extensively applied in problems of least squares, spectral estimation and system identification. In signal processing, the SVD plays a central role in subspace modeling or low-rank approximation (similar to our problem of estimating the number of hidden layer neurons) of signals.

4.2.1 Rank-revealing Decomposition

Consider the output matrix, H, of the hidden layer corresponding to the N training samples and n hidden neurons. The actual rank (say k) of H may be different from its numerical rank (say n), where $k \leq n$. Such a situation usually arises when the original matrix H is contaminated by E resulting in a matrix, $\tilde{H}$

$$\tilde{H} = H + E \tag{4.1}$$

with rank(H)=k and rank($\tilde{H}$)=n. This contamination, commonly referred to noise, obstructs the characterization of the true properties of the system or problem given the observed training samples. This phenomenon actually corresponds to the marginal role played by additional hidden layer neurons that tends to fit the features of the training samples which are not representative of the intrinsic underlying distribution of observations.

Given a real matrix $H \in R^{Nxn}$, applying the SVD results in the orthogonal transformation,

$$\begin{aligned} U^T HV &= [\sum; 0] \\ \sum = \text{diag}(\sigma_1, ...\sigma_n) & \end{aligned} \tag{4.2}$$

where $U \in R^{NxN}$ and $V \in R^{nxn}$ are known as the left and right singular vectors of H. $\sum \in R^{Nxn}$ is a diagonal matrix with unique, nonnegative entries ordered in decreasing magnitude. This decoupling technique of the SVD allows the expression of the original matrix as a sum of the first n columns of u and u^T, weighted by the singular values. The rank of H is determined by observing the n largest singular values that are non-zero.

While SVD do not reveal the actual rank of full-ranked $\tilde{H}$ [121, 190], through the structure of its zero elements, it provides information of the actual rank through the structure of small elements. Let the singular values of H and $\tilde{H}$ be $(\sigma_1, ...\sigma_n)$ and $(\tilde{\sigma_1}, ...\tilde{\sigma_n})$ respectively. From Schmidts Sub-space Theorem, we have

$$\tilde{\sigma}_{k+1}^2 + ... + \tilde{\sigma}_n^2 \leq ||E||_F^2 \tag{4.3}$$

where $|| \, ||_F$ denotes the Frobenius norm, revealing the rank of $\tilde{H}$ such that its $n-k$ smallest singular values are bounded by the Frobenius norm of E.

4.2.2 Actual Rank of Hidden Neuron Matrix

Every neuron in the hidden layer constructs a hyperplane in the input feature space [91] and the contribution of each hidden neuron to the separating capability of the ANN depends on its uniqueness. In a geometrical sense, the rank of H denotes the space in which the columns of H occupy, representing the number of separating hyperplanes in the system. In the case where $n = H - 1$ hidden neurons are used with a suitable nonlinear activation function such that H is full-ranked, the rank requirement [91, 177] is satisfied, giving rise to $N - 1$ separating hyperplanes for N training samples. This follows that, using simple matrix inverse, we are guaranteed perfect reconstruction for the training set. Intuitively, if H is of higher rank, H better fits the training data, at the expense of generalization when $\lim_{n \to N} \text{rank}(H)$. This full-rank condition also ensures that the patterns projected onto the hidden layer space are linearly independent; accordingly this is also known as ϕ-general position [33].

However, it should be noted that even if the transformed patterns produced in the hidden layer space are in general position, some of these patterns may be degenerate in the sense that certain patterns can be represented as a linear combination of other patterns. This leads to issue of whether this additional hidden neuron is contributing to the actual separability of the samples or merely compensating for the presence of noise in the observations. Clearly, there is a limit, for which introducing additional hidden neurons will tend to over-fit the training data. Therefore, the actual rank of the matrix H is more useful in estimating the appropriate number of hidden neurons in the Single-hidden Layer Feedforward Network (SLFN) for a given problem. In the context of representing the input patterns in hidden layer space, we can think of additional hidden layer neurons as causing degeneracy in this hidden layer space, for increasing the number of hidden layer neurons is akin to introducing noise into the system thus perturbing the hidden layer space such that the hidden layer space is now being represented by n hidden neurons which are of marginal benefit.

In Figure 4.1 shown below, using a 2-dimensional toy problem that is easily visualizable, we illustrate the problem of hyperplane construction in hidden layer space and its corresponding relationship with the singular values of the hidden layer output matrix H. From observation, we know that three appropriately placed hyperplanes should provide us with a good balance of network capacity and complexity without sacrificing the generalization capability of the resulting classifier. The placement of these hyperplanes is achieved through the use of learning algorithms (e.g. EA, or backpropagation using gradient descent) on the set of training data. These learning algorithms will usually attempt, to the best of their abilities, to position these hyperplanes such that their construction is as linearly independent as possible for the given set of training data. This usually requires that the learning algorithm has converged prior to using the SVD to decompose the matrix H and obtain the set of singular values. From the diagrams below, shown in input space, the singular values confirm that the use of three hidden layer neurons should be sufficient for the network capacity given the complexity of the problem. We arrive at this conclusion from the presence of a noticeable gap in the decay of singular values, indicating that most of the spectral energy of H can be attributed to the first three hidden layer neurons note that while we may not know the identity of these three most linearly independent hyperplanes (unless we carry out a recalculation of the SVD using a permutation of possible combinations of hidden neurons), we find that it is more efficient to retrain the whole network using the found number of hidden neurons, which in this case is three.

Theorem 4.1: Define the numerical ε-rank k_ε of the matrix H with respect to some tolerance ε [84] by

$$k_\varepsilon = k_\varepsilon(H, \varepsilon) \equiv \min_{||\ E||_2 \leq \varepsilon} \{\text{rank(H+E)}\} \tag{4.4}$$

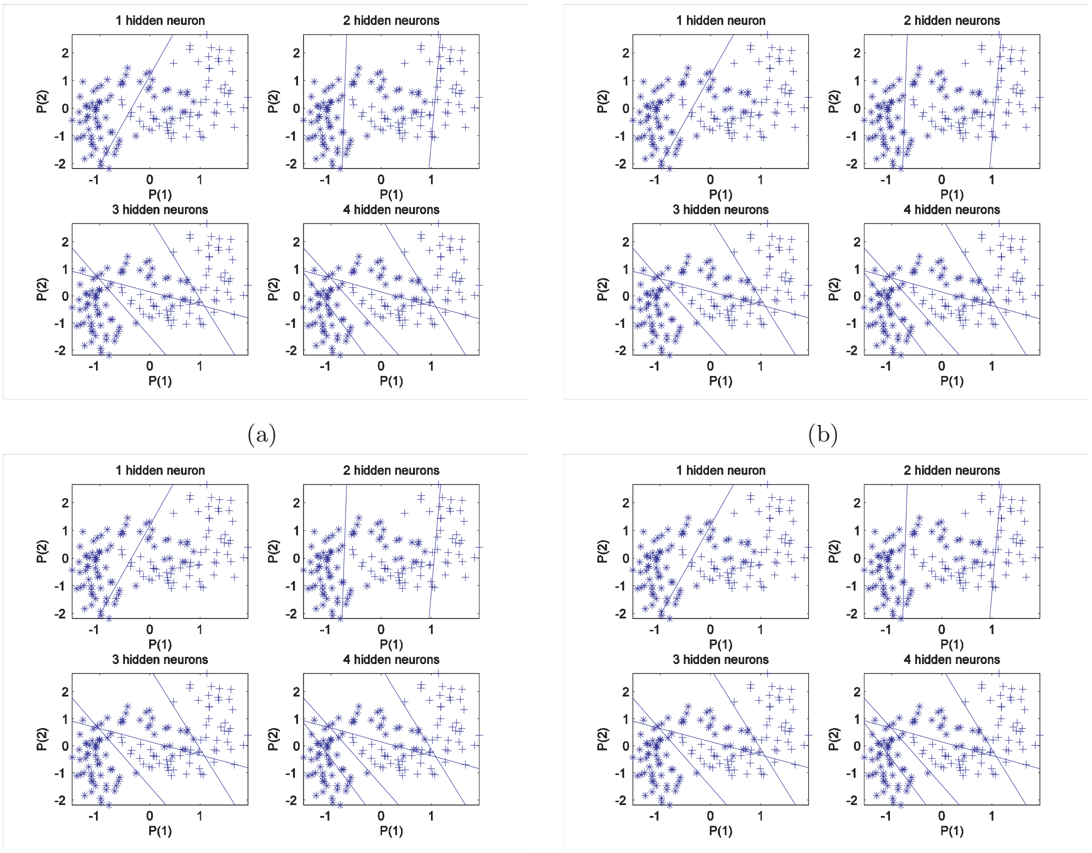
which states that if there is a gap between the k_ε-th and the $k_{\varepsilon+1}$-th singular values of size ε, then H has actual rank (ε-rank) k_ε. The larger this gap ε is, the more robust the matrix is to perturbation. To avoid possible problems when is itself perturbed, the definition of actual rank is refined by introducing δ as an upper bound for $\varepsilon$for which the numerical rank remains at least equal to k.

Theorem 4.2: The matrix H has a numerical rank of (δ, ε, r) with respect to the norm $||\ \cdot ||$ if δ, ε, and r satisfies the following:

$$\begin{aligned} k &= \inf\{\text{rank}(B) : ||\ A - B|| \leq \varepsilon \\ \varepsilon &< \delta \leq \sup\{\eta : ||\ A - B|| \leq \eta \Rightarrow \text{rank}(B) \geq k\} \end{aligned} \tag{4.5}$$

σ_k provides an upper bound for δ while σ_{k+1} provides a lower bound for ε.

The above definitions are equivalent to saying that the matrix H is linearly-independent when perturbed by E up to a threshold determined by $||E||^2 \leq \varepsilon$. This result also means that



(a)

(b)

(c)

(d)

Figure 4.1: (a), (b), (c): Diagram shows constructed hyperplanes in hidden layer space (1-12 hidden neurons); (d): corresponding decay of singular values as number of hidden layer neurons is increased.

the singular values of H satisfies $\sigma_{k_\varepsilon} > \varepsilon > \sigma_{k\varepsilon+1}$. As described in [84], a well-determined gap between the singular values of σ_{k_ε} and $\sigma_{k_{\varepsilon+1}}$, represented by ε should exist in order for the above definition to make much sense; k_ε should be, in other words, well-defined for small perturbations of the threshold ε and the singular values. Alternatively, the numerical ε-rank is the smallest integer k for which (Schmidts Subspace Theorem),

$$\sum_{j=k+1}^{n} \sigma_j^2 \leq \varepsilon^2 \tag{4.6}$$

This result suggests that as more neurons are added to the hidden layer, the contribution of each additional hidden neuron decreases after a certain threshold. From a geometric point of view, additional hyperplanes constructed by these newly introduced hidden neurons are not unique, or different as compared to existing hyperplanes (these new hyperplanes may be almost parallel to existing ones). The significance of these new hyperplanes can be quantified by examining the singular values of the matrix H, as more hidden layer neurons are added. For detailed proofs of these theorems, the interested reader is directed to [121, 190].

In our context, this suggests that we should use a SLFN with a fewer number of hidden neurons since a higher number of neurons in the hidden layer may unnecessarily fit the noise that is inherently present in the samples. As noted in [84], E is typically unknown, but what we do have knowledge of is the source of E, which in turn can be used to estimate the norm of E.

These singular values indicate the degree of mutual correlation between features in the hidden layer space with column degeneracy resulting when these hidden space features are highly correlated, which in turn leads to the conclusion that these additional neurons are redundant. While the singular values do not provide information on which of these features are correlated (the identity of the neurons are not explicitly known), the presence of small singular values would indicate that these additional hidden neurons can be removed without affecting the performance of the SLFN significantly.

4.2.3 Estimating the Threshold

A long-standing problem in the use of the SVD as a tool in determining the actual, or effective rank of a perturbed matrix is in the distinguishing of significantly small and insignificantly large singular values [125].

Suppose $H_n \in R^{Nxn}$ represents the hidden layer output activation matrix of a SLFN with n neurons in the hidden layer. As n increases, the input-output space mapping that is discovered by the MLP better approximates the training data. Increasing n can be seen as increasing the complexity (and hence capacity) of the network, but since all problems would have an inherent degree of complexity, which is essentially unknown estimating this complexity, characterized by k that is in some sense close to the inherent complexity of the problem, is our objective. However, as n is increased, the better fitting of the training data by the MLP gives rise to a lesser ability to generalize on unseen examples (i.e. the training set). Let $\sigma_i(H_n)$ denote the i-th singular value of the SLFN with n hidden neurons. A way to measure the contribution of the i-th singular value to say, the separability of classes, is to relate its value to other singular values in $\sum$.

If ε is large, we can assume that the matrix H is relatively robust to perturbations; conversely, if ε is selected to be small (but not too small such that the numerical ε-rank does not make sense, external noise that is introduced to the system may cause the matrix $H$to be rank-degenerate. Often, there is no clear value for k where $\sigma_k - \sigma_{k+1}$ is obvious. If the SLFN has been well-trained, and has converged (there is little change in its weight values), the decay of the singular values is gradual and not very distinct, and hence we cannot conclude confidently that the numerical rank of matrix H is less than its actual rank. This has been explored in further detail in [202].

4.2.4 Moore-Penrose Generalized Pseudoinverse

To resolve a linear system of the general form $H\beta = T$ is straightforward if the matrix H is square and non-singular. However, under many practical circumstances, the matrix H is usually singular and likely to be rectangular. The Moore-Penrose generalized inverse simplifies the treatment by providing the solution to the linear system in a least-squares sense. The pseudoinverse of H is defined differently depending on the rank and dimensionality of H.

In most practical problem, the system is over-determined and hence would want find the least-square error of $||H\beta - T||_2$ in the presence of the inconsistencies introduced by the additional equations. Thus, β is obtained from $||H\beta - T||_2$. The pseudoinverse can be shown to be the minimum norm least-squares solution of the system, i.e. the pseudoinverse of β, which is $\beta^{\dagger}$, minimizes $||H\beta - T||_2$. For further details on the pseudoinverse, readers are directed to [183].

4.3 Hybrid MO Evolutionary Neural Networks

4.3.1 Algorithmic flow of HMOEN

To design an EANN that is capable of evolving the architecture and weights of the ANN simultaneously, a few features such as variable-length chromosome representation, specialized genetic operator in the form of the SVD-based Architectural Recombination (SVAR), and micro-hybrid genetic algorithm (μHGA) for effective local search are incorporated in HMOEN. The flow of the HMOEN is shown in Figure 4.2. The design process begins with the initialization of population. All individuals will be evaluated according to the objective functions and ranked according to their dominance relationship in the population. The objective functions and ranking scheme will be described in Section 4.3.2.

After the ranking process, non-dominated solutions will be updated into the archive. This paper applies the fixed-sized archive applied in Chapter 2. The selected ANNs will then undergo the process of SVAR, which adapts the network architecture, and the mutation process. In order to reduce the noise presented by the change in network architecture as well as to improve convergence, the offspring are allocated to the for local exploitation. The evolution process repeats until the stopping criterion is satisfied. The mechanisms of SVAR and μHGA are described in Section 4.3.4 and 4.3.5 respectively.

4.3.2 MO Fitness Evaluation

ANN design is cast as a MO optimization problem where a number of objectives such as training accuracy and degree of complexity can be specified. The conflicting objectives

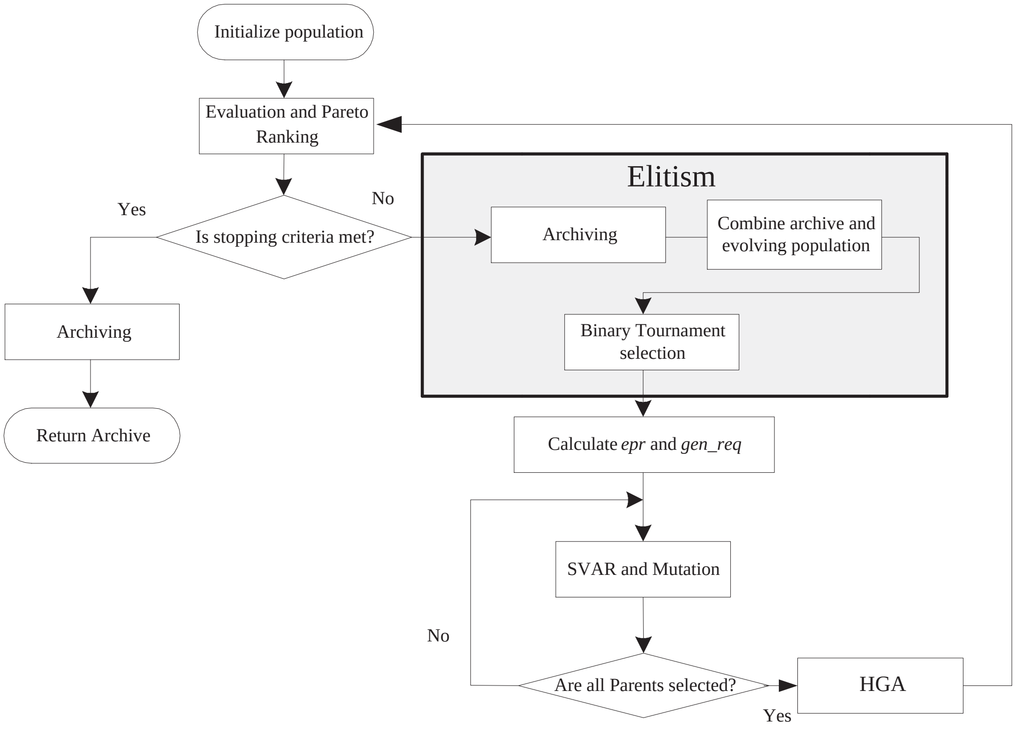

Initialize population
Evaluation and Pareto Ranking
Is stopping criteria met?
Yes
No
Archiving
Return Archive
Elitism
Archiving
Combine archive and evolving population
Binary Tournament selection
Calculate *epr* and *gen_req*
SVAR and Mutation
No
Are all Parents selected?
Yes
HGA

Figure 4.2: Algorithmic Flow of HMOEN.

of maximizing network capacity and minimizing network complexity is manifested in the tradeoffs between training and test accuracy. As before, the Pareto ranking scheme which assigns the same smallest cost for all non-dominated individuals is applied.

One of the primary reasons why a weighted objective is not favored is due to the fact that it is difficult to properly apportion the weights that should be associated with each objective, in converting a MO problem into a SO problem. Most objectives that are considered, such as training accuracy and size of neural network weights are not commensurable (not of the same dimensional quantity) which makes it rather difficult to place these two objectives on a similar platform for comparison.

In this chapter, we consider the simultaneous evolution of both the neural architecture as well as the synaptic weights. Further, this problem is distinguished from previous work by formulating the problem as a MO problem where the twin objectives of classification accuracy and network complexity are conflicting in nature. Therefore, the optimization problem for the ANNs generalization on unseen data can be written as

$$\begin{aligned} f_1 &= \min\Big\{\textstyle\sum_{\theta=1}^{N}\sum_{k=1}^{C}(y_k(\theta)-d_k(\theta))^2\Big\} \\ f_2 &= \min\{NH_k\} \\ f_3 &= \min\{||W_k||_2\} \end{aligned} \tag{4.7}$$

where f_1 refers to the sum-of-squared (SSE) errors of the classification errors. In our problem formulation, we use only one hidden layer of neurons (k=1), as dictated by the Universal Approximation Capability (UAC) theorem for neural networks. N is the number of samples, C is the number of class and d_k is the desired output.

The two other objectives that we consider in our MO approach are firstly, the minimization of the number of neurons in the hidden layer (f_2), and secondly, the minimization of the L$_2$-norm of the hidden layer weights (f_3). We consider each in turn, as will be demonstrated later in our experimental results typically, the use of f_2 and f_3 in addition to f_1 leads to improved generalization performance (as compared to the use of f_1 as the sole objective to me maximized) as the size, or complexity of the network is now controlled, although there is little distinction, empirically, of which of the two additional objectives to be minimized (f_2 or f_3) leads to better testing accuracies. We will address this issue in a later section of this chapter.

4.3.3 Variable Length Representation for ANN Structure

EAs process a set of encoded parameters, providing EA designers with the flexibility to design an appropriate representation of the potential solutions. Appropriate representation implies that it fulfills some criteria such as ease of implementation or exploitation of the problem structure. For simplicity, the chromosome is often represented as a fixed-structure and the embedded variables are usually assumed to be independent and context insensitive.

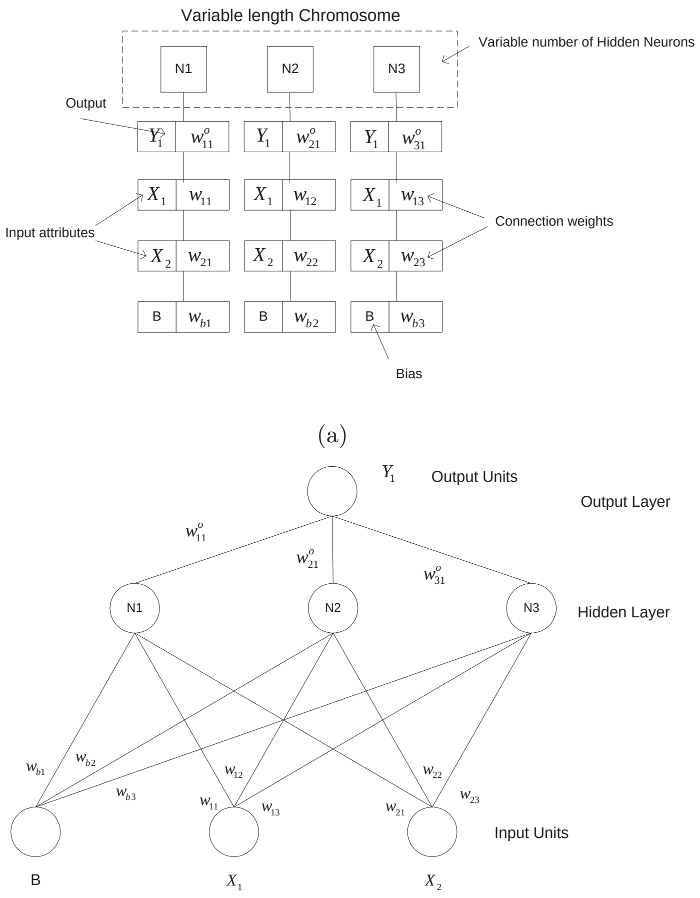
In EANNs, a hybrid structure between binary and real-number representation is commonly used for the simultaneous optimization of weights and architecture. However, such a representation is not suitable for ANN design problem where flexibility is essential for the simultaneous evolution of architecture and connection weights.

In this chapter, a variable-length chromosome representation is adopted to represent the ANN topology including the number of neurons in the hidden layers and the connection weights linking the input, hidden and output layer as illustrated in Figure 4.3(a). Figure 4.3(b) is the instantiation of the representation in Figure 4.3(a). Each neuron is coupled with its associated weights, thus allowing easy manipulated by search operators for the addition or deletion of neurons. The chromosome may consist of different number of neurons which reflects the complexity of the ANN but the number of connections is fixed by the number of input attributes. Such a representation is efficient and facilitates the design of problem-specific genetic operators.

4.3.4 SVD-based Architectural Recombination

In EANNs, the recombination process between two ANNs is unlikely to produce a good offspring due to the lack of a clear definition of a building block in the framework of ANN [219]. However, the lack of recombination to facilitate the exchange of information between candidate solutions implies that each individual is expected to adapt independently by making best use of all available local information. This motivates the development of the SVAR approach which is based on the fact that each neuron constructs a hyperplane in the input feature space and hence contributes to the resulting separating capability of the ANN. It follows that each neuron and its associated connections defines a building block which contributes to the capacity of the ANN.

The issues considered in the design of the SVAR operator include the selection of the appropriate neuron and its associated weights for recombination as well as the decision to remove or add an appropriate neuron to the candidate ANN design. SVAR is performed



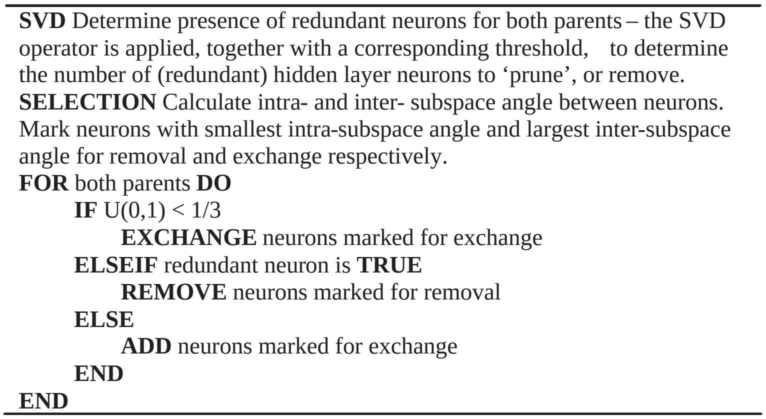
(a)

(b)

Figure 4.3: An instance of the variable chromosome representation of ANN and (b) the associate ANN.

between two parent ANNs and the procedure is outlined in the pseudocode shown in Figure 4.4.

For our proposed approach, the building blocks of each network are the set of neurons (together with its incoming weights from the previous layer). These, we call the building blocks, as they are the smallest units for which we operate on (such as performing crossover). The SVD is used as the tool to determine the presence of redundant neurons while the calculation of inter- and intra- subspace angles is used for the selection of neurons to be



```
SVD Determine presence of redundant neurons for both parents – the SVD operator is applied, together with a corresponding threshold,  to determine the number of (redundant) hidden layer neurons to ‘prune’, or remove.
SELECTION Calculate intra- and inter- subspace angle between neurons. Mark neurons with smallest intra-subspace angle and largest inter-subspace angle for removal and exchange respectively.
FOR both parents DO
    IF U(0,1) < 1/3
        EXCHANGE neurons marked for exchange
    ELSEIF redundant neuron is TRUE
        REMOVE neurons marked for removal
    ELSE
        ADD neurons marked for exchange
    END
END
```

Figure 4.4: Pseudocode of SVAR.

removed or exchanged the SVD operator (together with the corresponding threshold) will decide the number of hidden layer neurons to be removed.

The number of neurons in the hidden layers that are deemed redundant by the SVD operator is in effect, a function of the threshold that is used, with ε assuming the role of the SVD threshold. In deciding *which* hidden layer neuron to remove or prune, we use a geometrical approach, where the algorithm examines the subspace spanned between the hidden layer neurons such that the neuron(s) which is (are) most linearly correlated with the other hidden layer neurons are consequently removed. This is to prevent unnecessary removal of a neuron at the initial stages where the weights are not yet adapted to the problem.

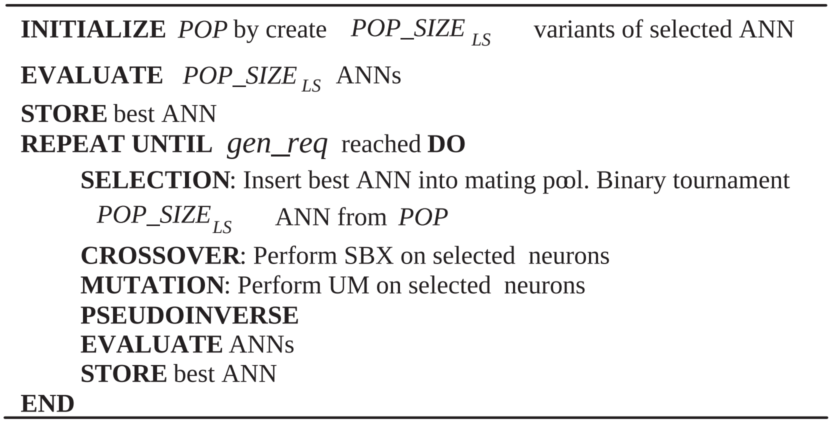
The rationale for utilizing subspace angles as the selection criterion for pruning and exchange is to encourage the linear independency between the neurons. In [202], the authors use the SVD operator to first determine the appropriate, or necessary number of hidden layer neurons on an initially large structured feedforward neural network, after which, the network is retrained using the same learning algorithm, but using the reduced set of hidden layer neurons. In our approach, however, we adopt an online method, in that during the evolutionary process, the identity of the hidden layer neuron(s) to be removed are determined geometrically, by the subspace approach as described above. From the flowchart, it is observed that it is possible for a candidate ANN with redundant neurons to retain the same structure via the exchange of neurons from the other parent.

4.3.5 Micro-Hybrid Genetic Algorithm

As mentioned in the introduction, the optimization of neural network structure is an inherently noisy problem, i.e., the immediate neural network fitness after the recombination process may not be a good indicator of the new network structure due to an inappropriate set of weights. Thus, it is necessary to optimize the synaptic weights with respect to the new ANN structure after any structural changes. The mutation operator offers a simple option for local fine-tuning [103]. However, domain information cannot be easily incorporated and its stochastic nature tends to render the search operation inefficient. Intuitively, the change in genetic structure should be ordered instead of being left to chance in order for the local search to be robust. While the well-known back-propagation (BP) algorithm is a directed by means of gradient descent, it is prone to being trapped in local optima. In view of these concerns, the EANN is hybridized with the μHGA .

μHGA

The μHGA exploits the synergy between a μGA [113] and the pseudoinverse operator to decompose the large and complex search space. Specifically, the μGA performs local fine-tuning of the hidden layer weights while the pseudoinverse optimizes output weights in the least squares sense based on the weights found by the μGA. The pseudocode of the μHGA is shown in Figure 4.5 where POP_SIZE_{LS} is the population size of the μHGA. In this paper, simulated binary crossover (SBX) and uniform mutation (UM) is applied to evolve



```
INITIALIZE POP by create $POP\_SIZE_{LS}$ variants of selected ANN
EVALUATE $POP\_SIZE_{LS}$ ANNs
STORE best ANN
REPEAT UNTIL gen_req reached DO
    SELECTION: Insert best ANN into mating pool. Binary tournament
      $POP\_SIZE_{LS}$ ANN from POP
    CROSSOVER: Perform SBX on selected neurons
    MUTATION: Perform UM on selected neurons
    PSEUDOINVERSE
    EVALUATE ANNs
    STORE best ANN
END
```

Figure 4.5: Pseudocode of μHGA.

the desired set of connection weights. The mutation strength of UM is adapted as,

$$s \quad = \quad 0.1 \cdot (uppbd_w - lowbd_w) \tag{4.8}$$

where $uppbd_w$ and $lowbd_w$ corresponds to the minimum, maximum of the associated weights in the population.

Balance between Evolution and Learning

While the incorporation of local search can accelerate convergence of the evolutionary optimization process, hybrid EAs also give rise to issues pertinent to the tradeoffs between evolution and learning. Apart from the obvious the challenge posed by limited computational resources, balance between exploration and exploitation is necessary to maintain diversity in the evolving population for the approximation of the Pareto optimal front. Consequently, these concerns have lead to the recent development of resource utilization schemes such as local search probability [103] and simulated heating [9].

While local search probability can reduce the computational time utilized for local fine-tuning, the exploration-exploitation dilemma is not explicitly considered. The fundamental idea behind simulated heating is based on simulated annealing, where the intensity of local search increases with time. Although, it is intuitive that more computational time for local search should be allocated in the later stages, online search requirements are not considered in simulated heating. In contrast to existing methods which allocate resources based on a predetermined schedule, this paper adapts the allocation of resources based on the feedback of an online performance measure, the evolutionary progress rate [196]. The evolutionary progress rate (epr(n)) can be defined as the ratio of the number of new non-dominated solutions discovered in generation n, $new_nondomsol(n)$, to the total number of non-dominated solutions, $total_nondomsol(n)$,

$$\mathrm{epr}(n) \quad = \quad \frac{new\_nondomsol(n)}{total\_nondomsol(n)} \tag{4.9}$$

The set of new non-dominated individuals discovered at each generation is basically composed of individuals that dominate the non-dominated individuals of the previous generation and individuals that contribute to the diversity of the solution set.

In this adaptive scheme, the number of individuals allocated for LS is adapted based on the epr(n) in every generation. Mathematically, the adaptation of computational resource allocation can be written as,

$$gen\_req(n) \quad = \quad (1 - \mathrm{epr}(n)) \cdot (upp\_bd_{com} - low\_bd_{com}) + low\_bd_{com} \tag{4.10}$$

where gen_req is the number of generations performed by μHGA while upp_bd_{com} and low_bd_{com} denote the upper and lower limits of available computational resource. The rationale is that a high value of epr(n) means that the algorithm is still in the exploratory stage and the need for local fine-tuning is low. Likewise, a low value of epr(n) is an indication of convergence and more resources are required to meet the requirements of local fine-tuning. In this chapter, upp_bd_{com} and low_bd_{com} are set as 20 and 10 of the total population size respectively.

Table 4.1: Parameter settings of HMOEN for the simulation study

Parameter	Settings
Population	Main Population size: 20; Archive size: 20
	μHGA: 4.
Chromosome	HMOEN: Variable Length real number representation;
	μHGA: Real number representation;
Selection	Binary tournament selection
Crossover operator	HMOEN: SVAR
	μHGA: SBX
Crossover rate	0.9
Distribution index	10
Mutation operator	Normally distributed mutation
Mutation rate	0.01
Mutation strength	Adaptive
Diversity operator	Niche count with radius 0.01 in the normalized objective space.

4.4 Experimental Study

4.4.1 Experimental Setup

In order to evaluate the effectiveness of the proposed methods, a detailed empirical study is carried out on seven different datasets. HMOEN is implemented using the MATLAB technical computing platform, and corresponding simulations are performed on an Intel Pentium 4 2.8 GHz computer. Thirty independent runs are performed for each of the dataset to obtain statistical information such as consistency and robustness of the algorithms. The various parameter settings of HMOEN are tabulated in Table 4.1.

In the training phase for the classifiers, we use 30-fold cross-validation, partitioning the data into two independent training and testing sets. 60% of the available samples are randomly selected as training data, with the remaining 40% as testing data. Prior to training, pre-processing is carried on the samples of each dataset. All input features are scaled and transformed such that the resulting input features have a mean of 0 and a variance of 1, as it has been shown that convergence is usually faster if the average of each input variable over the training set is close to zero [135]. For the outputs, since we consider classification problems, we use binary target values with a 1-out-of-C encoding where for a C-class problem, the largest output i is assigned to class i, with i={1,2,,C}. For the th training sample, the desired class output c where $d_k(\theta) = \{0, 1\}$ is 1 for $k = c$ and 0 otherwise. This is essentially a winner-take-all approach for the output layer neurons, and is a common approach used for classification purposes. Hidden layer neurons use a hyperbolic tangent nonlinearity, while the output nodes use a linear output activation function.

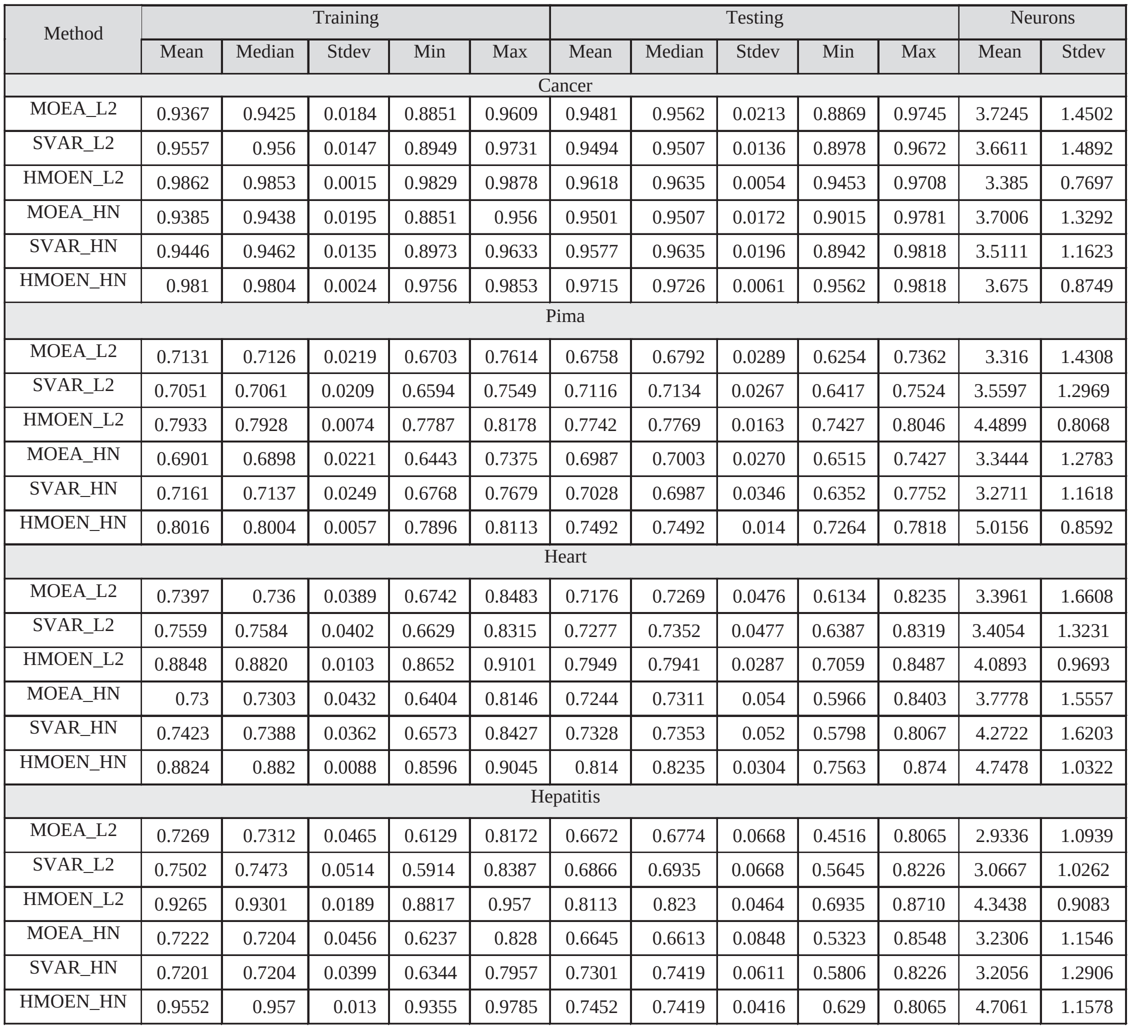
The real-world datasets used in this paper, represent some of the most challenging problems in machine learning, were obtained from the UCI machine learning database (http://www.ics.uci.edu/ mlearn/MLRepository.html). Many researchers have used these datasets in validating the performances of their algorithms, and thus these datasets provide a good test suite of problem for evaluation of the proposed approach. The key characteristics of these problems and their associated learning tasks are summarized in Table 4.2.

4.4.2 Experimental Results

The objective of this section is to establish the effectiveness of the proposed features of both SVAR and μHGA in the design of ANNs. By "design", we mean both the training of the network connection weights as well as the evolved structure of the network. In order to demonstrate their effectiveness of the individual features, the ANNs are evolved with the same MOEA with SVAR and μHGA incorporated incrementally in different setups. The different setups optimizing f_1 and f_2 for MOEA (without both SVAR and μHGA), MOEA (with only SVAR) and HMOEN are denoted as MOEA_HN, SVAR_HN, and HMOEN_HN respectively. Likewise, the different setups optimizing f_1 and f_3 for MOEA (without both SVAR and μHGA), MOEA (with only SVAR) and HMOEN are denoted as MOEA_L2, SVAR_L2, and HMOEN_L2 respectively. For all setups, evolution of the ANNs is terminated

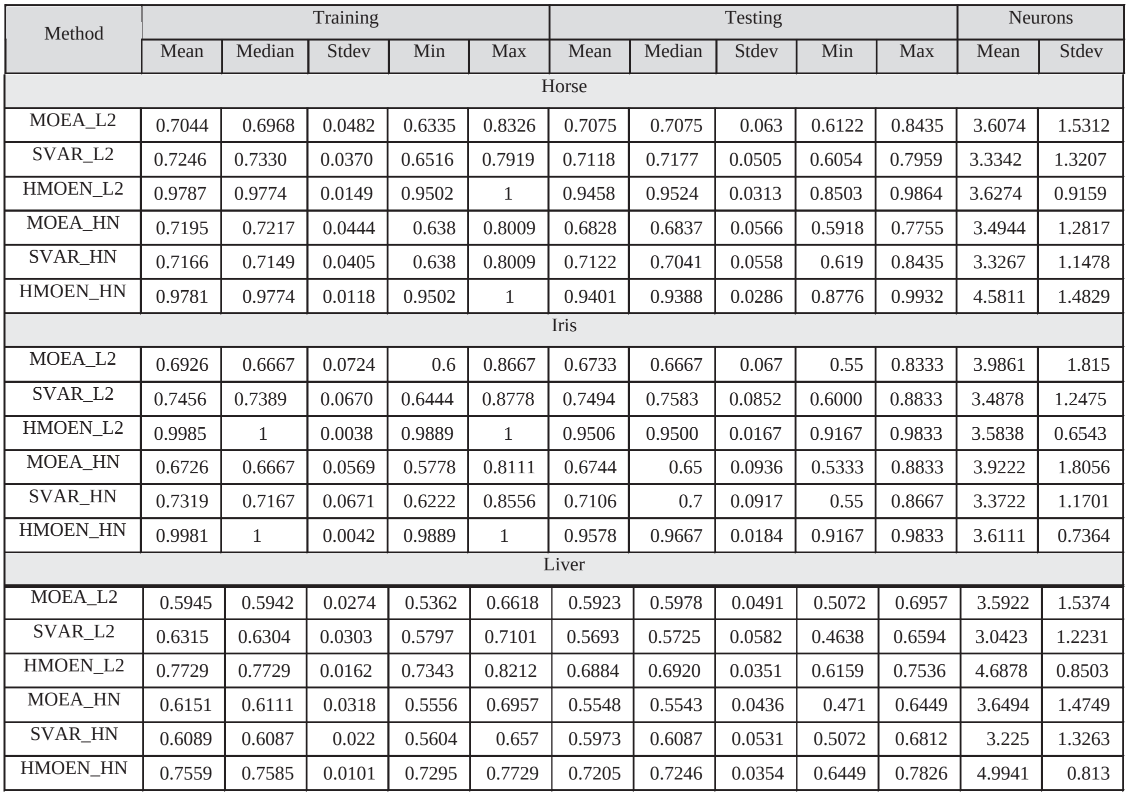
Table 4.2: Characteristics of Data Set

Dataset	Samples	Attributes	Classes	Remarks
Cancer	699	9	2	Determine the patients for whom the tumour is benign or malignant
Pima	768	8	2	Determine whether a patient shows sign of diabetes according to World Health Organization Criteria
Heart	297	13	2	The learning task is to predict the presence or absence of heart disease given the results of various medical tests carried out on a patient.
Hepatitis	155	19	2	The hepatitis problem is a complex and noisy dataset as it contains a large number of missing data (there are 167 missing values in total in this dataset). The learning task is to predict whether a patient with hepatitis will live or die.
Horse	368	27	2	The objective here is to determine, based on the physical ailments and attributes of a particular horse, if it should have surgery performed on it.
Iris	150	4	3	This dataset is perhaps the best-known database to be found in pattern recognition literature. One class is linearly separable from the other two; the latter are NOT linearly separable from each other.
Liver	345	7	2	The learning task for this dataset is to determine, if the adult male that is tested using blood tests suffer from liver disorders that might arise from excessive alcohol consumption



Method	Training					Testing					Neurons	
	Mean	Median	Stdev	Min	Max	Mean	Median	Stdev	Min	Max	Mean	Stdev
Cancer												
MOEA_L2	0.9367	0.9425	0.0184	0.8851	0.9609	0.9481	0.9562	0.0213	0.8869	0.9745	3.7245	1.4502
SVAR_L2	0.9557	0.956	0.0147	0.8949	0.9731	0.9494	0.9507	0.0136	0.8978	0.9672	3.6611	1.4892
HMOEN_L2	0.9862	0.9853	0.0015	0.9829	0.9878	0.9618	0.9635	0.0054	0.9453	0.9708	3.385	0.7697
MOEA_HN	0.9385	0.9438	0.0195	0.8851	0.956	0.9501	0.9507	0.0172	0.9015	0.9781	3.7006	1.3292
SVAR_HN	0.9446	0.9462	0.0135	0.8973	0.9633	0.9577	0.9635	0.0196	0.8942	0.9818	3.5111	1.1623
HMOEN_HN	0.981	0.9804	0.0024	0.9756	0.9853	0.9715	0.9726	0.0061	0.9562	0.9818	3.675	0.8749
Pima												
MOEA_L2	0.7131	0.7126	0.0219	0.6703	0.7614	0.6758	0.6792	0.0289	0.6254	0.7362	3.316	1.4308
SVAR_L2	0.7051	0.7061	0.0209	0.6594	0.7549	0.7116	0.7134	0.0267	0.6417	0.7524	3.5597	1.2969
HMOEN_L2	0.7933	0.7928	0.0074	0.7787	0.8178	0.7742	0.7769	0.0163	0.7427	0.8046	4.4899	0.8068
MOEA_HN	0.6901	0.6898	0.0221	0.6443	0.7375	0.6987	0.7003	0.0270	0.6515	0.7427	3.3444	1.2783
SVAR_HN	0.7161	0.7137	0.0249	0.6768	0.7679	0.7028	0.6987	0.0346	0.6352	0.7752	3.2711	1.1618
HMOEN_HN	0.8016	0.8004	0.0057	0.7896	0.8113	0.7492	0.7492	0.014	0.7264	0.7818	5.0156	0.8592
Heart												
MOEA_L2	0.7397	0.736	0.0389	0.6742	0.8483	0.7176	0.7269	0.0476	0.6134	0.8235	3.3961	1.6608
SVAR_L2	0.7559	0.7584	0.0402	0.6629	0.8315	0.7277	0.7352	0.0477	0.6387	0.8319	3.4054	1.3231
HMOEN_L2	0.8848	0.8820	0.0103	0.8652	0.9101	0.7949	0.7941	0.0287	0.7059	0.8487	4.0893	0.9693
MOEA_HN	0.73	0.7303	0.0432	0.6404	0.8146	0.7244	0.7311	0.054	0.5966	0.8403	3.7778	1.5557
SVAR_HN	0.7423	0.7388	0.0362	0.6573	0.8427	0.7328	0.7353	0.052	0.5798	0.8067	4.2722	1.6203
HMOEN_HN	0.8824	0.882	0.0088	0.8596	0.9045	0.814	0.8235	0.0304	0.7563	0.874	4.7478	1.0322
Hepatitis												
MOEA_L2	0.7269	0.7312	0.0465	0.6129	0.8172	0.6672	0.6774	0.0668	0.4516	0.8065	2.9336	1.0939
SVAR_L2	0.7502	0.7473	0.0514	0.5914	0.8387	0.6866	0.6935	0.0668	0.5645	0.8226	3.0667	1.0262
HMOEN_L2	0.9265	0.9301	0.0189	0.8817	0.957	0.8113	0.823	0.0464	0.6935	0.8710	4.3438	0.9083
MOEA_HN	0.7222	0.7204	0.0456	0.6237	0.828	0.6645	0.6613	0.0848	0.5323	0.8548	3.2306	1.1546
SVAR_HN	0.7201	0.7204	0.0399	0.6344	0.7957	0.7301	0.7419	0.0611	0.5806	0.8226	3.2056	1.2906
HMOEN_HN	0.9552	0.957	0.013	0.9355	0.9785	0.7452	0.7419	0.0416	0.629	0.8065	4.7061	1.1578

Figure 4.6: Performance Comparison between the Different Experimental Setups. The Figure Shows the Classification Accuracy and Mean Number of Hidden Neurons in the Archive for Cancer, Pima, Heart and Hepatitis Datasets.



Method	Training					Testing					Neurons	
	Mean	Median	Stdev	Min	Max	Mean	Median	Stdev	Min	Max	Mean	Stdev
Horse												
MOEA_L2	0.7044	0.6968	0.0482	0.6335	0.8326	0.7075	0.7075	0.063	0.6122	0.8435	3.6074	1.5312
SVAR_L2	0.7246	0.7330	0.0370	0.6516	0.7919	0.7118	0.7177	0.0505	0.6054	0.7959	3.3342	1.3207
HMOEN_L2	0.9787	0.9774	0.0149	0.9502	1	0.9458	0.9524	0.0313	0.8503	0.9864	3.6274	0.9159
MOEA_HN	0.7195	0.7217	0.0444	0.638	0.8009	0.6828	0.6837	0.0566	0.5918	0.7755	3.4944	1.2817
SVAR_HN	0.7166	0.7149	0.0405	0.638	0.8009	0.7122	0.7041	0.0558	0.619	0.8435	3.3267	1.1478
HMOEN_HN	0.9781	0.9774	0.0118	0.9502	1	0.9401	0.9388	0.0286	0.8776	0.9932	4.5811	1.4829
Iris												
MOEA_L2	0.6926	0.6667	0.0724	0.6	0.8667	0.6733	0.6667	0.067	0.55	0.8333	3.9861	1.815
SVAR_L2	0.7456	0.7389	0.0670	0.6444	0.8778	0.7494	0.7583	0.0852	0.6000	0.8833	3.4878	1.2475
HMOEN_L2	0.9985	1	0.0038	0.9889	1	0.9506	0.9500	0.0167	0.9167	0.9833	3.5838	0.6543
MOEA_HN	0.6726	0.6667	0.0569	0.5778	0.8111	0.6744	0.65	0.0936	0.5333	0.8833	3.9222	1.8056
SVAR_HN	0.7319	0.7167	0.0671	0.6222	0.8556	0.7106	0.7	0.0917	0.55	0.8667	3.3722	1.1701
HMOEN_HN	0.9981	1	0.0042	0.9889	1	0.9578	0.9667	0.0184	0.9167	0.9833	3.6111	0.7364
Liver												
MOEA_L2	0.5945	0.5942	0.0274	0.5362	0.6618	0.5923	0.5978	0.0491	0.5072	0.6957	3.5922	1.5374
SVAR_L2	0.6315	0.6304	0.0303	0.5797	0.7101	0.5693	0.5725	0.0582	0.4638	0.6594	3.0423	1.2231
HMOEN_L2	0.7729	0.7729	0.0162	0.7343	0.8212	0.6884	0.6920	0.0351	0.6159	0.7536	4.6878	0.8503
MOEA_HN	0.6151	0.6111	0.0318	0.5556	0.6957	0.5548	0.5543	0.0436	0.471	0.6449	3.6494	1.4749
SVAR_HN	0.6089	0.6087	0.022	0.5604	0.657	0.5973	0.6087	0.0531	0.5072	0.6812	3.225	1.3263
HMOEN_HN	0.7559	0.7585	0.0101	0.7295	0.7729	0.7205	0.7246	0.0354	0.6449	0.7826	4.9941	0.813

Figure 4.7: Performance Comparison between the Different Experimental Setups. The Figure Shows the Classification Accuracy and Mean Number of Hidden Neurons in the Archive for Horse, Iris and Liver datasets.

once the average f_1 value of the archived solutions stops improving. The results are shown in Figure 4.6 and Figure 4.7.

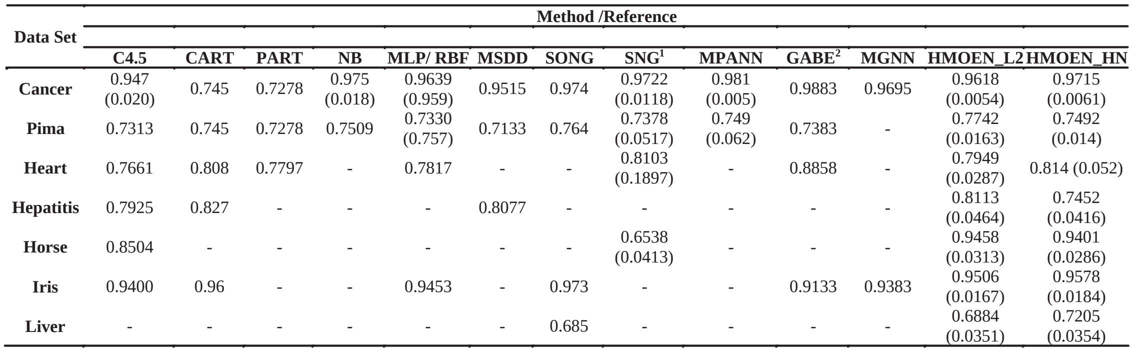
From the classification results, we see that the HMOEN_HN and HMOEN_L2, combining μHGA and SVAR clearly perform much better than that of MOEA_HN and MOEA_L2. Obviously, without the use of the SVD as a form of capacity control in SVAR, the performance of MOEA_HN and MOEA_L2 is inferior to SVAR_L2 and SVAR_HN for most of the problem. These results substantiate our earlier hypothesis that each (hidden) neuron, together with its corresponding hidden layer weights (leading from the input layer to the hidden layer), functions as a building block for an EANN. The specialized recombination operator acts specifically on these neuronal building blocks. This notion is intuitively appealing because when viewed from the perspective of hidden layer space, each hidden neuron and its input set of weights (for the single hidden layer) constructs a separating hyperplane in hidden layer space; thus, each hidden neuron together with its corresponding set of input weights, which are treated as a set of building blocks, are accountable for determining the separation of the training samples in hidden layer space.

The introduction of μHGA provides substantial improvements to the classification performance on the testing sets of all the datasets. By comparing the performances of between SVAR_HN and HMOEN_HN and between SVAR_L2 and HMOEN_L2 from Figures 4.6 and 4.7, it can be noted that the local search ensures that the final network is sufficiently well-trained such that the SVD is able to operate on the hidden layer activation matrix effectively. Recall that the use of the SVD, as described earlier requires the network to be "well-trained". In other words, without μHGA, the SVAR tend to remove neurons excessively as reflected in the low number of hidden neurons (large number of neurons are pruned). Therefore, it is also evident that SVAR and μHGA are complementary mechanisms in HMOEN.

Having validated the effectiveness of μHGA and SVAR, the performance of HMOEN_L2 and HMOEN_HN are compared against other works in the literature using these datasets. These works includes some well-known algorithms (C4.5 [5,164,194], CART [146,194], PART [64,194], NB [112,194], MSDD, SONG [98]) as well as recent EANN approaches (SNG [1] [66], MPANN [2,3], GABE [2] [24] and MGNN [158]). The summary of results is shown in Figure 4.8. We note that comparisons between the results obtained from different approaches have to be made cautiously, as there are numerous ways in which the experimental and simulation setups are done, for example, the training/testing ratio, the pre/post-processing, the cross-validation runs, etc. The results that are presented here are not fine-tuned in any manner, i.e., the same parameter and experimental settings are used for all the datasets. With the

[1] Results recorded are based on the performance of a SiNGle ANN (SNG) as opposed to an ensemble

[2] Results recorded are based on the performance of the ANNs using Genetic Algorithm with Baldwinian Evolution (GABE)



Data Set	Method /Reference												
	C4.5	CART	PART	NB	MLP/ RBF	MSDD	SONG	SNG[1]	MPANN	GABE[2]	MGNN	HMOEN_L2	HMOEN_HN
Cancer	0.947 (0.020)	0.745	0.7278	0.975 (0.018)	0.9639 (0.959)	0.9515	0.974	0.9722 (0.0118)	0.981 (0.005)	0.9883	0.9695	0.9618 (0.0054)	0.9715 (0.0061)
Pima	0.7313	0.745	0.7278	0.7509	0.7330 (0.757)	0.7133	0.764	0.7378 (0.0517)	0.749 (0.062)	0.7383	-	0.7742 (0.0163)	0.7492 (0.014)
Heart	0.7661	0.808	0.7797	-	0.7817	-	-	0.8103 (0.1897)	-	0.8858	-	0.7949 (0.0287)	0.814 (0.052)
Hepatitis	0.7925	0.827	-	-	-	0.8077	-	-	-	-	-	0.8113 (0.0464)	0.7452 (0.0416)
Horse	0.8504	-	-	-	-	-	-	0.6538 (0.0413)	-	-	-	0.9458 (0.0313)	0.9401 (0.0286)
Iris	0.9400	0.96	-	-	0.9453	-	0.973	-	-	0.9133	0.9383	0.9506 (0.0167)	0.9578 (0.0184)
Liver	-	-	-	-	-	-	0.685	-	-	-	-	0.6884 (0.0351)	0.7205 (0.0354)

Figure 4.8: Summary of Results Comparing the Performances of HMOEN_L2 and HMOEN_HN against Existing Works. The Figure shows the Reported Mean Classification Accuracy of the Various Works (Standard Deviations are shown in the Brackets Whenever Available).

exception of Hepatitis, HMOEN_L2 and HMOEN_HN have similar performances. It can be observed that the proposed approach is better or at least competitive for Pima, Hepatitis, Horse, Iris and Liver. Cancer results are outperformed by MPANN and GABE while Heart result is outperformed by GABE. On the other hand, GABE and MPANN perform poorly for Pima with respect to HMOEN_L2 and HMOEN_HN.

Abbass [2] reported that the average network sizes of the ANN with the lowest classification error for MPANN for the Cancer and Pima datasets were 4.125 and 6.6 respectively. In the case of single network in [66], the mean network sizes for the datasets of Cancer, Pima, Heart, and Horse are 5.89, 7.9, 7.28, and 20.3 respectively. GABE [24] fixes the number of neurons in the hidden layer to be 5. Using our proposed approach, the size of the networks that were evolved are correspondingly, for the datasets of Cancer, Pima, Heart, and Horse: (1) 3.385, 4.490, 4.089, and 3.627 respectively when the L2-norm is used as the second objective, and (2) 3.675, 5.016, 4.748, and 4.581 respectively when the number of hidden layer neurons is used as the second objective. Generally, the use of the L2-norm as the second objective to be minimized leads to smaller network sizes as compared to the use of the number of hidden layer neurons as the capacity control objective. In terms of classification accuracies, both perform similarly well, although we have to note that theoretically, for networks which possess similar training accuracies, networks which are smaller in size will tend to perform better in generalization.

4.4.3 Effects of Multiobjectivity on ANN Design and Accuracy

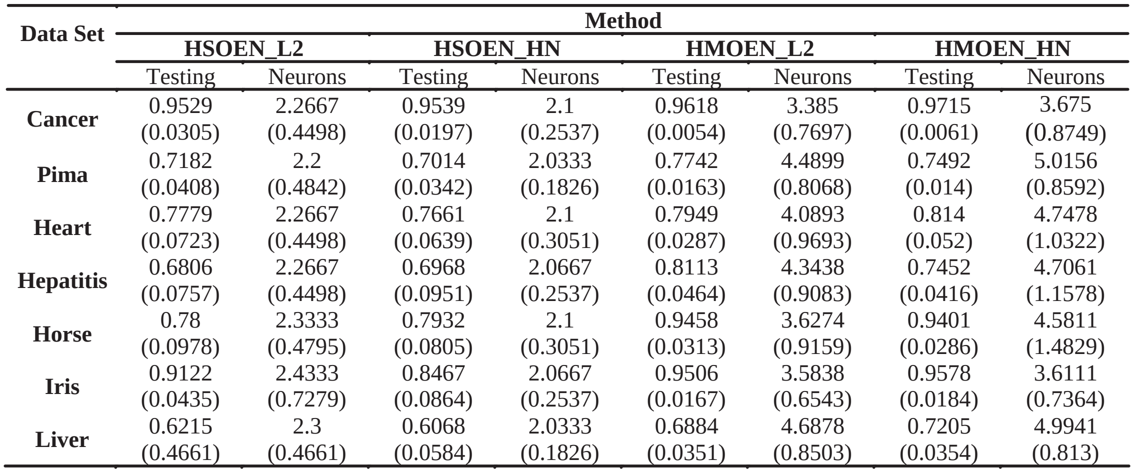
This section investigates the effects of modeling ANN design as a MO problem. In order to demonstrate the merits are the multi-objectivity, the following two single fitness functions are employed.

$$f_{SO,1} = 0.5 \cdot f_1 + 0.5 \cdot f_2 \tag{4.11}$$

$$f_{SO,1} = 0.5 \cdot f_1 + 0.5 \cdot f_3 \tag{4.12}$$

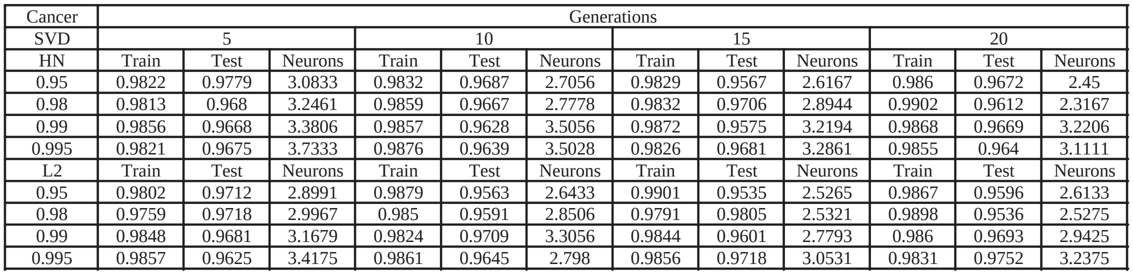
where f_1, f_2 and f_3 have been defined previously. However, as mentioned earlier, f_2 and f_3 are not of similar scale with f_1 which necessitates scaling considerations. In both cases, the second objective component is normalized adaptively according to the largest and smallest networks found in each generation. The algorithm optimizing $f_{SO,1}$ is denoted as HSOEN_HN while the algorithm optimizing $f_{SO,1}$ is denoted as HSOEN_L2. Both SO are incorporated with the features of SVAR and μHGA.

The simulation results are summarized in Figure 4.9. The aggregation of classification and error and network complexity is very effective in limiting the number of hidden neurons at the expense of classification accuracy. This is what we have come to expect, as in the SO method, it is very difficult to assign appropriate weights, in this case, even after online normalization is performed. Generally, it is noted that the explicit use of a second objective in a Pareto sense to control the size of the feedforward neural network tends to perform better than the use of the aggregation of weighted objective values.



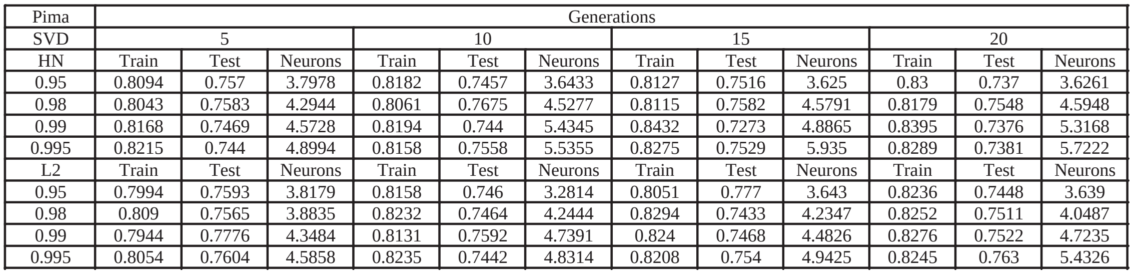
Data Set	Method							
	HSOEN_L2		HSOEN_HN		HMOEN_L2		HMOEN_HN	
	Testing	Neurons	Testing	Neurons	Testing	Neurons	Testing	Neurons
Cancer	0.9529 (0.0305)	2.2667 (0.4498)	0.9539 (0.0197)	2.1 (0.2537)	0.9618 (0.0054)	3.385 (0.7697)	0.9715 (0.0061)	3.675 (0.8749)
Pima	0.7182 (0.0408)	2.2 (0.4842)	0.7014 (0.0342)	2.0333 (0.1826)	0.7742 (0.0163)	4.4899 (0.8068)	0.7492 (0.014)	5.0156 (0.8592)
Heart	0.7779 (0.0723)	2.2667 (0.4498)	0.7661 (0.0639)	2.1 (0.3051)	0.7949 (0.0287)	4.0893 (0.9693)	0.814 (0.052)	4.7478 (1.0322)
Hepatitis	0.6806 (0.0757)	2.2667 (0.4498)	0.6968 (0.0951)	2.0667 (0.2537)	0.8113 (0.0464)	4.3438 (0.9083)	0.7452 (0.0416)	4.7061 (1.1578)
Horse	0.78 (0.0978)	2.3333 (0.4795)	0.7932 (0.0805)	2.1 (0.3051)	0.9458 (0.0313)	3.6274 (0.9159)	0.9401 (0.0286)	4.5811 (1.4829)
Iris	0.9122 (0.0435)	2.4333 (0.7279)	0.8467 (0.0864)	2.0667 (0.2537)	0.9506 (0.0167)	3.5838 (0.6543)	0.9578 (0.0184)	3.6111 (0.7364)
Liver	0.6215 (0.4661)	2.3 (0.4661)	0.6068 (0.0584)	2.0333 (0.1826)	0.6884 (0.0351)	4.6878 (0.8503)	0.7205 (0.0354)	4.9941 (0.813)

Figure 4.9: Performance Comparison between SO and MO Approach for all Datasets. The Table Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive. (Standard Deviations are shown in Brackets).



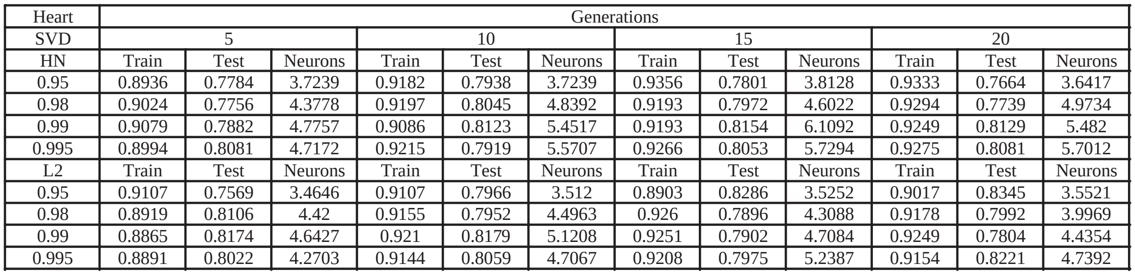
Cancer	Generations											
SVD	5			10			15			20		
HN	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9822	0.9779	3.0833	0.9832	0.9687	2.7056	0.9829	0.9567	2.6167	0.986	0.9672	2.45
0.98	0.9813	0.968	3.2461	0.9859	0.9667	2.7778	0.9832	0.9706	2.8944	0.9902	0.9612	2.3167
0.99	0.9856	0.9668	3.3806	0.9857	0.9628	3.5056	0.9872	0.9575	3.2194	0.9868	0.9669	3.2206
0.995	0.9821	0.9675	3.7333	0.9876	0.9639	3.5028	0.9826	0.9681	3.2861	0.9855	0.964	3.1111
L2	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9802	0.9712	2.8991	0.9879	0.9563	2.6433	0.9901	0.9535	2.5265	0.9867	0.9596	2.6133
0.98	0.9759	0.9718	2.9967	0.985	0.9591	2.8506	0.9791	0.9805	2.5321	0.9898	0.9536	2.5275
0.99	0.9848	0.9681	3.1679	0.9824	0.9709	3.3056	0.9844	0.9601	2.7793	0.986	0.9693	2.9425
0.995	0.9857	0.9625	3.4175	0.9861	0.9645	2.798	0.9856	0.9718	3.0531	0.9831	0.9752	3.2375

Figure 4.10: Performance Trend for Cancer over Different threshold and Number of Generation Settings. The Figure Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive.



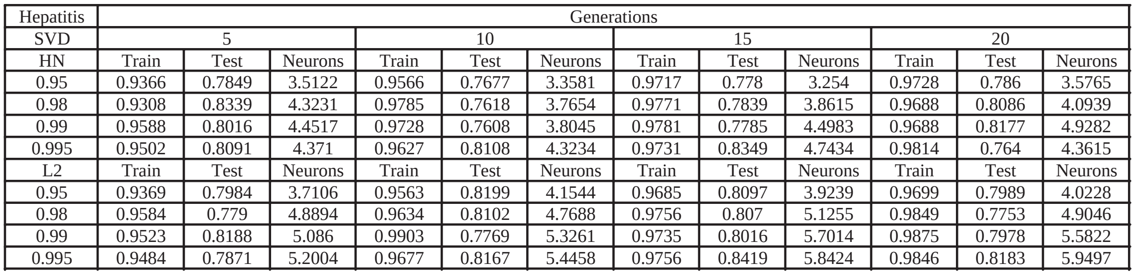
Pima	Generations											
SVD	5			10			15			20		
HN	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.8094	0.757	3.7978	0.8182	0.7457	3.6433	0.8127	0.7516	3.625	0.83	0.737	3.6261
0.98	0.8043	0.7583	4.2944	0.8061	0.7675	4.5277	0.8115	0.7582	4.5791	0.8179	0.7548	4.5948
0.99	0.8168	0.7469	4.5728	0.8194	0.744	5.4345	0.8432	0.7273	4.8865	0.8395	0.7376	5.3168
0.995	0.8215	0.744	4.8994	0.8158	0.7558	5.5355	0.8275	0.7529	5.935	0.8289	0.7381	5.7222
L2	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.7994	0.7593	3.8179	0.8158	0.746	3.2814	0.8051	0.777	3.643	0.8236	0.7448	3.639
0.98	0.809	0.7565	3.8835	0.8232	0.7464	4.2444	0.8294	0.7433	4.2347	0.8252	0.7511	4.0487
0.99	0.7944	0.7776	4.3484	0.8131	0.7592	4.7391	0.824	0.7468	4.4826	0.8276	0.7522	4.7235
0.995	0.8054	0.7604	4.5858	0.8235	0.7442	4.8314	0.8208	0.754	4.9425	0.8245	0.763	5.4326

Figure 4.11: Performance Trend for Pima over Different threshold and Number of Generation Settings. The Figure Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive.



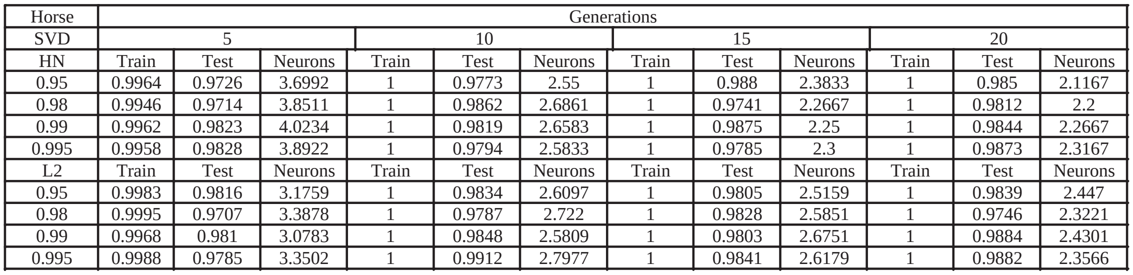
Heart	Generations											
SVD	5			10			15			20		
HN	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.8936	0.7784	3.7239	0.9182	0.7938	3.7239	0.9356	0.7801	3.8128	0.9333	0.7664	3.6417
0.98	0.9024	0.7756	4.3778	0.9197	0.8045	4.8392	0.9193	0.7972	4.6022	0.9294	0.7739	4.9734
0.99	0.9079	0.7882	4.7757	0.9086	0.8123	5.4517	0.9193	0.8154	6.1092	0.9249	0.8129	5.482
0.995	0.8994	0.8081	4.7172	0.9215	0.7919	5.5707	0.9266	0.8053	5.7294	0.9275	0.8081	5.7012
L2	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9107	0.7569	3.4646	0.9107	0.7966	3.512	0.8903	0.8286	3.5252	0.9017	0.8345	3.5521
0.98	0.8919	0.8106	4.42	0.9155	0.7952	4.4963	0.926	0.7896	4.3088	0.9178	0.7992	3.9969
0.99	0.8865	0.8174	4.6427	0.921	0.8179	5.1208	0.9251	0.7902	4.7084	0.9249	0.7804	4.4354
0.995	0.8891	0.8022	4.2703	0.9144	0.8059	4.7067	0.9208	0.7975	5.2387	0.9154	0.8221	4.7392

Figure 4.12: Performance Trend for Heart over Different threshold and Number of Generation Settings. The Figure Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive.



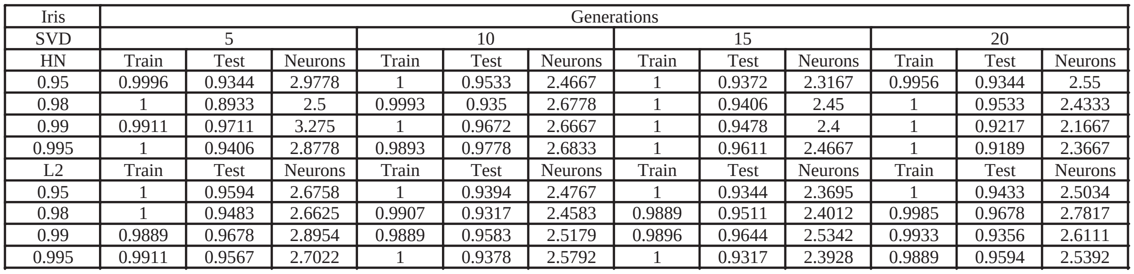
Hepatitis	Generations											
SVD	5			10			15			20		
HN	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9366	0.7849	3.5122	0.9566	0.7677	3.3581	0.9717	0.778	3.254	0.9728	0.786	3.5765
0.98	0.9308	0.8339	4.3231	0.9785	0.7618	3.7654	0.9771	0.7839	3.8615	0.9688	0.8086	4.0939
0.99	0.9588	0.8016	4.4517	0.9728	0.7608	3.8045	0.9781	0.7785	4.4983	0.9688	0.8177	4.9282
0.995	0.9502	0.8091	4.371	0.9627	0.8108	4.3234	0.9731	0.8349	4.7434	0.9814	0.764	4.3615
L2	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9369	0.7984	3.7106	0.9563	0.8199	4.1544	0.9685	0.8097	3.9239	0.9699	0.7989	4.0228
0.98	0.9584	0.779	4.8894	0.9634	0.8102	4.7688	0.9756	0.807	5.1255	0.9849	0.7753	4.9046
0.99	0.9523	0.8188	5.086	0.9903	0.7769	5.3261	0.9735	0.8016	5.7014	0.9875	0.7978	5.5822
0.995	0.9484	0.7871	5.2004	0.9677	0.8167	5.4458	0.9756	0.8419	5.8424	0.9846	0.8183	5.9497

Figure 4.13: Performance Trend for Hepatitis over Different threshold and Number of Generation Settings. The Figure Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive.



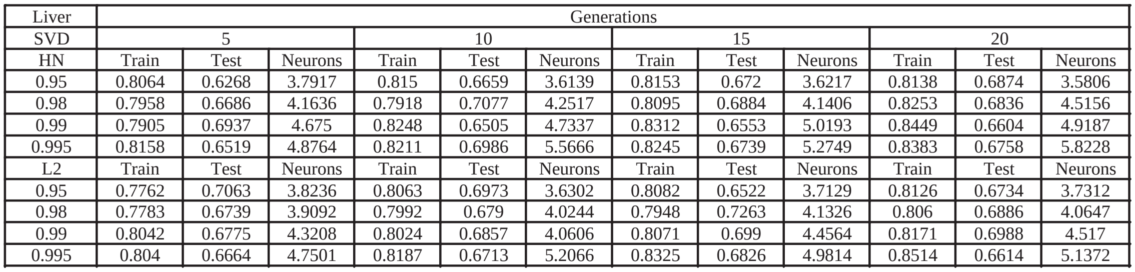
Horse	Generations											
SVD	5			10			15			20		
HN	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9964	0.9726	3.6992	1	0.9773	2.55	1	0.988	2.3833	1	0.985	2.1167
0.98	0.9946	0.9714	3.8511	1	0.9862	2.6861	1	0.9741	2.2667	1	0.9812	2.2
0.99	0.9962	0.9823	4.0234	1	0.9819	2.6583	1	0.9875	2.25	1	0.9844	2.2667
0.995	0.9958	0.9828	3.8922	1	0.9794	2.5833	1	0.9785	2.3	1	0.9873	2.3167
L2	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9983	0.9816	3.1759	1	0.9834	2.6097	1	0.9805	2.5159	1	0.9839	2.447
0.98	0.9995	0.9707	3.3878	1	0.9787	2.722	1	0.9828	2.5851	1	0.9746	2.3221
0.99	0.9968	0.981	3.0783	1	0.9848	2.5809	1	0.9803	2.6751	1	0.9884	2.4301
0.995	0.9988	0.9785	3.3502	1	0.9912	2.7977	1	0.9841	2.6179	1	0.9882	2.3566

Figure 4.14: Performance Trend for Horse over Different threshold and Number of Generation Settings. The Figure Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive.



Iris	Generations											
SVD	5			10			15			20		
HN	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.9996	0.9344	2.9778	1	0.9533	2.4667	1	0.9372	2.3167	0.9956	0.9344	2.55
0.98	1	0.8933	2.5	0.9993	0.935	2.6778	1	0.9406	2.45	1	0.9533	2.4333
0.99	0.9911	0.9711	3.275	1	0.9672	2.6667	1	0.9478	2.4	1	0.9217	2.1667
0.995	1	0.9406	2.8778	0.9893	0.9778	2.6833	1	0.9611	2.4667	1	0.9189	2.3667
L2	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	1	0.9594	2.6758	1	0.9394	2.4767	1	0.9344	2.3695	1	0.9433	2.5034
0.98	1	0.9483	2.6625	0.9907	0.9317	2.4583	0.9889	0.9511	2.4012	0.9985	0.9678	2.7817
0.99	0.9889	0.9678	2.8954	0.9889	0.9583	2.5179	0.9896	0.9644	2.5342	0.9933	0.9356	2.6111
0.995	0.9911	0.9567	2.7022	1	0.9378	2.5792	1	0.9317	2.3928	0.9889	0.9594	2.5392

Figure 4.15: Performance Trend for Iris over Different threshold and Number of Generation Settings. The Figure Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive.



Liver	Generations											
SVD	5			10			15			20		
HN	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.8064	0.6268	3.7917	0.815	0.6659	3.6139	0.8153	0.672	3.6217	0.8138	0.6874	3.5806
0.98	0.7958	0.6686	4.1636	0.7918	0.7077	4.2517	0.8095	0.6884	4.1406	0.8253	0.6836	4.5156
0.99	0.7905	0.6937	4.675	0.8248	0.6505	4.7337	0.8312	0.6553	5.0193	0.8449	0.6604	4.9187
0.995	0.8158	0.6519	4.8764	0.8211	0.6986	5.5666	0.8245	0.6739	5.2749	0.8383	0.6758	5.8228
L2	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons	Train	Test	Neurons
0.95	0.7762	0.7063	3.8236	0.8063	0.6973	3.6302	0.8082	0.6522	3.7129	0.8126	0.6734	3.7312
0.98	0.7783	0.6739	3.9092	0.7992	0.679	4.0244	0.7948	0.7263	4.1326	0.806	0.6886	4.0647
0.99	0.8042	0.6775	4.3208	0.8024	0.6857	4.0606	0.8071	0.699	4.4564	0.8171	0.6988	4.517
0.995	0.804	0.6664	4.7501	0.8187	0.6713	5.2066	0.8325	0.6826	4.9814	0.8514	0.6614	5.1372

Figure 4.16: Performance Trend for Liver over Different threshold and Number of Generation Settings. The Figure Shows the Mean Classification Accuracy and Number of Hidden Neurons in the Archive.

4.4.4 Analyzing Effects of Threshold and Generation settings

In Section 4.2, we have discussed about the issues of threshold settings and dangers of over-training. In order to investigate the relationship between the threshold settings and the degree of training, experiments are conducted for the different datasets over threshold settings of 0.95, 0.98, 0.99, 0.995 and the number of generations that HMOEN is permitted to run. Simulation results are summarized in Figure 4.10-Figure 4.16.

From the results shown in the corresponding tables, we observe two trends, firstly, that increasing the training, through the number of generations in which the network is allowed to evolve, the training accuracy will increase, as will the testing accuracy, up to a certain point, after which the testing accuracy will decrease due to overtraining. Overtraining arises from two main factors the use of an overly complex network coupled and/or excessive training times for a particular problem set that can be solved using simpler network architectures. The use of the SVD in this respect is aimed to “control” the size of the evolved network by removing extraneous neurons in the hidden layer during the evolutionary process. For increasing SVD threshold, the trend is such that the training accuracy would increase (all else being constant) as a higher SVD threshold tends to retain more hidden layer neurons. In training the population of networks for a longer period (as measured by the number of generations for which the population of networks is allowed to evolve), allows the SVD to perform better in that a larger threshold would give better classification accuracies. Examining the Cancer and Pima results, as the SVD threshold is increased progressively from 0.95 to 0.995, we observe that the size of the evolved network increases, as we expect, since a larger amount of the spectral energy of the singular values are retained. Structurally, the networks become more complex, as can be seen from examining the average size of the hidden layer neurons in the evolving population.

Secondly, by examining “simpler” datasets (by “simpler” we mean datasets whose samples are more easily classifiable in that an obvious hyperplane can be constructed between the classes in feature space) for example, Cancer and Pima, we observe that as the training time increases (as measured by the number of generations for which the learning process is allowed to take place), the classification accuracies remain steady, and over-training is prevented, or at least mitigated by including the structural complexity (network size) as a second objective to be minimized.

4.5 Conclusion

In this chapter, a hybrid MO evolutionary approach to ANN design is proposed. To address the issue of network architectural development, we consider the use of an simple, but robust information measure based on the SVD to estimate the necessary number of neurons to be used in training a single hidden layer feedforward network,. Subsequently, the SVD-based architectural recombination is presented for the purpose of facilitating the exchange of neuronal information between candidate neural network designs and adaptation of the number of neurons for each individual, based on a geometrical approach in identifying hidden layer neurons to prune. In addition, two other problem specific operators comprising a variable length representation and a μHGA with adaptive local search intensity are also proposed to handle the fundamental issues of structural adaptation and local fine-tuning. It has been shown that neural network classifiers evolved by the proposed approach provides competitive, if not better, performances over the set of datasets employed in the comparative study as compared to existing approaches. Experimental studies are also performed to examine the effectiveness of the proposed methods with respect to real-life datasets to illustrate that the both SVAR and μHGA models assume different, but nonetheless significant roles in the evolution of effective ANN designs. While we have demonstrated the effectiveness of our proposed approach for classification problems, we believe that the methods that we have employed in this article are sufficiently flexible and robust to be extended to handle a variety of problem domains, such as regression, prediction as well as system identification problems, all of which we hope to investigate in future works.

Chapter 5

Dynamic Multi-Objective Optimization

Apart from noise, many real-world problems involve time-dependent components. Instances of such problems can be found in the areas of control, scheduling, vehicle routing, autonomous path planning, and economics, just to name a few. For such problems, it is unlikely that the optimal Pareto set and the Pareto front will remain invariant and the previous solution must be adapted to reflect the current requirements. Therefore, the optimization goal is not only to evolve a near optimal and diverse PF^A but also to track it as it changes with time.

In a certain sense, the dynamic MO problem can be considered simply as the consecutive optimization of different time-constrained MO problems with varying complexities. On the other hand, one of the challenges of evolutionary dynamic optimization is to exploit past information to improve tracking performance. It is simply too inefficient to restart the optimization process every time a change in landscape is detected, especially when the new PS^* is somewhat similar to the previous solutions. It is also imperative that the MOEA must be capable of high convergence speeds in order to find the optimal solution set before it changes. However, when MOEA converges to the PS^*, the problem is that there will be a lack of search space diversity necessary to explore the search space for the new PF^* and PS^* when landscape changes.

5.1 Dynamic Multi-Objective Optimization Problems

Dynamism in real-world problems may be due to a variety of factors, some of which are due to human intervention while the rest are inherent to the problem: a change in preference by the decision-maker, a new job in the production line, an obstacle in the path of a robot, and breakdown of vehicle in vehicle routing. In certain cases, the number of objectives or decision variable to be optimized may be changed requiring a drastic change in the ranking or representation on the part of the MOEA. In this work, we will be focused on dynamic MO problem with fixed objective and design space dimensionality and which requires the MOEA to track the changing fitness landscape.

For subsequent discussions, we will affix the time variable to the MO optimization notations described in Chapter 1 to distinguish dynamic MO optimization from static MO optimization. The terms PF_t^* and PS_t^* refer to the desired Pareto front and solution set at time t while the set of tradeoffs and nondominated solutions evolved by the dynamic MOEA at time t will be termed as PF_t^A and PS_t^A respectively.

5.1.1 Dynamic MO Problem Categorization

In dynamic SO problem, a solution can either deteriorate due to a shift in the fitness landscape or become obsolete due to the emergence of a new optimum. Likewise such traits can be found in dynamic MO problems, except that we are now dealing with a set of solutions which makes the tracking process a lot trickier. Another distinct characteristic of dynamic MO problems is that the shape and distribution of PF^* are susceptible to change as well. This makes it necessary to consider the dynamics of both feature spaces in the investigation of dynamic MOEAs.

Accordingly, Farina *et al* [52] identified four different types of dynamic MO problem according to the changes affecting the optimal Pareto front and the optimal Pareto set,

- Type I where PS_t^* changes while PF_t^* remains invariant,

- Type II where both PS_t^* and PF_t^* changes,
- Type III where PF_t^* changes while PS_t^* remains invariant and
- Type IV where both PS_t^* and PF_t^* remain invariant.

Farina *et al* further noted that, even though both PS_t^* and PF_t^* are time-invariant in Type IV problems, it is possible that the fitness topology is changing with time. This alone may pose sufficient challenge to the dynamic MOEA in finding the desired solutions.

The above classification scheme is only applicable to dynamic MO optimization problems. There also exist other appropriate but more general categorization of dynamic problems. Deb *et al* [38] pointed out that the changes in a dynamic MO problem can take place in the objective functions, the constraint functions, and the variable boundaries. In [108], dynamic problems are classified according to how the optimal solutions move after a landscape change. Jin and Sendoff stated that the location of the optimum can 1) move linearly, 2) move nonlinearly, 3) oscillate among a few points, and 4) move randomly in the decision space. Another different but important perspective of dynamic problems can be found in the SO domain. Branke [17] proposed the categorization of dynamic problems based on 1) frequency of change, 2) severity of change, 3) predictability of change and 4) periodicity of change.

These classification are complementary schemes. The first and second schemes categorize the dynamic MO problem based on the physical aspects of change, the third considers how the optimum behaves with time while the fourth considers how the changes are effected. A more general approach would be to decompose the dynamic problem into its spatial and temporal components. Table 5.1 shows the list of spatial features and their attributes while Table 5.2 summarizes the different temporal features. Note that the spatial component is further decomposed into physical and non-physical attributes of change. Physical attributes refer to *physical aspects* of problem change such as PS_t^* and PF_t^*. Non-physical attributes refer to the *manner* in which these spatial physical attributes are changed. Further note

Table 5.1: Spatial Features of Dynamic MO problem

	Type	Attributes
Physical	PS_t^*	The whole or part of PS_t^* moves to a new location
	PF_t^*	The shape of PF_t^* changes or a part of PF_t^* disappears
	Fitness landscape	Fitness landscape changes without affecting PS^* and PF^*
	Random	Random PS_t^*, PF_t^* and fitness landscape changes. Aspect of change may occur at the same time or may not happen at all.
Non-Physical	Random	Changes to physical attributes are random
	Trend	Changes to physical attributes follow a fixed pattern. Past physical topology may or may not be revisited again
	Periodic	Changes to physical attributes are periodic. Changes within each period may or may not follow a fixed pattern

Table 5.2: Temporal Features of Dynamic MO problem

Type	Attributes
None	No change is triggered at all
Random	Change is triggered randomly
Fixed	Change is triggered at a fixed interval
Scheduled	Change is triggered based on a predetermined schedule such that it may follow a trend or even appear random.
Conditional	Change is triggered after some predefined condition is satisfied

that these physical spatial attributes are unique to dynamic MO problems since we are dealing with a set of solution in contrast to a single solution in SO optimization.

A dynamic MO problem may be characterized by more than one specific instance of spatial and temporal attributes. For example, a dynamic MO test problem can exhibit PF_t^* and PS_t^* changes simultaneously as in the case of Type II problem described earlier. At the same time, PF^* changes may followed a fixed trend that is triggered randomly while PS^* changes may be periodic that is triggered based on a fixed schedule. In the event of random physical changes, it is still possible that, whenever a particular aspect such as PF_t^* changes, the change may be following a certain trend.

5.1.2 Dynamic MO Test Problems

Since dynamic MO problems are essentially MO problems, guidelines and desirable properties of deterministic benchmarks and test suites suggested in the EMOO literature are applicable and should be taken into account. Some specific features pertinent to the dynamic domain that should be considered in a dynamic MO test suite include

- Cyclic spatial changes;
- Predictable spatial changes;
- Landscape changes such as emergent landscapes that are difficult to detect;
- Landscape properties that allow very fast convergence or no exploitable spatial changes at all, i.e. memory has no significant advantage at all;

In general, any dynamic test suite should include features that challenge the dynamic MOEA capability to 1) detect a change in the environment, 2) maintain or generate the necessary diversity to explore the search space upon any changes, 3) exploit past information and 4) converge on the new $\mathrm{PS}{*}_t$ quickly.

The spatial and temporal features described in Table 5.1 and Table 5.2 provide different challenges in the design of dynamic MOEAs. For example, the storage of past PS_t^* in the form of memory will improve algorithmic performance for problems exhibiting periodic non-physical attributes. In cases where spatial and temporal features follow some trend, the presence of predictive elements can prepare the evolutionary process in anticipation of the problem's future behavior. On the other hand, we can expect that these mechanisms will have little or no significant advantage for problems which do not revisit previous PS_t^* or exhibit any trend. Furthermore, it is always possible that the reintroduction of previous solutions or prediction strategy may mislead the optimization process instead.

TLK2 is one of the earliest dynamic MO test problems to be applied in the literature. It is based on moving peaks function [17], which provides an artificial multidimensional landscape consisting of several peaks, where the height, width, and position of each peak are varied as the environment changes. The problem of TLK2 is given as

$$\min f_1(\vec{x}) = x_1 \tag{5.1}$$

$$\min f_2(\vec{x}) = \frac{1}{x_1 \cdot g(t)} \tag{5.2}$$

$$g(t) = \max_{i=1,\ldots,5}\left[\frac{H_i(t)}{1+W_i(t)\sum_{j=1}^{5}(x_{j+1}-X_{ij}(t))^2}\right] \tag{5.3}$$

At a predefined frequency, τ_T, the height and width of each peak are changed by adding a random Gaussian variable. The location of each peak is moved by a vector v of fixed length s in a random direction, i.e., the parameter s controls the severity of change. A change in the peak is governed by the following equations

$$\sigma \in N(0,1) \tag{5.4}$$

$$H_i(t) = H_i(t-1) + 7\cdot\sigma \tag{5.5}$$

$$W_i(t) = W_i(t-1) + 0.01\cdot\sigma \tag{5.6}$$

$$X_i(t) = X_i(t-1) + v\cdot\sigma \tag{5.7}$$

The change in multimodal function g may result in a shift of the optimum to a different location. In this case, the evolutionary search needs to jump or cross a valley in order to find the new optimum.

The FDA test suite proposed by Farina *et al* [52] is built upon the ZDT and DTLZ framework described in Chapter 1. This test suite has been used in [38, 85, 144, 222]. Formally, the two-objective FDA test problems have the following functional structure.

$$
\begin{aligned}
\min f_1(\vec{x_I}, t) &= \sum_{x_i \in \vec{x_I}} x_i^{F(t)}, F(t) > 0 && (5.8)\\
\min f_2(\vec{x_{II}}, \vec{x_{II}}, t) &= g(\vec{x_{II}}, t) \cdot h(\vec{x_{III}}, f_1, g) && (5.9)\\
g(\vec{x_{II}}, t) &= 1 + G(t) + \sum_{x_i \in \vec{x_{II}}} (x_i - G(t))^2 && (5.10)\\
h(\vec{x_{III}}, t) &= 1 - (\frac{f_1}{g})^{H(t)} && (5.11)
\end{aligned}
$$

where F, H and G are time-dependent functions which controls how the density of Pareto solutions, shape of the PF_t^* and PS_t^* changes with time.

The M-objective FDA test problems have the following functional structure.

$$
\begin{aligned}
\min f_1(\vec{x}, t) &= (1 + g(\vec{x}_{II})) \cdot \cos(0.5\pi y_1) \cdots \cos(0.5\pi y_{M-1}) && (5.12)\\
\min f_2(\vec{x}, t) &= (1 + g(\vec{x}_{II})) \cdot \cos(0.5\pi y_1) \cdots \sin(0.5\pi y_{M-1}) && (5.13)\\
&\vdots\\
\min f_M(\vec{x}, t) &= (1 + g(\vec{x_{II}})) \cdot \sin(0.5\pi y_1) && (5.14)\\
g(\vec{x}_{II}, t) &= G(t) + \sum_{x_i \in \vec{x}_{II}} (x_i - G(t))^2 && (5.15)\\
y_i &= x_i^{F(t)} && (5.16)
\end{aligned}
$$

where F controls how the density of Pareto solutions while G controls the changes in the PF_t^* and PS_t^* over time.

The dynamics of the FDA test functions is governed by the equation,

$$t = \frac{1}{n_T} \lfloor \frac{\tau}{\tau_T} \rfloor \tag{5.17}$$

where n_T and τ_T specifies the severity and frequency of landscape change respectively. Interestingly, due to the sinusoidal behavior of $G(t)$ and $H(t)$, n_T also determines the periodicity of similar solution sets emerging. In particular, a smaller n_T implies that the number of different PS^*_t is small. Both n_T and τ_T has a lower bound of 1.0. Setting a value of $n_t < 1.0$ will result in a magnitude change that is out of range, while increasing values of n_T produces decreasing magnitudes of change. Likewise, increasing values of τ_T will result in increasingly static environments.

This work applies four different dynamic MO test functions to validate the capability of dynamic MOEA in tracking the changing MO fitness landscape. The first problem is FDA1 [52], a Type I problem where only the PS^*_t is dynamic. The other three test functions are based on the construction guidelines provided by Farina *et al* [52]. The second test function, dMOP1, is a Type III problem where only the PF^*_t is dynamic while dMOP2 is a Type II problem where both PS^*_t and the PF^*_t change with time. Like FDA1, dMOP3 is a Type I problem. However, the variable that controls the spread of the PF^*_t changes as well. The definitions of these dynamic MO test functions are summarized in Table 5.3.

Mehnen *et al* [144] proposed the DSW test suite to facilitate theoretical analysis in dynamic MO optimization.

$$\min f_1(\vec{x}) = \Big(a_{11}x_1 + a_{12}x_1 - b_1 \cdot G(t)\Big)^2 + \sum_{i=2,\dots,|\vec{x}|} x_i^2 \tag{5.18}$$

$$\min f_2(\vec{x}) = \Big(a_{21}x_1 + a_{22}x_1 - b_2 \cdot G(t) - 2\Big)^2 + \sum_{i=2,\dots,|\vec{x}|} x_i^2 \tag{5.19}$$

where the type of spatial changes is determined by setting of appropriate a_{11}, a_{12}, a_{21}, a_{22}, b_1 and b_2 values. The main limitation of the DSW test problems is that it is not as intuitive as

Table 5.3: Definition of Dynamic Test Functions

	Test	Definition
1	FDA1	$f_1(x_1) = x_1,$ $f_2(x_2, ...x_m) = g \cdot h,$ $g(x_2, ...x_m) = 1 + \sum_{i=2}^{m}(x_i - G(t))^2,$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \sin(0.5\pi \cdot t)$ where $m = 10$, $x_1 \in [0, 1]$, $-1 \leq x_i < 1$, $\forall i = 2, ..., 10$
2	dMOP1	$f_1(x_1) = x_1,$ $f_2(x_2, ...x_m) = g \cdot h,$ $g(x_2, ...x_m) = 1 + 9 \cdot \sum_{i=2}^{m} {x_i}^2,$ $h(f_1, g) = 1 - (\frac{f_1}{g})^{H(t)}$ $H(t) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ where $m = 10$, $x_i \in [0, 1]$
3	dMOP2	$f_1(x_1) = x_1,$ $f_2(x_2, ...x_m) = g \cdot h,$ $g(x_2, ...x_m) = 1 + \sum_{i=2}^{m}(x_i - G(t))^2,$ $h(f_1, g) = 1 - (\frac{f_1}{g})^{H(t)}$ $H(t) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ $G(t) = \sin(0.5\pi \cdot t)$ where $m = 10$, $x_i \in [0, 1]$
4	dMOP3	$f_1(x_r) = x_r,$ $f_2(\vec{x} \backslash x_r) = g \cdot h,$ $g(\vec{x} \backslash x_r) = 1 + \sum_{i=1}^{\vec{x} \backslash x_r}(x_i - G(t))^2,$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $H(t) = 0.75 \cdot \sin(0.5\pi \cdot t) + 1.25$ $G(t) = \sin(0.5\pi \cdot t)$ where $m = 10$, r=U(1,2,...,m), $x_i \in [0, 1]$

compared to TLK2 and the FDA test functions when it comes to the configuring of dynamic spatial features.

An interesting approach of aggregating objective functions of existing test problems through dynamically changing weights to form a lower dimensional dynamic problem is proposed in [108]. For example, a three-objective problem can be reformulated to form a two-objective dynamic problem in the following way,

$$\min f_1'(\vec{x}, t) = w(t) \cdot f_1 + (1 - w(t)) \cdot f_2 \tag{5.20}$$

$$\min f_2'(\vec{x}, t) = w(t) \cdot f_1 + (1 - w(t)) \cdot f_3 \tag{5.21}$$

where $0 \leq w \leq 1$ is a function of time that give rise to the dynamic properties of the reformulated problem, f_1, f_2 and f_3 are the original objective functions. w can be defined as either a linear or nonlinear function to produce different test properties. Simplicity and ease of construction are the main advantages of this approach.

5.2 Performance Metrics for dynamic MO Optimization

As mentioned earlier, the objective of dynamic MO optimization is not only to evolve a near optimal and diverse PF_t^A but also to track the dynamic PF_t^*. Performance metrics of dynamic MOEAs must be able to indicate:

- how effective the dynamic MOEA is in attaining the MO optimization goals (mentioned in Chapter 1) in the face of changing physical spatial attributes and
- how fast the dynamic MOEA is capable of converging on the new solution set since there may be a restriction on time.

Static performance metrics described in Chapter 1 will not make any sense because PF^* and PS^* change with time. However, they can be easily adapted to provide an accurate assessment of the performance of MOEAs in a dynamic environment.

The simplest way is to illustrate the performance trend over time [38,52,222] by assessing the evolved solutions with respect to static metrics at different time instances. For example, Zeng *et al* [222] calculated and tabulated the GD and spread [43] of PF_t^A at time instances just before the next landscape change is triggered. Farina *et al* applied and plotted the convergence traces of the following measures in the decision and objective space,

$$e_x(t) = \frac{1}{np}\sum_{j=1}^{np} \min_{i=1:nh} ||\frac{\mathrm{pf}_i^*(t) - \mathrm{pf}_j^A(t)}{\mathrm{R}(t) - \mathrm{U}(t)}|| \tag{5.22}$$

$$e_f(t) = \frac{1}{np}\sum_{j=1}^{np} \min_{i=1:nh} ||\mathrm{ps}_i^*(t) - \mathrm{ps}_j^A(t)|| \tag{5.23}$$

where np is $|PF_t^A|$, $\mathrm{R}(t)$ is the time-dependent nadir point and $\mathrm{U}(t)$ is the time-dependent utopia point. $\mathrm{pf}_i^*(t)$ and $\mathrm{pf}_j^A(t)$ are the i-th member of PF_t^* and PF_t^A respectively while $\mathrm{ps}_i^*(t)$ and $\mathrm{ps}_j^A(t)$ are the i-th member of PS_t^* and PS_t^A respectively.

However, as pointed out by Branke [17], it is desirable to have a more compact form to describe algorithmic performances. One way of achieving this is to extend the idea of offline error applied in dynamic SO optimization and calculate the time averages of static metrics such as e_x and e_f in the case of [85]. The sampling of performance metrics should be done at instances just before the next landscape change to eliminate unnecessary penalty on dynamic MOEAs employing diversity introduction schemes such as random restart or hypermutation in situations where change is small.

In [74], we extended variable space GD (VD) and MS in the following form,

$$\mathrm{VD}_{offline} = \frac{1}{\tau}\sum_{t=1}^{\tau} \mathrm{VD} \cdot \mathrm{I}(t) \tag{5.24}$$

$$\mathrm{MS}_{offline} = \frac{1}{\tau}\sum_{t=1}^{\tau} \mathrm{MS} \cdot \mathrm{I}(t) \tag{5.25}$$

$$\mathrm{I}(t) = \begin{cases} 1, & \text{if } \ t\%\tau_t = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.26}$$

where % is the modulus operator. Similar to the metric of GD in static environment, a low value of VD$_{offline}$ is desirable and reflects good tracking capability. Likewise, a higher value of MS$_{offline}$ reflects that the MOEA is capable of evolving a diverse PF$_t^A$ in a dynamic environment. VD measures the degree of convergence between PS^{*t} and PS^A.

$$\text{VD} = \frac{1}{n_{PS}} \cdot \left(n_{PS} \sum_{i=1}^{n_{PS}} d_i^2 \right)^{\frac{1}{2}} \tag{5.27}$$

where $n_{PS} = |\text{PS}_t^A|$, d_i is the Euclidean distance (in decision space) between the i-th member of PS$_t^A$ and the nearest member of PS^{*t}.

5.3 Evolutionary Dynamic Optimization Techniques

Although there is a number of studies on evolutionary optimization in dynamic environments, most of them are restricted to the domain of SO problems and comprehensive discussions on dynamic SO evolutionary algorithms (SOEAs) can be found in [19, 148]. On the contrary, the application of MOEAs to dynamic MO problems is explored only recently.

Nonetheless, from the available literature, it is clear that EAs for dynamic optimization in any problem domain must be capable of detecting the change in fitness landscape and maintaining diversity within the evolving population. Different techniques proposed to handle the issue of diversity are based the following three classes.

- Diversity Introduction: This approach introduces diversity upon the detection of landscape change [27, 80, 213]. Random restart or reinitialization is one of the simplest techniques for generating diversity. Other common techniques include hypermutation where mutation is increased dramatically and the variable local search where mutation is increased gradually if no improvement is achieved. These approaches can be easily extended to MOEAs. The main drawback is that information gained is lost after the introduction of diversity.

- Diversity Maintenance: This approach sought to maintain diversity throughout the run [69, 79, 147]. One of the techniques that can be easily incorporated in MOEAs is the random immigrant which is conceptually similar to the idea of random restart. However, in random immigrant, random individuals are introduced into the evolving population at fixed intervals and only a part of the population is replaced. Diversity preservation techniques described in Chapter 1 can also be used, except that diversity assessment should be performed in the decision space.

- Multiple Population: The basic idea of applying multiple populations is to conduct simultaneous exploration in different regions to track any change or emergence of new optimal solutions [23, 210, 218]. Typically, this approach involves a population which exploits the current optimal solution while the other populations are encouraged to explore the search space.

In order to improve performance, many dynamic EAs also incorporate some form of memory to store past solutions in anticipation of eventual reuse. Moreover, it should be noted that adaptation to the MOEA design must be made to account for:

- Outdated elitist solutions: One potential problem of MOEAs in dynamic environment is their exploitation of nondominated solutions. When the landscape changes, the current solution set may not be indicative of the optimal Pareto front and will misguide the optimization process.

- Diversity loss: Although various diversity preservation techniques are adopted in MOEAs, diversity are maintained in the objective space to obtain a well-distributed and well-spread solution set. Unless the new optimal solution set is within the vicinity of the previous optimal solution set, it is unlikely that MOEA is able to track any landscape changes.

Deb *et al* [38] extended the NSGAII for the optimization of the dynamic hydro-thermal power scheduling problem. In order to detect problem changes, 10% of the population are selected randomly and re-evaluated in every generation. When a change is detected, all outdated solutions are re-evaluated and diversity is introduced either through random initialization or mutation. Contrary to the norm in SO optimization, only a portion of the evolving population undergo the diversity enhancement process. The effects of the population ratio to be mutated or replaced by random individuals through random initialization are also investigated. The key finding is that random initialization is more susceptible to the setting of population ratio.

In [222], Zeng *et al* proposed a dynamic orthogonal MOEA (DOMOEA) as the baseline MOEA for dynamic MO optimization. The DOMOEA treats the dynamic MO problem as a new problem instance after every landscape change. However, it exploits past information by using the PS_t^A prior to change as the new initial population. Diversity is maintained in the evolving population through a linear crossover operator, which generates an offspring different from its parents.

Instead of reintroducing past optimal solutions into the evolving population, information is exploited to predict the future behavior of the dynamic MO problem in [85]. An autoregressive model is employed to estimate the location of PS_{t+1}^* and PF_{t+1}^* and the generated individuals are used to seed the population when a change in the problem landscape is detected.

Chapter 6

A Competitive-Cooperation Coevolutionary Paradigm for Dynamic MO Optimization

As pointed out in the previous chapter, it is imperative that the MOEA must be capable of attaining high convergence speeds in order to find the optimal solution set before it changes and becomes obsolete. However, high convergence speed often implies a rapid loss of diversity during the optimization process, which inevitably leads to the inability to track the dynamic Pareto front. Therefore, it is necessary to maintain or generate sufficient diversity to explore the search space when the MO problem changes.

In these two regards, the notion of coevolution is very attractive. The coevolutionary paradigm, inspired by the reciprocal evolutionary change driven by the cooperative [162] or competitive interaction [170] between different species, has been extended successfully to MO optimization recently [31, 99, 114, 139, 142, 195].

- On the former issue of high convergence speed, several studies [161, 212] have shown that the introduction of ecological models and coevolutionary architectures are effective methods to improve the efficacy of canonical evolutionary algorithms. As a specific instance, Tan *et al* [195] demonstrated that high convergence speeds can be achieved while maintaining a good diversity of solutions. MO coevolutionary algorithms (MOCAs) seem particularly suitable for dynamic MO optimization, where the high speed of convergence can potentially be exploited for adapting quickly to the changing environment.

- On the latter issue of diversity, the works in [10, 162] demonstrated that both competitive and cooperative coevolution have its own unique mechanisms for maintaining diversity in the species subpopulation.

On the other hand, before these two attractive features can be exploited for dynamic MO optimization, it is necessary to consider the issue of appropriate problem decomposition which is crucial to successful implementation of coevolution. In this chapter, we are concerned with the decomposition of the search space. The best way of handling problem decomposition may not be known *a priori* and may change with time in dynamic MO problem. This paper proposes a new coevolutionary paradigm that incorporates both competitive and cooperative mechanisms observed in nature to solve MO optimization problems and to track the Pareto front in a dynamic environment. The main idea of competitive-cooperation coevolution is to allow the decomposition process of the optimization problem to adapt and emerge rather than being hand designed and fixed at the start of the evolutionary optimization process. In particular, each species subpopulation will compete to represent a particular subcomponent of the MO problem while the eventual winners will cooperate to evolve the better solutions. Through this iterative process of competition and cooperation, the various subcomponents are optimized by different species subpopulations based on the optimization requirements of that particular time instant, enabling the MOCA to handle both the static and dynamic MO problems. A competitive-cooperation coevolutionary algorithm (COEA) for static environment is designed based on the proposed coevolutionary paradigm and subsequently extended as dynamic COEA (dCOEA) to handle dynamic MO optimization problems.

6.1 Competition, Cooperation and Competitive-cooperation in Coevolution

Existing evolutionary techniques based on this paradigm can be classified into two main classes: competitive coevolution and cooperative coevolution. Regardless of the approach adopted, the design of coevolutionary algorithms for MO optimization requires researchers to address many issues that are unique to MO problems. In this respect, insights such as the incorporation of the various elitist and diversity mechanisms gained from the design of MOEAs can be similarly exploited in the design of MOCA. On the other hand, successful implementation of coevolution requires the explicit consideration of design issues [161] such as problem decomposition, the handling of parameter interactions and credit assignment. It should be noted that the first two issues are problem dependent and the best way of handling them may not be known *a priori*, motivating this work on an alternative coevolutionary model.

This section begins with a review of competitive and cooperative evolutionary algorithms for MO optimization, highlighting the key features and the limitations of these existing approaches. The proposed competitive-cooperation model is then described along with detailed discussions of how the different design issues are addressed.

6.1.1 Competitive Coevolution

The model of competitive coevolution is often compared to predator-prey or host-parasite interactions, where preys (or hosts) implement the potential solutions to the optimization problem, while the predators (or parasites) implement individual "fitness-cases". When applying this idea into optimization [4, 170], there are usually two subpopulations and an inverse fitness interaction exists between the two subpopulations. To survive, the losing subpopulation adapts to counter the winning subpopulation in order to become the new winner.

Although the competitive coevolution has been applied in many SOEA studies [86,163], this model is rarely investigated in the domain of EMOO. Lohn *et al* [139] embodied the model of competitive coevolution in MO optimization through a competitive coevolutionary model which contains population of candidate solutions and target population with the target objective vectors. A distinct characteristic of this algorithm is the lack of any explicit diversity preservation mechanism to guide the coevolutionary optimization process. Empirical studies are conducted with well-known MOEAs such as SPEA and NSGA, and performance of this competitive MOCA is found to be better than these test algorithms.

There are several limitations to this coevolutionary model which probably explains the lack of work in this area, at least in the case of MO numerical optimization. While competitive coevolution is a natural model for evolving objects such as game playing programs for which it is difficult to write an external fitness function, the need to hand-decompose the problem into antagonistic subcomponents places severe limitation on its range of applicability. Adding to its complexity is the need to adapt the predator population, which is the population of target vectors in the case of [139], such that it exerts appropriate convergence pressure. In the context of MO optimization, this pressure must be exerted to promote individuals in a direction that is normal as well as tangential to the tradeoff region at the same time. Intuitively, competitive coevolutionary approaches may be sensitive to the shape of PF*.

6.1.2 Cooperative Coevolution

Cooperative coevolution is inspired by the ecological relationship of symbiosis where different species live together in a mutually beneficial relationship. The basic idea of cooperative coevolution is to divide and conquer [162]: divide a large system into many modules, evolve the modules separately and then combine them together to form the whole system. The cooperative coevolutionary algorithm involves a number of independently evolving species that together form complex structures for solving difficult problem. The fitness of an individual depends on its ability to collaborate with individuals from other species and favors the development of cooperative strategies and individuals. In addition, these techniques can be implemented at two basic levels depending on the type of modules that are evolved simultaneously [115]. In the case of single-level coevolution [31, 99, 114, 142], each evolving subpopulation represents a subcomponent of the problem to be solved. On the other hand, a two-level coevolutionary process involves simultaneous optimization of the system and modules in separate subpopulations [10, 66].

An explicit way of implementing cooperative coevolution behavior in optimization techniques is to split a solution vector into different subcomponents and assign multiple evolving subpopulations to optimize the individual subcomponents [162]. Contrary to SO optimization, MO optimization is associated with a set of nondominated solutions which inevitably leads to the issue of fitness assignment and representative selection. Therefore, appropriate representatives are crucial for the search of a diverse and uniformly distributed solution set while suitable cooperative schemes must be incorporated to drive the subpopulations in tandem towards the PF*.

An early attempt to integrate the cooperative model for MO optimization is based on this method of decomposing the problem along the decision space and each subpopulation is optimized by MO genetic algorithm (MOGA) [63]. In this MO cooperative coevolutionary genetic algorithm (MOCCGA) [114], each individual is evaluated twice in collaboration with either a random or the best representative from the other subpopulations and the best Pareto rank is assigned as fitness. However, MOCCGA is limited by the lack of elitism and the localized perception of Pareto optimality.

These limitations are partially rectified by Maneeratana *et al* in [142] which incorporates elitism in the form of a fixed sized archive to store the set of nondominated solutions. In addition, the same cooperative model is successfully extended to other MOEAs such as Niched Pareto GA [90] and NSGA [188] with significant improvements over their canonical counterparts. Like MOCCGA, these MOCAs also suffers from the problem that fitness assignment conducted within a species may not be a good indicator of optimality.

In [99], Iorio and Li presented the nondominated sorting cooperative coevolutionary algorithm (NSCCGA) which is essentially the coevolutionary extention of NSGAII. NSCCGA is different from the previous two works in the sense that elitist solutions are reinserted into the subpopulations and fitness assignment takes into account the set of nondominated solutions found via the nondominated sorting. Instead of selecting nondominated individuals with the best degree of crowding, representatives are selected randomly from the best nondominated front.

Contrary to the trend of integrating the cooperative model with well-known MOEAs, Tan *et al* [195] implemented a cooperative-coevolution evolutionary algorithm (CCEA) that is based on a basic MOEA. Although the same ranking scheme [63] implemented in MOCCGA is adopted here, each individual is ranked against the nondominated solutions stored in the archive instead of within the subpopulation. In addition, an extending operator which reinserts nondominated solutions with the best niche count into the evolving subpopulation is implemented in CCEA to improve diversity and distribution of the PF. The authors also investigated the effects of various representative selection and observed that robust performances can be better achieved by conducting cooperation with two representative from each subpopulations and retaining the better collaboration.

Iorio and Li [99] also highlighted that coevolutionary algorithms are susceptible to parameter interactions, although higher mutation rates can improve algorithmic performance when handling rotated problems. Apparently, there is an inherent tradeoff between the fine-grain search capability and lack of diversity in the relatively smaller sized subpopulations of coevolutionary algorithms. In this regard, the game-theoretic approach of modeling cooperation in [185] alleviates the problem of parameter dependencies somewhat by decomposing the problem into only two subpopulations. Without restricting to a single computational paradigm, an interesting approach of switching iteratively between canonical particle swarm optimization (PSO) and cooperative PSO is proposed by Van den Bergh and

Engelbrecht [211] for SO optimization.

Employing a variant of the cooperative model discussed so far, Coello Coello and Sierra [31] proposed the coevolutionary MOEA (CO-MOEA) where the different subpopulations cooperates to form the PF instead of a valid solution. CO-MOEA starts with a single evolving population and adaptively assigns different regions of the decision variable space to new subpopulations. This assignment process is based on the analysis of the contribution of each decision variable to the PF stored in the adaptive grid [127]. Furthermore, subpopulation size is also changed in proportion to the contribution to the discovery of new nondominated solutions and subpopulations without any significant contribution are eliminated. While such an approach effectively removed design considerations such as representative selection and parameter interactions, it can be easily noted that CO-MOEA do not have fine-grain search capability of the MOCAs adopting Potter and Jong's model especially for high-dimensional problems.

6.1.3 Competitive-Cooperation Coevolution

One major issue present in the previous works is that problem decomposition places severe restrictions on algorithmic design and performance of both competitive and cooperative models. In retrospect, this problem should not arise in the context of coevolutionary algorithms since the role at which each species play is an emergent property in nature. On the other hand, it should be noted that collaboration and competition among the different species are modeled independently in coevolutionary algorithms, but the two different types of interaction are rarely exclusive within an ecological system. For example, there is competition even in the veneer of seemingly perfect plant-pollinator coevolution in nature [180], where different species of bees will compete for nectar and different species of flowers will compete to attract more bees. By incorporating both elements of cooperation and competition, the proposed model represents a more holistic view of the coevolutionary forces in nature.

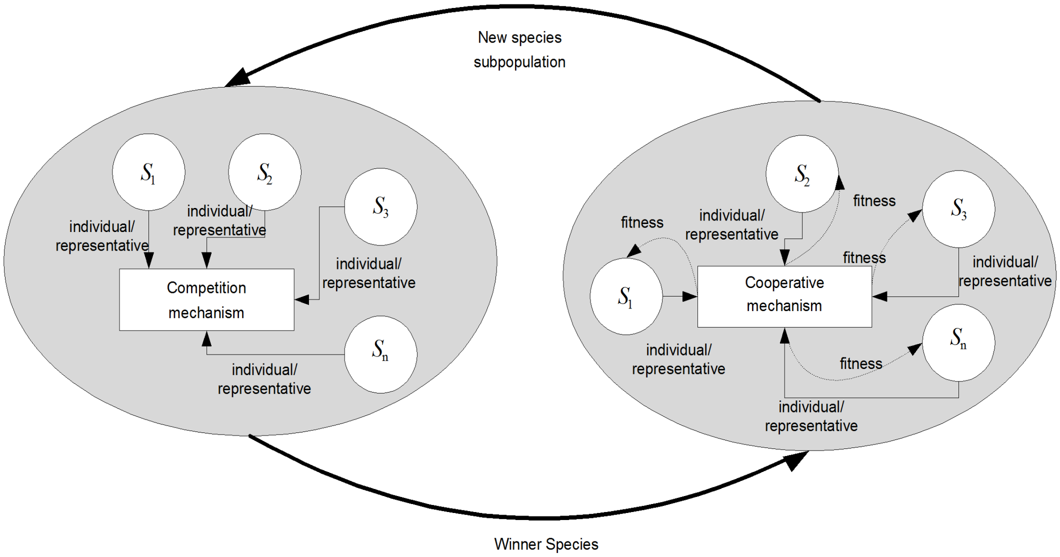


Figure 6.1: Framework of the competitive-cooperation model

The proposed model involves two tightly-coupled coevolutionary processes and the relationship between them is illustrated in Figure. 6.1. As in the case of conventional cooperative coevolutionary algorithms, individuals from the different species collaborate to solve the problem at hand during the cooperative process. Each subpopulation evolves in isolation and there is no restriction on the form of representation or the underlying EA. On the other hand, the cooperative species will enter into competition with other subpopulations for the right to represent the various subcomponents of the problem.

Although Figure. 6.1 shows that the interaction between the cooperative and competitive processes take place iteratively after each generation, this frequency can be determined by the designer accordingly. For the ensuing discussions, we consider the situation where the problem at hand is decomposed along the decision variables. Also, each decision variable may be assigned to a number of subpopulations and a subpopulation may be optimizing more than one decision variable.

Credit Assignment

Credit assignment for the competitive and cooperative process is performed at the subpopulation and individual level respectively. Following the situation given above, the different objectives of the MO problem at the cooperative process is evaluated by assembling each individual along with the representatives of the other species to form a valid solution. Accordingly, appropriate fitness assignment such as Pareto ranking can be computed for the particular individual. In the competitive process, the fitness of a particular species is computed by estimating how well it performs in a particular role relative to its competitors in the cooperation with other species to produce good solutions. For example, the species selected out of N competing subpopulations to represent a particular variable is given a higher probability of representing it in the later generations, while the losing species of the competition is penalized and given a lower probability.

Problem Decomposition and subcomponent interdependency

As mentioned earlier in the section, problem decomposition is one of the primary issues to be addressed in coevolutionary optimization. The difficulties lies in the fact that information pertinent to the number or role of subcomponents are usually not known *a priori* and many problems can only be decomposed into subcomponents exhibiting complex interdependencies. To this end, the competitive-cooperation coevolutionary model will addresses this issue through emergent problem decomposition.

As illustrated by the example given above, the competitive process leads to a potential “arms race”among the cooperative species to improve their contribution to the associated subcomponents. Notice that the collaboration between the two coevolutionary models has led to the natural formations of competitive subpopulations rather than subcomponents. In addition, it facilitates the interactions between different species, in possibly various roles, right at the onset of the optimization process and the benefits of this interactions include the discovery of interdependencies between the species. Therefore, the interplay of competition and cooperation can provide an environment in which interdependent subcomponents end up within the similar species and reasonable problem decompositions emerge due to evolutionary pressure rather than being specified by the user.

By comparison, the emergent attribute of competitive-cooperation coevolutionary model is distinctively different from that present in the cooperative model proposed by Potter and Jong [162], at least at the conceptual level. While the participation of any subpopulation is dependent on the contribution made to the collaboration between species in both approaches, this property is the result of the emergence of better fit species for a particular problem subcomponent in the proposed model. One limitation of the approach adopted in [162] is that stagnant subpopulations are simply replaced by randomly initialized subpopulations, implying that any possible information gained previously is discarded.

Diversity

The competitive-cooperation coevolutionary model provides a means of exploiting the complementary diversity preservation mechanisms of both competitive and cooperative models. In the case of the cooperative model, the evolution of isolated species tends to produce more diversed individuals across the different subpopulations, although this property does not necessarily extend to within each subpopulation. On the other hand, a diverse subpopulation is driven by the necessity to deal with the different situations posed by the other subpopulations in the competitive model. Furthermore, the competitive process in competitive-cooperation coevolutionary model allows a more diversified search as the optimization of each subcomponent is no longer restricted to one species. The competing species provides another round of optimization for each subcomponent, which increases the extent of the search while maintaining low computational requirements.

6.2 Applying Competitive-Cooperation Coevolution for MO optimization (COEA)

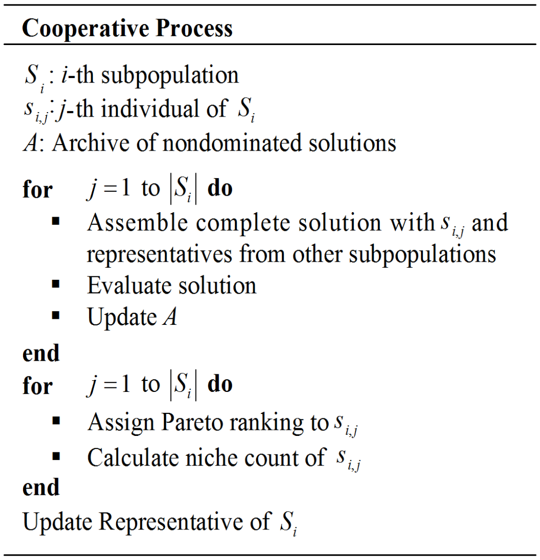
Based on the competitive-cooperation coevolutionary paradigm described in Section 6.1, this section presents the competitive-cooperation coevolutionary algorithm (COEA) for MO optimization. The mechanism for cooperative coevolution is described in 6.2.1 while the competitive element of the proposed paradigm is presented in Section 6.2.2. Finally, the implementation details of COEA is given in 6.2.3.

6.2.1 Cooperative Mechanism

The cooperative mechanism of the proposed COEA is extended from the model introduced in Tan *et al* [195]. By adopting this strategy, the algorithm can exploit the fine-grained search capability desirable in many applications while maintaining diversity across the subpopulations.

The pseudocode of the cooperative mechanism is shown in Figure. 6.2. At the start of the optimization process, the i-th subpopulation is initialized to represent the i-th variable. Concatenation between individuals in S_i and representatives from the other subpopulation is necessary to form a valid solution for evaluation. As an example, consider a 3-decision variable problem where subpopulations, S_1, S_2 and S_3, represent the variables x_1, x_2 and x_3 respectively. When assessing the fitness of $s_{1,j}$, it will combine with the representatives of S_2 and S_3 to form a valid solution.

Archive update is conducted after each individual evaluation and the archiving process has been described in Chapter 2. After which, Pareto ranking and niche count computation of individual, $s_{i,j}$ are conducted with respect to the archive. Note that only the fitness of individuals from S_i is updated at the i-th cycle. Similar to the ranking process, the niche count (nc) of each individual is calculated with respect to the archive of nondominated solutions. The dynamic sharing proposed in [197] is employed in this paper.



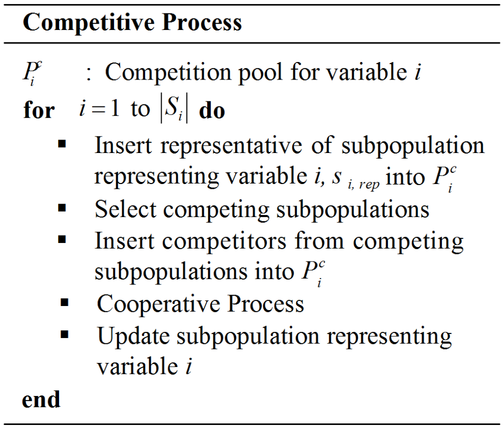
```
Cooperative Process
$S_i$: $i$-th subpopulation
$s_{i,j}$: $j$-th individual of $S_i$
$A$: Archive of nondominated solutions
for   $j = 1$ to $|S_i|$ do
    ▪ Assemble complete solution with $s_{i,j}$ and
      representatives from other subpopulations
    ▪ Evaluate solution
    ▪ Update $A$
end
for   $j = 1$ to $|S_i|$ do
    ▪ Assign Pareto ranking to $s_{i,j}$
    ▪ Calculate niche count of $s_{i,j}$
end
Update Representative of $S_i$
```

Figure 6.2: Pseudocode for the adopted Cooperative Coevolutionary mechanism.

The cooperative process is carried out in turn for all n_x subpopulations where n_x is the number of decision variables. Before proceeding to the evaluation of the next subpopulation, the representative of S_i denoted as $s_{i,\ rep}$ is updated in order to improve convergence speed. This updating process is based on a partial order such that ranks will be considered first followed by niche count in order to break the tie of ranks. For any two individuals, $s_{i,j}$ and $s_{i,k}$, $s_{i,j}$ is selected over $s_{i,k}$ if $rank(s_{i,j}) < rank(s_{i,k})$ or $\{rank(s_{i,j}) = rank(s_{i,k})$ and $nc(s_{i,j})$ $< nc(s_{i,k})\}$. The rationale of selecting a nondominated representative with the lowest niche count is to promote the diversity of the solutions using the approach of cooperation among multiple subpopulations.

6.2.2 Competitive Mechanism

Given the cooperative scheme of optimizing a single variable in each subpopulation, one simple approach is to allow the different subpopulations to take up the role of a particular problem subcomponent in a round-robin fashion. The most competitive subpopulation is then determined and the subcomponent will be optimized by the winning species in the



```
Competitive Process
$P_i^c$  : Competition pool for variable $i$
for  $i = 1$ to $|S_i|$ do
    ▪ Insert representative of subpopulation
      representing variable $i$, $s_{i,\ rep}$ into $P_i^c$
    ▪ Select competing subpopulations
    ▪ Insert competitors from competing
      subpopulations into $P_i^c$
    ▪ Cooperative Process
    ▪ Update subpopulation representing
      variable $i$
end
```

Figure 6.3: Pseudocode for the adopted Competitive Coevolutionary mechanism.

next cooperative process. Ideally, the competition depth is such that all individuals from a particular subpopulation compete with all other individuals from the other subpopulations in order to determine the extent of its suitability. However, such an exhaustive approach requires extensive computational effort and it is practically infeasible. A more practical approach is to conduct competition with only selected individuals among a certain numbers of competitor subpopulations to estimate the species fitness and suitability.

The pseudocode of the competitive mechanism is shown in Figure. 6.3. The competitive process to discover the most suitable subpopulation is carried for each variable in an iterative manner. For the i-th variable, the representative of the associated subpopulation, i.e. $s_{i,\ rep}$, is selected along with competitors from the other subpopulations to form a competition pool. With regard to the issue of competitor selection, COEA adopts a simple scheme of selecting a random individual from each competing subpopulation. Intuitively, the selection of a random competitor will enable the COEA to explore the relationships between the different variables. Other competition schemes will be presented and analyzed in Section 6.4.2. In the case where $n_x > |S_i|$, i.e. the number of subpopulations is larger than subpopulation size, the participating subpopulations are selected randomly before the start of the competition process. This provides the other subpopulations left out in this instance the opportunity to participate in future competitions.

These competitors will then compete via the cooperative mechanism described before to determine to extent of cooperation achieved with the representative of the other subpopulations. The winning species can be determined by simply checking the originating subpopulation of the representative after the representative update. At the end of the competitive process, S_i will remain unchanged if its competition representative wins the competition. In the case where a winner emerges from other subpopulations, S_i will be replaced by the individuals from the winning subpopulation. The rationale of replacing the losing subpopulation instead of associating the winning subpopulation directly with the variable is that different variables may have similar but not identical properties. Therefore, it would be more appropriate to seed the losing subpopulation with the desirable information and allow it to evolve independently.

By embedding the competitive mechanism within the cooperative process, the adaptation of problem decomposition and the optimization process are conducted simultaneously. Hence, no additional computation cost is incurred by the competition. It has the further advantage of providing the chance for the different subpopulations to solve for a single component as a collective unit, with the competitors as a source of diversity.

6.2.3 Implementation

As illustrated by the algorithmic flow of the proposed COEA in Figure. 6.4, the competitive mechanism is activated at a fixed frequency of $C_{freq} = 10$. In the proceeding recombination process, the subpopulation individuals are shuffled randomly before undergoing uniform crossover and bit-flip mutation. The reason for not performing selection based on some fitness measure is that the replacement individuals have not been assessed and may not perform in an identical manner in their new role optimizing another subcomponent. On the other hand, binary tournament selection of individuals for the mating pool will be conducted on the subpopulations after the cooperative process. Note that the subpopulations are

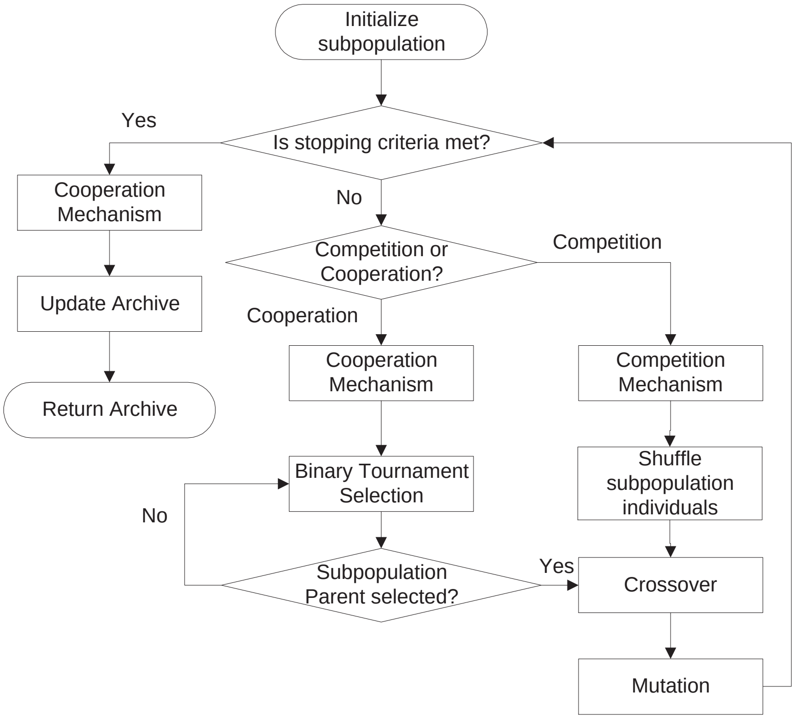


Figure 6.4: Flowchart of COEA.

evolved in isolation for both competitive and cooperative processes. The algorithm employs an fixed-size archive to store non-dominated solutions along the evolution. As mentioned in the prior sections, the archive is updated at each cycle within the cooperative or competitive mechanism. A complete solution formed by the subpopulations will be added to the archive if it is not dominated by any archived solutions. Likewise, any archive members dominated by this solution will be removed. When the predetermined archive size is reached, a recurrent truncation process [117] based on niche count is used to eliminate the most crowded archive member.

6.3 Adapting COEA for Dynamic MO optimization

Although the proposed COEA is capable of adapting to the different requirements of the MO problems, it has to be adapted for dynamic optimization. In particular, the issues of diversity and outdated archived solutions have to be addressed before the dynamic COEA is capable of dealing with the environmental variations. Section 6.3.1 describes a diversity scheme which allow diversity to be introduced while exploiting useful past information. Section 6.3.2 describes a simple temporal memory which stores outdated nondominated solution and introduces these solutions back into the archive at the appropriate moment.

6.3.1 Introducing Diversity Via Stochastic Competitors

The diversity necessary for the tracking of the dynamic PS_t^* by COEA can be either be introduced explicitly through mechanisms such as random restart and hypermutation or maintained by means of niching methods and other elaborate diversity preservation schemes. Note that the third approach of using multiple populations to explore the different regions of the search space is not applicable because the application of subpopulations in COEA serves in another purpose of optimizing a specific subcomponent of the problem. Explicit generation of diversity will enable the algorithm to react faster to severe environmental changes but it is limited by failure to utilize any past information. On the other hand, the potential for information exploitation in diversity preservation schemes is attained at the expense of slower convergence rates. This is also known as the exploration-exploitation dilemma for dynamic optimization [19].

In order to solve this problem, a diversity scheme which exploits the competitive mechanism of COEA is implemented. In every generation, a fixed number of archived solutions are re-evaluated and the current objective values are checked against previous values for discrepancies. Any environmental variation will result in the subsequent activation of the competitive mechanism, in addition to its fixed schedule. The rationale of this strategy is to allow the algorithm to assess the potential of existing information within the various subpopulations for exploitation in the new problem landscape.

Furthermore, the competitive process provides a natural conduit in which the introduction of diversity into the subpopulations can be regulated. Instead of reinitialization or subjecting the entire subpopulation to hypermutation, a set of stochastic competitors is introduced together with the competitors from the other subpopulations where the ratio between the two types of competitors is given by the parameter, SC_{ratio}. The idea is to compare the potential of new regions in the search space and past information to decide whether the subpopulation should be initialized. Latin hypercube sampling is applied to generate individuals uniformly along each dimension. In the case where the stochastic competitor emerges as the winner, the particular subpopulation is reinitialized in the region that the winner is sampled from. Therefore, diversity is introduced into the subpopulations only if it presents an advantage over the current information at hand.

6.3.2 Handling Outdated Archived Solutions

After an environmental change, it is unlikely that the archived solutions will remain nondominated. If left unchecked, these solutions will keep out the true nondominated solutions at that particular time instance. Therefore appropriate measures must be taken to minimize the detrimental effects of outdated archived solutions. One simple approach is to re-evaluate all the outdated solutions and remove only the dominated solutions from the archive. Since most MOEAs are elitist in nature, this approach may have the disadvantage of misleading the optimization process with these nondominated but outdated archived solutions. Furthermore, re-evaluation results in additional computation cost. Another approach is to simply discard all archived solutions. While this approach will not incur any extra computation cost and there is no risk of misguiding the evolutionary process, the information about past PF_t cannot be exploited in the case where the PS_t^* is cyclic in nature.

In order to store these potentially useful information in dCOEA, an additional external population which is denoted as the temporal memory is used in conjunction with the archive. In the ideal situation, the temporal memory will be a repository of all the nondominated solutions prior to any environmental variation. However, in the light of limited computational resources, decision must be made on what solutions and how solutions are to be stored. On the latter issue of how outdated solutions are stored, a fixed number R_{size} of the archive is added to the temporal memory upon a landscape change. When the upper bound of the temporal memory is reached, the oldest set of R_{size} outdated solutions is removed to make room for newer solutions. As for the former issue of selecting R_{size} outdated solutions, the dCOEA stores the extreme solutions along each dimension in the objective space. In the case where R_{size} is greater than the number of extreme solutions, the rest of the solutions to be stored are randomly selected from the archive. On the other hand, if R_{size} is smaller than the number of extreme solutions, R_{size} extreme solutions will be randomly selected into the temporal memory. Note that R_{size} actually controls the tradeoff between the storage of information across the different environmental changes and information for a particular instance of landscape change. In particular, a smaller R_{size} allows for a more diverse range of past solutions.

After the R_{size} outdated archived individuals have been added to the temporal memory, all archived solutions will be discarded. Subsequently, the temporal memory will be re-evaluated and archive updating is conducted on this external population. The computational cost incurred by this re-evaluation process is necessary to exploit any possible information regarding the current PS_t^*. However, to address the possibility that solutions updated into the archive through this scheme may misguide the optimization process, no archived solutions will be reinserted back to the subpopulations in the generation immediately after the environmental change.

Table 6.1: Parameter setting for different algorithms

Parameter	Settings

Populations	Population size 100 in NSGAII, SPEA2, PAES, and IMOEA;
	Subpopulation size 10 in COEA and CCEA;
	Archive (or secondary population) size 100.

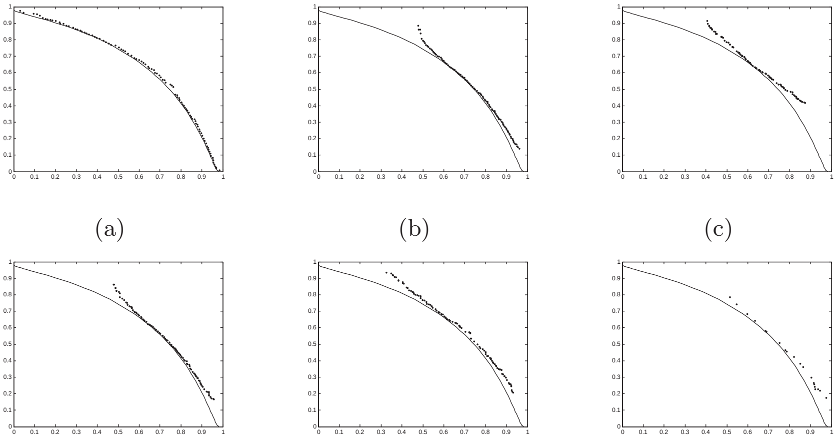
Chromosome	Binary coding; 30 bits per decision variable.
Selection	Binary tournament selection
Crossover operator	Uniform crossover
Crossover rate	0.8
Mutation operator	Bit-flip mutation
Mutation rate	$\frac{1}{L}$ for DTLZ3 where L is the chromosome length; $\frac{1}{B}$ for FON and KUR where B is the bit size per decision variable;
Niche Radius	Dynamic sharing.

6.4 Static Environment Empirical Study

This section starts with a comparative study between COEA and MOEAs that are representative of the state-of-the-arts will be conducted in Section 6.4.1. This section concludes with further investigations to gain better insights to the dynamics of competitive-cooperation evolution in Section 6.4.2 and Section 6.4.3.

6.4.1 Comparative Study of COEA

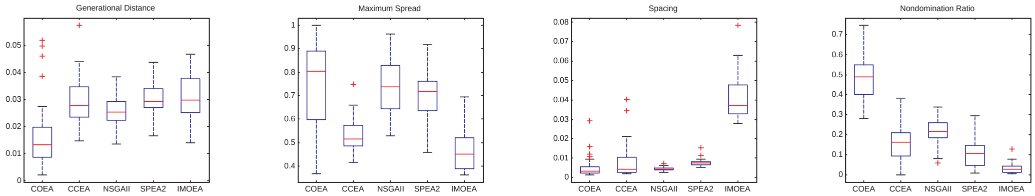
In order to examine the effectiveness of COEA, a comparative study with CCEA [195], SPEA2 [228], NSGAII [43], and IMOEA [199] is carried out based on FON, KUR and DTLZ3. The simulations are implemented in C++ on an Intel Pentium 4 2.8 GHz computer and thirty independent runs are performed for each of the test functions in order to obtain the statistical information, such as consistency and robustness of the algorithms. In order to assess statistical difference of the simulation results, Kolmogorov-Smirnov (KS)



(a) (b) (c)

(d) (e) (f)

Figure 6.5: The evolved Pareto front from (a) COEA, (b) CCEA, (c) PAES, (d) NSGAII, (e) SPEA2, and (f) IMOEA for FON.



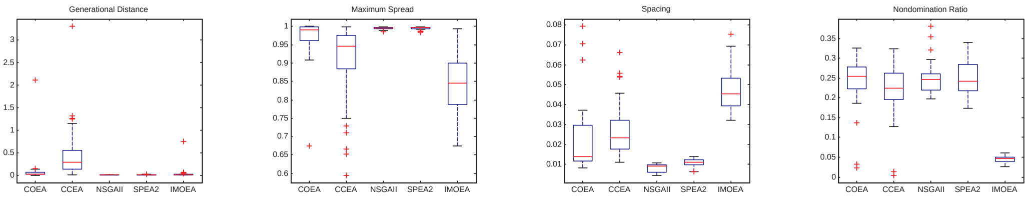
(a) (b) (c) (d)

Figure 6.6: Performance metrics of (a) GD, (b) MS, (c) S, and (d) NR for FON.

test is applied to the different performance metrics. The various parameter settings for each algorithm are listed in Table 6.1. All the algorithms are implemented using the same binary coding scheme, tournament selection, uniform crossover, and bit flip mutation.

FON

FON challenges the algorithms ability to find and maintain the entire tradeoff curve uniformly. Since the tradeoff curve is non-convex and nonlinear in nature, it is difficult for the algorithms to maintain a stable evolving population for FON. A stopping criterion of



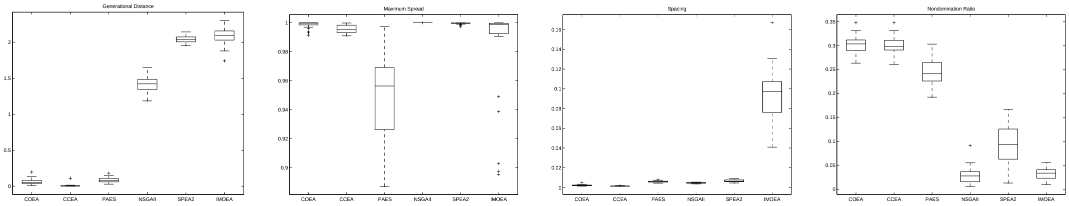
(a) (b) (c) (d)

Figure 6.7: Performance metrics of (a) GD, (b) MS, (c) S, and (d) NR for KUR.

20,000 evaluations is used for this problem. The PF from the different algorithms using the same random seed is showed in Figure. 6.5(a)-(f) while the distribution of the different performance metrics is represented by box plots in Figure. 6.6(a)-(d). The advantages of the proposed competitive-cooperation model over traditional cooperative model in handling parameter interactions is evident from Figure. 6.6 and by comparing the evolved PF in Figure. 6.5(a) and Figure. 6.5(b). It can be noted that CCEA and IMOEA performed relatively worse as compared to the other algorithms in the aspects of GD and MS. Although MOCAs are known to be susceptible to parameter interactions, CCEA has competitive performance with NSGAII and SPEA2 in the metric of NR. The KS test revealed that COEA and other algorithms are statistically different in terms of GD and HR. On the other hand, the performance of COEA in terms of MS is statistically indifferent from NSGAII and SPEA2. In general, IMOEA has the worst performance.

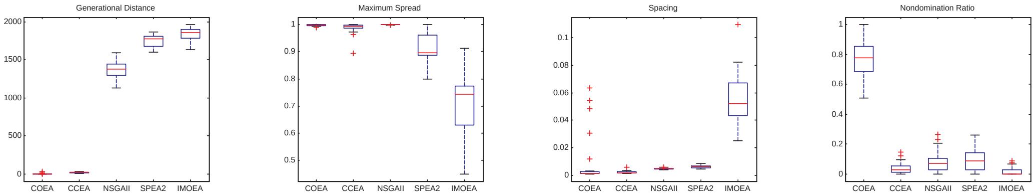
KUR

KUR is characterized by an PF* that is non-convex and disconnected, i.e., it contains three distinct disconnected regions on the final tradeoff. The decision variables correspond to the global tradeoff for KUR are difficult to be discovered, since they are disconnected in the decision variable space. Like FON, there are high interactions between the decision variables which will pose problems to MOCAs. A stopping criterion of 30,000 evaluations is used for this problem. The distribution of the different performance metrics is represented by box



(a) (b) (c) (d)

Figure 6.8: Performance metrics of (a) GD, (b) MS, (c) S, and (d) NR for DTLZ2.



(a) (b) (c) (d)

Figure 6.9: Performance metrics of (a) GD, (b) MS, (c) S, and (d) NR for DTLZ3.

plots in Figure. 6.7(a)-(d). The main difficulty stemming from high parameter interactions in this problem is the discovery of all four disconnected regions of PF. Although CCEA is able of evolving a PF that is close to PF*, it can be observed that it faced difficulty in finding a diverse PF from Figure. 6.7(b) and Figure. 6.7(c). In this sense, the competitive-cooperation paradigm allows COEA to find a more diverse solution set as compared to CCEA, PAES and IMOEA as reflected from the metric of MS. The KS test also reveals that the performances of COEA, NSGAII and SPEA2 are similar in performance for the various MO optimization metrics.

High Dimensional Problems

DTLZ3 is used to challenge the various MOEA capability to produce adequate pressure in driving the individuals towards the high-dimensional PF*. In addition, DTLZ3 is also characterized by multi-modality. A stopping criterion of 28,000 evaluations is used for both problems. The distribution of the different performance metrics for DTLZ3 is showed in Figure. 6.9(a)-(d). It can be noted that SPEA2, NSGAII and IMOEA are unable to find any solutions near the PF*. While SPEA2 and NSGAII are able to find a good spread of solutions, IMOEA is unable to evolved a diverse and well-distributed solution set. On the other hand, COEA scales well with increasing objectives, producing competitive performances in the aspects of GD, S and MS with CCEA. Furthermore, the metric of NR clearly shows that COEA outperforms CCEA.

6.4.2 Effects of the Competition Mechanism

It can be observed from the comparative studies that COEA is capable of evolving a near-optimal, diverse and uniformly distributed Pareto front for the different benchmark problems. In this section, experiments are conducted at C_{freq}={1, 5, 10, 30, 50, inf} in order to study the effects and dynamics of the incorporating both competitive and cooperative process in a common framework upon the benchmark problems of FON, KUR and DTLZ3. As mentioned earlier, FON and KUR have severe parameter interactions and it important to consider the effects of competition in improving the performance of COEA. DTLZ3 is used in the study here since it has been observed in previous section that most algorithms are unable to deal with this benchmark problem effectively.

The performance of COEA with C_{freq}={1, 5, 10, 30, 50, inf} for FON, KUR and DTLZ3 are summarized in Table 6.2, Table 6.3 and Table 6.4 respectively. Note that no competition takes place when C_{freq}=inf, effectively reducing the competitive-cooperation paradigm to a conventional cooperative model. From the tables, it can be observed that COEA performs the best for the three benchmark problems at lower settings of C_{freq} while it performs the worst in the absence of the competitive mechanism. By comparing the results over the different C_{freq}, it is clear that increasing C_{freq} allows COEA to adapt faster to the problem requirements and evolve a near optimal and more diverse PF. On the other hand, improvements in the aspects of MS is attained at the expense of GD suffers for FON and

Table 6.2: Performance of COEA for FON with different C_{freq}. The best results are highlighted in bold.

		1	5	10	30	50	Inf
GD	1st quartile	0.0080	**0.0050**	0.0086	0.0107	0.0119	0.0235
	Median	0.0116	**0.0075**	0.0133	0.0157	0.0207	0.0276
	3rd quartile	0.0171	**0.0090**	0.0198	0.0217	0.0243	0.0347
MS	1st quartile	**0.9492**	0.5394	0.5991	0.6313	0.6121	0.4857
	Median	**0.9741**	0.8916	0.8036	0.7510	0.6882	0.5159
	3rd quartile	**0.9975**	0.9466	0.8891	0.8547	0.7280	0.5732

Table 6.3: Performance of COEA for KUR with different C_{freq}. The best results are highlighted in bold.

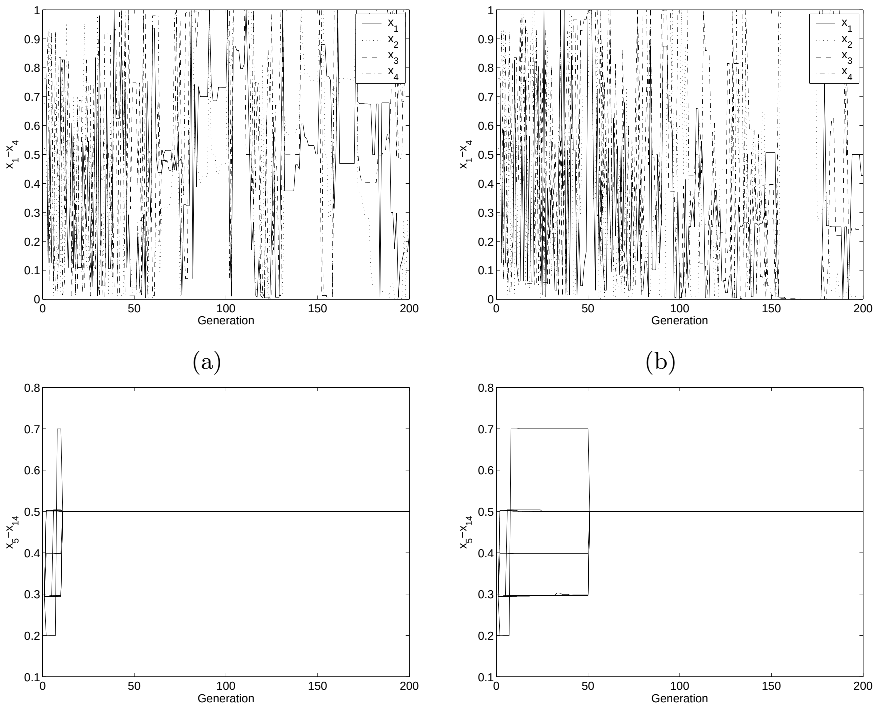
		1	5	10	30	50	Inf
GD	1st quartile	0.0349	**0.0256**	0.0329	0.0370	0.0521	0.1414
	Median	0.0425	**0.0365**	0.0376	0.0864	0.2946	0.2941
	3rd quartile	**0.0499**	0.0549	0.0807	0.3078	0.4924	0.5592
MS	1st quartile	**0.9995**	0.9822	0.9608	0.9458	0.9214	0.8841
	Median	**0.9998**	0.9939	0.9902	0.9678	0.9610	0.9461
	3rd quartile	**1.0000**	0.9988	0.9987	0.9906	0.9730	0.9752

KUR while performance deteriorates sharply at $C_{freq} = 1$ in the case of DTLZ3. This is probably because constant competition restricts the time necessary for the subpopulations to adapt to the decision variables. Nonetheless, it can be noted that the mere inclusion of competition with reasonable C_{freq} brings about significant improvement to both the convergence and diversity in FON, KUR and DTLZ3.

Figure. 6.10 shows the dynamics of the best solution for each variable at $C_{freq} = 10$ and $C_{freq} = 50$ in DTLZ3. In order to evolve a near-optimal, diverse and uniformly distributed

Table 6.4: Performance of COEA for DTLZ3 with different C_{freq}. The best results are highlighted in bold.

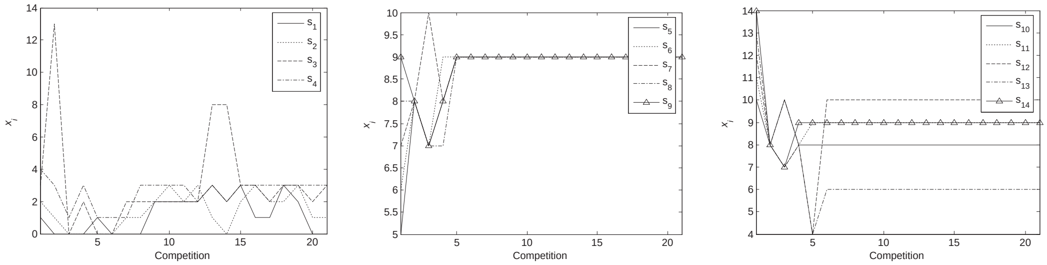
		1	5	10	30	50	Inf
GD	1st quartile	28.6021	0.0000	0.0000	0.0000	0.0000	15.0409
	Median	58.4115	0.0039	0.0009	0.0000	0.0000	18.4015
	3rd quartile	100.8232	0.0252	**0.0248**	0.0271	0.1414	23.4576
MS	1st quartile	0.6744	**0.9990**	0.9972	0.9950	0.9958	0.9860
	Median	0.7575	**0.9998**	0.9990	0.9987	0.9979	0.9933
	3rd quartile	0.8702	**1.0000**	0.9998	0.9996	0.9995	0.9986



(a) (b)

(c) (d)

Figure 6.10: Dynamics of variables x_1-x_4 (top) and x_5-x_{14} (bottom) along the evolutionary process for DTLZ3 at (a) $C_{freq} = 10$ and (b) $C_{freq} = 50$.



(a) (b) (c)

Figure 6.11: Dynamics of subpopulations emerging as the winner during the competitive process for variables (a) x_1-x_4, (b) x_5-x_9, and (c) x_{10}-x_{14}.

PF, the algorithm must be able maintain a wide range of values for x_1-x_4 while finding the optimal value of 0.5 for x_5-x_{14}. For both settings, x_1-x_4 oscillates continuously along the evolutionary process in order to span the entire range of feasible values. Likewise, x_5-x_{14} eventually converges to the optimal value of 0.5. However, by comparing Figure. 6.10(a) and Figure. 6.10(b), it can be seen that COEA at $C_{freq} = 10$ converges to the optimal value of 0.5 at the tenth generation while COEA at $C_{freq} = 50$ only converges at the fiftieth generation. It should be noted that the convergence of the algorithm coincides with each competition process.

In order to analyze the influence of the competitive process on the emergent decomposition process, the winning subpopulation for each round of competition is shown in Figure. 6.11. To facilitate the introduction of diversity for variables x_1-x_{14}, it is observed that S_1-S_3 have emerged as the most suitable subpopulations for that purpose and each takes over the role of optimizing a variable within x_1-x_4 in an almost iterative manner. In the case of variables x_5-x_{14}, it is observed that S_8 took over the rest of the subpopulation at the first competition. Although subsequent winners include S_4, S_7, S_8, S_9 and S_{10}, S_9 is the dominant subpopulation for these variables. Taking a closer look at the subpopulation distribution also reveals that the individuals of S_1-S_3 are distributed throughout the search space while the individuals of S_4-S_{14} are concentrated about the point 0.5.

6.4.3 Effects of Different Competition Schemes

Just as the choice of representative is an important design issue in any cooperative coevolutionary algorithms, it can be further expected that the different schemes of competition will have an significant impact on competitive-cooperation coevolution algorithms. In this section, three different competition models are incorporated in COEA and their effectiveness for MO optimization are investigated.

- Random: Before the start of each competition process, an individual will be selected randomly from each competing subpopulation as the participant. These set of competitors will remain fixed during the whole course of competition for the particular subcomponent. This is also the scheme that is implemented in COEA for the comparative study.
- Elitist: Before the start of each competition process, each competing subpopulation will select the best individual for their associated subcomponents as the participant. These set of competitors will remain fixed during the whole course of competition for the particular subcomponent. This scheme can be expected to wok well in situations where the different subcomponents have very similar properties.
- Hybrid: Before the start of each competition process, each competing subpopulation will randomly select either the best individual or random individual as the participant. These set of competitors will remain fixed during the whole course of competition for the particular subcomponent. The Hybrid scheme represents the tradeoff between random and elitist scheme.

Experiments are conducted for COEA with different competition schemes at $C_{freq} = 10$ and the results of thirty independent runs for FON, KUR and DTLZ3 are summarized in Table 6.5, Table 6.6 and Table 6.7 respectively. It can be seen that the elitist scheme is capable of evolving PF with very good convergence for all three problems, performing the

Table 6.5: Performance of COEA for FON with different competitors types. The best results are highlighted in bold.

		Random	Elitist	Hybrid
	1st quartile	0.0086	**0.0069**	0.0071
GD	Median	0.0133	**0.0083**	0.0102
	3rd quartile	0.0198	0.0190	**0.0158**
	1st quartile	0.5991	**0.671203**	0.6646
MS	Median	0.8036	0.7565	**0.8288**
	3rd quartile	0.8891	0.8796	**0.9125**

Table 6.6: Performance of COEA for KUR with different competitors types. The best results are highlighted in bold.

		Random	Elitist	Hybrid
	1st quartile	0.0329	**0.0264**	0.0306
GD	Median	**0.0376**	0.0400	0.0537
	3rd quartile	**0.0807**	0.1056	0.0918
	1st quartile	**0.9608**	0.8244	0.9491
MS	Median	**0.9902**	0.9691	0.9868
	3rd quartile	**0.9986**	0.9948	0.9955

best in the metric of GD for DTLZ3. This is expected since the optimal values for variables x_5-x_{14} are identical and the elitist scheme is able to exploit this relationship very quickly. On the other hand, it is observed that the random scheme and hybrid scheme demonstrates better performances when parameter interactions are present. The limitation of high selection pressure introduced by the elitist scheme is also evident from the relatively poor performance in the metric of MS for all problems. While the random scheme demonstrates the best for KUR where the PS^* is discontinuous in the decision space, it produces rela-

Table 6.7: Performance of COEA for DTLZ3 with different competitors types. The best results are highlighted in bold.

		Random	Elitist	Hybrid
GD	1st quartile	0.0000	0.0000	0.0000
	Median	**0.0009**	0.0010	0.0033
	3rd quartile	0.0248	**0.0125**	0.0172
MS	1st quartile	**0.9972**	0.9956	0.9956
	Median	0.9990	0.9979	**0.9995**
	3rd quartile	0.9998	0.9992	**0.9999**

tively poor results for in terms of convergence for FON and DTLZ3. In contrast, the hybrid scheme is capable of producing competitive results in all cases. Furthermore, the hybrid scheme has the best performance in terms of MS for the problems of FON and DTLZ3.

The elitist scheme is the greediest method which may restrict the exploration of possible relationships between the variables. This explains why it performs well for problems with low variable interactions but provides relatively poor results for problems with high variable interactions. In contrast, the random scheme is the least greedy approach that is likely to consider the different variable relationships and maintain diverse solutions in the evolution. Hence it performs well for problems with high variable interactions but the random nature of competitor selection is unable to exploit fully the fact that the optimal solutions for FON and DTLZ3 lies in the same region. Nonetheless, it is also such a property that allows the random scheme to evolve a more diverse PF as compared to the elitist scheme. On the other hand, the hybrid scheme demonstrates characteristics of both random and elitist scheme allowing it to attain competitive results that are at least comparable to the other two schemes. Although the three competition schemes behaves differently for the different problems, it is clear that the proposed coevolutionary model is capable of producing better performances as compared to the conventional coevolutionary models. Note that these three

Table 6.8: Parameter setting for different algorithms

Parameter	Settings

Populations	Population size 100 in dMOEA;
	Subpopulation size 10 in dCOEA and dCCEA;
	Archive (or secondary population) size 100.

Chromosome	Binary coding; 30 bits per decision variable.
Selection	Binary tournament selection
Crossover operator	Uniform crossover
Crossover rate	0.8
Mutation operator	Bit-flip mutation
Mutation rate	$\frac{1}{L}$ for FDA1, dMOP1, dMOP2 and dMOP3;
Niche Radius	Dynamic sharing.
Evaluation number	20,000

schemes are only examples of how different competition models can be applied and other variants can be considered.

6.5 Dynamic Environment Empirical Study

6.5.1 Comparative Study

In order to compare the relative ability of the proposed dCOEA, two different dynamic MOEAs based on a basic MOEA and CCEA are used as test algorithms. In both dynamic MOEA (dMOEA) and dynamic CCEA (dCCEA), a fixed number of archived solutions are re-evaluated in every generation. In the case where a change in landscape is detected, the temporal memory described previously will be applied and random restart is incorporated to generate diversity within the evolving population.

Thirty independent simulation runs are performed for each of the test problems, and the values of the various parameter settings in the algorithm are tabulated in Table 6.8.

Experiments are conducted at different severity levels of $n_T = \{1,\ 10,\ 20\}$ and different frequencies of $\tau_T = \{5,\ 10,\ 25\}$ in order to study the impact of dynamism on EMOO. Since each generation involves 100 evaluations, the setting of $\tau_T = 5$ means that the landscape will change in very 500 evaluations. As before, a random initial population is created for each of the simulation runs in every test problem. In this section, SC_{ratio} and R_{size} is set as 0.5 and 5 respectively.

FDA1

FDA1 is convex and as a static problem, should not pose any difficulty to the state-of-the-arts MOEA. On the other hand, as a dynamic problem, it challenges the dynamic MOEA ability to track and converge upon PF_t^* with every landscape change. One interesting aspect of this problem is that the distribution and diversity of the solutions along PF_t is not affected by the landscape change. The simulation results of the algorithms with respect to $VD_{offline}$ and $MS_{offline}$ with various settings of τ_T and n_T are summarized in Table 6.9. In general, the coevolutionary paradigm seemed more appropriate than canonical MOEA in handling dynamic landscapes. In addition, it is evident that dCOEA outperforms dCCEA in both aspects of tracking and finding a diverse solution set. From Table 6.9, it can be observed that the performances of dMOEA, dCCEA and dCOEA in the aspects of convergence and diversity improves with increasing τ_T, i.e. less frequent landscape changes. This is expected as a larger value of τ_T allows the algorithms more time to evolve a better PF. While dMOEA demonstrated better convergence properties with larger values of n_T, i.e less severe landscape changes, the performance of dCCEA and dCOEA actually improves with the severity of change.

dMOP1

Unlike FDA1, the convexity of dMOP1 changes with time while the location of PS^* remains fixed and it challenges the dynamic MOEA ability to maintain a diverse PF_t^* with every

Table 6.9: Performance of MOEA, dCCEA and dCOEA for FDA1 at different settings of τ_T and n_T. The best results are highlighted in bold only if it is statistically different based on the KS test.

			VD$_{offline}$			MS$_{offline}$	
(τ_t, n_T)		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
(5,10)	1st quartile	0.666	0.243	**0.107**	0.789	0.829	**0.939**
	Median	0.683	0.255	**0.110**	0.801	0.834	**0.944**
	3rd quartile	0.695	0.264	**0.113**	0.801	0.841	**0.953**
(10,10)	1st quartile	0.489	0.154	**0.034**	0.870	0.863	**0.963**
	Median	0.508	0.163	**0.038**	0.878	0.873	**0.970**
	3rd quartile	0.521	0.167	**0.039**	0.890	0.882	**0.977**
(25,10)	1st quartile	0.485	0.080	**0.001**	0.876	0.926	**0.979**
	Median	0.528	0.091	**0.002**	0.894	0.939	**0.985**
	3rd quartile	0.583	0.102	**0.003**	0.914	0.947	**0.989**
(10,1)	1st quartile	1.008	0.135	**0.020**	0.535	0.857	**0.973**
	Median	1.031	0.149	**0.022**	0.585	0.866	**0.981**
	3rd quartile	1.064	0.156	**0.025**	0.599	0.883	**0.984**
(10,20)	1st quartile	0.542	0.152	**0.039**	0.847	0.858	**0.970**
	Median	0.584	0.162	**0.042**	0.868	0.875	**0.975**
	3rd quartile	0.606	0.171	**0.044**	0.881	0.888	**0.979**

landscape change. The simulation results of the algorithms with respect to VD$_{offline}$ and MS$_{offline}$ with various settings of τ_T and n_T are summarized in Table 6.10. As in the problem of FDA1, dCOEA outperforms dMOEA and dCCEA in both aspects of tracking and finding a diverse solution set. However, it should be noted that dMOEA outperforms dCCEA in both performance metrics when $\tau_T = 5$ and $\tau_T = 10$. The evolutionary trace of VD$_{offline}$ and MS$_{offline}$ at these settings are plotted in Figure. 6.12. While dCOEA and dCCEA behaves similarly in the initial generations before the first landscape change, it is observed that dCCEA is greatly affected by the change in PF shape. On the other hand, dMOEA is capable of finding PS$_t^*$ as well as a diverse PF$_t$ despite the slower convergence

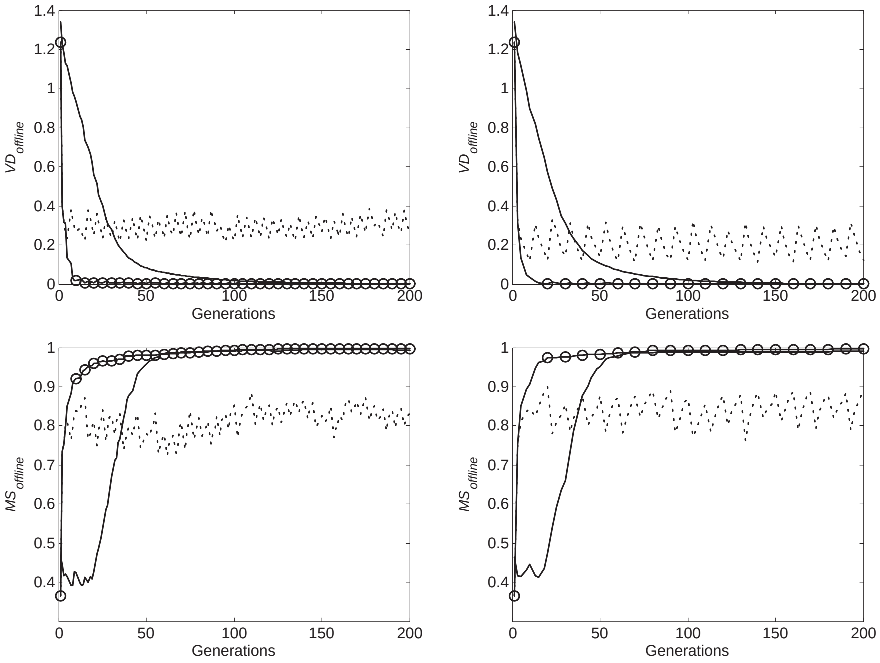
Table 6.10: Performance of MOEA, dCCEA and dCOEA for dMOP1 different settings of τ_T and n_T. The best results are highlighted in bold only if it is statistically different based on the KS test.

			$\mathrm{VD}_{offline}$			$\mathrm{MS}_{offline}$	
(τ_t, n_T)		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
	1st quartile	0.114	0.230	**0.005**	0.891	0.825	**0.977**
(5,10)	Median	0.128	0.242	**0.007**	0.911	0.838	**0.983**
	3rd quartile	0.137	0.252	**0.008**	0.933	0.846	**0.989**
	1st quartile	0.103	0.111	**0.002**	0.916	0.880	**0.988**
(10,10)	Median	0.114	0.121	**0.003**	0.916	0.880	**0.988**
	3rd quartile	0.131	0.132	**0.004**	0.935	0.888	**0.994**
	1st quartile	0.065	0.023	**0.001**	0.916	0.931	**0.989**
(25,10)	Median	0.077	0.026	**0.00**	0.940	0.948	**0.991**
	3rd quartile	0.093	0.030	**0.001**	0.962	0.962	**0.996**
	1st quartile	0.106	0.120	**0.002**	0.891	0.870	**0.986**
(10,1)	Median	0.116	0.126	**0.003**	0.914	0.877	**0.990**
	3rd quartile	0.128	0.137	**0.004**	0.934	0.893	**0.992**
	1st quartile	0.101	0.115	**0.002**	0.904	0.871	**0.982**
(10,20)	Median	0.117	0.123	**0.003**	0.921	0.881	**0.988**
	3rd quartile	0.130	0.133	**0.003**	0.939	0.890	**0.993**

speed. Based on previous studies in dynamic SO optimization, diversity schemes such as random restart tend to perform poorly in situations where change is minimal. Nonetheless, comparing to the problem of FDA1, increasing the severity of change has relatively less impact on the metric of $\mathrm{VD}_{offline}$ for the three algorithms. This is probably due to the incorporation of temporal memory which allows the algorithm to rediscover PS^* quickly, even though random restart is utilized in dMOEA and dCCEA.

dMOP2

The convexity and PS_t^* of dMOP2 changes with time and it challenges the dynamic MOEA ability to track the PS_t^* and maintain a diverse PF_t^* simultaneously with every landscape



(a)

(b)

Figure 6.12: Evolutionary trace of dMOEA (-), dCCEA (–) and dCOEA (o) for (a) $\tau_T = 5$ and $n_T = 10$ and (b) $\tau_T = 10$ and $n_T = 10$.

change. The simulation results of the algorithms with respect to VD$_{offline}$ and MS$_{offline}$ with various settings of τ_T and n_T are summarized in Table 6.11. In contrast to the previous two problems, dCOEA is outperformed by dCCEA in the aspect of VD$_{offline}$ when (τ_T, n_T) is set as (5,10) and (10,10). Since random restart is applied by dCOEA, it will be interesting to note that further investigations in the next section demonstrates that a lower SC_{ratio} will actually allow dCOEA to attain better performances. On the other hand, dCOEA outperforms both dMOEA and dCCEA in tracking and attaining better diversity at the other settings. In addition, by comparing the metric of MS$_{offline}$ in Table 6.9, Table 6.10 and Table 6.10, it can be observed that dCCEA is unable to find a diverse PF$_t$ when the shape of PF$_t^*$ is dynamic.

Table 6.11: Performance of MOEA, dCCEA and dCOEAS for dMOP2 at different settings of τ_T and n_T. The best results are highlighted in bold only if it is statistically different based on the KS test.

			VD$_{offline}$			MS$_{offline}$	
(τ_t, n_T)		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
(5,10)	1st quartile	0.642	**0.285**	0.352	0.973	0.852	**0.988**
	Median	0.666	**0.291**	0.372	0.981	0.861	**0.991**
	3rd quartile	0.680	**0.300**	0.384	0.986	0.871	**0.994**
(10,10)	1st quartile	0.495	0.159	0.173	0.976	0.886	**0.991**
	Median	0.517	0.169xx	0.180	0.980	0.902	**0.993**
	3rd quartile	0.535	0.187	0.192	0.987	0.915	**0.996**
(25,10)	1st quartile	0.462	0.069	**0.059**	0.9817	0.949	**0.991**
	Median	0.514	0.075	**0.063**	0.989	0.958	**0.994**
	3rd quartile	0.557	0.093	**0.071**	0.993	0.964	**0.997**
(10,1)	1st quartile	1.137	0.176	**0.140**	0.965	0.881	**0.991**
	Median	1.166	0.186	**0.152**	0.978	0.899	**0.996**
	3rd quartile	1.188	0.202	**0.176**	0.985	0.912	**0.998**
(10,20)	1st quartile	0.466	0.166	0.162	0.966	0.889	**0.991**
	Median	0.487	0.177	0.170	0.979	0.899	**0.992**
	3rd quartile	0.519	0.185	0.184	0.986	0.916	**0.996**

dMOP3

dMOP3 have similar characteristics to FDA1. However, because the variable that determines the spread of the solution set is not fixed and changes with time, the dynamic MOEA faces the additional challenge in tracking a diverse PF$_t^*$ as well. The simulation results of the algorithms with respect to VD$_{offline}$ and MS$_{offline}$ with various settings of τ_T and n_T are summarized in Table 6.12. Indeed, by comparing Table 6.9 and Table 6.10, it can be observed that the three algorithms attain lower performances in MS$_{offline}$ at $\tau_T = 5$ and $\tau_T = 10$. Nonetheless, as in the cases of FDA1 and dMOP2, it is clear that dCOEA outperforms both dMOEA and dCCEA in both aspects of tracking and finding a diverse solution set for all

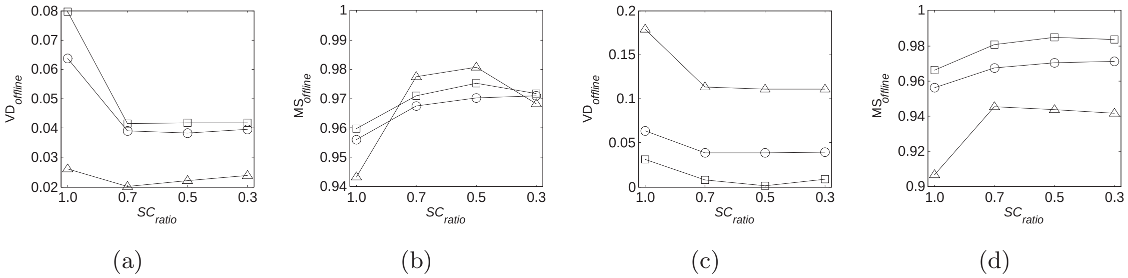
Table 6.12: Performance of MOEA, dCCEA and dCOEAS for dMOP3 at different settings of τ_T and n_T. The best results are highlighted in bold only if it is statistically different based on the KS test.

			VD$_{offline}$			MS$_{offline}$	
(τ_t, n_T)		MOEA	dCCEA	dCOEA	MOEA	dCCEA	dCOEA
	1st quartile	0.679	0.226	**0.083**	0.619	0.824	**0.906**
(5,10)	Median	0.701	0.2398	**0.087**	0.637	0.835	**0.913**
	3rd quartile	0.727	0.249	**0.09**	0.658	0.841	**0.927**
	1st quartile	0.460	0.140	**0.013**	0.802	0.856	**0.943**
(10,10)	Median	0.482	0.149	**0.017**	0.822	0.867	**0.957**
	3rd quartile	0.507	0.162	**0.021**	0.843	0.880	**0.965**
	1st quartile	0.424	0.068	**0.001**	0.903	0.927	**0.976**
(25,10)	Median	0.467	0.078	**0.002**	0.914	0.9338	**0.983**
	3rd quartile	0.515	0.096	**0.003**	0.927	0.949	**0.987**
	1st quartile	1.055	0.129	**0.011**	0.505	0.861	**0.977**
(10,1)	Median	1.087	0.138	**0.014**	0.539	0.873	**0.981**
	3rd quartile	1.108	0.15	**0.018**	0.565	0.886	**0.987**
	1st quartile	0.477	0.138	**0.019**	0.837	0.855	**0.946**
(10,20)	Median	0.505	0.147	**0.022**	0.857	0.865	**0.954**
	3rd quartile	0.538	0.155	**0.025**	0.866	0.883	**0.966**

settings of τ_T and n_T.

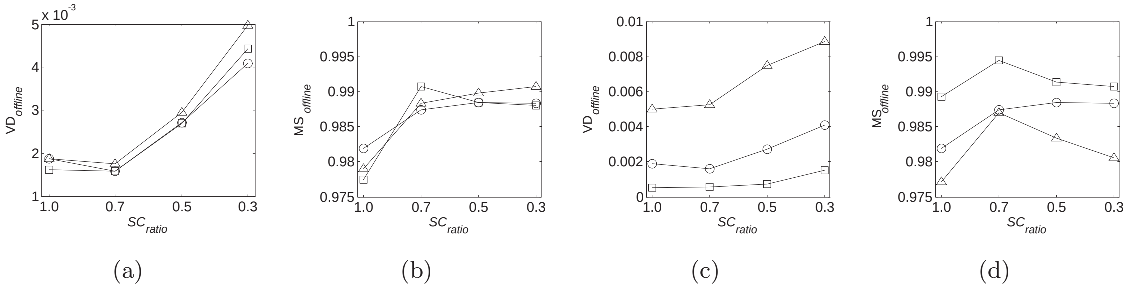
6.5.2 Effects of Stochastic Competitors

The SC_{ratio} determines the degree of diversity introduced into the proposed dCOEA after every landscape change, and hence plays a crucial role in the tracking capability of the algorithm. The relationship between $SC_{ratio} = \{0.3, 0.5, 0.7, 1.0\}$ with various settings of n_T and τ_T for FDA1 are shown in Figure. 6.13. These relationships are similarly investigated for dMOP1, dMOP2 and dMOP3 as illustrated in Figure. 6.14-6.16 respectively. Note that no stochastic competitors are introduced at $SC_{ratio} = 1.0$.



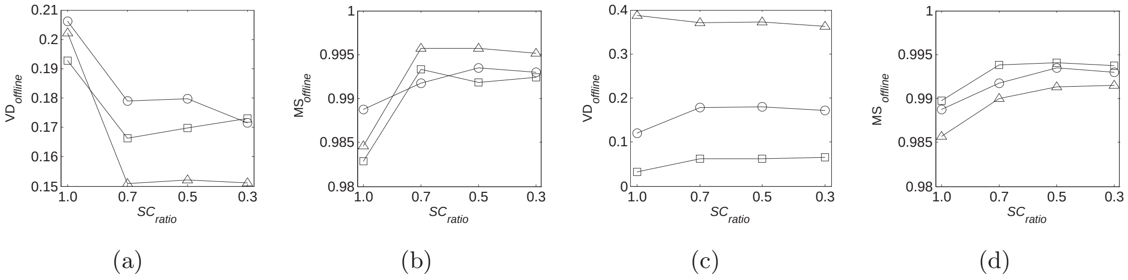
(a) (b) (c) (d)

Figure 6.13: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for FDA1 over different settings of SC_{ratio}



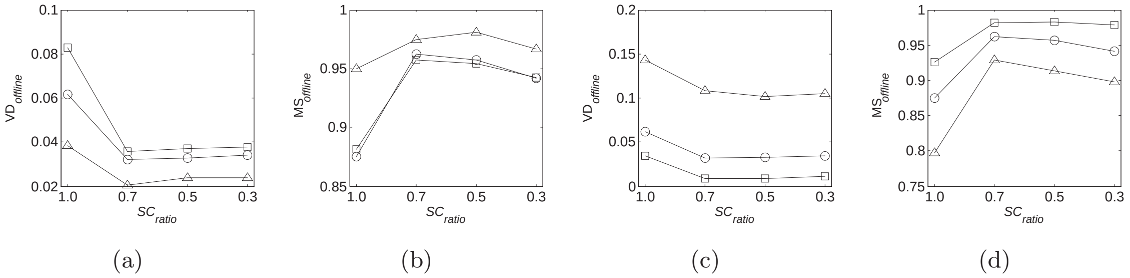
(a) (b) (c) (d)

Figure 6.14: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for dMOP1 over different settings of SC_{ratio}



(a) (b) (c) (d)

Figure 6.15: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for dMOP2 over different settings of SC_{ratio}

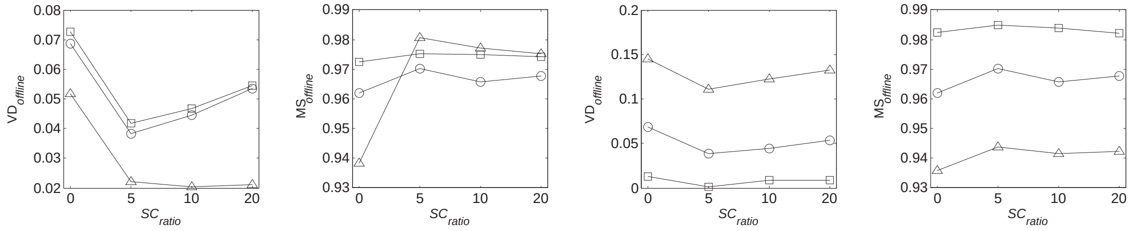


(a) (b) (c) (d)

Figure 6.16: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for dMOP3 over different settings of SC_{ratio}

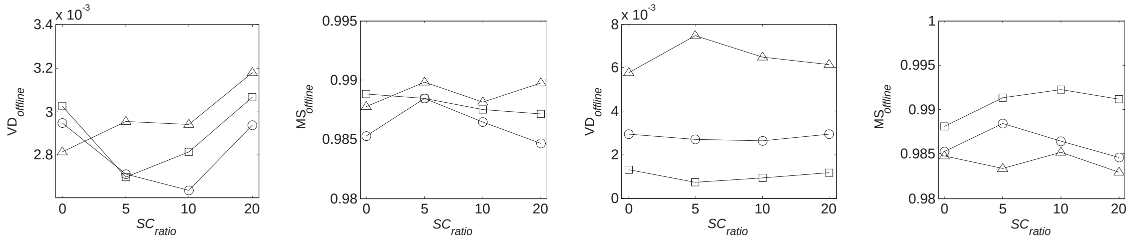
As evident from the metric of MS$_{offline}$, the diversity of the evolved PF$_t$ generally improves with the introduction of stochastic competitors. However, the diversity introduced by this mechanism seem to have a detrimental impact on the tracking ability for dMOP1. Remember that the location of PS$_t^*$ remains unchanged for this benchmark and the only task is for the dynamic MOEA to redistribute the solutions along the dimension of x_1. Intuitively, dCOEA is more able to find higher quality solutions to refill the archive since information within each subpopulation is retained when $SC_{ratio} = 1$. On the other hand, it is unlikely that the set of nondominated solutions brought about by stochastic competitors is better or even comparable to the archived solutions before the change in PF$_t^*$ shape. Nonetheless, it is clear that stochastic competitors play an important role in the tracking of dynamic PS$_t^*$ for the problems of FDA1, dMOP2 and dMOP3.

In a similar way, dCOEA demonstrates the best performance at high level of severity when $n_T = 1$ for FDA1, dMOP2 and dMOP3 while performing worst for dMOP1 at the same setting. This observation is in agreement with past investigations conducted in the realm of dynamic SO optimization that indicate higher degree of change is required with severe environmental changes. On the other hand, the introduction of large number of stochastic competitors at $SC_{ratio} = 0.3$ does not provide any significant improvements at $n_T = 1$.



(a) (b) (c) (d)

Figure 6.17: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for FDA1 over different settings of R_{size}



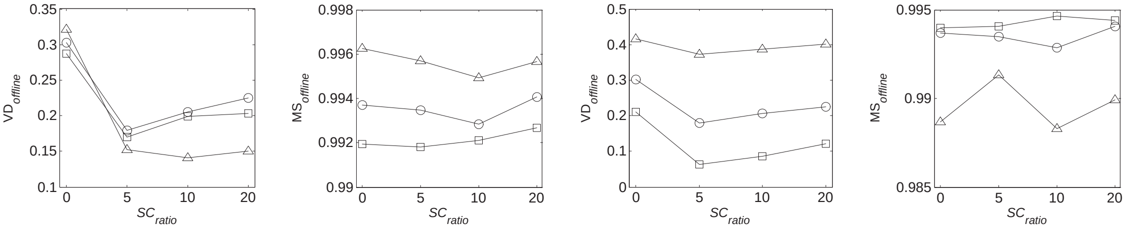
(a) (b) (c) (d)

Figure 6.18: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for dMOP1 over different settings of R_{size}

6.5.3 Effects of Temporal Memory

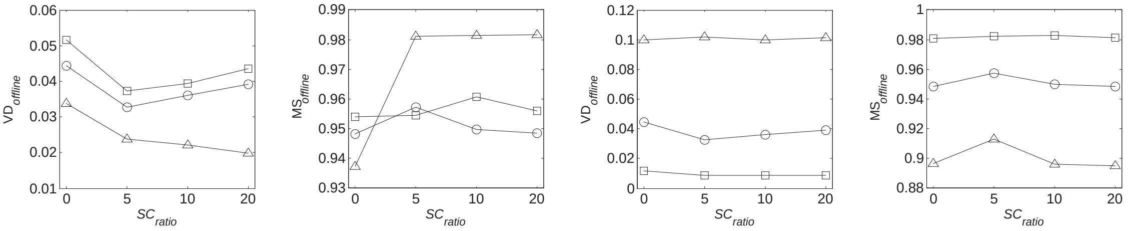
The R_{size} determines the extent in which information about past PS$_t^*$ is stored. A large R_{size} allows a higher degree of information exploitation at the expense of a more diverse repertoire of past PS$_t^*$. On the other hand, very limited information regarding each past PS_t^* is avaliable when R_{size} is small. The relationship between $R_{size} = \{0, 5, 10, 20\}$ with various settings of n_T and τ_T for FDA1 are shown in Figure. 6.16. Note that no memory is retained at $R_{size} = 0$. These relationships are similarly investigated for dMOP1, dMOP2 and dMOP3 and illustrated in Figure. 6.18-6.20.

Similar observations made previously in Section 6.5.2 such as better tracking perfor-



(a) (b) (c) (d)

Figure 6.19: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for dMOP2 over different settings of R_{size}



(a) (b) (c) (d)

Figure 6.20: Performance metrics of (a) VD$_{offline}$ and (b) MS$_{offline}$ at n_t=1.0 (△), n_t=10.0 (○), and n_t=20.0 (□) and (c) VD$_{offline}$ and (d) MS$_{offline}$ at τ_T=5.0 (△), τ_T=10.0 (○), and τ_T=25.0 (□) for dMOP3 over different settings of R_{size}

mances at higher τ_T and at $n_T = 1.0$ for FDA1, dMOP2 and dMOP3 can also be observed over the different R_{size} settings. Considering the contribution of temporal memory to the tracking capability of dCOEA, Figure. 6.18-6.20 show that the incorporation of appropriately sized memory tends to improve convergence as indicated by the metric of VD$_{offline}$. The only exception occurs for the case of dMOP1 at the setting of $n_T = 1.0$ and $\tau_T = 5.0$. The tradeoff between exploration and exploitation of information is also evident from the performance trend with increasing R_{size}. For instance, when repetition of similar PS_t^* is very frequent as in the case of $n_T = 1.0$, a large R_{size} can be used to mine information from past PS_t^* since the number of different PS$_t^*$ that needs to be represented in the memory is small. Vice versa, only a small R_{size} should be applied when number of different PS_t^* over time is higher.

6.6 Conclusion

This chapter presented a new coevolutionary paradigm that incorporates both competitive and cooperative mechanisms observed in nature to solve MO optimization problems and to track the Pareto front in a dynamic environment. The proposed competitive-cooperation coevolution is capable of overcoming the limitations of conventional coevolutionary models by allowing the decomposition process of the optimization problem to emerge based on problem requirements as well as exploiting the high speed of convergence to allow the algorithm to adapt quickly to the changing environment. Based on this coevolutionary model, a competitive-cooperation coevolutionary algorithm (COEA) is proposed for multi-objective optimization. Subsequently, this algorithm is extended as a dynamic COEA (dCOEA) and incorporated the features of stochastic competitors that allows the algorithm to track the changing solution set and temporal memory that allows the algorithm to exploit past information. Extensive studies upon three benchmark problems demonstrates that COEA is capable of evolving near-optimal, diverse and uniformly distribution Pareto fronts even for problems with severe parameter interactions. The parameter settings and working dynamics of the competitive mechanism as well as different competitive schemes are also examined, illustrating the robustness and importance of both competitive and cooperative elements in a common framework. Likewise, extensive studies are performed to investigate the performances of dCOEA over different settings of change severity and change frequency. Simulation results shows that dCOEA is capable of tracking the different environmental changes in the test functions employed effectively and efficiently. In addition, the contribution and parameter settings of the diversity scheme and the temporal memory are also analyzed over various problem settings.

Chapter 7

An Investigation on Noise-Induced Features in Robust Evolutionary Multi-Objective Optimization

Branke [19] considered robust optimization as a special case of dynamic optimization where solutions cannot be adapted fast enough to keep pace with environmental changes. In such cases, it would be desirable to find solutions that perform reasonably well within some range of change. Many real-world applications are susceptible to decision or environmental parameter variation which results in large or unacceptable performance variation. Robust optimization of MO problems is the third and final type of uncertainty considered in this work and it involves the optimization of a set of Pareto optimal solutions that remain satisfactory in the face of parametric variations. This chapter addresses the issue of robust MO optimization by presenting a robust continuous MO test suite with features of noise-induced solution space, fitness landscape and decision space variation. In addition, the vehicle routing problem with stochastic demand (VRPSD) is presented a practical example of robust combinatorial MO optimization problems.

7.1 Robust measures

In order to avoid any confusion in the subsequent discussions, it will be instructive to make a distinction between the notations used for deterministic MO and robust MO optimization. The terms PF* and PS* refer to the desired Pareto front and solution set in the general sense, without representing any specific case. The optimal Pareto front and the corresponding Pareto solution set of a particular deterministic MO problem will be denoted as PF$^*_{det}$ and PS$^*_{det}$ respectively. Note that PF$^*_{det}$ may not be known *a priori* and it is fixed for any particular MO problem. The final set of nondominated solutions evolved by MOEA will be termed as PF$^A_{det}$.

In the case of robust MO optimization, the optimal robust Pareto front and solution set are also dependent on the noise model and the robust measure. This implies that, contrary to PF* and PS*, the optimal robust Pareto set is not fixed. Furthermore, the structure of the Pareto front, i.e. its dimensionality may change as well due to the additional optimization criteria of robustness. Therefore, the notation should reflect the noise model and the robust measure used. In this paper, the optimal robust Pareto front and optimal solution set are denoted as PF$^*_{rm,\sigma}$ and PS$^*_{rm,\sigma}$ respectively. The terms rm and σ refers to the robust measure and noise model in consideration. Accordingly, PF$^A_{rm,\sigma}$ refers to the final set of nondominated solutions evolved by robust MOEA based on the robust measure, rm and noise model, σ.

There are several possible notions of robustness and many different robust measures have been applied in the literature. The most popular and straight-forward measure is the optimization of the expected performance over the possible disturbances, i.e. $E(f_i) = \frac{1}{N} \cdot \sum_{i=1}^{N} f_i(\vec{x} + \vec{\sigma_i})$. Solutions that are optimized based on expected fitness are known as effective solutions. Hence, for MO optimization, the resulting Pareto front is known as the effective Pareto front (PF$^A_{eff,\sigma}$). Other measures includes the optimization of the worst case scenario [155], as a constraint to be satisfied [40], and various forms of variances.

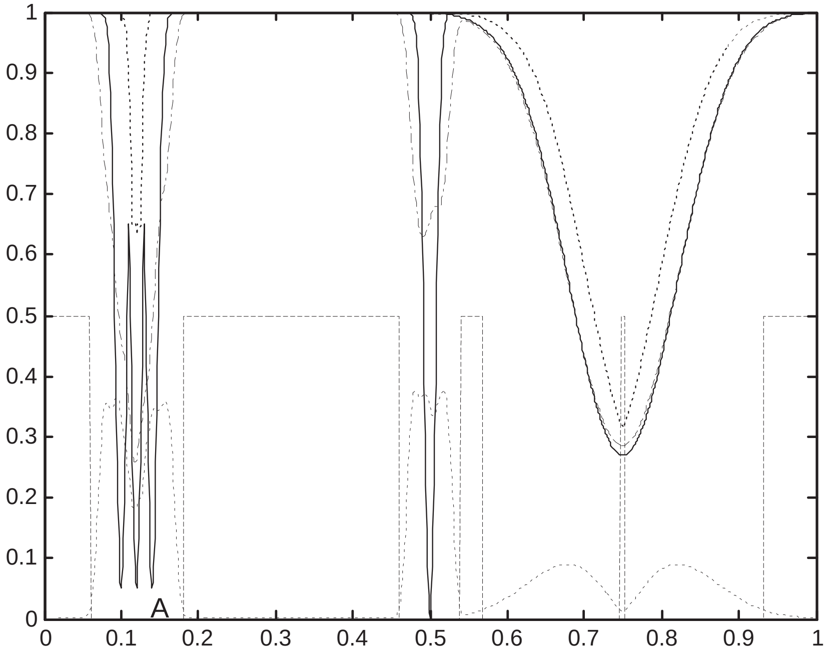


Figure 7.1: Illustration of the different robust measures, constrained (– –), standard deviation (- - -), effective (-·-·) and worst case (· · ·), with respect to the deterministic landscape (—)

Each of these robust measures reflects the different aspects of robustness and Figure 7.1 illustrates the behavior of the different robust measures for an arbitrary function of varying sensitivities in the search space. The various plots are generated by sampling the values of x with uniform distribution of $[-0.025, 0.025]$. If the model is known with absolute certainty and the solution can be implemented exactly, then the global optimal represented by the deterministic solution at $x = 0.5$ is the ideal solution. However, if variable x is stochastic, then the solutions presented by the other approaches will be more viable and the location of the optimal is also different. In particular, it can be noted the expected mean approach will favor the solution at $x = 0.11$ while the approaches based on variance and worst case will favor the solution at $x = 0.75$. On the other hand, the constrained approach indicates the feasible solutions which satisfies the pre-defined criteria.

7.2 Evolutionary Robust Optimization Techniques

While MOEAs have been demonstrated to be capable of discovering good tradeoff solutions for various MO problems, it is necessary to ensure that these solutions are implementable in practice. However, conventional MOEAs are unable to identify robust solutions and lack the necessary mechanisms to find $\mathrm{PS}^*_{rm,n}$ unless it coincides with the PS^*_{det}.

The role of a robust MOEA is to find a set of $\mathrm{PF}^A_{rm,n}$ that is more robust than that evolved by MOEA while maintaining relatively good solution quality. It can be noted that studies on evolutionary robust optimization are mainly conducted in the domain of robust SO optimization and it is unlikely that these techniques are suitable for robust MO problems, as in the case of SOEAs for MO optimization. Nevertheless, the robust measures and uncertainty handling mechanisms adopted in these works are generally applicable for robust MO optimization; subsequent discussions are largely based on these studies and on its suitability in the context of robust MOEA. It should also be noted that only three studies addressing robust MO optimization [40, 73, 81] are known to exist in the literature. Specific issues such as diversity preservation and fitness assignment must be considered in robust MOEA design.

Based on the state-of-the-arts, EAs for robust optimization can be classified into SO and MO approaches depending on how the various measures are incorporated into the EA.

1. The *SO* approach optimizes the selected robust measures in place of the original objectives.

2. The *MO* approach considers the various MO objective functions and selected robust measures as separate entities during optimization.

As noted by Jin and Branke [107], the former is the more popular approach. This is perhaps because of its ease of implementation whereas there is a need to consider the implications brought about by the increase in problem dimensionality for the latter.

7.2.1 SO approach

Since it is usually difficult to compute the various robust measures analytically, this approach is also characterized by the stochastic evaluation of the adopted robust measure to account for uncertainties, i.e. these measures are usually estimated over a number of randomly sampled perturbations. The optimization of the expected objective values estimated from the mean of the sampled points is also known as *explicit averaging* and has been applied successfully for robust MO optimization [40]. In the same work, the effects of sample size and noise level on $\mathrm{PF}^{A}_{eff,n}$ are investigated.

Although simple to implement, stochastic evaluation is computationally intensive since additional solution evaluations are required. It can be expected that this situation will be exacerbated by the presence of multiple objectives in MO problems. Therefore, suitable methods for reducing the number of evaluations will be required to lower total computational cost. To this end, the Latin hypercube sampling is applied by [19, 40] to get a better estimate fitness estimate. Other methods in the robust SO optimization literature that are appropriate for reducing computational cost of robust MOEAs include:

- Allocation of more computational resource for the evaluation of Pareto-optimal solutions,
- Sampling of neighborhood solutions,
- Adaptation of computational resource allocation for evaluation through the evolutionary process, and
- Use of approximate models in place of the original objective functions.

A viable option for the efficient optimization of expected objective values is the method of *implicit averaging* [207, 208] where individual solution is perturbed once before evaluation. This approach is based on concept that solutions are implicitly averaged over a set of perturbed samples as the MOEA tends to revisit promising regions of the search space. Tsutsui and Ghosh also showed, by means of the Schema theorem, that an EA with infinite population size working on perturbed evaluations has the same effects as working on the effective fitness.

7.2.2 MO approach

The MO approach involves both deterministic and stochastic evaluation of the various objectives and robust measures respectively. Therefore, computational cost is also an important issue as in the case of the SO approach.

At present, there are two variants of the MO approach for robust MOEA. The first approach optimizes the selected robust measures on top of the existing deterministic objective functions and sought to discover the inherent tradeoff between optimality and robustness. This is also known as the MO approach [109] and various combinations of different measures such as expected fitness and variance-based measures have been applied in the evolutionary robust SO optimization literature. In [167], Ray utilized three objectives, the deterministic objective value, the effective objective value and the standard deviation, to evolve designs that remains feasible under decision variable variations. In [136], Lim *et al* also presented a SO/MO inverse evolutionary optimization methodology for robust design. In contrast to conventional forward robust optimization, the inverse approach avoid making assumptions about the uncertainty when insufficient field data exists for estimating its structure. Apart from the objectives of nominal fitness and robustness, Lim *et al* consider the possible benefits as the uncertainty prevails by introducing an opportunity criterion in the inverse search scheme as the third objective. The second variant is proposed by Deb and Gupta [40] as a more practical approach to the SO method and treats the selected robust measures as hard constraints. The goal is to evolve the best PF^{A}_{det} that satisfies the tolerable bounds on performance deviation.

7.3 Robust Optimization Problems

This section presents a set of guidelines for the construction of robust MO test problems. Based on the existing literature on robust optimization, Section 7.3.1 reviews the different categorization of robust problems and presents a classification schemes applicable to MO optimization. Desirable properties of robust MO problems are highlighted and some existing test problems are analyzed empirically in Section 7.3.2. Subsequently, the robust landscape generator and detailed construction guidelines are presented in Section 7.3.4. Finally, a vehicle routing problem with stochastic demand (VRPSD) test suit is proposed as an example of a real-world representation of combinatorial robust MO problem in Section 7.3.5.

7.3.1 Robust MO Problem Categorization

Robust optimization is very similar to noisy optimization and often considered in the same context. However, there are significant differences between these two forms of uncertainties. For noisy optimization, uncertainty is *inherent* to the objective functions and it tends to mislead the optimization process, resulting in convergence to sub-optimal solutions. In the case of robust optimization, noise is *incorporated* into the objective functions to guide the optimization process to regions that are less sensitive to parametric variations.

Different categorization of robust problems have been considered in the literature. Based on the source of uncertainty, Jin and Branke [107] states that robust optimization can be considered from the perspective of solution sensitivity to decision variable variation or environmental variation. Decision variable variation stems from the fact that deviations from design specifications are inevitable in manufacturing. On the other hand, environmental variable variation arises from variations in operational or environmental conditions. Instances of environmental variations include temperature changes in circuit design [204], speed changes in aerodynamic shape and turbine blade design, and machine breakdowns in manufacturing scheduling.

An alternative classification of SO problems based on the relationship between the efficient and effective fitness landscapes is presented recently in [157]. Paenke *et al* proposed four categories: 1) identical optimum where efficient and robust optimum are identical, 2) neighborhood optimum where efficient and robust optimum are located on the same peak or trough, 3) local-global flip where one of the local optimum corresponds to the robust optimum, and 4) max-min flip where the global maximum corresponds to the robust optimum. Deb and Gupta [40] considered a similar classification that is specific to the context of MO optimization: 1) the global efficient front is robust, 2) a part of the global efficient front is not robust, 3) the robust front is represented by a local efficient front, and 4) the robust front is represented by both the global and local efficient fronts.

Robust MO problems are certainly much more complex than SO problems as both decision space and objective space are susceptible to change due to uncertainties. Recent studies [14,182] have shown that some problems have the interesting property of demonstrating fitness topological changes in the presence of noise. To be precise, topological variation strictly refers to the introduction of new problem features to the deterministic problem under the influence of noise. For the two classification schemes described above, problems of the first category are typically considered to be less interesting as compared to problems of the other classes. On the other hand, it is possible that noise-induced landscape variation can actually result in a more challenging optimization problem even if the location of the optimum remains the same. Moreover, a landscape transformation may result from the addition of different robust criteria as objectives to be solved. Therefore, it will certainly be more interesting to classify robust MO problems according to the aspects of change under the influence of noise, i.e, how the decision space and objective space behaves in the face of uncertainties.

Most benchmark problems in the literature are commonly characterized by the emergence of a local optimum as the most robust solution in the presence of noise, signifying a change in the location of the optimum, and in the context of MO problem, a change in

PS*. Moreover, as mentioned above, it is distinctly possible that the whole fitness topology changes leading to two distinct types of search space variation. As noted by Deb and Gupta [40], the PF* is also susceptible to changes. For the classification of robust MO problems, this paper defines a three-bit binary number where the bits, in decreasing significance, represents the presence of PF*, PS*, and landscape changes respectively. As a specific instance, a MO problem that demonstrates landscape and PS* changes is a class 5 problem under this classification.

The above classification will be useful in the investigation of the various problem characteristics impact on evolutionary MO optimization as well as identifying the suitability of the different robust handling techniques. Other aspects of robust MO problem that are worth considering includes the effect of the different robust measures on the landscape transformation and the degree of change with increasing noise levels. As shown in Figure 7.1, the various robust measures results will result in different transformation. For the latter case, the change in landscape properties such as the height of each peak may change gradually with noise or it may be a sudden change of landscape feature once a certain noise threshold is reached.

7.3.2 Empirical Analysis of Existing Benchmark Features

Several desirable properties of deterministic benchmarks and test suites have been suggested in the EA literature. In addition to these guidelines, the following issues should be considered in the development of robust benchmark problems in the context of MO optimization:

- Robust MO problems are essentially MO problems and guidelines for the construction of MO benchmark problems established in previous research should be taken into account;
- The PF* of the test functions should not be any more difficult to find compared to PF$^*_{rm,n}$ when conventional MOEAs are applied;

- Some test problems should contain existing or emergent features that pose more difficulty when robust MO optimization techniques are employed;
- The “sensitive ” component of the benchmark problems should be scalable;
- Some test problems contains possible tradeoffs in robustness between different objectives;

In general, any test function should be simple enough to allow for analysis of algorithmic behavior but, at the same time, complex enough to allow conjectures to the real-world [17]. However, a quick survey of past works will reveal the lack of problem characteristics beyond the basic landscape featuring contrastive sharp and broad peaks or troughs in the evaluation of uncertainty-handling techniques. In particular, some robust SO test functions may be too simplistic for proper algorithmic evaluation with the apparent lack of difficulties that may hinder the selection of robust MO solutions. Furthermore, some robust benchmarks are distinctly multi-modal in nature and it may be difficult to ascertain whether the robust solution found is the consequence of premature convergence or the effectiveness of the particular robust optimization technique.

Therefore, empirical investigations are conducted in this section to analyze the behavior of four existing benchmark problems found in the literature. Three of the problems studied are extended from SO benchmark problems in [17, 20, 157] using the ZDT framework [229], which allows the easy incorporation of problem characteristics that hinder MOEA progress to the Pareto front. The fourth is a robust MO problem proposed in [40]. All four benchmark problems are class 2 test functions, i.e only the PF* changes. The definitions of these extended benchmarks are summarized in Table 7.1. To examine the scalability of the “sensitive” components of these problems, experiments are conducted at $n_{x,r}$ = {2,5,10}.

In the simulation studies, two state-of-the-arts MOEA, NSGAII and SPEA2, are applied to determine the difficulty of finding PF$^*_{det}$. Both algorithms are implemented using the same binary coding scheme of 15 bits, binary tournament selection, uniform crossover, and bit

Table 7.1: Definition of robust Test Problems

Problem	Definition
rMOP1 [20]	$f_1(x_1) = x_1$, $f_2(x_2, ...x_{n_{x,r}}) = g \cdot h$, $g(x_2, ...x_{\lvert n_{x,r}\rvert}) = 1 + \sum_{i=2}^{n_{x,r}} (2 - \sin\sqrt{\lvert 40x_i\rvert} + \frac{20-\lvert x_i\rvert}{20})$, $h(f_1) = 1 - \sqrt{f_1}$ where $x_1 \in [0, 1]$, $-20 \leq x_i < 20$, $\forall i = 2, ..., n_{x,r}$
rMOP2 [17]	$f_1(x_1) = x_1$, $f_2(x_2, ...x_{n_{x,r}}) = g \cdot h$, $g(x_2, ...x_{n_{x,r}}) = 1 + \sum_{i=2}^{n_{x,r}} G(x_i)$, $\mathrm{G}(x_i) = \begin{cases} 0.2 - (x_i + 1)^2 - 0.8 \cdot \lvert\sin(6.283x_i)\rvert, & \text{if } -2 \leq x_i < 0 \\ 0.6411 + 0.6 \cdot 2^{-8\cdot\lvert x_i - 1\rvert} - 0.8 \cdot \lvert\sin(6.283x_i)\rvert, & \text{if } 0 \leq x_i < 2 \end{cases}$ $h(f_1) = 1 - \sqrt{f_1}$ where $x_1 \in [0, 1]$, $-2 \leq x_i < 2$, $\forall i = 2, ..., n_{x,r}$
rMOP3 [157]	$f_1(x_1) = x_1$, $f_2(x_2, ...x_{n_{x,r}}) = g \cdot h$, $g(x_2, ...x_{n_{x,r}}) = 1 + \sum_{i=2}^{n_{x,r}} G(x_i)$, $\mathrm{G}(x_i) = \begin{cases} 0.6 - 0.5\exp(-0.5 \cdot \frac{(x_i-0.4)^2}{0.05^2}), & \text{if } x_i < 0.4693 \\ 0.6 - 0.6\exp(-0.5 \cdot \frac{(x_i-0.5)^2}{0.02^2}), & \text{if } 0.4693 \geq x_i \leq 0.5304 \\ 0.6 - 0.5\exp(-0.5 \cdot \frac{(x_i-0.6)^2}{0.05^2}), & \text{if otherwise} \end{cases}$ $h(f_1) = 1 - \sqrt{f_1}$ where $x_i \in [0, 1]$, $\forall i = 2, ..., n_{x,r}$
rMOP4 [40]	$f_1(x_1) = x_1$, $f_2(x_2, ...x_m) = h \cdot (g + S)$, $g(x_{\lvert n_{x,r}\rvert+2}, ...x_m) = \sum_{i=\lvert n_{x,r}\rvert+2}^{m} 50x_i^2$, $S(f_1) = 1 - \sqrt{f_1}$ $h(x_2, ..., x_{\lvert n_{x,r}\rvert+1}) = 2 - 0.8\exp(\sum_{i=2}^{\lvert n_{x,r}\rvert+1}(\frac{x_i-0.35}{0.25})^2)$ $-\exp(\sum_{i=2}^{\lvert n_{x,r}\rvert+1}(\frac{x_i-0.85}{0.03})^2)$ where $x_1 \in [0, 1]$, $-20 \leq x_i < 20$, $\forall i = 2, ..., m$

Table 7.2: Empirical Results of NSGAII and SPEA2 for the different robust MO test functions

			NSGAII			SPEA2	
		Ratio	Space	Maximum Spread	Ratio	Space	Maximum Spread
rMOP1	2-D	0.8867	0.5411	1.0	0.7280	0.6582	1.0
	5-D	0.0	0.5431	1.0	0.0	0.6783	1.0
	10-D	0.0	0.5626	0.9999	0.0	0.6773	0.9999
rMOP2	2-D	0.9947	0.5314	1.0	0.9890	0.6362	1.0
	5-D	0.9883	0.5369	1.0	0.9840	0.6278	1.0
	10-D	0.9850	0.5781	0.9999	0.9807	0.6849	0.9999
rMOP3	2-D	0.9853	0.5332	1.0	0.9833	0.6354	1.0
	5-D	0.7193	0.4965	1.0	0.6203	0.6484	1.0
	10-D	0.0	0.5039	0.9999	0.0	0.6268	1.0
rMOP4	2-D	0.5250	0.5066	0.9999	0.4243	0.6500	0.9999
	5-D	0.0920	0.5012	0.9997	0.0	0.6442	0.9999
	10-D	0.0	04900	0.9997	0.0	0.6174	0.9998

flip mutation. The simulations are implemented in C++ on an Intel Pentium 4 2.8 GHz computer and thirty independent runs are performed for each of the test functions in order to obtain the statistical information, such as consistency and robustness of the algorithms. The simulation results with respect to the metrics of ratio of convergence, S and MS are shown in Table 7.2. The ratio of convergence is based on the average number of nondominated solutions in each run that are located in the vicinity of PS^*_{det}. A solution is considered to be in the vicinity of the PS^*_{det} if it has a Euclidean distance of less than 0.05 difference from the nearest point in the PS^*_{det}.

From the simulation results, it is observed that NSGAII and SPEA2 generally performs similarly for the set of benchmark problems. It is evident from the metrics of S and MS in Table 7.2 that both algorithms are capable of consistent performance in the aspects of solution distribution and diversity. This is due to the manner in which the test problems are constructed where the distribution and diversity of the solution set is optimized only through the $h()$ function. With the exception of rMOP2, NSGAII and SPEA2 are unable to locate the PF$^*_{det}$ consistently and this situation worsens with increasing solution space dimensionality. Since the algorithms converges to regions that are less sensitive to parametric variation readily even without the incorporation of any robust handling mechanisms, it is clear that the characteristics exhibited by rMOP1, rMOP3 and rMOP4 are not suitable for the evaluation of robust MOEA techniques and the sensitive component of these problems are clearly not scalable. On the other hand, the problem of rMOP2 has been specially designed by Branke such that the basin of attraction and areas under the curve of the peaks are the same. Consequently, it is the only problem that allows NSGAII and SPEA2 to converge to the PF$^*_{det}$.

7.3.3 Robust MO Test Problems Design

The fundamental component of the robust MO test functions proposed in this paper is a Gaussian landscape generator that introduces various parametric sensitivities to the deterministic fitness landscape. It generates a set of $n_{x,r}$-dimensional minima throughout the fitness landscape and it is given by:

$$b(\vec{x_r}) = 1 - \frac{1}{|x_r|}\sum_{i \in x_r} \max_{j \in J}\Big\{ h_{ij} \cdot \exp[(\frac{x_i - \mu_{ij} \cdot E_{ij}(\sigma, s_{ij})}{w})^2]\Big\} \tag{7.1}$$

$$E_{ij}(\sigma, s_{ij}) \quad = \quad 1 + s_{ij} \cdot U(-\sigma, \sigma) \tag{7.2}$$

where J is the number of basis functions, d_j, μ_{ij}, and w denote the amplitude, location and the width of the basis functions. E_{ij} is function that controls how the environmental variable behaves with noise, σ and the degree of sensitivity, s_{ij}. Intuitively, the robustness of a particular basin will depend on the associated E_{ij} function while the amplitude will determine the optimality of the solution. From (7.1) and (7.2), it can be noted that test functions designed using this landscape generator is different from most previous works in two aspects:

- Any solution space or objective space transformation is a consequent of *environmental* variation. Although environmental parameter variation is rarely considered in the literature, it is definitely more flexible compared to decision parameter variation when it comes to the design of different possible scenarios.
- As observed from the simulation studies conducted in the previous section, it is important for the basin of attraction of the various troughs to be very similar. This ensures that there is no initialization bias towards any particular region of the search space.

The max function has been used successfully in previous work [65, 73] to combine the different Gaussian components, and it ensures that the landscape feature at any one point is determined and influenced only by the dominant basin. Without the overlapping influences from the other basis functions, this allows each basis function to be considered independently and facilitates the design and analysis of the robust test function. In particular, it is possible to define explicitly the location and depth of the different basins to create different test functions with specific characteristics. For the purpose of evaluating algorithmic performance, it is necessary to know the relative degree of robustness for each minima. Assuming σ is uniformly distributed that is independent for each $R_{ij}()$, the theoretical values for each basis function can be easily worked to be:

$$B_j = d_j \cdot \Big(\frac{w\sqrt{\pi}}{2s_{ij}\sigma} \cdot erf(\frac{s_{ij}\sigma_{ij}}{w}) \Big) \tag{7.3}$$

One desirable property of this test generator is that it provides a means to extend existing MO test problems to robust MO test functions without changing the original problem characteristics. The rationale is to allow researchers to investigate the impact of robust optimization on test functions with different characteristics such as deception, multi-modality and discontinuities. As a specific instance, consider the i-th objective function of an arbitrary MO benchmark problem. The corresponding objective function of the extended robust MO test function can be written as:

$$f_i'(\vec{x}) = f_i(\vec{x_d}) + b(\vec{x}_r) \tag{7.4}$$

where $\vec{x}_d$ represents the subset of decision variables associated with original problem while $\vec{x}_r$ represents the subset of decision variables of the robust component of the problem.

In this paper, the robust MO test problems are built upon the ZDT framework, which has been applied earlier in Section 7.3.2 to extend the robust SO problems. The flexibility of this framework has also been demonstrated by the development of a suite of dynamic MO problems by Farina *et al* in [52]. The guidelines for the construction of the deterministic ZDT test functions are formally described by the following

$$\begin{array}{rcl} \min f_1(\vec{x_{d1}}) & = & x_1 \\ \min f_2(\vec{x_{d2}}) & = & g(\vec{x_{d2}}) \cdot h(f_1, g) \end{array} \tag{7.5}$$

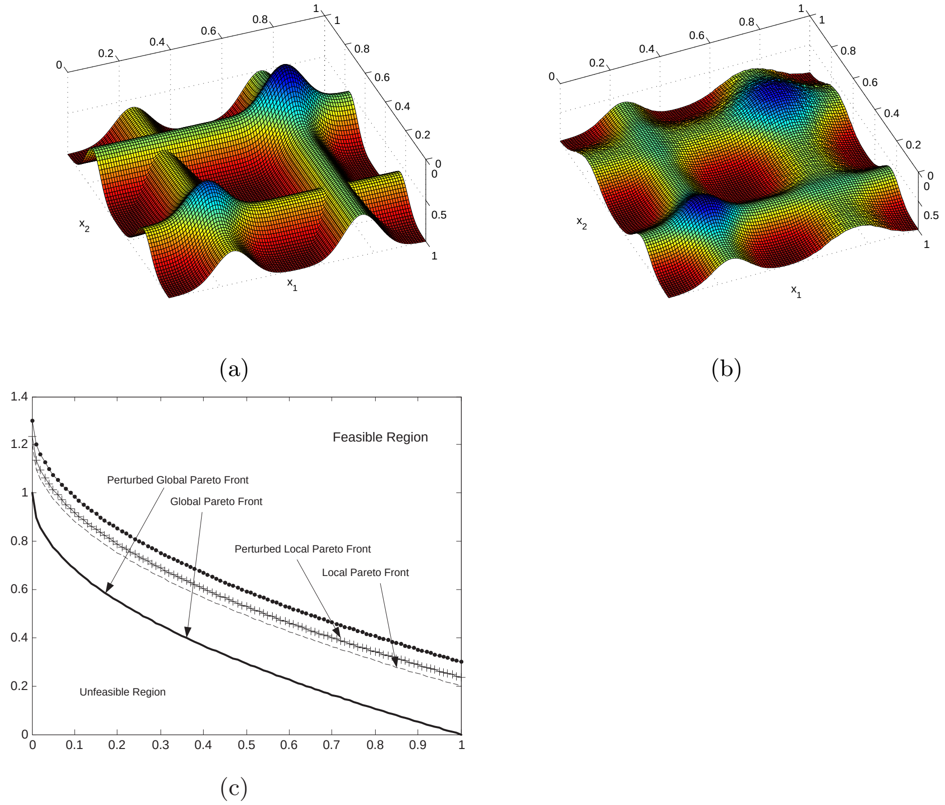
where $\vec{x_{d1}}, \vec{x_{d2}} \in \vec{x}$, and the g and h functions control the problem difficulty and the shape of the Pareto front respectively. For the ensuring discussions, we assume that the particular ZDT problem to be extended have the following functional form,

$$\begin{array}{rcl} g(\vec{x_{d2}}) & = & 1 + \sum_{x \in x_{d2}} x_i \\ h(f_1, g) & = & 1 - (\frac{f_1}{g})^{\alpha}. \end{array} \tag{7.6}$$

7.3.4 Robust MO Test Problems Design

Basic landscape generation

Noise-induced changes to the PF*, PS*, and fitness landscape can be introduced by incorporating b into either the h and/or g functions to construct different classes of robust test



(a) (b)

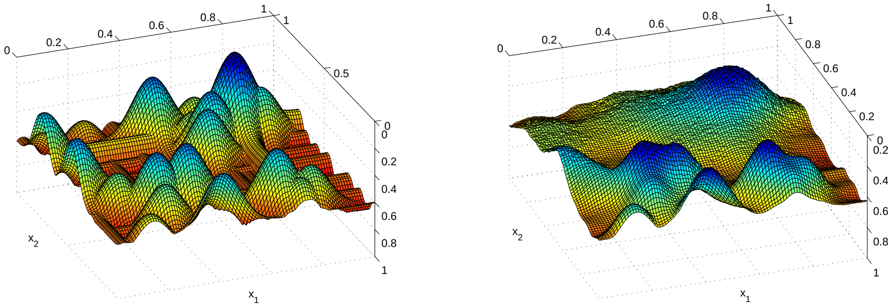
(c)

Figure 7.2: An example of a 2-D landscape with two basins with $s = 1$ at (a) $\sigma = 0.0$ and (b) $\sigma = 0.15$. The minima at (0.75,0.75) is optimal under a deterministic setting while the minima at (0.25,0.25) emerges as the global robust minima at $\sigma = 0.15$. The corresponding Pareto fronts of the resulting problem in (c) shows the relationship between the two minima.

problems. A straight forward approach of introducing robust features into the problem is to change g in the form of $g(\vec{x}) = 1 + b(\vec{x_r})$, with h and f_1 unchanged. $\vec{x_r}$ is also a subset of $\vec{x}$. Let us consider a two-dimensional landscape generated by

$$\begin{aligned} b(\vec{x_r}) &= 1 - \tfrac{1}{|x_r|}\textstyle\sum_{i\in\vec{x_r}} \max\Big\{0.8\exp[(\tfrac{x_i - 0.25E_{i1}(\sigma,s)}{0.1})^2] \\ & \quad , \exp[(\tfrac{x_i - 0.75E_{i2}(\sigma,s)}{0.1})^2]\Big\}. \end{aligned} \tag{7.7}$$

The problem landscape presented by b at $\sigma = \{0, 0.15\}$, and the resulting Pareto fronts are shown in Figure 7.2. The minima located at (0.75,0.75) is the global minima in a



(a) (b)

Figure 7.3: An example of a arbitrary 2-D landscape with $J = 40$ at (a) $\sigma = 0.0$ and (b) $\sigma = 0.15$. The minima at (0.75,0.75) is optimal under a deterministic setting while the minima at (0.25,0.25) emerges as the global robust minima at $\sigma = 0.15$.

deterministic setting and failure to converge to this point will result in a dominated solution. With $s = 1$, notice that the effects of noise on the basin at (0.75,0.75) is actually three time more than that of the basin at (0.25,0.25). Thus, when noise is incorporated into the problem, the local minima at (0.25,0.25) constitutes to the $\mathrm{PF}^*_{rm,n}$ as it is more robust and its performance is less affected by noise. Since the only change induced by noise is the location of PS^*, this is a simple instance of a class 2 robust problem.

On the other hand, if the g and b function are combined such that

$$g(\vec{x}) = (1 + \sum_{x \in x_{d2}} x_i) + b(\vec{x_r}), \tag{7.8}$$

the resulting problem is also a class 2 problem. However, such a formulation allows the analysis of the effects of robust optimization on the original problem. Thus the robust MOEA must be capable of finding the global robust minima associated with b as well dealing with the difficulties posed by the deterministic problem in order to find $\mathrm{PF}^*_{rm,n}$. It is also possible to redefine f_1 as $f_1(\vec{x}) = x_1 + b(\vec{x_r})$ to construct a class 2 problem with similar properties.

Instead of hand designing b, the landscape generator can be parameterized to generate arbitrary landscapes by specifying key geometric properties such as the number of basins, the noise level at which investigations are conducted and the locations of the optimal deterministic and robust minima. The geometric properties of the other $J-2$ basins must be selected such that their effective minima computed by (7.3) are worse than the predefined global robust optima by some predefined ratio. A basin j is considered to be more robust than a basin k if the following criteria is met: $s_{ij} \cdot \mu_{ij} < s_{ij} \cdot \mu_{ik}$. In addition, the sensitivity of each basin to noise should adhere to the condition of $s_{1j} \cdot \mu_{1j} = s_{2j} \cdot \mu_{2j} = ... =_{ij} \cdot \mu_{ij}$ for the landscape to behave properly.

Accordingly, the general form of the landscape generator can be written as

$$\langle RLS : [\vec{\mu}_r, h_r], [\vec{\mu}_g, h_g], \sigma, J, w, \beta\rangle \tag{7.9}$$

where $\vec{\mu}_r$ and h_r, $\vec{\mu}_g$ and h_g specifies the location and depth of the global robust and deterministic minima respectively while β is the factor at which the next best robust optima is worse compared to the global robust optima. An example of a 2-D landscape generated using the specification of

$$\langle[(0.25, 0.25), 0.8], [(0.75, 0.75), 1.0], 0.1, 40, 0.1, 0.1\rangle \tag{7.10}$$

is shown in Figure 7.3. Note that the landscape illustrated in Figure 7.2 can be generated by specifying $J = 2$.

Changing the decision space

When combined with g in the ways described above, the b function give rise to the element of noise-induced changes to the PS* and results in class 2 test problems. Features of noise-induced search space variation can be easily incorporated into the problem by changing g

in the following form,

$$g(\vec{x}) = 1 + (\sum_{x \in x_{d2}} x_i)^{\beta - b(\vec{x_r})}, \tag{7.11}$$

which forces the distribution of the solutions to change. Notice that g is now a function of b. In this particular instance, it is possible to apply (7.7) as the b function but finding its optimal minima will has no direct contribution to solution optimality. Interestingly, finding the optimal for b will improve the distribution of the solutions near PS* and hence simplifies the problem somewhat. Thus the resulting problem will be considered as a class 1 test problem.

More complex fitness topology variation can be induced by making h a function of g instead. In particular, consider the scenario where we define b such that the width, i.e. size of the basin of attraction, of the selected minima is a function of g and replace the g function by

$$g'(\vec{x}) = g(\vec{x_{d2}}) + b(\vec{x_r}, g). \tag{7.12}$$

The corresponding problem depends on the characteristics of the b function; it is a class 1 test problem if $J = 1$ and class 3 test problem if $J \geq 1$ and deterministic and robust optimal is different. In any event, the robust MOEA must be able to deal with the features that arises due to noise in order to find PF$^*_{rm,n}$.

Changing the solution space

Since the shape of the PF$^*_{det}$ is determined by the h function in the ZDT framework, PF$^*_{rm,n}$ can be easily controlled by combining the b and h in some appropriate way. The simplest way to introduce PF* change is to control its convexity:

$$h(f_1, g, \vec{x_r}) = 1 - (\frac{f_1}{g})^{\alpha + b(\vec{x_r})}. \tag{7.13}$$

If the g function is unchanged and b defines a single basin, only the convexity of the PF* is affected by noise while PS* remains the same. Thus, the resulting problem is a class 4 test problem and the robust MOEA must be capable of distributing the solutions along PF* with varying noise-induced convexity. However, if b is characterized by multiple basins as illustrated in Figure 7.2 or Figure 7.3, both the PF* and PS* will change and the corresponding problem becomes a class 6 test problem instead.

It is also possible to redefine the h function as,

$$h(f_1, g, b) = b(\vec{x_r}, f_1) - \sqrt{(\frac{f_1}{g})}. \tag{7.14}$$

where b is now a function of f_1 as well. One interesting implication of such a formulation, particularly if sensitivity of the relevant basins increases with f_1, is the resulting tradeoffs between the robustness and optimality of f_2. Therefore, a part of the PF* will become dominated in the presence of noise and hence only part of x_1 makes up the PF$^*_{rm,n}$. Intuitively, the corresponding problem is class 6 test problem.

Example of a robust MO test suite

Having described the possible modifications to extend the ZDT test problems, we are now in the position to suggest a suite of five robust MO test problems summarized in Table 7.3 and Table 7.4 that satisfies the requirements described in Section 7.3.2. Although not all seven classes of problems are represented, these problems embody the most challenging aspects of robust MO optimization that have been described previously. Nonetheless, interested readers are encouraged to construct more interesting problems based on the guidelines made in the previous sections. At this point, it is worth mentioning that the proposed b function can also be employed as a non-optimizable component of the problem and as a noise-sensitive environment variable instead, i.e. $b(R)$.

GTCO1 utilizes the effects of (7.12) to bring about a change from unimodal at $\sigma = 0.0$ to multimodal fitness landscape at $\sigma = 0.2$ as shown in Figure 7.4. The PS$^*_{det}$ and PS$^*_{rm,n}$ is the same at all noise levels and corresponds to $x_i \in \vec{x_{d1}} = 0$ and $x_i \in \vec{x_r} = 0$. The

Table 7.3: Definitions of the GTCO test suite

Problem	Definition
GTCO1	$f_1(\vec{x_{d1}}) = x_1$ $g(\vec{x_{d2}}) = 1 + \sum_{i=2}^{\lvert\vec{x_{d2}}\rvert}(x_i - 0.5)^2 + b(\vec{x_r}, x_i)$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $b(\vec{x_r}, x_i) = 1 - \frac{1}{\lvert\vec{x_r}\rvert}\sum_{j \in \vec{x_r}} \exp[(\frac{x_j - E_i(\sigma,s)}{W(x_i)})^2]$ $W(x_i) = 0.1 + 0.1\cos(20(x_i - 0.5)\pi) \cdot (1 - \lvert x_i - 0.5\rvert)^5$ $E_i(\sigma, s) = U(-\sigma, \sigma), \quad \vec{x_{d1}}, \vec{x_{d2}}, \vec{x_r} \in [0,1]$
GTCO2	$f_1(\vec{x_{d1}}) = x_1$ $g(\vec{x}) = 1 + b(\vec{x_r})$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $b(\vec{x_r}) = 1 - \frac{1}{\lvert\vec{x_r}\rvert}\sum_{i \in \vec{x_r}} \max\Big\{0.8\exp[(\frac{x_i - 0.25E_{i1}(\sigma,s)}{0.1})^2]$ $, \exp[(\frac{x_i - 0.75E_{i2}(\sigma,s)}{0.1})^2]\Big\}$ $E_{ij}(\sigma, s) = 1 + U(-\sigma, \sigma), \quad \vec{x_{d1}}, \vec{x_{d2}}, \vec{x_r} \in [0,1]$
GTCO3	$f_1(\vec{x_{d1}}) = x_1$ $g(\vec{x}) = 1 + 10(\sum_{i=2}^{\lvert\vec{x_{d2}}\rvert} \frac{x_i}{\lvert\vec{x_{d2}}\rvert - 1})^{1.25 - b_1(\vec{x_{r1}})} + b_2(\vec{x_{r2}})$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $b_1(\vec{x_{r1}}) = 1 - \frac{1}{\lvert\vec{x_{r1}}\rvert}\sum_{i \in \vec{x_{r1}}} \exp[(\frac{x_i^5 - E_{1,i}(\sigma,s)}{0.05})^2]$ $b_2(\vec{x_{r2}}) = 1 - \frac{1}{\lvert\vec{x_{r2}}\rvert}\sum_{i \in \vec{x_{r2}}} \max\Big\{0.8\exp[(\frac{x_i - 0.25E_{2,i1}(\sigma,s)}{0.1})^2]$ $, \exp[(\frac{x_i - 0.75E_{2,i2}(\sigma,s)}{0.1})^2]\Big\}$ $E_{1,i}(\sigma, s) = U(-\sigma, \sigma),$ $E_{2,ij}(\sigma, s) = 1 + U(-\sigma, \sigma), \quad \vec{x_{d1}}, \vec{x_{d2}}, \vec{x_{r1}}, \vec{x_{r2}} \in [0,1]$

Table 7.4: Definitions of the GTCO test suite

Problem	Definition
GTCO4	$f_1(\vec{x_{d1}}) = x_1$ $g(\vec{x_{d2}}) = 1 + 10\sum_{i=2}^{\lvert\vec{x_{d2}}\rvert} x_i$ $h(f_1, g, \vec{x_r}) = 1 - (\frac{f_1}{g})^{\alpha}, \quad \alpha = 0.5 + b(\vec{x_r})$ $b(\vec{x_r}) = 1 - \frac{1}{\lvert\vec{x_r}\rvert}\sum_{i\in\vec{x_r}} \max\Big\{0.8\exp[(\frac{x_i - 0.25E_i(\sigma,s)}{0.05})^2]$ $, \exp[(\frac{x_i - 0.75E_{ij}(\sigma,s)}{0.05})^2]\Big\}$ $E_{ij}(\sigma, s) = 1 + U(-\sigma, \sigma), \quad \vec{x_{d1}}, \vec{x_{d2}}, \vec{x_r} \in [0,1]$
GTCO5	$f_1(\vec{x_{d1}}) = x_1$ $g(\vec{x}) = 1 + \sum_{i=2}^{\lvert\vec{x_{d2}}\rvert} x_i + 5b_1(\vec{x_{r1}})$ $h(f_1, g) = 1 + 2b_2(\vec{x_{r2}}, f_1) - \sqrt{\frac{f_1}{g}}$ $b_1(x_{r1}, x_i) = 1 - \frac{1}{\lvert\vec{x_{r1}}\rvert}\sum_{i\in\vec{x_{r1}}} \exp[(\frac{x_i - E_i(\sigma,s)}{W_1(x_i)})^2]$ $b_2(\vec{x_{r2}}, f_1) = 1 - \frac{1}{\lvert\vec{x_{r2}}\rvert}\sum_{i\in\vec{x_{r2}}} \exp[(\frac{x_i - E_i(\sigma,s)}{W_2(f_1)})^2]$ $W_1(x_i) = \begin{cases} 0.2, & \text{if } x_i < 0.05 \\ 0.1x_i + 0.05, & \text{otherwise} \end{cases}$ $W_2(f_1) = 0.2 \cdot (1.1 - \sqrt{f_1})$ $E_i(\sigma, s) = U(-\sigma, \sigma), \quad \vec{x_{d1}}, \vec{x_{d2}}, \vec{x_{r1}}, \vec{x_{r2}} \in [0,1]$

problem becomes increasingly multimodal with increasing σ and this is an instance where the problem becomes more challenging and the robust MOEA will face difficulties finding PF* due to the landscape change. The settings of $\lvert\vec{x_{d2}}\rvert = 10$, $\lvert\vec{x_r}\rvert = 5$ and $\sigma = 0.2$ are recommended for GTCO1.

GTCO2 is an instantiation of the two-minima scenario considered in Section 7.3.4. This problem is similar to the problem of rMOP4 in the sense that the deterministic global and local minima switches when noise is increased beyond threshold as shown in Figure 7.5. However, as mentioned before, the basins of attraction for both minima are the same which eliminates initialization bias. The PS$^*_{det}$ corresponds to $\vec{x_r} = 0.75$ while PS$^*_{rm,n}$ corresponds to $\vec{x_r} = 0.25$. The settings of $\lvert\vec{x_r}\rvert = 10$ and $\sigma = 0.2$ are recommended for GTCO2.

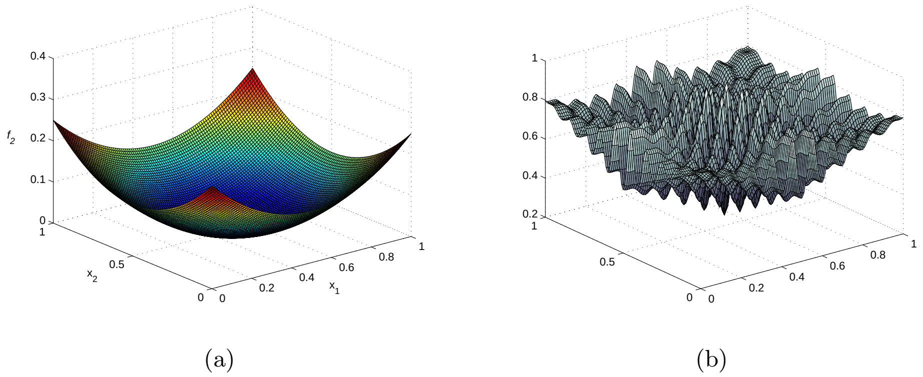


Figure 7.4: Fitness landscape of GTCO1 with $|x_r| = 2$ at (a) $\sigma = 0.0$ and (b) $\sigma = 0.2$. GTCO1 is unimodal under a deterministic setting and becomes increasingly multimodal as noise is increased.

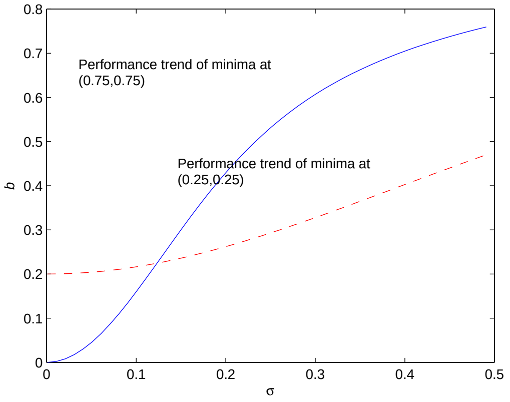
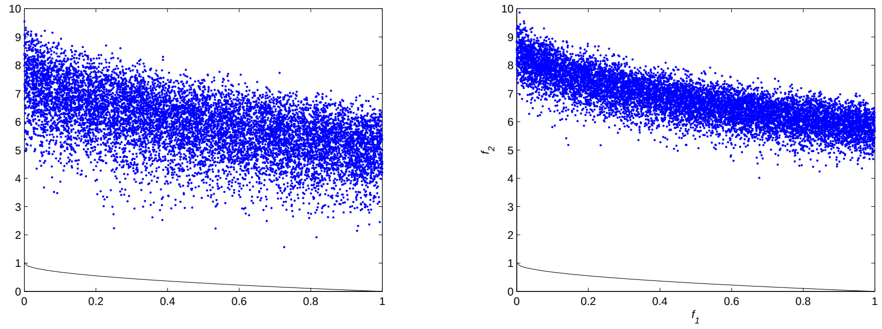


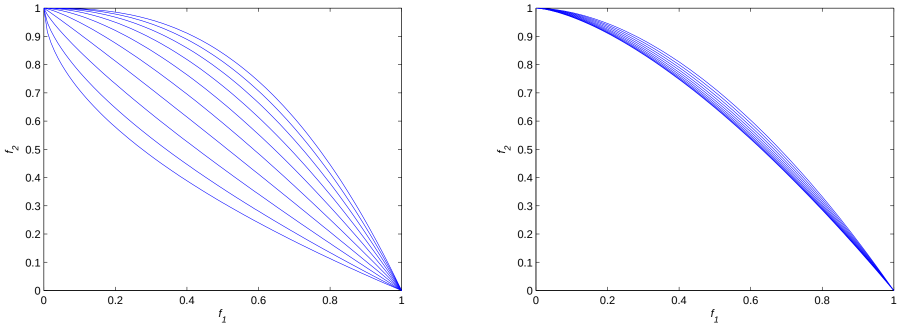
Figure 7.5: Performance variation of the two minima with increasing σ for GTCO2.



(a)

(b)

Figure 7.6: 10000 random solutions for GTCO3 at (a) $\sigma = 0.0$ and (b) $\sigma = 0.2$. The density of the solutions near the Pareto front is adversely affected in the presence of noise and deteriorates with increasing uncertainties.

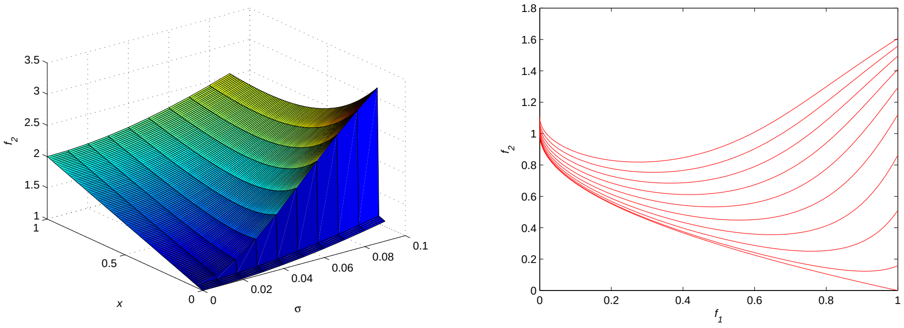


(a)

(b)

Figure 7.7: The resulting Pareto front of GTCO4 at (a) $\vec{x_r} = 0.75$ and (b) $\vec{x_r} = 0.75$ for $\sigma = [0.01, 0.1]$.

GTCO3 represents a combination of GTCO2 and the effects of (7.11) to induce both fitness landscape and PS* changes in the presence of noise. Noise-induced changes to the decision space is similar to GTCO2 except that the density of the Pareto optimal solutions is now adversely affected by noise. The behavior of the solution space at $\sigma = 0.0$ and $\sigma = 0.2$ is shown in Figure 7.6, where it can be noted that the bias away from the PS* will be attenuated with increasing σ values. The density of Pareto optimal solutions is



(a) (b)

Figure 7.8: Effects of (a) decision space variation and (b) solution space variation across different σ values for GTCO5.

at its highest and, hence easiest to find, when $\vec{x_{r2}} = 0.0$. The settings of $|\vec{x_{d2}}| = 10$, $|\vec{x_{r1}}| = |\vec{x_{r2}}| = 5$ and $\sigma = 0.2$ are recommended for this problem.

Noise-induced PS* and PF* change in GTCO4 is achieved through the implementation of (7.13). Once again, the b function governed by (7.7) is applied to generate PS* changes and the corresponding Pareto fronts at different σ levels are shown in Figure 7.8. At low levels of σ, PS* corresponds to $\vec{x_r} = 0.75$ and the PF* becomes increasingly nonconvex with noise. At sufficiently high σ levels, the PS* corresponds to $\vec{x_r} = 0.25$. Note that nonconvexity is one of the problems that posed considerable difficulty to early MO algorithms. Therefore, the robust MOEA have be capable of distributing the discovered solutions uniformly along the Pareto front for the various degrees of convexity. The settings of $|\vec{x_{d2}}| = |\vec{x_r}| = 10$ and $\sigma = 0.2$ are recommended for this problem.

GTCO5 is based on (7.14) which introduces noise-induced PS* and PF* changes. Robustness of the solutions are correlated to f_1 and this presents a conflict with the optimality of f_2. Considering the effects of this tradeoff alone, the region of PF* that remains becomes increasingly smaller with noise as illustrated in Figure 7.8(a). Fitness topological changes are based on the principle adopted in GTCO1. However, the associated b function give rise to a deceptive landscape in this instance as shown in Figure 7.8(b). The only decision variable associated with PS* that varies with σ is x_1 while the other remains at $\vec{x_r} = 0.0$ and $\vec{x_{d2}} = 0.0$. The settings of $|\vec{x_{d2}}| = |\vec{x_r}| = 10$ and $\sigma = 0.08$ are recommended for this problem.

7.3.5 Vehicle Routing Problem with Stochastic Demand

This section presents the vehicle routing problem with stochastic demand (VRPSD) as a practical example of robust MO optimization problems. The VRPSD is a variant of the classical vehicle routing problem, where customers' demands are stochastic and all other parameters are known *a priori*. The demands are treated as random variables whose distributions are known and the actual demand of each customer is revealed only when a vehicle arrives at the customer's location. This combinatorial optimization problem appears in the delivery of home heating oil [48], trash collection, sludge disposal [131], beer and soft drinks distribution, the provision of bank automates with cash, and the collection of cash from bank branches [128]. The VRPSD has been shown to be naturally MO [193] and involves not only generating minimal cost solutions but also robust solutions whose expected costs are good approximation of the actual costs of implementation. Due to the stochastic nature of the customers demands, the cost of a particular solution cannot be known with certainty and various robust measures such as the expected cost, have to be employed.

One common assumption made in this problem model is the homogeneity of all vehicles in the fleet and each one has a capacity limit which acts as a hard constraint. In the case of a route failure, i.e. a vehicle finds that a customer's demand cannot be satisfied upon reaching the customer, a recourse policy is employed to maintain the feasibility of solutions [67, 68, 129, 130, 200, 201]. The recourse policy requires the vehicle to unload all remaining goods at the particular customer, return to the depot to restock before going back to complete the service and/or continue with the scheduled route. These recourse actions will of course incur additional transportation cost, in terms of the travel distance and time for the to and fro trips to the depot. Additional service times will also be incurred when a vehicle visits a customer more than once or returns to the depot for restocking. As a matter of practicality, note that each customer can only be serviced by one vehicle.

The VRPSD is characterized by many factors, which can influence the behavior of MO routing and scheduling algorithms. As a robust MO test function, we suggest the following parameterization of the problem landscape, $\langle \text{VRPSD} : \vec{F}, N, \vec{L_c}, L_d, \vec{D}, \vec{s} \rangle$

- *Topology of customers,* $\vec{L_c}$ If it is possible to obtain real-world information about the problem, then the actual geographical distribution of the customers can be easily used to construct the problem. Otherwise, the locations of the customers can be generated randomly based on some reasonable probability distributions. The spatial distribution of customers can be categorized into three main classes [187]: 1) Type-R where all the customers are remotely located, 2) Type-C problems where the customers are grouped into clusters and 3) Type-RC which is a mixture of remote and clustered customers.

- *Customer demand distribution,* $\vec{D}$ The demand distribution determines the extent to which the robust problem deviates from the deterministic one. A uniform distribution assumes a fixed range of demands and the problem can be solved conservatively by optimizing on worst case demands. On the other hand, there is an outside chance for the occurrence of outlier demands if a normal distribution is adopted, which results in a more challenging problem. One common approach of generating the normal demand distribution model is to use the original demand quantity from some existing VRP datasets as the mean demand and generating the standard deviation of the demand distribution of each customer randomly such that it falls between zero and one-third of the mean demand of the customer [47, 193].

- *Location of depot,* L_d The location of the depot has a significant impact on the optimization process. For instance, a depot that is located at the centre of a map would give the depot better proximity from all the customers and allow shorter trips back to the depot for restocking in the event of route failures. The other extreme case would be to locate the depot at a corner of the map. This would make recourse actions more expensive and emphasize the importance of robust solutions to the stochastic problem.

- *Service duration,* $\vec{s}$ A convenient way of specifying the service duration is to set the service times of all customers to be the same [193]. However, it is quite unlikely that this would be the case in reality. It would be more appropriate if service times were given physical meanings such as the time required for loading and unloading of cargo or the time required for clearing the customs. In either case, the service time could be proportional to the amount of cargo to be loaded at the depot and unloaded at the customer. Problems with longer service times would also amplify the need for robust solutions that minimize the occurrences of route failures since multiple trips to the depot for restocking would be more costly.

- *Number of customers,*$|V|$ V is the set of all customers. It is clear that the problem gets more difficult as the number of customers increases. Typical problem sizes that have been adopted in conventional vehicle routing problem with time withdrawals (VRPTW) test problems range from 100 [187] to 1000 [89] customers and serves as a good guide for VRPSD.

- *Optimization criteria,* $\vec{F}$ As in all real-world optimization problems, it is desirable to minimize overall operational cost which includes factors such as travel distance (C_d), the number of vehicles involved (C_v) and monetary cost such as driver remuneration (C_m). Thus, the MO VRPSD is to find the routing schedule S such that it:

$$\min \vec{F}(S) = \{C_d(S), C_v(S), C_m(S)\}. \tag{7.15}$$

A simple example of the routing plan with $|V| = 6$ is illustrated in Fig. 7.9. The routing schedule S is given as S={R$_1$, R$_2$} where R$_1$ is represented by R$_1$= $\langle v_1, v_6, v_2, v_3\rangle$ and R$_2$ is represented by R$_2$= $\langle v_5, v_4\rangle$. The depot is omitted since all vehicles must depart and return

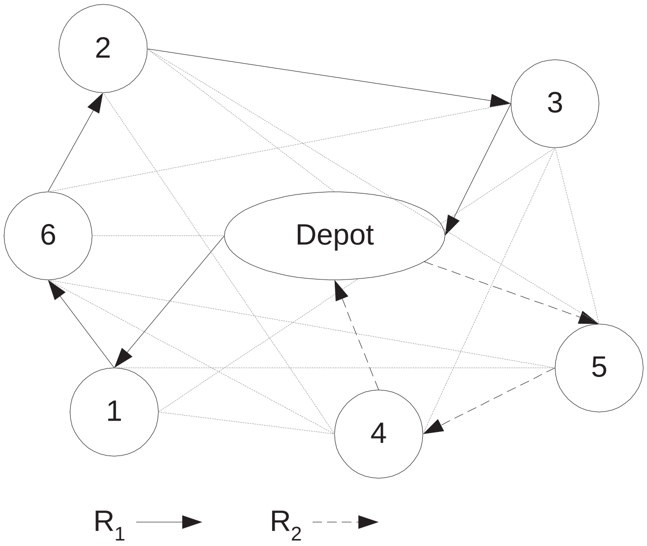


Figure 7.9: Graphical representation of a simple vehicle routing problem.

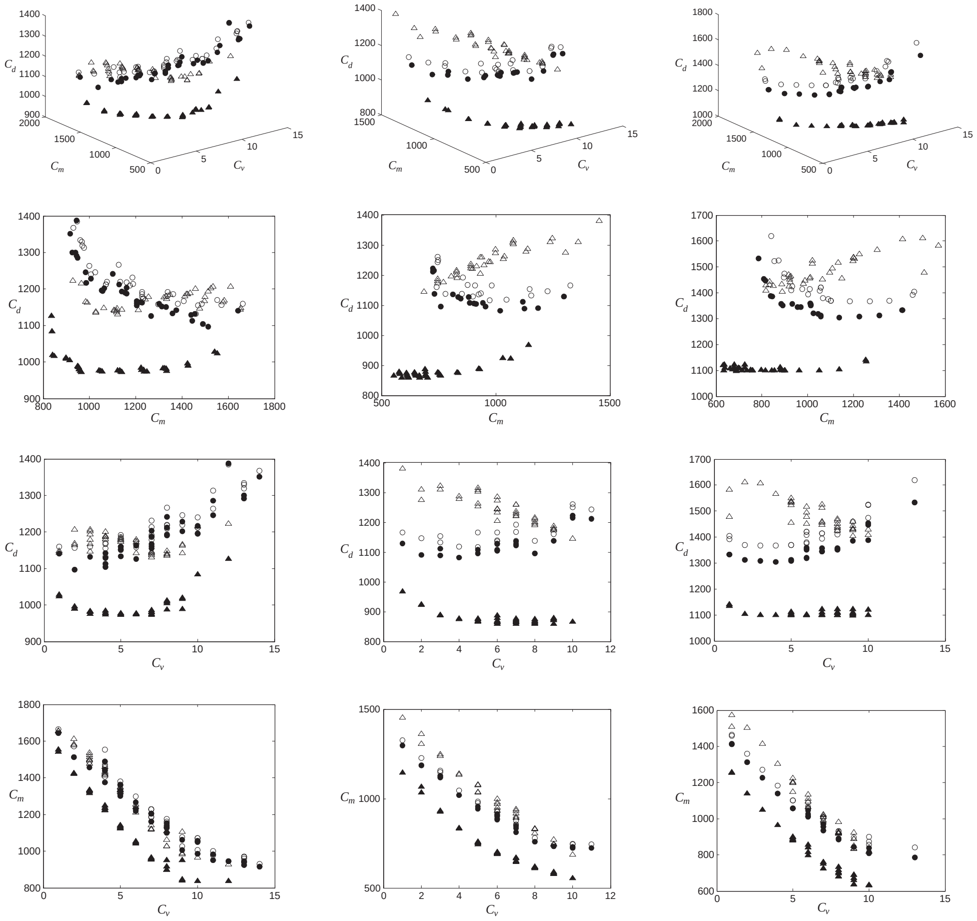
to the depot. It can be seen that the number of routes $|S|$ is equal to the number of vehicles (C_v) used in the plan. The condition of $\bigcup_i^{|S|} R_i = V$, i.e. all customers are routed, must be satisfied.

Figure 7.10 shows the PF^A_{det} $\text{PF}^A_{eff,n}$ for three different instances of the VRPSD problem. The Pareto front obtained for Figure 7.10(a) is based on

$$\begin{aligned} \langle \text{VRPSD1} \;&:\; [C_d, C_m, C_v],\ 100,\ \text{Type-R},\ (50,50) \\ &,\ N(\mu, U(0, \tfrac{1}{3}\mu),\ 10\rangle, \end{aligned} \tag{7.16}$$

the Pareto front obtained for Figure 7.10(b) is based on

$$\begin{aligned} \langle \text{VRPSD2} \;&:\; [C_d, C_m, C_v],\ 100,\ \text{Type-C},\ (50,50) \\ &,\ N(\mu, U(0, \tfrac{1}{3}\mu),\ 10\rangle, \end{aligned} \tag{7.17}$$



(a) (b) (c)

Figure 7.10: Pareto fronts for (a) VRPSD1, (b) VRPSD2, (c) VRPSD3 test problems. The first row shows the 3-dimensional Pareto fronts, the second row shows the same fronts along C_d and C_m, the third row shows the same fronts along C_d and C_v and the fourth row shows the same front along C_m and C_v. ○ denote solutions evolved using averaging while △ denote solution evolved deterministically. ● and ▲ represent the corresponding solutions after averaging over 5000 samples.

while the Pareto front obtained for Figure 7.10(c) is based on

$$\begin{aligned}\langle \text{VRPSD3} \ : \ & [C_d, C_m, C_v],\ 100,\ \text{Type-RC},\ (50,50) \\ & ,\ N(\mu, U(0, \tfrac{1}{3}\mu),\ 10\rangle,\end{aligned} \tag{7.18}$$

Evidence of landscape changes and solution sensitivity to demand variation is indicated by the different degrees of robustness between the PF^A_{det} and $\text{PF}^A_{eff,n}$. Different problem behavior can also be observed as VRPSD2 seemed to be most susceptible to noise. The Pareto fronts observed in the second and third rows of the figure also implies that PF^*_{det} $\text{PF}^*_{eff,n}$ have very different shapes. In particular, the PF^A_{eff} shows that the C_d decreases with increasing C_m but the PF^A_{det} shows the exact opposite relationship, i.e. the C_d increases with increasing C_m. Additionally, the PF^A_{eff} show that the C_d increases with the C_v, while the PF^A_{det} shows that the reverse relationship. This finding demonstrates that an algorithm designed to find the PF^*_{det} may result in routing schedules with unacceptable performance variation.

7.4 Empirical Analysis

In this section, simulation studies are conducted to analyze the performances of NSGAII and SPEA2 on the proposed GTCO test suite. In particular, we investigate the performance of NSGAII and SPEA2 over the number of samples, H={1, 5, 10, 20} and σ={0.0, 0.05, 0.1, 0.2} for the different problems. Thirty independent runs of 500 generations are performed for each of the test problems. Monte Carlo integration with number of samples H is applied.

Apart from the additional goal of solution robustness, the MO optimization goals of convergence, distribution and diversity must be considered. As before, the metrics of MS and S will be applied to assess algorithmic performance with respect to solution set diversity and distribution respectively. Since we are interested in robust designs, the metric of VD described in Chapter 5 is used in this section.

GTCO1 is a class 1 problem where multimodality is introduced into the fitness landscape with noise while PS$^*_{det}$=PS$^*_{eff,\sigma}$. From the metric of VD in Figure. 7.11(a) as well as the distribution of the solutions in Figure 7.12(a)-(d), NSGAII and SPEA2 encounters no difficulty finding the PS$^*_{det}$ but both algorithms face increasing difficulties with increasing σ. The diversity of PF$^A_{eff,\sigma}$ is also affected. This clearly demonstrates how robust optimization can be more difficult in the face of noise-induced landscape features.

Figure 7.13 shows the performance trend of SPEA2 and NSGAII for GTCO2. GTCO2 is a class 2 problem which exhibits a change in PS* once σ exceeds a certain threshold. The variation of PS$^A_{eff,\sigma}$ for both algorithms can be seen in Figure 7.14. Interesting, the effects of implicit averaging can be observed in Figure 7.14(a) and both algorithms are capable of finding PS$^*_{eff,\sigma}$ as well as a diverse PF$^A_{eff,\sigma}$ given sufficient number of samples. While both algorithms exhibit similar performances, it can be noted from Figure 7.13(b) and Figure 7.14(d) (along the x_1 axis), that the diversity maintenance mechanism of SPEA2 are more susceptible to noise.

GTCO3 is a class 3 problem which exhibits noise-induced landscape and PS* changes and the performance trends of NSGAII and SPEA2 are shown in Figure 7.15. As in the problem of GTCO2, NSGAII and SPEA2 are capable of finding PS$^*_{eff,\sigma}$ as well as a diverse PF$^A_{eff,\sigma}$ given sufficient number of samples. Nonetheless, even though GTCO2 and GTCO3 undergo the same PS* transformation, it is obvious that the change in solution density results in different sampling requirements. This is evident by comparing Figure. 7.14(b)-(c) and Figure 7.16(b)-(c).

Similar to GTCO2 and GTCO3, GTCO4 shows similar performance trends over different σ and H values in Figure 7.17. However, since the change in PS* is closely linked to the change in PF*, the number of samples necessary to find PS* is also higher. The PF* changes with σ as shown in Figure 7.18. Once again, it can be observed from Figure 7.17(b) and Figure 7.18(d) (along the x_1 axis) that noise poses considerable challenge to the diversity maintenance mechanism of SPEA2 as it is unable to distribute the solutions uniformly along

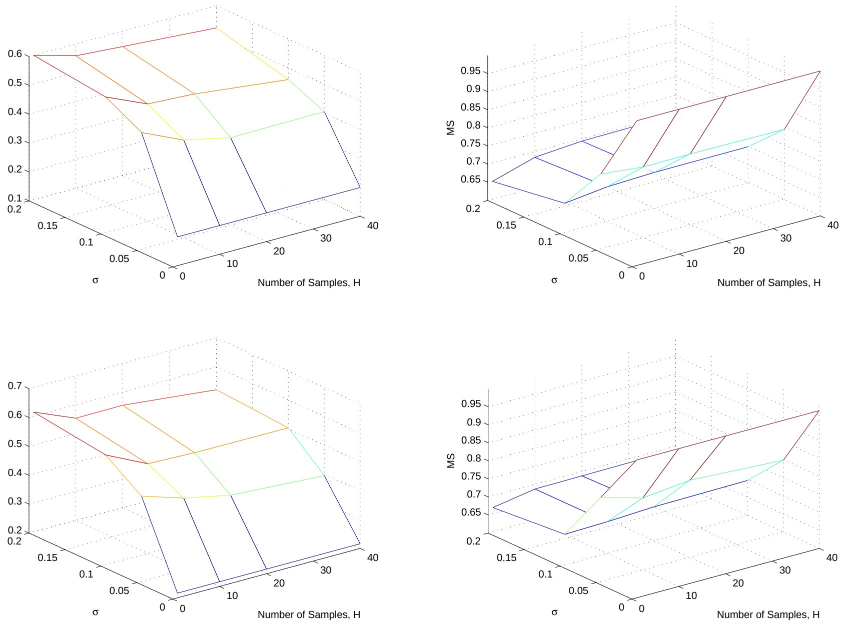
$\mathrm{PF}^{A}_{eff,\sigma}$.

As in the case of GTCO1, NSGAII and SPEA2 have problems finding PS* as evident from the metric of VD in Figure. 7.19(a). From Figure. 7.19(b), it appears that algorithmic performances in terms of MS seem to have improved beyond $\sigma = 0.1$. However, it is due to the fact that the span of $\mathrm{PF}^{A}_{eff,\sigma}$ has reduced considerably. Moreover, further investigation reveals that the number of solutions found is very small, even at H=40.

From the simulation results, it is clear that the problems of the GTCO test suite pose different difficulties to NSGAII and SPEA2. Generally speaking, both algorithms exhibit similar performances for the five problems over different H and σ settings. Nonetheless, it is also noted the SPEA2 did not fare as well in terms of the discovery of a uniformly distributed and diverse $\mathrm{PF}^{A}_{eff,\sigma}$. Together with the observations that NSGAII and SPEA2 are unable to handle noise-induced features of multi-modality and deception, this suggests that the state-of-the-arts MOEAs may not be able to evolve robust solutions effectively through simplistic extensions.

7.5 Conclusion

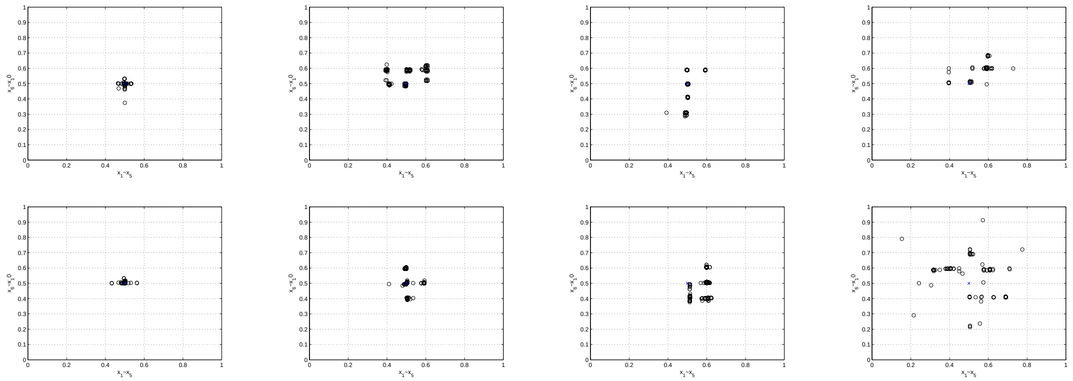
Apart from the need to satisfy several competing objectives, many real-world applications are also sensitive to decision or environmental parameter variation which results in large or unacceptable performance variation. Although the application of evolutionary multi-objective optimization to real-world problems are gaining popularity from researchers in different fields, there is a distinct lack in studies investigating the issues of robust optimization in the literature. This chapter examines the suitability of existing robust test problems for MO optimization and presents a set of guidelines for the construction of robust MO test problems. The fundamental component of the robust test problems is a Gaussian landscape generator that facilitates the specification of robust optimization-specific features such as noise-induced solution space, fitness landscape and decision space variation. This generator



(a)

(b)

Figure 7.11: GTCO1 Performance trend of NSGAII (first row) and SPEA2 (second row) over H={1, 5, 10, 20} and σ={0.0, 0.05, 0.1, 0.2} for (a) VD and (b) MS.



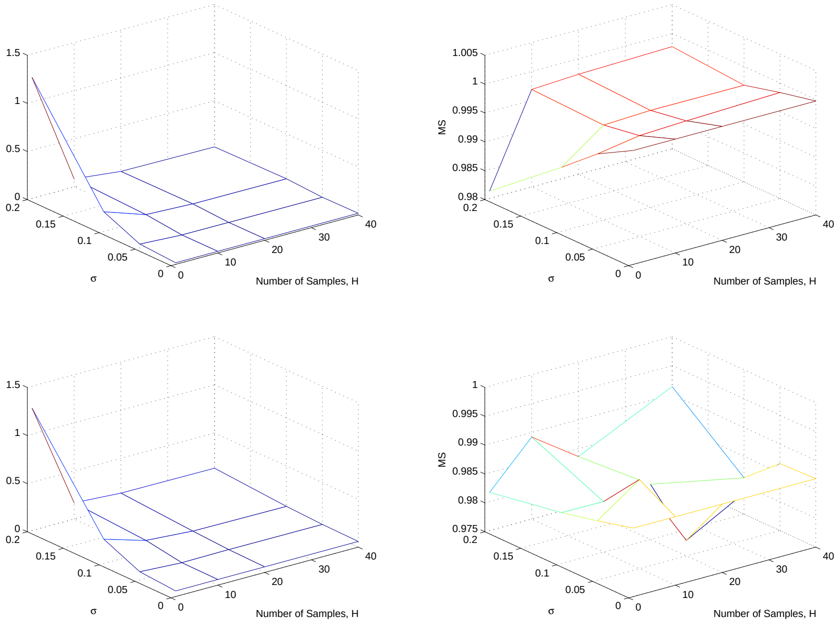
(a)

(b)

(c)

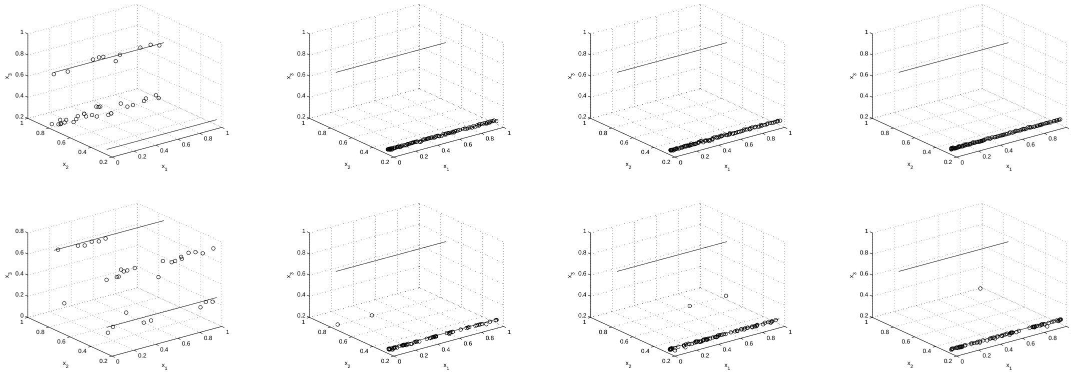
(d)

Figure 7.12: The evolved solutions of NSGAII (first row) and SPEA2 (second row) with number of samples H=1 for GTCO1 along $\vec{x_{d2}}$ with number of samples (a) $\sigma = 0$, (b) $\sigma = 0.05$, (c) $\sigma = 0.1$, and (d) $\sigma = 0.2$. The PS* is represented by (x) while the evolved solutions are represented by ($\circ$).



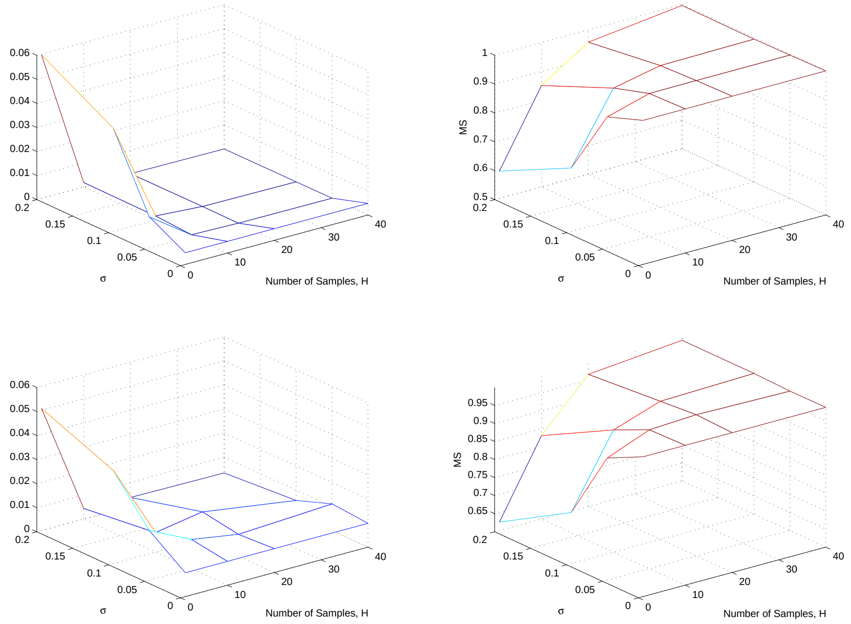
(a) (b)

Figure 7.13: GTCO2 Performance trend of NSGAII (first row) and SPEA2 (second row) over H={1, 5, 10, 20} and σ={0.0, 0.05, 0.1, 0.2} for (a) VD and (b) MS.



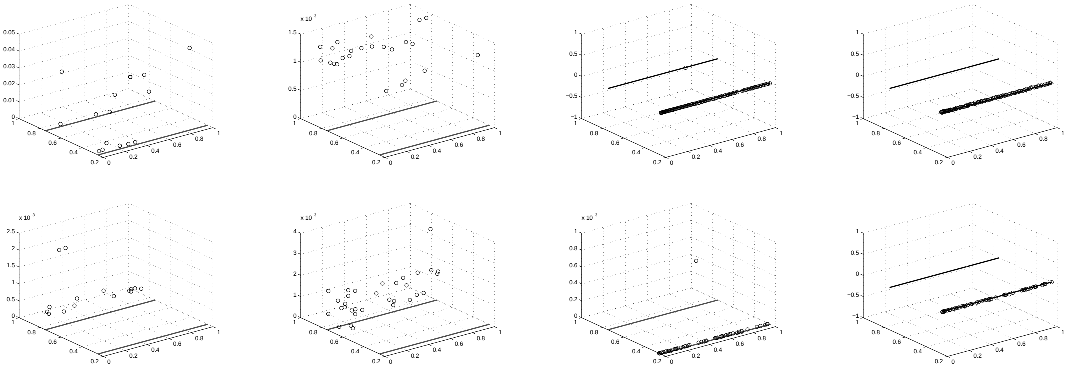
(a) (b) (c) (d)

Figure 7.14: The evolved solutions of NSGAII (first row) and SPEA2 (second row) at $\sigma = 0.2$ for GTCO2 as seen in the decision space with number of samples (a) H0, (b) H=5, (c) H=10 , and (d) H=20. The PS* is represented by (-) while the evolved solutions are represented by ($\circ$).



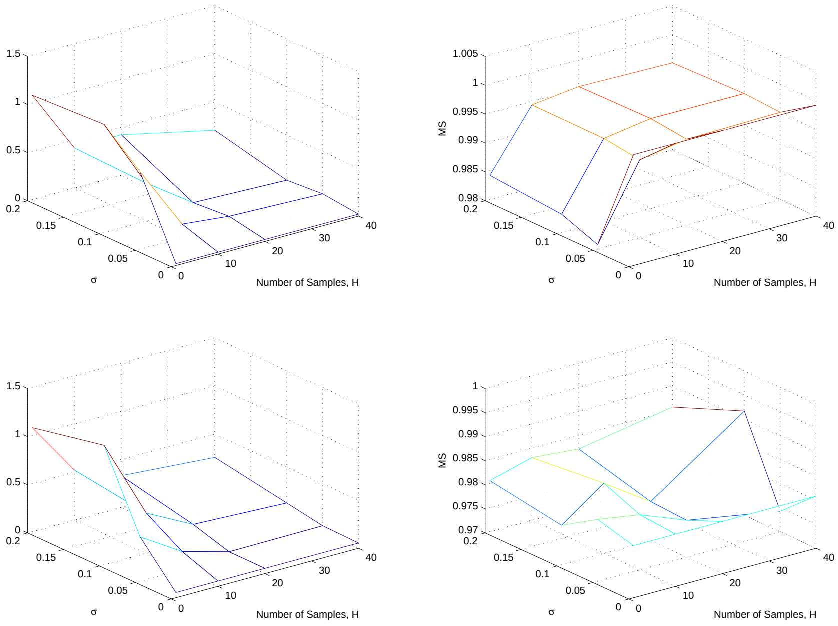
(a) (b)

Figure 7.15: GTCO3 Performance trend of NSGAII (first row) and SPEA2 (second row) over H={1, 5, 10, 20} and σ={0.0, 0.05, 0.1, 0.2} for (a) VD and (b) MS.



(a) (b) (c) (d)

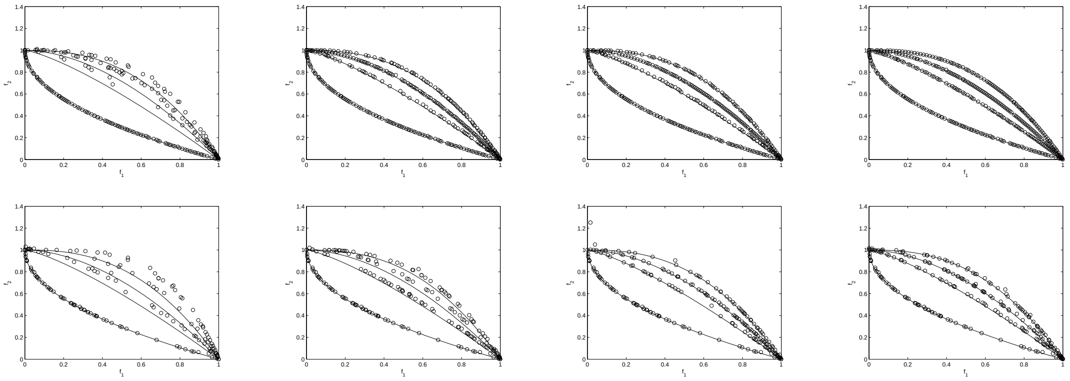
Figure 7.16: The evolved solutions of NSGAII (first row) and SPEA2 (second row) at $\sigma = 0.2$ for GTCO3 as seen in the decision space with number of samples (a) H0, (b) H=5, (c) H=10 , and (d) H=20. The PS* is represented by (-) while the evolved solutions are represented by ($\circ$).



(a)

(b)

Figure 7.17: GTCO4 Performance trend of NSGAII (first row) and SPEA2 (second row) over H={1, 5, 10, 20} and σ={0.0, 0.05, 0.1, 0.2} for (a) VD and (b) MS.



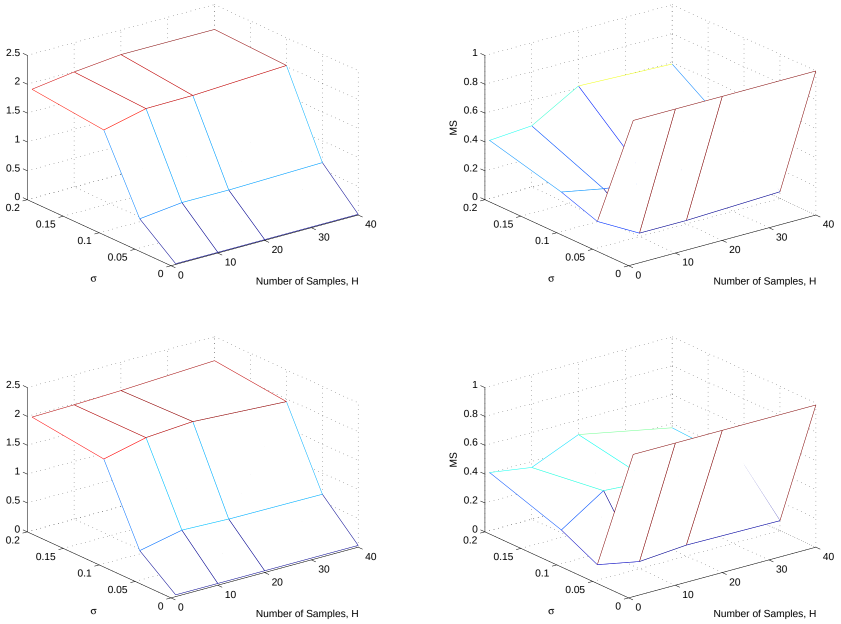
(a)

(b)

(c)

(d)

Figure 7.18: The PF^A of NSGAII (first row) and SPEA2 (second row) at various σ = 0.2 values for GTCO4 as seen in the decision space with number of samples (a) H0, (b) H=5, (c) H=10 , and (d) H=20. The PF* is represented by (-) while the evolved solutions are represented by ($\circ$).



(a) (b)

Figure 7.19: GTCO5 Performance trend of NSGAII (first row) and SPEA2 (second row) over H={1, 5, 10, 20} and σ={0.0, 0.05, 0.1, 0.2} for (a) VD and (b) MS.

is developed with the purpose of generating noise-sensitive landscapes in conjuction with existing MO test problems, and due to its independent nature, it can be used to generate robust single objective test problems as well. Subsequently, a robust MO test suite is built upon the ZDT framework. Additionally, the vehicle routing problem with stochastic demand (VRPSD) is presented a practical example of robust combinatorial MO optimization problems. In order to demonstrate the difficulties posed by the proposed test problems, NSGAII and SPEA2 are applied on all five continuous test problems. The study suggests that robust MO problems can offer greater challenges to the optimization algorithms when noise is introduced. Furthermore, it highlights the necessity to design more effective MOEAs as well as more rigorous simulation studies.

Chapter 8

Conclusions

MOEAs are a class of stochastic search methods that have been found to be very efficient and effective in solving sophisticated MO problems where conventional optimization tools fail to work well. MOEAs' advantage can be attributed to it's capability of sampling multiple candidate solutions simultaneously, a task that most classical MO optimization techniques are found to be wanting. Since MOEA draw its inspiration from nature where uncertainty is a common phenomenon, these algorithms are also expected to be inherently robust to uncertainties. Much work has been done to the development of these algorithms in the past decade and it is finding increasingly application to the fields of bio informatics, logical circuit design, control engineering and resource allocation. Interestingly, many researchers in the field of EMOO assume that the optimization problem can be modeled and determined exactly. Consequently, the issues of uncertainties are rarely examined.

8.1 Contributions

This work contributes towards the design of effective MOEAs in the presence of uncertainties. We have investigated the impact of noisy fitness functions on the performance of MOEAs. An in-depth empirical analysis is first carried out to understand how MOEA perform in the presence of different noise levels. In addition, three noise-handling features, including an experiential learning directed perturbation operator, a gene adaptation selection strategy, and a possibilistic archiving model are proposed to handle noisy fitness functions. Simulation studies showed the effectiveness of the proposed mechanisms on the test functions employed in handling both static and noisy environments. The extendibility of the experiential learning directed perturbation operator and gene adaptation selection strategy is validated in SPEA2 and NSGAII. In particular, the incorporation of the two mechanisms allow SPEA2 and NSGAII to perform better for both static and noisy environments. Analysis of population distribution reveals why the operator works.

Chapter 4 considers the noisy problem of neural network classifiers adaptation. Noise is introduced as a consequence of synaptic weights that are not well trained for a particular network structure. Therefore, it is necessary to optimize the synaptic weights after any structural changes. This chapter starts with the proposal of a geometrical measure based on the singular value decomposition (SVD) to estimate the necessary number of neurons to be used in training a single hidden layer feedforward neural network is presented. Subsequently, an architectural recombination procedure based on the geometrical measure that adapts the number of necessary hidden neurons and facilitates the exchange of neuronal information between candidate designs is presented. In order to reduce the effects of noise due to inappropriate instances of synaptic weights, a μHGA with an adaptive local search intensity scheme for local fine-tuning is developed. The importance of μHGA to reduce the noise of the synaptic weights with respect to each network structure is demonstrated in the empirical studies.

Chapter 6 examines the effects of fitness landscape dynamism. The coevolutionary paradigm can be exploited to achieve high convergence speeds to track the changing Pareto front. However, the conventional model is limited by the need to hand-design the decomposition process which is dependent on the problem characteristic. Therefore, a new coevolutionary paradigm which incorporates both competitive and cooperative elements is proposed to allow the decomposition process of the optimization problem to adapt and emerge. In particular, each species subpopulation will compete to represent a particular subcomponent of the MO problem while the eventual winners will cooperate to evolve the better solutions. Simulation results demonstrates the ability of the competitive-Cooperation coevolutionary paradigm to handle high parameter interactions of KUR and FON as well as the changing requirements of applied dynamic MO test problems. Further investigations are conducted to reveal how and why the proposed coevolutionary model works.

Chapter 7 investigates the suitability of existing robust test problems for MO optimization and demonstrates that most of these test functions have a bias towards the robust optimal. Thus if MOEA are evaluated on the extensions of these problems, it will not be a good indicator the robust-handling mechanisms. This chapter presents a Gaussian landscape generator to generate different landscapes that are sensitive to environmental variation. In addition, a set of guidelines for the specification of robust optimization-specific features such as noise-induced solution space, fitness landscape and decision space variation is presented. Based on this framework, a robust MO test suite is built upon the ZDT framework and empirical studies demonstrated the difficulties that is posed by the noise-induced features for SPEA2 and NSGAII. In addition, the vehicle routing problem with stochastic demand (VRPSD) is presented a practical example of robust combinatorial MO optimization problems.

8.2 Future Works

Although we have studied different means of handling the various forms of uncertainties, these works barely scratched the surface of what is left to be addressed. Although detailed analysis of issues related to the application of MOEAs in uncertain environments have been provided in this work, most discussions are based exclusively on empirical results on benchmark test problems with 2-5 objectives. In addition, computational efficiency is considered only in terms of the number of function evaluation calls. Since the number of evaluation is not the best indicator of efficiency especially when evaluation cost is low, one immediate extension would be to consider the computational time complexity of the proposed mechanisms. Certainly, it would also be desirable to apply the proposed techniques to benchmark functions with larger number of objectives and real-world problems in the near future.

Like most existing work, this research has concentrated on the effects of Gaussian noise. On the other hand, it has been shown that other noise models such as Cauchy and χ^2 have different and significant impacts on the optimization process. Therefore, this is definitely one area that should be dealt with in the near future. Another possible implication of different noise impact is that the heuristical approaches proposed in this work may not be extended to handle the various noise models. On the other hand, the use of the alternative method of averaging is computationally expensive. While surrogate models have been applied in the domain of SO optimization to find robust solutions, it has yet been studied for MO problems. For that matter, it will be interesting to note that it has not been utilized for other forms of uncertainties as well. One possible application of surrogate models is in the optimization of noisy fitness functions where noise may be filtered out through the approximation process-an advantage that may have been overlooked thus far.

In the aspects of dynamic MO optimization, the categorization of dynamic MO problems actually reveals that there are many possible types of test functions that have yet been explored. In addition, one common assumption in evolutionary dynamic optimization is that changes in the landscape can be detected easily by checking for discrepancies between the old and re-evaluated objective values. This may not be the case in the event where new peaks are introduced without affecting the fitness values of existing nondominated solutions. In addition, the re-evaluation of past solutions are computationally expensive. Therefore, more effective and efficient detection mechanisms must be designed. Another issue is the exploitation and storage of past solutions. In this work, we adopted a very simple first-in-first-out approach of storing past nondominated solutions prior to each change. However, more intelligent storage methodology that, perhaps take into account the utility of past information, should be explored.

In the case of robust MO optimization, we have only started the ball rolling with the presentation of a new robust MO test suite. Note that, unlike most existing research, sensitivities are introduced to the environmental variables instead of the decision variables. This necessitates the design of robust MOEAs that are capable of handling such parametric sensitivities.

Bibliography

[1] H. Abbass and R. Sarker and C. Newton, “PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems,” in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pp. 27-30, 2001.

[2] H. Abbass, “A Memetic Pareto Evolutionary Approach to Artificial Neural Networks,” in *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pp. 1-12, 2001.

[3] H. A. Abbass, “Speeding up backpropagation using multiobjective evolutionary algorithms,” *Neural Computation*, vol. 15, pp. 2705-2726, 2003.

[4] P. J. Angeline and J. B. Pollack, “Competitive environments evolve better solutions for complex tasks,” in *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 264-270, 1993.

[5] M.-L. Antonie, O. R. Zaiane and, R.C. Holte, “Learning to Use a Learned Model: A Two-Stage Approach to Classification,” in *Proceedings of the Sixth International Conference on Data Mining*, pp. 33-42, 2006.

[6] D. V. Arnold and H. G. Beyer, “Local performance of the (1+1)-ES in a noisy environment,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 30-41, 2002.

[7] D. V. Arnold and H. G. Beyer, “A General Noise Model and Its Effects on Evolution Strategy Performance,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 380-391, 2006.

[8] T. Back and U. Hammel, “Evolution strategies applied to perturbed objective functions,” in *Proceedings of the First IEEE Conference on Evolutionary Computation*, vol. 1, pp. 40-45, 1994.

[9] N. K. Bambha, S. S Bhattacharyya, J. Teich, and E. Zitzler, “Systematic Integration of Parameterized Local Search Into Evolutionary Algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 137-154, 2004.

[10] H. J. C. Barbosa and A. M. S. Barreto, "An Interactive Genetic Algorithm with Co-evolution of Weights for Multiobjective Problems," in *Proceedings of the 2001 Genetic and Evolutionary Computation Congress*, pp. 203-210, 2001.

[11] M. Basseur and E. Zitzler, "Handling Uncertainty in Indicator-Based Multiobjective Optimization, " *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 255-272, 2006.

[12] T. Beielstein and S. Markon, "Threshold selection, hypothesis tests, and DOC methods," in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 777-782, 2002.

[13] H. G. Beyer, "Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, pp. 239-267, 2000.

[14] H-G. Beyer and B. Sendhoff, "Evolution Strategies for Robust Optimization," in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pp. 1346-1353, 2006.

[15] P. Bosman and D. Thierens, "The balance between proximity and diversity in multi-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174-188, 2003.

[16] P. Bosman and D. Thierens, "The naive MIDEA: A baseline multi-objective EA," in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, vol. 3410, pp. 428-442, 2005.

[17] J. Branke, Evolutionary Optimization in Dynamic Environments. Norwell, MA: Kluwer, 2001.

[18] J. Branke, C. Schmidt and H. Schmeck, "Efficient fitness estimation in noisy environments," in *Proceedings of Genetic and Evolutionary Computation*, San Francisco, CA, pp. 243-250. Morgan Kaufmann, 2001.

[19] J. Branke, "Reducing the sampling variance when searching for robust solution,"in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 235-424, 2001.

[20] J. Branke, "Creating robust solutions by means of evolutionary algorithms," in *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, pp. 119-128, 1998.

[21] D. Buche, P. Stoll, R. Dornberger, and P. Koumoutsakos, "Multiobjective Evolutionary Algorithm for the Optimization of Noisy Combustion Processes," *IEEE Transactions*

on Systems, Man, and CyberneticsPart C: Applications and Reviews, vol. 32, no. 4, pp. 460-473, 2002.

[22] L. T. Bui, H. A. Abbass and D. Essam, "Fitness Inheritance For Noisy Evolutionary Multi-Objective Optimization, " in *Proceedings of the 2005 Genetic and Evolutionary Computation Congress*, pp. 779-785, 2005.

[23] J. Branke and H. Schmeck, "Designing evolutionary algorithms for dynamic optimization problems," in Theory and Application of Evolutionary Computation: Recent Trends, (eds.) S. Tsutsui and A. Ghosh, Springer-Verlag, pp. 239-262, 2002.

[24] E. Cant-Paz and C. Kamath, "An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, pp. 915-927, 2005.

[25] A. Chandra and X. Yao, "Ensemble learning using multi-objective evolutionary algorithms," *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 4, pp. 417-445, 2006.

[26] A. Chandra and X. Yao, "Evolving hybrid ensembles of learning machines for ensemble generation," *Neurocomputing*, vol. 69, no. 7-9, pp. 686-700, 2006.

[27] H. G. Cobb, "An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments, " Technical Report AIC-90-001, Naval Research Laboratory, Washington, DC, 1990.

[28] C. A. Coello Coello, "Evolutionary Multiobjective Optimization: A Historical View of the Field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28-36, 2006.

[29] C. A. Coello Coello and N. Cruz Corts, "Solving Multiobjective Optimization Problems using an Artificial Immune System," Genetic Programming and Evolvable Machines, vol. 6, no. 2, pp. 163-190, 2005.

[30] C. A. Coello Coello, G. T. Pulido and M. S. Lechuga, "Handling Multiple Objectives With Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256-279, 2004.

[31] C. A. Coello Coello and M. R. Sierra, "A Coevolutionary Multi-Objective Evolutionary Algorithm," in *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 482-489, 2003.

[32] D. W. Corne, J. D. Knowles, and M. J Oates, "The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization," in *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature*, pp. 839-848, 2000.

[33] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Transactions on Electronic Computation*, vol. 14, pp. 326-34, 1965.

[34] A. J. Chipperfield and P. J. Fleming, "Multiobjective Gas Turbine Engine Controller Design Using Genetic Algorithms," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 5, pp. 583-587. 1996.

[35] X. Cui, M. Li and T. Fang, "Study of Population Diversity of Multiobjective Evolutionary Algorithm Based on Immune and Entropy Principles," in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1316-1321, 2001.

[36] A. G. Cunha, P. Oliviera, and J. Covas, "Use genetic algorithms in multicriteria optimization to solve industrial problems," in *Proceedings of the Seventhth International Conference on Genetic Algorithms*, pp. 682-688, 1997.

[37] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problem," *Evolutionary Computation*, vol. 7, no. 3, pp. 205-230, 1999.

[38] K. Deb, N. Udaya Bhaskara Rao and S. Karthik, "Dynamic Multi-Objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-Thermal Power Scheduling, " Kanpur Genetic Algorithms Lab. (KanGAL), Indian Institue of Technology, Kanpur, India, Technical Report 2006008, 2006.

[39] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 825-830, 2002.

[40] K. Deb and H. Gupta, "Introducing robustness in multiobjective optimization," Kanpur Genetic Algorithms Lab. (KanGAL), Indian Institue of Technology, Kanpur, India, Technical Report 2004016, 2004.

[41] K. Deb and T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 67-81, 2001.

[42] K. Deb, P. Zope and A. Jain, "Distributed computing of Pareto-optimal solutions with evolutionary algorithms," in *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, pp. 534-549, 2003.

[43] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[44] K. A. De Jong, An analysis of the behaviour of a class genetic adaptive systems, Ph.D thesis, University of Michigan, 1975.

[45] V. Devireddy and P. Reed, "Efficient and Reliable Evolutionary Multiobjective Optimization Using ϵ-Dominance Archiving and Adaptive Population Sizing" in *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, pp. 130-131, 2004.

[46] P. Di Barba, M. Farina and A. Savini, "An improved technique for enhancing diversity in Pareto evolutionary optimization of electromagnetic devices," *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 20, no. 2, pp. 482-496, 2001.

[47] M. Dror and P. Trudeau, "Stochastic vehicle routing with modified savings algorithm," *European Journal of Operational Research*, vol. 23, no. 2, pp. 228-235, 1986.

[48] M. Dror, M. O. Ball, and B. L. Golden, "Computational comparison of algorithms for inventory routing," *Annals of Operations Research*, vol. 4, pp. 3-23, 1985.

[49] D. Dubois and H. Prade, Possibility Theory: An Approach to Computerized Processing and Uncertainty, Plenum Press, New York, 1988.

[50] M. Emmerich, N. Beume, and B. Naujoks, "An EMO Algorithm Using the Hypervolume Measure as Selection Criterion," in *Proceedings of the Third Conference on Evolutionary Multi-Criterion Optimization*, pp. 62-76, 2005.

[51] R. M. Everson and J. E. Fieldsend, "Multiobjective Optimization of Safety Related Systems: An Application to Short-Term Conflict Alert," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 2, pp. 187-198, 2006.

[52] M. Farina, K. Deb and P. Amato "Dynamic Multiobjective Optimization Problems : Test Cases, Aproximations, and Applications," *IEEE Transactions on Evolutionary Computation*, vol.8, no.5, pp 425-442, 2004.

[53] M. Farina and P. Amato, "A fuzzy definition of "optimality" for many-criteria optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 3, pp. 315-326, 2003.

[54] M. Farina, "A Minimal Cost Hybrid Strategy for Pareto Optimal Front Approximation," *Evolutionary Optimization*, vol. 3, no. 1, pp. 41-52, 2001.

[55] J. E. Fieldsend, R. M. Everson and S. Singh, "Using Unconstrained Elite Archives for Multiobjective Optimization" *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 305-323, 2003.

[56] J. E. Fieldsend, and S. Singh, “Pareto evolutionary neural networks,” *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 338-354, 2005.

[57] J. E. Fieldsend and R. M. Everson, “Multi-objective Optimisation in the Presence of Uncertainty,” in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 243-250, 2005.

[58] M. Fleischer, “The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics,” in *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, vol. 2632, pp. 519-533, 2003.

[59] L. J. Fogel, A. J. Owens, M. J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley, 1966.

[60] D. B. Fogel, E. C. Wasson, and E. M. Boughton, “Evolving neural networks for detecting breast cancer,” *Cancer letters*, vol. 96, no. 1, pp. 49-53, 1995.

[61] C. M. Fonseca and P. J. Fleming, “Multi-objective genetic algorithm made easy: Selection, sharing and mating restriction,” in *International Conference on Genetic Algorithm in Engineering Systems: Innovations and Application*, pp. 12-14. 1995.

[62] C. M. Fonseca and P. J. Fleming, “Multiobjective Optimal Controller Design with Genetic Algorithms,” in *Proceedings on IEE Control*, pp. 745-749, 1994.

[63] C. M. Fonseca and P. J. Fleming, “Genetic algorithm for multiobjective optimization, formulation, discussion and generalization,” in *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416-423, 1993.

[64] E. Frank and I.H. Witten, “Generating accurate rule sets without global optimization,” in *Proceedings of the Fifteenth International Conference Machine Learning*, vol. 22, pp. 144-151, 1998.

[65] M. Gallagher and B. Yuan, “A General-Purpose Tunable Landscape Generator,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 590-603, 2006.

[66] N. Garcia-Pedrajas, C. Hervas-Martinez, and D. Ortiz-Boyer, “Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification,” *IEEE Transactions on Evolutionary Computation*, vol.9, no.3, pp 271-302, 2005.

[67] M. Gendreau, G. Laporte, and R. Sguin “A tabu search heuristic for the vehicle routing problem with stochastic demands and customers,” *Operations Research*, vol. 44, no. 3, pp. 469-477, 1996.

[68] M. Gendreau, G. Laporte, and R. Sguin “An exact algorithm for the vehicle routing problem with stochastic demands and customers,” *Transportation Science*, vol. 29, pp. 143-155, 1995.

[69] A. Ghosh, S. Tstutsui, and H. Tanaka, "Function optimization in non-stationary environment using steady state genetic algorithms with aging of individuals," in *Proceedings of 1998 IEEE Congress on Evolutionary Computation (CEC 1998)*, pp. 666-671, 1998.

[70] R. Ghosh and B. Verma, "Finding Optimal Architecture and Weights Using Evolutionary Least Square Based Learning," in *Proceedings of Neural Information Processing*, vol. 1, pp. 528-532, 2002.

[71] O. Giustolisi and V. Simeone, "Optimal design of artificial neural networks by a multi-objective strategy: groundwater level predictions," *Hydrological Sciences Journal*, vol. 51, no. 3, pp. 502-523, 2006.

[72] C. K. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 354-381, 2007.

[73] C. K. Goh and K. C. Tan, "Evolving the tradeoffs between Pareto-optimality and Robustness in Multi-Objective Evolutionary Algorithms," Evolutionary Computation in Dynamic and Uncertain Environments, (eds.) S. Yang, Y. S. Ong and Y. Jin, Springer, pp. 457-478, 2007.

[74] C. K. Goh and K. C. Tan, "A Competitive-Cooperation Coevolutionary Paradigm for Dynamic Multi-objective Optimization," IEEE Transactions on Evolutionary Computation, in press.

[75] D. E. Goldberg, The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Kluwer Academic Publishers, 2002.

[76] D. E. Goldberg, Genetic Algorithms for Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[77] D. E. Goldberg, "Sizing populations for serial and parallel genetic algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 70-79, 1989.

[78] D. E. Goldberg, and J. Richardson, "Genetic algorithms with sharing for multi-modal function optimization," in *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41-49, 1987.

[79] J. J. Grefenstette, "Genetic algorithms for changing environments, "in *Proceedings of the Second International Conference on Parallel Problem Solving from Nature*, pp. 137-144, 1992.

[80] J. J. Grefenstette and C. L. Ramsey, "An approach to anytime learning, " in *Proceedings of the Ninth International Conference on Machine Learning*, pp. 41-49, 1987.

[81] H. Gupta and K. Deb, "Handling constraints in robust multi-objective optimization," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 25-32, 2005.

[82] P. Hajela and C. Y. Lin, "Genetic Search Strategies in Multicriterion Optimal Design," *Structural Optimization*, vol. 4, pp. 99-107, 1992.

[83] N. Hallam, P. Blanchfield, and G. Kendall, "Handling Diversity in Evolutionary Multiobjective Optimisation," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 2233-2240, 2005.

[84] P.C. Hansen, Rank-deficient and discrete ill-posed problems: Numerical aspects of linear inversion, SIAM, 1998.

[85] I. Hatzakis and D. Wallace, "Dynamic Multi-Objective Optimization with Evolutionary Algorithms: A Foward-Looking Approach,"in *Proceedings of the 2006 Genetic and Evolutionary Computation Congress*, pp. 1201-1208, 2006.

[86] D. W. Hillis,"Coevolving parasites improve simulated evolution as an optimization procedure," Artificial Life 2, (eds.) C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, pp. 313-324, 1991.

[87] T. Hiroyasu, S. Nakayama and M. Miki,"Comparison Study of SPEA2+, SPEA2, and NSGA-II in Diesel Engine Emissions and Fuel Economy Problem," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 236-242, 2005.

[88] J. H. Holland, Adaptation in Natural Artificial Systems: An Introductory Analysis with Applocations to Biology, Control, and Artificial Intelligence, MIT press, 1992.

[89] J. Homberger and H. Gehring, "Two evolutionary meta-heuristics for the vehicle routing problem with time windows," *INFORMS Journal on Computing*, vol. 37, no. 3, pp. 297-318, 1999.

[90] J. Horn and N. Nafpliotis,"Multiobjective optimization using the niched Pareto genetic algorithm," Technical Report No. 930005, Illinois Genetic Algorithms Laboraatory (IlliGAL), University of Illinois at UrbanaChampaign, 1993.

[91] S.C. Huang and Y.F. Huang,"Bounds on number of hidden neurons of multilayer percep-trons in classification and recognition," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 2500-2503, 1990.

[92] E. J. Hughes, "Evolutionary Many-Objective Optimisation: Many Once or One Many?," in *Proceedings of 2005 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 222-227, 2005.

[93] E. J. Hughes, "Multiple single objective pareto sampling, " in *Proceedings of 2003 IEEE Congress on Evolutionary Computation*, pp. 26782684, 2003.

[94] E. J. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, pp. 329-343, 2001.

[95] E. J. Hughes, "Constraint handling with uncertain and noisy multi-objective evolution," in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, vol. 2, pp. 963-970, 2001.

[96] S. Huband, L. Barone, L. White and P. Hingston, "A Scalable Multi-objective Test Problem Toolkit, " in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, pp. 280-294, 2005.

[97] K. Ikeda, H. Kita, and S. Kobayashi, "Does Non-dominated Really Mean Near to Optimal? " in *Proceedings of the 2001 IEEE Conference on Evolutionary Computation*, vol. 2, pp. 957-962, 2001.

[98] H. Inoue and H. Narihisa, "Self-Organizing Neural Grove and Its Applications, " in *Proceedings of International Joint Conference on Neural Networks*, pp. 1205-1210, 2005.

[99] A. W. Iorio and X. Li, "A Cooperative Coevolutionary Multiobjective Algorithm Using Non-dominated Sorting," in *Proceedings of the 2004 Genetic and Evolutionary Computation Congress*, pp. 537-548, 2004

[100] H. Ishibuchi and Y. Shibata, "An Empirical Study on the Effect of Mating Restriction on the Search Ability of EMO Algorithms," in *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, pp. 433-447, 2003.

[101] H. Ishibuchi and Y. Shibata, "A Similarity-Based Mating Scheme for Evolutionary Multiobjective Optimization," in *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, pp. 1065-1076, 2003.

[102] H. Ishibuchi and T. Murata, "A Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling," *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, vol. 28, no. 3, pp. 392-403, 1998.

[103] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop,"*IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204-223, 2003

[104] A. Jaszkiewicz, "On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem-A Comparative Experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402-412, 2002.

[105] A. Jaszkiewicz, "Do multi-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 133-143, 2003.

[106] L. Jiao, M. Gong, R. Shang, H. Du and B. Lu, "Clonal Selection with Immune Dominance and Anergy Based Multiobjective Optimization," in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, pp. 474-489, 2005.

[107] Y. Jin and J. Branke,"Evolutionary Optimization in Uncertain EnvironmentsA Survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303-317, 2005.

[108] Y. Jin and B. Sendhoff, "Constructing Dynamic Optimization Test Problems Using the Multi-objective Optimization Concept," in *Proceedings of the 2004 EvoWorkshops*, pp. 525-536, 2004.

[109] Y. Jin and B. Sendhoff, "Tradeoff between performance and robustness: An evolutionary multiobjective approach," in *Proceedings of the Second Conference on Evolutionary Multi-Criterion Optimization*, pp. 237-251, 2003.

[110] Y. Jin, T. Okabe and B. Sendhoff, "Adapting Weighted Aggregation for Multiobjective Evolution Strategies," in *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, pp. 96-110, 2001.

[111] Y. Jin, M. Olhofer and B. Sendhoff, "Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How?," in *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, pp. 1042-1049, 2001.

[112] G. H. John and P.Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338-345, 1995.

[113] S. A. Karzrlis, S. E. Papadakis, J. B. Theocharis, and V. Petridis, , "Microgenetic Algorithms as Generalized Hill-Climbing Operators for GA Optimization," *IEEE Transactions On Evolutionary Computation*, vol. 5, no. 3, pp. 204-217, 2001.

[114] N. Keerativuttiumrong, N. Chaiyaratana and V. Varavithya, "Multiobjective cooperative coevolutionary genetic algorithm," in *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature*, pp. 288-297, 2002.

[115] V. Khare, X. Yao and B. Sendhoff, “Credit assignment among neurons in co-evolving populations,” in *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature*, pp. 882-891, 2004.

[116] V. Khare, X. Yao and K. Deb, “Performance scaling of multi-objective evolutionary algorithms,” in *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, pp. 376-390, 2003.

[117] E. F. Khor, K. C. Tan, T. H. Lee, and C. K. Goh, “A study on distribution preservation mechanism in evolutionary multi-objective optimization,” *Artificial Intelligence Review*, vol. 23, no. 1, pp. 31-56, 2005.

[118] KE. F. Khor, K. C. Tan, and T. H. Lee, “Tabu-based exploratory evolutionary algorithm for effective multi-objective optimization,” in *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, pp. 344-358, 2001.

[119] W. Kinnebrock, “Accelerating the standard backpropagation method using a genetic approach,” *Neurocomputing*, vol. 6, no. 5-6, pp. 583-588, 1994.

[120] H. Kita, Y. Yabumoto, N. Mori, and Y. Nishikawa, “Multi- Objective Optimization by Means of the Thermodynamical Genetic Algorithm,” in *Proceedings of the Fourth Parallel Problem Solving from Nature*, pp. 504-512, 1996.

[121] V. Klema and A. Laub, “The Singular Value Decomposition: Its Computation and Some Applications,” *IEEE Transanction on Automatic Control*, vol. 2, pp. 164-176, 1980.

[122] J. D. Knowles, D. W. Corne and M. Fleischer, “Bounded archiving using the Lebesgue measure,” in *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, vol. 4, pp. 2490-249, 2003.

[123] J. D. Knowles, and D. W. Corne, “Properties of an adaptive archiving algorithm for storing nondominated vectors,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100-116, 2003.

[124] J. D. Knowles, and D. W. Corne, “On Metrics for Comparing Nondominated Sets,” in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 711-716, 2002.

[125] K. Konstantinides and K. Yao, “Statistical analysis of effective singular values in matrix rank determination,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 5, pp. 757-763, 1988.

[126] F. Kursawe, “A Variant of Evolution Strategies for Vector Optimization,” in *Proceedings of the Firsth International Conference on Parallel Problem Solving from Nature*, vol. 496, pp. 193-197, 1991.

[127] J. D. Knowles, and D. W. Corne, “Approximating the non-dominated front using the Pareto archived evolution strategy,”*Evolutionary Computation*, vol. 8, no. 2, pp. 149-172, 2000.

[128] V. Lambert, G. Laporte, and F. V. Louveaux, “Designing Collection Routes through Bank Branches,”*Computers and Operations Research*, vol. 20, pp. 783-791, 1993.

[129] G. Laporte and F. V. Louveaux “Solving stochastic routing Problems with the Integer L-shaped Method,” Fleet Management and Logistics, (eds.)T. G. Crainic and G. Laporte, Kluwer Academic Publishers, Boston, pp. 159-167, 1998.

[130] G. Laporte and F. V. Louveaux “The integer L-shape method for stochastic integer problems with complete recourse,” *Operations Research Letters*, vol. 13, pp. 133-142, 1993.

[131] R. C. Larson, “Transportation of sludge to the 106-mile site: An inventory routing algorithm for fleet sizing and logistic system design,” *Transportation Science*, vol. 22, pp. 186-198, 1988.

[132] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb “Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 439-447, 2002.

[133] M. Laumanns, E. Zitzler, and L. Thiele, “On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization,” in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 181-196, 2001.

[134] M. Laumanns, E. Zitzler, and L. Thiele, “A unified model for multi-objective evolutionary algorithms with elitism,” in *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 46-53, 2000.

[135] Y. LeCun, L. Bottou, G. Orr, and K. Muller, “Efficient BackProp,”Neural Networks: Tricks of the trade, Springer-Verlag, 1998.

[136] D. Lim, Y.-S. Ong, M.-H. Lim, and Y. Jin, “Single/Multi-objective Inverse Robust Evolutionary Design Methodology in the Presence of Uncertainty,” Evolutionary Computation in Dynamic and Uncertain Environments, (eds.) S. Yang, Y. S. Ong and Y. Jin, Springer, in press.

[137] Y. Liu, X. Yao, and T. Higuchi, "Evolutionary Ensembles with Negative Correlation Learning," *IEEE Transactions On Evolutionary Computation*, vol. 4, no. 4, pp. 380-387, 2000.

[138] T. H. Liu and K. J. Mills, "Robotic Trajectory Control System Design for Multiple Simultaneous Specifications: Theory and Experimentation," in *Transactions on ASME*, vol. 120, pp. 520-523. 1998.

[139] J. D. Lohn, W. F. Kraus and G. L. Haith, "Comparing a coevolutionary genetic algorithm for multiobjective optimization," in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1157-1162, 2002.

[140] H. Lu and G. G. Yen, "Rank-based multiobjective genetic algorithm and benchmark test function study," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 325-343, 2003.

[141] G. C. Luh, C. H. Chueh, and W. W. Liu, "MOIA: Multi-Objective Immune Algorithm," *Engineering Optimization*, vol. 35, no. 2, pp. 143-164, 2003.

[142] K. Maneeratana, K. Boonlong and N. Chaiyaratana, "Multi-objective Optimisation by Co-operative Co-evolution," in *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature*, pp. 772-781, 2004.

[143] V. Maniezzo, "Genetic evolution of the topology and weight distribution of neural networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 39-53, 1994.

[144] J. Mehnen, T. Wagner, and G. Rudolph, "Evolutionary Optimization of Dynamic Multi-objective Test Functions," in *Proceedings of the Second Italian Workshop on Evolutionary Computation*, 2006.

[145] P. Merz and B. Freisleben, "A comparison of memetic algorithms, Tabu search, and ant colonies for the quadratic assignment problem," in *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 2063-2070, 1999.

[146] D. Michie, D.J. Spiegelhalter and C.C. Taylor, Machine Learning, Neural and Statistical Classification, London: Ellis Horwood, 1994.

[147] N. Mori, H. Kita, and Y. Nishikawa, "Adaptation to a changing environment by means of the thermodynamical genetic algorithm," in *Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature*, pp. 513-522, 1996.

[148] R. Morrison, Designing Evolutionary Algorithms for Dynamic Environments. Berlin, Germany: Springer-Verlag, 2004.

[149] C. L. Mumford, “A Hierarchical Solve-and-Merge Framework for Multi-Objective Optimization,” in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 2241-2247, 2005.

[150] T. Murata and H. Ishibuchi, “MOGA: Multi-objective genetic algorithms,” in *Proceedings of the 1995 IEEE Congress on Evolutionary Computation*, pp. 289-294, 1995.

[151] V. Nissen and J. Propach, “On the robustness of population based versus point-based optimization in the presence of noise,” *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 3, pp. 107-119, 1998.

[152] T. Okabe, Y. Jin, B. Sendhoff, and M. Olhofer, “Voronoi-based estimation of distribution algorithm for multi-objective optimization,” in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pp. 1594-1601, 2004.

[153] T. Okuda, T. Hiroyasu, M. Miki, S. Watanabe, “DCMOGA: Distributed Cooperation model of Multi-Objective Genetic Algorithm” in *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature*, pp. 155-160, 2002.

[154] Y. S, Ong and A. J. Keane, “Meta-Lamarckian Learning in Memetic Algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99-110, 2004.

[155] Y. S. Ong, P. B. Nair , K. Y. Lum, “Min-Max Surrogate Assisted Evolutionary Algorithm for Robust Aerodynamic Design,”*IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 392-404, 2006.

[156] A. Osyczka and S. Krenich, “Evolutionary Algorithms for Multicriteria Optimization with Selecting a Representative Subset of Pareto Optimal Solutions,” in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 141-153, 2001.

[157] I. Paenke, J. Branke and Y. Jin, “Efficient Search for Robust Solutions by Means of Evolutionary Algorithms and Fitness Approximation, ” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 405-420, 2006.

[158] P.P. Palmes, T. Hayasaka, and S. Usui, “Mutation-Based Genetic Neural Network, ” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 587-600, 2005.

[159] G. Parks, J. Li, M. Balazs and I. Miller, “An empirical investigation of elitism in multiobjective genetic algorithms,” *Foundations of Computing and Decision Sciences*, vol. 26, no. 1, pp. 51-74, 2001.

[160] E. Parzen, “On the estimation of a probability density function and mode,” *Annals of Mathematical Statistics*, vol. 33, pp. 1065-1076, 1962.

[161] M. A. Potter, “The Design and Analysis of a Computational Model of Cooperative Coevolution,” Ph.D Thesis, George Mason University, 1997.

[162] M. A. Potter and K. A. De Jong, “Cooperative coevolution: An architecture for evolving coadapted subcomponents,” *Evolutionary Computation*, vol. 8, no. 1, pp. 1-29, 2000.

[163] J. Paredis, “Coevolutionary constraint satisfaction,” in in *Proceedings of the Third International Conference on Parallel Problem Solving from Nature*, pp. 46-55, 1994.

[164] J. R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann, 1993.

[165] S. Rana, D. Whitney, and R. Cogswell, “Searching in the presence of noise,” in *Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature*, pp. 198-207, Springer, 1996.

[166] M. Rattray and J. Shapiro, “Noisy fitness evaluations in genetic algorithms and the dynamics of learning,” *Foundations of Genetic Algorithms 4*, (eds.) R. K. Belew and M. D. Vose, pp. 117-139. Morgan Kaufmann, 1997.

[167] T. Ray, “Constrained robust optimal design using a multiobjective evolutionary algorithm,” in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 419-424, 2002.

[168] I. Rechenberg, Evolutionsstrategie, Frommann-Holzboog, 1994.

[169] C. R. Reeves, Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publication, 1993.

[170] C. D. Rosin and R. K. Belew, “New methods for competitive coevolution,” *Evolutionary Computation*, vol. 5, no. 1, pp. 1-29, 1997.

[171] J. Rowe, K. Vinsen and N. Marvin, “Parallel GAs for Multiobjective Functions,” in *Second Nordic Workshop on Genetic Algorithms and Their Applications*, pp. 61-70, 1996.

[172] G. Rudolph, “A partial order approach to noisy fitness functions,” in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 318-325, 2001.

[173] G. Rudolph and A. Agapie, “Convergence Properties of Some Multi-Objective Evolutionary Algorithms,”in *Proceedings of the 2000 Conference on Evolutionary Computation*, pp. 1010-1016, 2000.

[174] G. Rudolph, “On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set,” in *Proceedings of the 1998 Conference on Evolutionary Computation*, pp. 511-516, 1998.

[175] Y. Sano and H. Kita, “Optimization of noisy fitness functions by means of genetic algorithms using history of search with test of estimation,” in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 360-365, 2002.

[176] R. Sarker, K. Liang, and C. Newton, “A New Evolutionary Algorithm for Multiobjective Optimization,” *European Journal of Operational Research*, vol. 140, no. 1, pp. 12-23, 2002.

[177] M. A. Sartori and P. J. Antsaklis, “A simple method to derive bounds on the size and to train multi-layer neural networks,” *IEEE Transactions on Neural Networks*, vol. 2, no. 4, pp. 467-471, 1991.

[178] H. Sato, H. E. Aguirre and K. Tanaka, “Enhanced Multi-objective Evolutionary Algorithms Using Local Dominance,” in *Proceedings of the 2004 RISP International Workshop on Nonlinear Circuits and Signal Processing*, pp. 319-322, 2004.

[179] J. D. Schaffer, “Multi-Objective Optimization with Vector Evaluated Genetic Algorithms,” in *Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100, 1985.

[180] W. M. Schaffer, D. W. Zeh, S. L. Buchmann, S. Kleinhaus, M. V. Schaffer, and J. Antrim, “Competition for nectar between introduced honeybees and native North American bees and ants,” *Ecology*, vol. 64, pp. 564-577, 1983.

[181] J. R. Scott, “Fault Tolerant Design Using Single and Multi-criteria Genetic Algorithms,”Masters Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[182] , B. Sendhoff, H-G. Beyer and M. Olhofer, “On Noise Induced Multi-modality in evolutionary algorithms,”Recent Advances in Simulated Evolution and Learning, (eds.) L. Wang, K. Tan, T. Furuhashi, K.-H. Kim and F. Sattar, pp. 219-224, 2002.

[183] D. Serre, Matrices: Theory and Applications, Springer-Verlag, New York, 2002

[184] C. E. Shannon, “A mathematical theory of communications,” *Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.

[185] K. B. Sim, J. Y. Kim and D. W. Lee, “Game Model Based Co-evolutionary Solution for Multiobjective Optimization Problems,” *International Journal of Contol, Automation, and Systems*, vol. 2, no. 2, pp. 247-255, 2004.

[186] A. Singh, “Uncertainty based Multi-objective Optimization of Groundwater Remediation Design,” Master’s Thesis, University of Illinois at Urbana-Champaign, 2003.

[187] M. M. Solomon, “Algorithms for the vehicle routing and scheduling problems with time window constraints,” *Operations Research*, vol. 35, no. 2, pp. 254-265, 1987.

[188] N. Srinivas, and K. Deb, “Multiobjective optimization using non-dominated sorting in genetic algorithms,” *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1994.

[189] K. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary Computation*, vol. 10, no. 2, pp. 99-127, 2002.

[190] G. Stewart, “Determining Rank in the Presence of Error,” Technical Report (TR-92-108) Institute for Advanced Computer Studies, (TR-2972) Department of Computer Science, University of Maryland, College Park, Oct 1992.

[191] P. D. Stroud, “Kalman-extended genetic algorithms for search in nonstationary environments with noisy fitness evaluations,” *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 66-77, 2001.

[192] K. C. Tan, C. K. Goh, A. A. Mamun and E. E. Zin, “An Evolutionary Artificial Immune System for Multi-Objective Optimization,” *European Journal of Operational Research*, in press.

[193] K. C. Tan, C. Y. Cheong and C. K. Goh, “Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation” *European Journal of Operational Research*, vol. 177, pp. 813-839, 2007.

[194] K. C. Tan, Q. Yu and J. H. Ang, “A coevolutionary algorithm for rules discovery in data mining,” *International Journal of Systems Science*, vol. 37, no. 12, pp. 835-864, 2006.

[195] K. C. Tan, Y. J. Yang, and C. K. Goh, “A distributed cooperative coevolutionary algorithm for multiobjective optimization,”*IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 527-549, 2006.

[196] K. C. Tan, C. K. Goh, Y. J. Yang, and T. H. Lee, “Evolving better population distribution and exploration in evolutionary multi-objective optimization,” *European Journal of Operational Research*, vol. 171, no. 2, pp. 463-495, 2006.

[197] K. C. Tan, E. F. Khor, T. H. Lee and R. Sathikannan, “An evolutionary algorithm with advanced goal and priority specification for multiobjective optimization,” *Journal of Artificial Intelligence Research*, vol. 18, pp. 183-215, 2003.

[198] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons," *Artificial Intelligence Review*, vol. 17, no. 4, pp. 251-290, 2002.

[199] K. C. Tan, T. H. Lee and E. F. Khor, "Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 6, pp. 565-588, 2001.

[200] D. Teodorovic and P. Lucic, "Intelligent vehicle routing system," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pp. 482487, 2000.

[201] D. Teodorovic and G. Pavkovic, "The fuzzy set theory approach to the vehicle routing problem when demand at nodes is uncertain," *Fuzzy Sets and Systems*, vol. 82, no. 3, pp. 307-317, 1996.

[202] E. J. Teoh, K. C. Tan and C. Xiang, "Estimating the number of hidden neurons in a feedforward network using the Singular Value Decomposition," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1623-1629, 2006.

[203] J. Teich, "Pareto-front exploration with uncertain objectives," in *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, Springer-Verlag, pp. 314-328, 2001.

[204] A. Thompson, "Evolutionary techniques for fault tolerance,"in *Proceedings of the UKACC International Conference on Control*, pp. 693-698, 1996.

[205] H. A. Thompson and P. J. Fleming,"An Integrated Multi-Disciplinary Optimisation Environment for Distributed Aero-engine Control System Arhitectures," in *Proceedings of the Fourteenth World Congress of International Federation of Automatic Control*, pp. 407-412. 1999.

[206] A. Toffolo and E. Benini,"Genetic Diversity as an Objective in Multi-Objective Evolutionary Algorithms," *Evolutionary Computation*, vol. 11, no. 2, pp. 151-167, 2003.

[207] S. Tsutsui and A. Ghosh, "A comparative study on the effects of adding perturbations to phenotypic parameters in genetic algorithms with a robust solution searching scheme," in *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 585-591, 1999.

[208] S. Tsutsui and A. Ghosh, "Genetic algorithms with a robust solution searching scheme," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 201-208, 1997.

[209] A. Turkcan and M. S. Akturk, “A problem space genetic algorithm in multiobjective optimization,” *Journal of Intelligent Manufacturing*, vol. 14, pp. 363-378, 2003.

[210] R. K. Ursem, “Mutinational GA optimization techniques in dynamic environments, ” in *Proceedings of the 2000 Genetic and Evolutionary Computation Congress*, pp. 19-26, 2000.

[211] F. Van den Bergh and A. P. Engelbrecht, “A cooperative approach to particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225-239, 2004.

[212] G. Venter and R. T. Haftka, “A Two Species Genetic Algorithm for Designing Composite Laminates Subjected to Uncertainty,” in *Proceedings of the37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pp. 1848-1857, 1996.

[213] F. Vavak, K. Jukes, and T. C. Fogarty, “Adaptive combustion balancing in multiple burner boiler using a genetic algorithm with variable range of local search, ”in *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 719-726, 1997.

[214] D. A. Van Veldhuizen, J. B. Zydallis and G. B. Lamont, “Considerations in engineering parallel multiobjective evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 144-173, 2003.

[215] D. A. Van. Veldhuizen and G. B. Lamont, “On measuring multiobjective evolutionary algorithm performance,” in *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 204-211, 2000.

[216] D. A. Van Veldhuizen and G. B. Lamont, “Multiobjective Evolutionary Algorithm Research: A History and Analysis,” Technical Report TR-98-03, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Ohio, 1998.

[217] B. Verma and R. Ghosh, “A novel evolutionary Neural Learning Algorithm,” in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 2 , pp. 1884-1889, 2002.

[218] M. Wineberg and F. Oppacher, “Enhancing the GAs ability to cope with dynamic environments,” in *Proceedings of the 2000 Genetic and Evolutionary Computation Congress*, pp. 3-10, 2000.

[219] X. Yao, “Evolving Artificial Neural Networks, ” *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423-1447, 1999.

[220] X. Yao and Y. Liu, “A new evolutionary system for evolving artificial neural networks, ” *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 694-713, 1997.

[221] X. Yao and Y. Liu, “Making use of population information in evolutionary artificial neural networks, ”*IEEE Transaction on Systems, Man, and Cybernetics- Part B: Cybernetics*, vol. 28, pp. 417-425, 1998.

[222] S. Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, and L. Kang, “A Dynamic Multi-objective Evolutionary Algorithm Based on an Orthogonal Design, ” in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pp. 573-580, 2006.

[223] X.-H. Zhang, H.-Y. Meng and L.-C. Jiao, “Intelligent Particle Swarm Optimization in Multiobjective Optimization,” in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 714-719, 2005.

[224] Q. Zhao and T. Higuchi, “Evolutionary learning of nearest-neighbor MLP,” *IEEE Transactions on Neural Networks*, vol. 7, pp. 762-767, 1996.

[225] Q. Zhao, “Stable on-line evolutionary learning of NN-MLP” *IEEE Transactions on Neural Networks*, vol. 8, pp. 1371-1378, 1997.

[226] E. Zitzler and S. Kunzli, “Indicator-Based Selection in Multiobjective Search,” in *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature*, pp. 832-842, 2004.

[227] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,”*IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117-132, 2003.

[228] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, 2001.

[229] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.

[230] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

[231] Zitzler, Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Ph.D Thesis, Swiss Federal Institute of Technology, Zurich, 1999.