Development of algorithms to solve different key challenges facing design optimization

Md Asafuddoula

A thesis submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

School of Engineering and Information Technology
University of New South Wales, Canberra, Australia

26 February 2014

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Asafuddoula

First name: Md   Other name/s:

Abbreviation for degree as given in the University calendar: PhD

School: School of Engineering and Information Technology   Faculty: University College

Title: Development of algorithms to solve different key challenges facing design optimization

Abstract (350 words maximum)

Optimization methods play an indispensable role in today's competitive environment and there are plenty of practical examples where such methods have been used to identify better performing designs (Boeing 787 Dreamliner and NASA ST5 antenna). Increasing complexity of the problems have also led to the development of sophisticated mathematical models that can only be solved using computationally expensive numerical simulations such as finite element methods (FEM), computational fluids dynamics (CFD), computational electromagnetics (CEM), etc. Repeated use of such numerical simulations is necessary in the context of optimization, i.e., to identify optimum products and processes with outstanding performance features. In reality, such problems often involve a large number of constraints and often demand multiple performance considerations.

Over the decades, population based metaheuristics have proven to be efficient, robust and versatile methods for numerical optimization as they are more amenable to deal with such black-box problems. The major downside of any of these population based metaheuristics is their extremely long run time. Therefore, it is no surprise that the development of fast and efficient metaheuristics is an actively pursued research area. In this thesis, an effort is made to address three key challenges facing the adoption of population based metaheuristics for practical design optimization. The first challenge relates to the development of an efficient and reliable optimization algorithm capable of dealing with constrained optimization problems. In particular, two novel constraint handling mechanisms are introduced i.e., one with the concept of partial evaluation using constraint sequencing and the other involving adaptive constraint handling. The second contribution reported in this thesis relates to the development of an algorithm to tackle optimization problems involving more than four objectives, i.e., many objective optimizations. The third contribution made in this thesis is in the area of robust design optimization where the effects of various formulations are studied in the framework of six-sigma quality. Four different problem formulations of robust design and methods to solve them have been proposed.

The performance of these algorithms/schemes is rigorously assessed using well established benchmark functions and a suite of engineering design optimization problems. The results assessed using various measures clearly indicate that the proposed develop ments offer competitive advantages over existing schemes.

Finally, a summary of the findings of the work is presented. In addition, future issues and directions which could be pursued with the aim of making the algorithms more efficient for handling various types of optimization problems are identified.

FOR OFFICE USE ONLY   Date of completion of requirements for Award:

COPYRIGHT STATEMENT

'I hereby grant the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.
I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstract International (this is applicable to doctoral theses only).
I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.'

Signed ....

Date 27/05/2014

AUTHENTICITY STATEMENT

'I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.'

Signed ....

Date 27/05/2014

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed ....

Date 27/05/2014 ....

Abstract

Optimization methods play an indispensable role in today's competitive environment and there are plenty of practical examples where such methods have been used to identify better performing designs (Boeing 787 Dreamliner and NASA ST5 antenna). Increasing complexity of the problems have also led to the development of sophisticated mathematical models that can only be solved using computationally expensive numerical simulations such as finite element methods (FEM), computational fluids dynamics (CFD), computational electromagnetics (CEM), etc. Repeated use of such numerical simulations is necessary in the context of optimization, i.e., to identify optimum products and processes with outstanding performance features. In reality, such problems often involve a large number of constraints and often demand multiple performance considerations.

Over the decades, population based metaheuristics have proven to be efficient, robust and versatile methods for numerical optimization as they are more amenable to deal with such black-box problems. The major downside of any of these population based metaheuristics is their extremely long run time. Therefore, it is no surprise that the development of fast and efficient metaheuristics is an actively pursued research area.

In this thesis, an effort is made to address three key challenges facing the adoption of population based metaheuristics for practical design optimization. The first challenge relates to the development of an efficient and reliable optimization algorithm capable of dealing with constrained optimization problems. In particular, two novel constraint handling mechanisms are introduced i.e., one with the concept of partial evaluation using constraint sequencing and the other involving adaptive constraint handling. The study is motivated by the fundamental question *should one evaluate all constraints of a solution even if it has violated one constraint?* and *what is the difference in the underlying search process if multiple constraint sequences are used?*. The second contribution reported in this thesis relates to the development of an algorithm to tackle optimization problems involving more than four objectives, i.e., many objective optimizations. In this context, an algorithm based on decomposition is introduced which extends the capability of the well-known MOEA/D to deal with many objective optimization problems. The algorithm incorporates a systematic sampling scheme and

the balance between convergence and diversity during the course of search is maintained via a simple preemptive distance comparison scheme. The third contribution made in this thesis is in the area of robust design optimization where the effects of various formulations are studied in the framework of six-sigma quality. Four different problem formulations of robust design and methods to solve them have been proposed.

The performance of these algorithms/schemes is rigorously assessed using well established benchmark functions and a suite of engineering design optimization problems. The results assessed using various measures clearly indicate that the proposed developments offer competitive advantages over existing schemes.

Finally, a summary of the findings of the work is presented. In addition, future issues and directions which could be pursued with the aim of making the algorithms more efficient for handling various types of optimization problems are identified.

Acknowledgements

In the name of Allah (the one and only one creator of everything), the most Merciful, the most Gracious. All praise is due to Allah; we praise Him, seek His help, and ask for His forgiveness. I am thankful to Allah, who supplied me with the courage, the guidance, and the love to complete this research. Also, I cannot forget the ideal man of the world and most respectable personality for whom Allah created the whole universe, Prophet Mohammad (Peace Be Upon Him).

I express my profound sense of reverence to my supervisor A/Prof. Tapabrata Ray who gave me the opportunity to work in his MDO (Multidisciplinary Design Optimization) group. His continuous support, motivation and untiring guidance have made this dream come true. His vast knowledge, calm nature and positive criticism motivated me in all the time of research and writing of this thesis. Thanks to him for bearing my mistakes and whenever I couldnt meet the deadlines. I am also grateful to my co-supervisor A/Prof. Ruhul Sarker for his cooperation with the valuable suggestions at time of difficulties.

I would also like to thank my wife Rafia Afrin for her mental support and great patience at all times. My parents, and sister have given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice.

I thank my fellow lab mates whos contact in MDO are remembered always by name: Dr. Hemant Kumar Singh, Dr. Ahmad Faisal Mohamad Ayob, Dr. Khairul Alam, Mohammad Sharif Khan for the stimulating discussions, for the helpful suggestions to improve myself, and for all the fun we have had in the last three and half years.

List of Publications

Journal Papers

[1] M. Asafuddoula, T. Ray, and R. Sarker. A decomposition based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*. Under review.

[2] M. Asafuddoula, H. K. Singh, and T. Ray. Six-sigma robust design optimization using a many-objective decomposition based evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*. In Press, (Accepted 11/04/2014).

[3] M. Asafuddoula, T. Ray, and R. Sarker. A computationally efficient constraint handling method for inequality constrained optimization problems. *Applied Soft Computing*. Under review.

[4] M. Asafuddoula, T. Ray, and R. Sarker. An adaptive hybrid differential evolution algorithm for single objective optimization. *Applied Mathematics and Computation*. In Press, (Accepted 06/01/2014).

Lecture Notes

[1] M. Asafuddoula, T. Ray, and R. Sarker. A decomposition based evolutionary algorithm for many objective optimization with systematic sampling and adaptive epsilon control. In *Seventh International Conference on Evolutionary Multi-Criterion Optimization*, volume 7811, pages 417–427. Springer, Lecture Notes in Computer Science, 2013.

[2] M. Asafuddoula, T. Ray, and R. Sarker. A self-adaptive differential evolution algorithm with constraint sequencing. In *AI 2012: Advances in Artificial Intelligence*, volume 7691, pages 182–193. Springer, Lecture Notes in Artificial Intelligence, 2012.

Conference Papers

[1] H. K. Singh, M. Asafuddoula, and T. Ray. Solving problems with a mix of hard and soft constraints using modified infeasibility driven evolutionary algorithm (IDEA-M). In *2014 IEEE Congress on Evolutionary Computation*. Accepted.

[2] T. Ray, M. Asafuddoula, and A. Isaacs. A steady state decomposition based quantum genetic algorithm for many objective optimization. In *IEEE Congress on Evolutionary Computation*, pages 2817–2824, Cancun, Mexico, 2013.

[3] M. Asafuddoula, T. Ray, and R. Sarker. An efficient constraint handling approach for optimization problems with limited feasibility and computationally expensive constraint evaluations. In *Genetic and Evolutionary Computation Conference Companion (GECCO 2013)*, pages 113–114, Amsterdam, The Netherlands, 2013. ACM.

[4] M. Asafuddoula, T. Ray, and R. Sarker. Evaluate till you violate: A differential evolution algorithm based on partial evaluation of the constraint set. In *IEEE Symposium Series on Computational Intelligence*, pages 31–37, Singapore, Singapore, 2013.

[5] M. Asafuddoula, T. Ray, and R. Sarker. A differential evolution algorithm with constraint sequencing. In *Third Global Congress on Intelligent Systems (GCIS)*, pages 68–71, Wuhan, China, 2012.

[6] M. Asafuddoula, T. Ray, R. Sarker, and K. Alam. An adaptive constraint handling approach embedded MOEA/D. In *IEEE Congress on Evolutionary Computation*, pages 2516–2513, Brisbane, Australia, 2012.

[7] M. Asafuddoula, T. Ray, and R. Sarker. An adaptive differential evolution algorithm and its performance on real world optimization problems. In *IEEE Congress on Evolutionary Computation*, pages 1057–1062, New Orleans,USA, 2011.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Overview of the Dissertation

1.1 Introduction

Many real-life problems can be modeled as an optimization problem. Optimization generally means to seek for an optimum design/solution (or a set of designs/solutions). Metaphorically, the optimum design refers to a design which is either "maximized" or "minimized" in terms of its performance (or objective) (i.e., a function of various design variables). In an optimization problem, the aim is to discover the best combination of values of such design variables which contributes towards the optimum. Some typical examples from the domain of engineering design include minimization of the cost of a welded beam, minimization of the weight of a tension/compression spring, minimization of the total weight of the speed reducer etc. It is important to highlight that most of these optimization problems involve a large number of (inequality and equality constraints) while some involve more than four objectives. These problems still pose significant challenges to any optimizer and there has been an active interest to develop efficient algorithms to solve such problems.

There are many methods to deal with these optimization problems. In particular, the work reported in this dissertation aims to introduce novel methods of constraint handling/constraint sequencing, efficient optimization algorithms based on differential evolution, novel methods to deal with many objective optimization problems and finally an approach to identify robust solutions.

The following sections provide the background of various optimization problems, solution schemes and means of performance assessment. The gaps are listed clearly to lay the foundations of this research. The contributions are presented in the following chapters.

1.2 Optimization problem

An optimization problem can mathematically be formulated as follows.

$$\begin{aligned} &\underset{(\mathbf{x})}{\text{Minimize}} \quad f_i(\mathbf{x}), i = 1, 2, ...M \\ &\text{Subject to} \\ &\qquad g_j(\mathbf{x}) \geq 0, j = 1, 2, ...p \\ &\qquad h_k(\mathbf{x}) = 0, k = 1, 2, ...q \end{aligned} \tag{1.1}$$

$$\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)} \tag{1.2}$$

where f_i is a objective function and M is the number of objectives. The number of inequality and equality constraints are denoted by p and q respectively. Depending upon the number of objectives the optimization problem is divided into three categories i.e., single objective problem ($M = 1$), multi-objective problem ($M \geq 2$), many-objective problem ($M \geq 4$). These problems are briefly discussed below. The notion of minimization is used throughout this dissertation.

1.2.1 Single-objective problem

Single-objective optimization refers to problems with only one objective. The aim is to find all solutions $\mathbf{x} \subset S$ such that $f(\mathbf{x})$ assumes the minimum value f^*. The solution x to such a problem can be either a unique solution, or there may be multiple values of

$\mathbf{x}$ for which the objective value is f^*.

1.2.2 Multi-/Many-objective problem

Multi-objective optimization refers to problems with $(M \geq 2)$. In the event $(M \geq 4)$, the optimization problem is referred as a many-objective optimization problem. In the absence of any preference information among the objectives, the aim is to identify the set of non-dominated solutions.

Definition 1. *For any two solutions $x_1, x_2 \in \mathbb{R}$, x_1 is said to be non-dominated with x_2, if and only if $f_i(x_1) \leq f_i(x_2)$ $\forall i = 1....M$ with strict inequality for at least one i.*

Definition 2. *For $x^* \in \Omega$, $\Omega = \{x \in \mathbb{R}\}$ is said to be a Pareto optimal solution if there exists no other feasible solution $x \in \Omega$ such that $f_i(x) \leq f_i(x^*)$, $\forall i = 1....M$ with strict inequality for at least one i.*

In Figure 1.1, four solutions, A, B, C and D are shown in the objective space. Among these solutions, C and D dominate B because they are better in both objectives, i.e., f_1,C $<$ f_1,B and f_2,C $<$ f_2,B; same relation can be drawn for solutions D and B. On the other hand, A and C are non-dominated with respect to each other, since A is better in f_2, whereas C is better in f_1, i.e., f_1,A $>$ f_1,C and f_2,A $<$ f_2,C. For the same reason, pairs A and C, C and D and D and A are non-dominated with respect to each other.

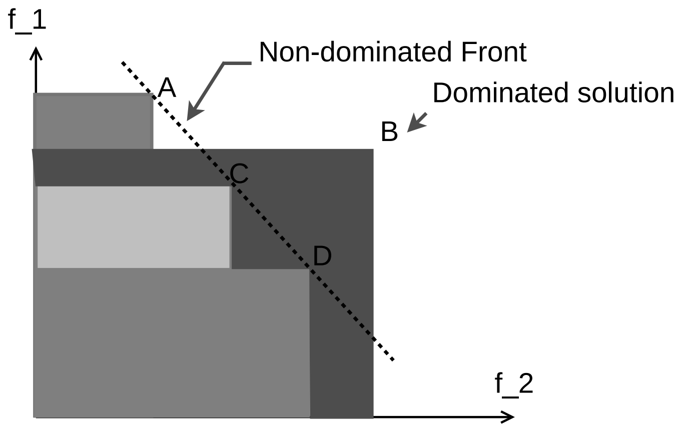


Figure 1.1: Dominance relationships for multi-objective optimization (C, D dominate B; A, C and D form a non-dominated set)

Among these four solutions, A, C and D form a non-dominated set since they are not dominated by any solution.

1.3 Optimization methods

While a plethora of optimization methods have been developed over the years, population based stochastic optimization methods are commonly used to solve complex problems involving multiple objectives and constraints. Such methods are typically attractive as they can deal with multimodal functions and do not require assumptions on functional continuity. Furthermore, such methods are also capable of delivering the nondominated set in a single run. A simple classification is presented in Figure 1.2 with the salient scope of population based methods.

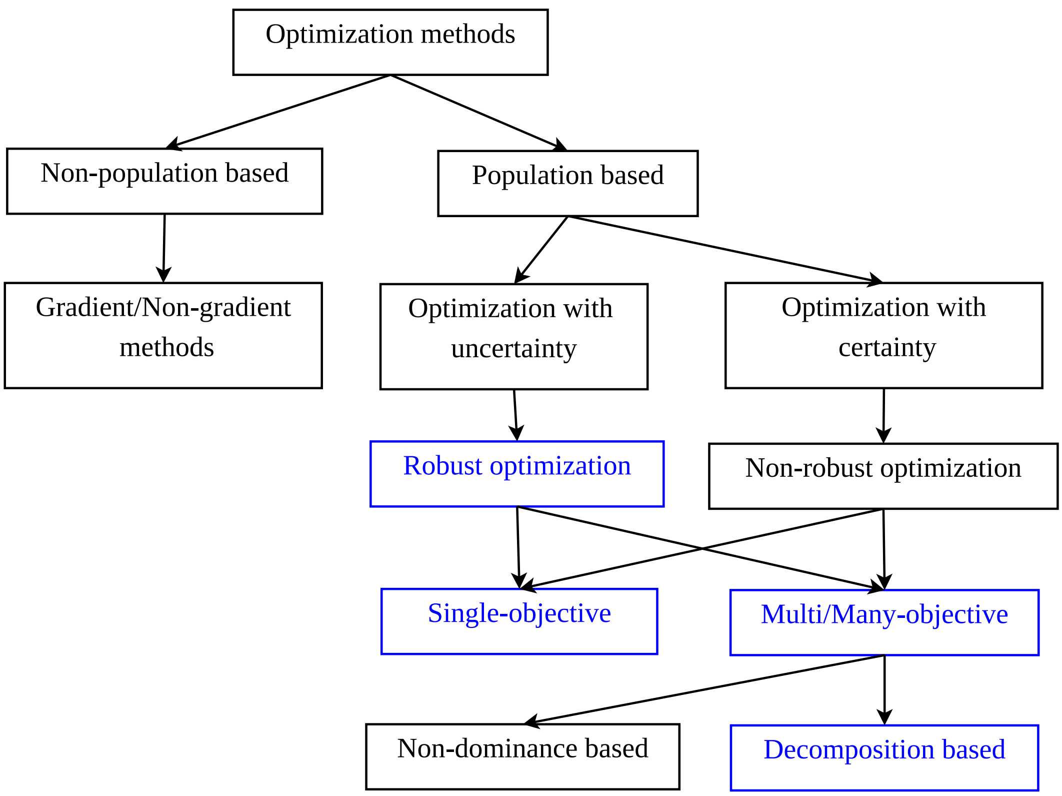


Figure 1.2: A simple classification of optimization methods

The basic concept of a population based method is to improve a set of solutions through the process of evolution [2]. Clearly, their advantages can be summarized as follows:

- They can be used to deal with combinatorial, continuous, or mixed problems with no or minimal customization. Since they do not require the objective function to be continuous or smooth or to possess any specific properties such as linearity, they can be applied to any *black-box* optimization problem.
- They are capable of identifying the global optimum as they do not rely on a single solution or gradients.
- Since such algorithms operate with a population of solutions, they can deliver the set of non-dominated solutions in a single run.
- Since they are inherently parallel in nature, such algorithms can be easily parallelized to reduce the time taken for an optimization exercise.
- In the event, the problem involves only continuous variables, the efficiency of such algorithms can be further improved using hybridization i.e., coupling a local search (i.e., gradient based) to expedite convergence.

While a population based scheme is attractive, there are a number of key issues to be resolved prior to an application. The challenges are summarized below.

- The major disadvantage of any population based method is its computational complexity (i.e., the need to evaluate numerous solutions prior to convergence). In the event, such evaluations involve iterative solvers (as in computational fluid dynamics, finite element analysis etc.), the computational time required for an optimization exercise could easily be prohibitive. Thus the underlying efficiency of the optimizer is of key importance.
- Since most real life problems involve constraints, efficient handling of constraints is necessary. In the event such constraints are evaluated using computationally expensive iterative solvers, decisions concerning the cost-benefits of evaluating of an infeasible solution needs to be carefully considered.

- In presence of more than four objectives, alternatives to non-dominance based schemes have to be deployed to enforce adequate selection pressure to drive the population of solutions towards convergence.

- Real life application demands identification of robust solutions i.e., solutions which are insensitive/less sensitive to variations in the variable values. Development of algorithms to solve robust optimization problems are far more challenging as it requires evaluation of numerous additional solutions.

This dissertation will try to address the above listed research questions i.e., *how can existing population based methods be enhanced to deal with the challenges?* To begin with, firstly, the major roadblocks are identified, and thereafter, mechanisms for countering them are proposed and documented in detail.

In this study, the enhancements are made to a canonical population based method, i.e., a basic differential evolution (DE) algorithm. The performance of the modified algorithms are studied using a number of single, multi-/many-objective optimization problems. The enhancements are fairly generic and can be implemented within other population based optimization algorithms.

1.3.1 Differential evolution algorithm

Differential Evolution (DE) is a subset of evolutionary algorithms and can be considered as a generic population-based meta-heuristic. DE imitates the natural evolutionary process of the species, as it goes through the biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solution is generated through the consummation of multiple randomly selected parent individuals. This solution is then evaluated for fitness. Out of the parents and the offspring solution, one with the best performance survives. This process continues until the convergence condition is met. A basic DE framework is shown in Algorithm 1.1.

Algorithm 1.1 Basic Differential Evolution algorithm

SET: Maximum number of generations GEN_{max}.

1: **Initialize** a population
2: **while** $gen \leq GEN_{max}$ **do**
3: Select three random parents s.t., $p_1 \neq p_2 \neq p_3$
4: Perform the **mutation**
5: Perform the **recombination**
6: Perform the **selection**
7: **end while**

Initialization

The search starts with a set of randomly generated solutions i.e., $\mathbf{x}_{i,gen}, i = 1, 2, 3, ...N$, where N is the number of individuals in a population. A uniform sampling is commonly used for the design variables between its lower and upper bounds.

Mutation

Offspring is generated from the mating of three parent solutions. These parent solutions are randomly selected from the population, such that, they are mutually different from each other. These parent solutions are individually called the *donor vector*. To generate the offspring a *trial vector* ($\mathbf{v}_{i,gen}$) i.e., $\mathbf{v}_{i,gen} = \mathbf{x}_{r_1,gen} + F(\mathbf{x}_{r_2,gen} - \mathbf{x}_{r_3,gen}), r_1 \neq r_2 \neq r_3$ is generated in the mutation process, where F is called the mutation factor/scaling factor.

Recombination

In the recombination process, the offspring is generated by observing each of the variables either from the *donor vector* or the *trial vector*. The *target vector* ($\mathbf{u}_{i,gen}$) is generated as follows.

$$u_{i,j,gen} = \begin{cases} v_{i,j,gen}, & \text{if} \quad rand[0,1) \leq CR \quad or \quad j = j_{rand}, \\ x_{i,j,gen}, & \text{otherwise} \end{cases} \tag{1.3}$$

where $j = 1, ..D$ is the number of variables and CR is called the recombination factor/crossover rate.

Selection

The generated offspring is selected over any parent solution if the fitness of i-th solution is less than the r-th parent solution i.e., $f(\mathbf{u}_{i,gen}) < f(\mathbf{u}_{r,gen})$, assuming that the aim is to minimize the fitness.

1.3.2 Non-dominance based method

Non-dominance based methods are commonly used for multi-objective problems. In this case, the final output is not a single solution but rather a set of solutions.

1.3.3 Decomposition based method

Decomposition based methods are normally used for the problems where the number of objectives is more than one. There are many decomposition approaches reported in the literature. Three most common decomposition approaches are briefly described below.

Weighted Sum Approach

The weighted sum approach is a straightforward decomposition method. It assigns the weighting factor for each objective and calculates the weighted sum of the objectives [3]. Thus, the objective vector is transformed into a single objective scalar value as follows:

$$\min_{x \in X} \phi^{ws}(x) = \sum_{i=1}^{M} w_i f_i(x) \tag{1.4}$$

where $\mathbf{w}$ is the weight vector with $w_i \geq 0, \forall_i = 1, 2, ..., M$. The major drawback of this method is the choice of weight directions; improper set of weight direction misleads the search to a premature convergence and it is unable to solve problems with a concave Pareto front.

Tchebycheff Approach

In this approach, the search is directed with respect to a reference point, which is known as *ideal point* ($\mathbf{z}^*$) or *uptopian point* ($\mathbf{u}^*$). An *ideal point* is a hypothetical point with the following property i.e., $z^* = [\min(f_{j,1}(x)), \min(f_{j,2}(x)), ..., \min(f_{N,M}(x))]$, where N is the size of population and M is the number of objectives. In addition, the *uptopian point* is referred as $u^* = z^* - \epsilon$ (ϵ is a very small quantity s.t., $\epsilon > 0$). The objective vector is transformed into a single objective scalar value as follows:

$$\min_{x \in X} \phi^{te}(x) = \max_{i=1}^{M} w_i |(f_i(x) - u_i^*)| \tag{1.5}$$

where $\mathbf{w}$ is the weight vector for M number of objectives. One of the drawbacks of this method is that the search continues to direct through the reference point i.e., *ideal point*, the improper scaling of the objectives often increases/decreases the selection pressure.

Normal Boundary Intersection Method

Normal boundary intersection method (NBI) [4] is yet another approach used in decomposition. The mathematical formulation of the method is derived as follows:

$$\begin{gathered} \max s \\ \text{subject to constraint} \\ \Psi w_i + s\bar{n} = f_i(x) - f_i^* \\ s \in \Re, x \in \Omega \end{gathered} \tag{1.6}$$

where $\mathbf{w}$ is the weight vector i.e., $\sum_{i=1}^{M} w_i = 1$, f_i^* denotes the *ideal point* and $\Psi \in \Re^{m \times m}$ is a matrix with the columns $f_i(x) - f_i^*$ for $i = 1, 2, ...m$. The i-th column of this matrix ($f_i(x) - f_i^*$) is referred as the *pay-off* matrix. Here, $\bar{n}$ refers to the quasi normal which calculated as $\bar{n} = -\phi e$, where e is the column vector of all ones. One of the drawbacks of this method is that the method has to deal with the equality constraint.

As illustrated in Figure 1.3, the constraint has to ensure that $f(x)$ is always in L, the line with direction w and passing through f^*.

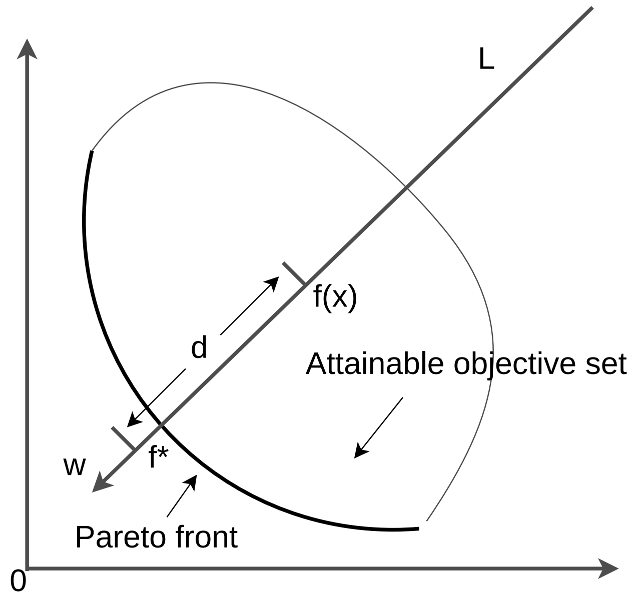


Figure 1.3: Illustration of normal boundary intersection method.

Variants of NBI use a penalty factor to deal with the constraint. Mathematically, the scalar optimization subproblem can be presented as follows

$$
\begin{aligned}
&\min_{x \in X} \phi^{BIP}(x) = d_1 + K d_2 \\
&\text{subject to constraint} \\
&\qquad d_1 = \frac{||(f^* - f(x))^T w||}{||w||} \\
&\text{and } \; d_2 = ||f(x) - (f^* - d_1 w)||
\end{aligned}
\tag{1.7}
$$

As shown in Figure 1.4, d_1 is the distance between f^* and p; d_2 is the distance between $f(x)$ and L. To match with the previous formulation of Equation 1.6, the penalty factor K for Equation 1.7 is needed.

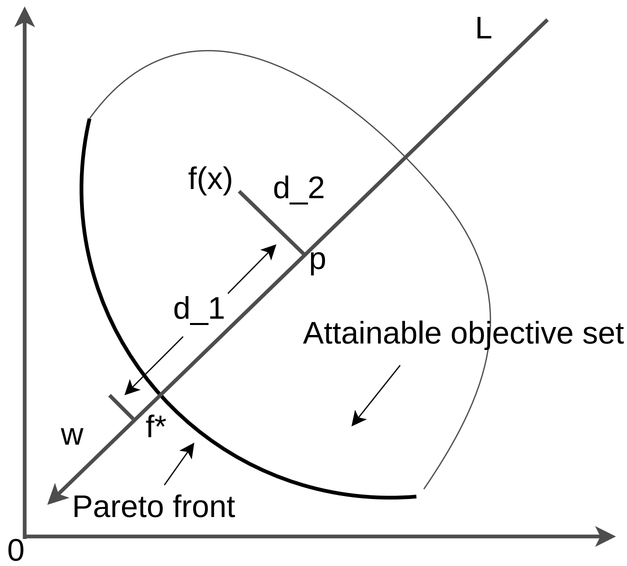


Figure 1.4: Illustration of boundary intersection approach using penalty factor.

1.3.4 Robust optimization method

Robust design optimization aims to find a solution (or a set of solutions) that is/are competent and reliable under given uncertainties. It can be seen from Figure 1.5 that, a deterministic optimal solution can be found at solution A using any conventional optimization algorithm assuming the minimization of the performance, which render the optimum solution prone to fail to maintain the desired performance with a slight variation in its variable values.

On the other hand, the solution B obtained from a robust optimization is moderately good in terms of optimality and also good in terms of robustness. It is clear that in order to solve a robust optimization problem, one needs to evaluate the neighborhood performance of the solution. Some of the early techniques relied on adding a safety factor to constraints/variables to come up with robust ("conservative") designs [5]. Recently, more involved research has focused on development of approaches to identify robust optimal solutions. The studies can be broadly classified into the areas that deal with (a) formulation of a robust optimization problem (b) quantification of robustness and (c) means to deal with such problems with affordable computing resources i.e., the search algorithms. A detailed description of each of these areas appears in Chapter 4.

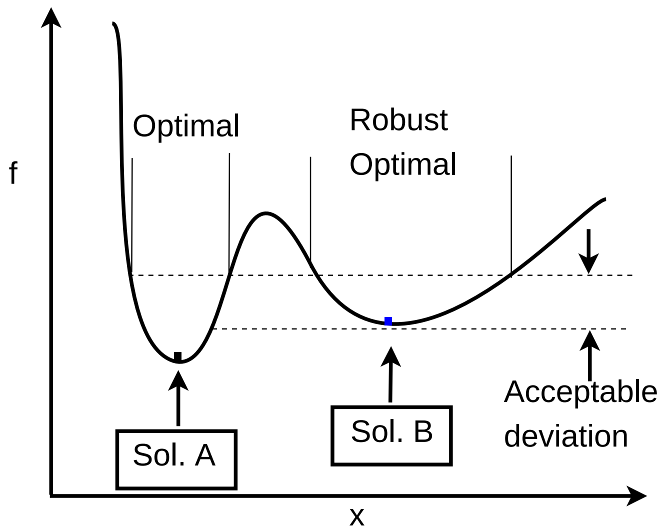


Figure 1.5: Comparison between conventional optimization and robust optimization (in minimization problem).

1.4 Performance assessment

The performance of any optimization problem can be evaluated by assessing the optimality of the solution (or set of solutions) achieved at the end of evolutionary process.

1.4.1 Performance assessment: Single objective optimization

In the context of single objective optimization, the problem of performance assessment is straightforward as the comparison only depends on a single performance criterion. Since the algorithm is stochastic in nature, it is necessary to report the results of multiple runs. In this thesis, multiple independent runs are carried out and the statistics are reported and compared across multiple algorithms.

1.4.2 Multi-/Many-objective optimization

The quality of solutions obtained from multi-objective optimization is often difficult to assess. Two measures are commonly used e.g., the spread of the solutions across the Pareto-optimal front and the ability to attain the Pareto-optimal front. Several metrics are available in the literature to compute the value of *convergence* and *diversity* of a set of solutions. In this study two common metrics have been used i.e., (a) hypervolume [6] (b) Inverted generational distance (IGD) [7, 8, 9].

- **Diversity:** assesses the spread across the Pareto-optimal front.
- **Convergence:** assesses convergence of the solutions i.e., closeness to the Pareto-optimal front.

Hypervolume

S-metric or Hypervolume (HV) was introduced by Zitzler and Theiele and then described by Coello Coello, Veldhuizen and Lamont [10]. It is the Lebesgue measure $\wedge$ of the union of hypercubes c_i for the non-dominated set m_i with respect to a reference point f_{ref}:

$$S(M) = \wedge(\bigcup_i c_i | m_i \in M) = \wedge(\bigcup_{m \in M} f | m \prec f \prec f_{ref}) \quad (1.8)$$

An example is shown in Figure 1.6 with a non-dominated set of solutions (i.e., A, B, C, D). The hyperarea refers to the union of all the individual areas of A, B, C and D. Hypervolume is used to capture the convergence and the diversity of a set of

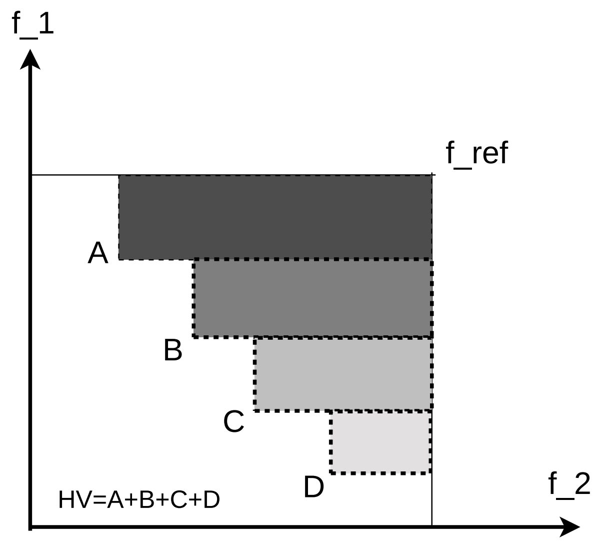


Figure 1.6: Calculation of hypervolume (both objectives are being minimized).

solutions. For a problem where minimization of the solutions is concerned, the large value of hypervolume demonstrates better convergence and also indicates the solutions are diverse.

IGD

Inverted generational distance (IGD) was introduced in [7, 11, 12] to measure the convergence and the spread of a set of solutions. The mathematical formulation is given below.

$$IGD = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} d(p, q) \tag{1.9}$$

where P is the reference set (or known Pareto front), Q is the Pareto front produced by any optimization algorithm and $\min_{q \in Q} d(p, q)$ is the minimum Euclidean distance between a point from the reference set to all the points found in the given Pareto front. The average value of this distance is the indication of IGD of that Pareto front. Figure 1.7 shows a Pareto front Q with a reference set of solutions.

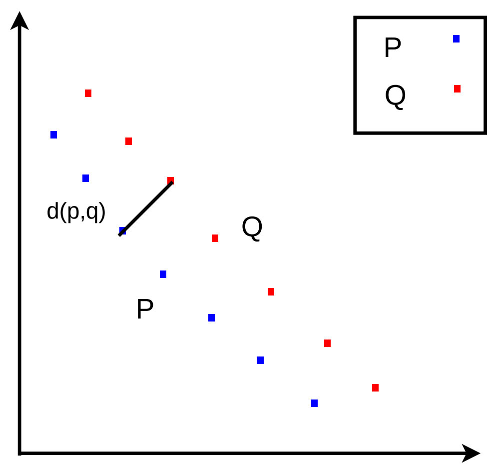


Figure 1.7: Calculation of IGD (P ≡ Pareto front, Q ≡ non-dominated set).

1.5 Scope of Research

In this thesis, efforts are made to address the challenges faced by existing optimization approaches. In particular, they include the presence of constraints, high numbers of objectives, and development of efficient means to identify solutions with desired levels of robustness. Although these strategies have been implemented in DE, it can be easily extended to other metaheuristic paradigms.

The purpose of the work presented in this thesis is not to invent an optimization algorithm, but rather to identify the potential areas where the algorithm can be enhanced and improved to deal with real world optimization problems (i.e., computational time, convergence, robustness). The working of the proposed methods is claimed through rigorous testing on mathematical benchmarks. The performance of the developed approaches has been compared with contemporary algorithms to objectively illustrate the benefits.

The primary focus of this thesis is to explore the efficiency and effectiveness of the approaches in solving engineering optimization problems. Although the emphasis of this work is on engineering design optimization, the methods developed are suitable as generic optimizers and can be applied to problems in other disciplines such as scheduling, finance and statistics. Dynamic problems, in which the objectives change with time, are not studied in this thesis.

1.6 Contributions of Thesis

The following contributions are made in this thesis:

1. The first contribution of this thesis relates to the development of an efficient and reliable optimization algorithm capable of dealing with constrained optimization problems. Towards this development, three different approaches are presented. The first proposal is to adaptively select the control parameters which hold the

key to performance of the algorithm. Secondly, a constraint handling method is introduced for inequality constraints. Finally, a general adaptive constraint handling method is proposed. These improvements are assessed by implementing the methods in the DE framework. Following is the outline of these algorithms.

(a) *Adaptive hybrid DE algorithm*: The proposed algorithm incorporates an adaptive crossover rate control mechanism, a combination of crossover types and a local search strategy. Binomial and exponential crossover mechanisms have been used in various stages of evolution to exploit their strengths in exploration and exploitation.

(b) *DE with constraint sequencing*: A novel constraint handling scheme has been introduced within the framework of differential evolution utilizing the concepts of partial evaluation and constraint sequencing. The utility of using multiple constraint sequences is highlighted using three illustrative examples. The approach is likely to provide significant computational benefits for problems involving computationally expensive constraints. Furthermore, since DE-CS attempts to reach the feasible space from different search directions, it is less likely to be trapped in local optima. Since the efficiency of the algorithm stems from handling constraints, the approach is likely to be less useful for problems with high feasibility ratio. While the approach presented in this thesis inherently assumes that the constraints can be evaluated independently, the method can be extended further to deal with blocks or sets of constraints as encountered in more realistic multidisciplinary optimization problems.

(c) *Adaptive constraint handling method*: An adaptive constraint handling approach has been presented. The constraint handling approach is embedded within the framework of multi-objective evolutionary algorithm based on decomposition (MOEA/D) to equip it to deal with constrained optimization problems. Since the constraint handling scheme is generic, it can be used in other forms of population based stochastic algorithms.

2. The second contribution is in the area of multi-/many-objective optimization problems. As reported in the previous studies, Pareto-dominance based methods are insufficient for dealing with the problems with high number of objectives. More often, these problems are tackled through the decomposition based methods when the convergence criterion is stricter. However, this method needs a careful consideration to balance between the convergence and diversity. This study proposes a decomposition based evolutionary algorithm. Its performance is demonstrated using unconstrained and constrained many objective optimization problems. The approach utilizes reference directions to guide the search, wherein the reference directions are generated using a systematic sampling. The algorithm is designed using a steady state form. In an attempt to alleviate the problems associated with decomposition (commonly encountered in the context of reference direction based methods), the balance between diversity and convergence is maintained using a simple preemptive distance comparison scheme.

3. The third contribution is in the area of optimization problems with uncertain variables (robust design optimization) for multi-/many-objective problems. Three relevant directions are pursued:

   (a) *Robust optimization problem formulation*: A very important step in a robust optimization exercise is to formulate a problem. Depending upon the robust formulation different optimization methods can be applied in order to get the robust solutions. A number of different approaches have been proposed in the literature, wherein additional objective(s) and/or constraint(s) have been added to the original formulation. In this thesis, a generic model of the robust design optimization problem is formulated which can easily be coupled with any evolutionary based optimization algorithm.

   (b) *Quantification of robustness*: The solution achieved by the robust optimization exercise needs to be quantified. Most of the approaches reported in the literature utilizes the *expected* value and variance of the solution. The major downside lies with the mechanism for estimating the *expected* fitness as it requires a large number of samples to compute the expected value with good accuracy. This study utilizes a six sigma based robust quantification with Latin hypercube sampling to approximate the *expected* value and the variance.

(c) *Search algorithms*: While the issues concerning the formulation of the problem and the measures of robustness have been addressed, the outcome of a robust optimization exercise is also dependent on the efficiency of the underlying search strategy. A simple decomposition based search algorithm is proposed to deal with the many-objective formulation of the robust optimization problem.

4. Finally, a number of numerical experiments are conducted on several benchmark test problems and engineering applications using the above mentioned algorithms. Comparison with previously reported studies is included in order to highlight the benefits.

1.7 Organization of Thesis

Following this introduction, this thesis is divided into five further chapters. While Chapter 1 lays some of the groundwork for the research, Chapters 2–4 present the principal technical contributions and describes the numerical experiments in detail. Chapter 5 provides a summary and a few future directions of the presented work. Since the thesis explores diverse disciplines within optimization, the relevant literature is included in each chapter instead of as one large unit. Individual contents of the chapters are outlined as follows.

- Chapter 1, provides an introduction to optimization with a brief and simple classification of various optimization techniques. Population based techniques, which form the center of optimization methods developed in this thesis, are particularly discussed in detail. Thereafter, various areas in which this thesis seeks to make improvements in existing approaches are highlighted. These include constraint handling, many-objective and the robust optimization problems. The detailed contributions in each of these areas are presented individually in further chapters.

- In Chapter 2, three new algorithms for dealing with constrained optimization problems are proposed. One of them, Adaptive Hybrid DE algorithm, is an improved DE that adaptively selects the control parameters and also incorporates the local search for further benefits. The other algorithm, DE with Constraint Sequencing (DE-CS), extends the conventional constraint handling techniques by introducing constraint sequencing schemes to improve the rate of convergence. Finally, in MOEA/D-ACH, an adaptive constraint handling method is presented in the framework of MOEA/D for multi-objective problems.

- In Chapter 3, a decomposition based evolutionary algorithm (DBEA) is proposed to deal with the many-objective problems (i.e., problems with four or more objectives). The benefits of using DBEA are highlighted with several real-world engineering benchmark problems.

- In Chapter 4, a robust optimization algorithm is introduced wherein a six sigma based robust measure is used to quantify the robustness. The underlying optimization problem is solved using the decomposition based evolutionary algorithm developed earlier.

- In Chapter 5, a summary of the findings of the work is presented. In addition, future issues and directions which could be pursued with the aim of making the algorithms more efficient for handling various types of optimization problems are identified.

Chapter 2

Constraint Handling

Part of this work has previously appeared in **Asafuddoula, M.** , Ray, T. and Sarker, R., “A self-adaptive differential evolution algorithm with constraint sequencing,” in Proceedings AI 2012: Advances in Artificial Intelligence, vol. 7691 of Lecture Notes in Artificial Intelligence, pp. 182-193, Springer, 2012..

Asafuddoula, M., Ray, T., and Sarker, R., and Alam, K., “An adaptive constraint handling approach embedded MOEA/D,” in IEEE Congress on Evolutionary Computation, pp. 2516-2513, Brisbane, Australia, 2012.

Asafuddoula, M., Ray, T., and Sarker, R., “An efficient constraint handling approach for optimization problems with limited feasibility and computationally expensive constraint evaluations”, in Proceeding of the 2013 Genetic and Evolutionary Computation Conference Companion (GECCO 2013), (Amsterdam, The Netherlands), pp. 113-114, 2013.

Asafuddoula, M., Ray, T., and Sarker, R., “An adaptive hybrid differential evolution algorithm for single objective optimization”, Applied Mathematics and Computation. In Press, (Accepted 06/01/2014).

Chapter Overview

This chapter introduces an adaptive parameter control strategy for DE and two constraint handling methods. These strategies are implemented in DE, however, the ideas can be extended to other similar frameworks. Comparisons with other published state-of-the-art algorithms on various benchmark problems are included in order to highlight the benefits.

2.1 Introduction

Over the years, a large number of real world optimization problems have been successfully solved using various forms of population based stochastic optimization algorithms. Many of these algorithms belong to the class of DE. The winning entries of *CEC-2006* [13] and *CEC-2010* [14] competitions relied on DE based algorithms. Although the performance of DE based algorithms has improved consistently, their performance is known to be largely dependent on the choice of its parameters i.e., mutation factor (F), crossover rate (CR). For constrained optimization problems, the underlying schemes for constraint handling play a major role in the efficiency of the optimization algorithm. The following discussion provides a brief background on existing approaches to deal with these.

- **Mutation factor:** The mutation factor is perhaps the first parameter which has been identified to affect the performance of DE. Earlier forms of DEs used a fixed value of $F = 0.5$, while there are reports of F ranging between 0.1 and 1.0 [15], use of Gaussian with mean of 0.5 and a standard deviation of 0.3 [16] or the use of a Cauchy distribution with a mean of 0.5 and a standard deviation of 0.1 [17].

- **Crossover rate:** The crossover rate (CR) is yet another parameter which has also been identified to affect the performance of DE. Numerical experiments have indicated that a linearly separable problem can be efficiently solved using low CR values, while higher CR values are preferred for non-separable and multi-modal problems [18, 19]. Various CR values have been used in past studies ranging from fixed values of 0.9 for solving linear, non-linear and multi-modal functions [20, 21, 22, 23], 0.7 for solving cubic, polynomial and quadratic functions [24] and 0.3 for noisy optimization problems [25, 26].

- **Mutation strategy:** There are also a number of mutation strategies that have been proposed over the years. Commonly adopted mutation strategies include *DE/best/1*, *DE/rand/1*, *DE/current-to-best/1* and *DE/rand/2* [27, 28]. Variants have also emerged in recent years, such as *DE/rand/1/-Either -Or-Algorithm* etc [29]. Observations have indicated that *DE/rand/1* performs well for linearly separable, unimodal or non-separable and noisy functions [30, 25]. Experiments also indicate that *DE/current-to-best/1* and *DE/rand/2* are effective for solving multi-modal and non-separable functions [27].

- **Crossover strategy:** In terms of crossover strategy, two most promising ones include the binomial crossover and the exponential crossover. Studies in [30] indicate that the binomial crossover undergoing a binomial gene-wise crossover [31] is less greedy and has the ability to solve linearly separable and multi-modal problems [27]. The exponential crossover undergoing sequential participation of multiple genes [32] exploit more and tend to be useful for solving non-linear functions [21]. Two other forms i.e., *trigonometric mutation* and *arithmetic recombination* have also been proposed in recent years.

- **Parent selection:** While in a native DE, three random parents are chosen for mutation [20, 33], a number of recent algorithms have modified this basic parent selection scheme. In the works of [21], two parents were randomly selected from an active population while the third was selected randomly from an archive. Instead of random selection of parents, a selection probability inversely proportional to its distance from the mutated individual was used for selecting multiple parents in [34] for the solution of unconstrained optimization problems. In an effort to further enhance the performance of DE variants, local search strategies have also been incorporated such as in the works of hybrid DE [35, 36, 37]. In such approaches, local searches are conducted sparingly and periodically from promising solutions.

- **Constraint-handling:** In addition to the parameters discussed above, the performance of the optimization algorithm is affected by the underlying mechanism of constraint handling. Constraint handling methods can be broadly categorized into six different types [38, 39] use of penalty functions, repair schemes, feasibility-first, ϵ-constrained method, dominance based method, adaptive method and ensembles. The discussion below provides a brief of each approach.

  - **Use of penalty functions:** Penalty function based methods are one of the most commonly adopted approaches. These methods penalize the infeasible solution with predefined penalty factor(s) and aggregate into a scalar value. In the context of minimization, the primary aim is to decrease the fitness of infeasible solutions in order to favor the selection of feasible solutions. Variants of the penalty function based approach include static penalties [40], dynamic penalties [41], annealing penalties [42], adaptive penalties [43, 44], death penalty method [40], superiority of feasible points [45], faster adaptive method [46]. While some of these methods have reported competitive performance, choice of an appropriate penalty factor(s) is non trivial.
  - **Repair schemes:** Evolutionary algorithms perform well for unconstrained or simple constrained optimization problems e.g., box constraints but are known to face difficulties in solving highly constrained problems. This is because the traditional search operators (i.e., crossover and mutation) are blind to constraints [47]. A number of repair schemes have been introduced in [48, 39, 49] wherein an infeasible solution is repaired to a feasible solution. Development of repair schemes are often problem dependent and may also

involve additional computational cost.

- **Feasibility-first:** Feasibility-first scheme is yet another form of constraint handling where a feasible solution is always preferred over an infeasible solution. The common form of preference rules are governed by the following: (a) any feasible solution is preferred over an infeasible solution (b) among two feasible solutions, the one with better objective is preferred (c) among two infeasible solutions, the one with lowest constraint violation is preferred [50]. While these feasibility-rules were originally developed in the context of evolutionary algorithms, they were introduced in DE algorithm [51, 52].

- **ϵ-constrained method:** One of the most recent constraint-handling techniques reported in the specialized literature is that of ϵ-constraint proposed in [53]. The method essentially designates a selected set of infeasible solutions as feasible i.e., by accepting a certain level of constraint violation. This acceptance level of constraint violation (v_ϵ) is referred as the ϵ-level which is calculated for each generation (G) as follows:

$$v_\epsilon(G) = \begin{cases} v_{\epsilon_0}(1 - \frac{G}{G_c})^R, & 0 < G < G_c \\ 0, & G \geq G_c \end{cases} \tag{2.1}$$

where G_c is a control generation up to which the ϵ-level is considered. The parameter R is changed according to the Equation 2.3. The ϵ level is adjusted to be a small value $\epsilon_\lambda = 10^{-5}$ at the generation $G_\lambda = 0.95G_c$.

$$\epsilon(G_\lambda) = v_{\epsilon_0}(1 - G_\lambda/G_c)^R = \epsilon_\lambda \tag{2.2}$$

$$\begin{aligned} R &= (\log \epsilon_\lambda - \log v_{\epsilon_0})/\log(1 - G_\lambda/G_c) \\ &= (-5 - \log v_{\epsilon_0})/\log 0.05 \end{aligned} \tag{2.3}$$

To avoid a too small value of R, a predefined value of $R_{min} = 3$ is assigned.

The value of R and scaling factor (F) are controlled as follows:

$$R = 0.3R + 0.7R_{min} \tag{2.4}$$

$$F = 0.3F_0 + 0.7 \tag{2.5}$$

The method delivered highly competitive results on a set of constraint optimization problems of CEC-2010 [14]. Motivated by its performance, the approach has been successfully adopted in the works such as in [54, 55, 56]. Although the method performed well on the benchmark suite, the prescribed values of the parameters could potentially affect the performance for an unknown problem.

– **Dominance based method:** In spite of the fact that dominance based methods face great difficulties with an increase of number of objectives, there are some competitive constraint handling methods which rely on such concepts. Infeasibility Driven Evolutionary Algorithm (IDEA) [57] utilizes constraint violation as an additional objective and preserves a set of infeasible solutions throughout the search. Notable concepts in this direction include the works reported in [58, 59, 60, 61]. Others include spherical-pruning multi-objective optimization differential evolution (sp-MODE) [62], Hybrid Constrained EA (HCOEA) [63], steady state EA [64], adaptive trade-off model (ATM) evolution strategy (ATMES) [65], ϵ-dominance concept [66], Adaptive Bacterial Foraging Algorithm (ABFA) [67].

– **Ensemble and other methods:** In the backdrop of No-Free-Lunch (NFL) theorem [68], ensemble of constraint handling schemes have also been introduced [24, 69, 70, 71]. Such schemes still require a number of user defined parameters.

In order to eliminate the need of user defined parameter(s)/penalty factors, a number of adaptive strategies have been proposed in the literature. They include adaptive penalty technique [72], parameter less penalty function [73], a self-adaptive fitness formulation [74], stochastic ranking [75] etc.

In this chapter, one adaptive hybrid parameter control strategy implemented as an Adaptive Hybrid DE Algorithm and two key ideas for dealing with constrained optimization problems are introduced. The two constraint-handling methods are described using the framework of DE i.e., DE with Constraint Sequencing (DE-CS) and MOEA/D with Adaptive Constraint Handling (MOEA/D-ACH).

2.2 Adaptive Hybrid DE algorithm

Adaptive Hybrid DE algorithm (AH-DEa) inherits the benefits of various proven strategies from the literature. The proposed algorithm has the following features.

- The first feature is its use of adaptive crossover rates from a given set of discrete values spanning a range from 0.1 to 1.0.
- The mutation factor is also adapted, following the scheme proposed by [21] and as for the diversity, two parents are selected from the active population and the third from the archive.
- The basic mutation strategy is employed e.g., DE/rand/1 with a combination of binomial and exponential crossover depending upon the search strategy in different stages of evolution.
- A local search is used at the end of the DE process in an attempt to further improve the best solution obtained through DE evolution [36].
- A restart mechanism is introduced during the local search phase, wherein the population is reinitialized if no improvement to the best solution is achieved during the course of local search.

The pseudo code for the proposed algorithm is presented in the following.

Algorithm 2.1 Pseudocode of an adaptive hybrid DE algorithm (AH-DEa)

Input: Scale factor F_0, Crossover rate $CR_k = \{1/N_{CR}, 2/N_{CR}, ...k/N_{CR}\}$, where $N_{CR} = 10$ is the number of discrete crossover rates, $k = \{1, 2....N_{CR}\}$, Size of archive population is M, Size of active population is N, Number of max generation Max_{gen}, Number of max function evaluation Max_{fevel}.

Set the generation count $G = 0$ and function evaluation $fevel = 0$.
Initialize $x_{j,i} = x_{j,min} + rand_j[0, 1).(x_{j,max} - x_{j,min})$, $j = 1, 2, ...D$.
Assign M individuals $x_{j,i}$ to *Archive_Pop*.
Assign $PR_k = \{\text{k}/N_{PR}\}$, $k = \{1, 2, 3........N_{PR}\}$.
Evaluate the *Archive_Pop* and sort the members using constraint violation.
Assign the top N individuals to *Active_Pop* from *Archive_Pop*.
while $(G \leq Max_{gen})$ & $(fevel \leq Max_{fevel})$ **do**
    Select the crossover rate CR using roulette wheel selection from the pool of crossover rates CR_k, where $k = \{1, 2....N_{CR}\}$.
    for *i=1:N* **do**
        Select three parents as described in Subsection 2.2.2 and generate the donor vector, $\mathbf{v}_{i,G}$ using $DE/rand/1$ mutation strategy.
        Generate the trial vector, $u_{j,i,G} = DE/rand/1/bin$ if $i \leq 0.2 * Max_{gen}$, otherwise $DE/rand/1/Exp$.
        Evaluate the trial vector, $u_{j,i,G}$.
        if $\xi(\mathbf{u}_{i,G}) \leq \xi(Active\_Pop_i)$ **then**
            $Active\_Pop_i = \mathbf{u}_{i,G}$
        else
            $Archive\_Pop_l = \mathbf{u}_{i,G}; l = rand[N, M]$.
        end if
        Function evaluation, $fevel = fevel + 1$.
    end for
    Generation count, $G = G + 1$.
    Update the PR_k by calculating the success ratio described using Equation 2.6.
    Invoke local search if $fevel >= 90\% Max_{fevel}$.
end while

Algorithm description:

The algorithm starts off with a predefined set of crossover rates ($N_{CR} = 10$ values selected in the study) with equal probability and the corresponding probability of the crossover rate is controlled based on the success ratio of each individual. In an attempt to maintain diversity, two parents are selected from the active population of size N randomly, while the third parent is selected from the archive of size M. The active population is the top N individuals which are initially copied from the archive. The fitness (ξ) of i^{th} solution consist the objective function value $f_i(\mathbf{x})$, $\mathbf{x} \in \Re^D$ and the value of constraints c_i, where c_i is the constraint violation measure based on the equality and inequality constraints and D represents the number of variables. Offspring solution is compared with the i^{th} individual in the active population for a possible replacement. If the offspring is unable to make it into the active population, it replaces a random solution in the archive. A local search is invoked with the best individual for the last 10% of the function evaluations. The replacement scheme is analogous to steady state models of evolutionary computation. The pseudocode of the algorithm is presented in Algorithm 2.1, while its components are described in sub-sections.

2.2.1 Adaptive CR strategy

The choice of crossover rate in a conventional DE is user defined. However, some predefined values have been suggested in the literature which varies across the range of problems. To alleviate this difficulty, few approaches have been developed over the last few years. In this proposed algorithm (AH-DEa), a roulette wheel based CR selection scheme has been used. Initially, the crossover rates are mapped to contiguous segments between $1/N_{CR}$ to k/N_{CR} where, N_{CR} is the number of CRs in the set and $k = \{1, 2, ...N_{CR}\}$. From this set, a CR value is selected based on their selection values (PR). These selection values represent a set PR_k = {k/N_{PR}}, $k = \{1, 2, 3........N_{PR}\}$. These PR_k values are updated based on success or failure of the offspring generated. Success refers to a situation when an offspring replaces an individual. The success ratio is calculated and used to update each CR selection value PR_k as follows:

$$PR_k = \frac{SR_k}{\sum_{i=1}^{N_{CR}} SR_i}, \tag{2.6}$$

where $k = \{1, 2, 3....N_{CR}\}$ is the number of crossover rates and SR_k is the success ratio of k^{th} crossover rate. The success ratio SR_k is calculated as follows:

$$SR_k = \frac{\sum_{i=1}^{N} ns_{fevel}}{\sum_{i=1}^{N} ns_{fevel} + \sum_{i=1}^{N} nf_{fevel}} \tag{2.7}$$

where, N is the size of active population, ns_{fevel} is the number of successes and nf_{fevel} is the number of failures. A value of 0.05 is used if the success ratio SR_k for k^{th} crossover rate becomes zero.

2.2.2 Parent selection strategy

The performance of DE is highly dependent on the underlying mechanism of parent selection. The earlier section provides a brief discussion on various parent selection schemes. In this algorithm, the population is logically divided into two sets, an active population and an archive population. Two parents are selected from the active population and one parent is selected form the archive population.

2.2.3 Analysis of mutation strategy

There are a number of alternative mutation strategies. The performance of the mutation strategies is dependent on the participating parents [27, 76]. For solving unimodal, separable and non-separable functions, $DE/rand/1$ has been used widely [29] while $DE/rand/1$ was found to perform well for multi-modal problems [34]. In this algorithm, $DE/rand/1$ has been used for the mutation process and the scaling factor (F) is controlled using the strategy proposed in [21].

2.2.4 Analysis of crossover strategy

In DE, two crossover strategies have been commonly used i.e., binomial and exponential. Depending upon the control parameter selection scheme, the behaviour of this crossover process can vary. As discussed earlier, $DE/rand/1/Bin$ is better for exploration while $DE/rand/1/Exp$ is known to be better for exploitation. Therefore, in this algorithm both strategies have been used in different stages of the search process. At the beginning of a search, a binomial crossover is used for offspring generation while in later stages exploitation is achieved through an exponential crossover. In this study, this transition point is defined as 80% of function evaluation budget.

1. **Binomial crossover (DE/rand/1/Bin):** The binomial crossover has consistently performed well on several benchmarks and are known to be explorative [27, 20]. In this algorithm, a binomial crossover is used for exploration. The binomial crossover is defined as follows:

$$u_{i,j,G} = \begin{cases} v_{i,j,G}, & \text{if} \quad rand[0,1) \leq CR \quad or \quad j = j_{rand}, \\ x_{i,j,G}, & \text{otherwise} \end{cases} \tag{2.8}$$

   In Equation 2.8 $v_{i,j,G}$ is referred as the *trial vector*. If a randomly generated number is less than or equal to the control parameter (CR), then the trial vector is copied to the target vector ($u_{i,j,G}$), otherwise, the variable remains unchanged. In binomial crossover this process is repeated for every variable.

2. **Exponential crossover (DE/rand/1/Exp):** The exponential crossover is yet another form which has been successful in exploitation [32, 21]. The exponential crossover is defined as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } j = \langle n \rangle_D, \langle n+1 \rangle_D, .., \langle n+L-1 \rangle_D \\ x_{j,i,G}, & \text{otherwise} \end{cases} \tag{2.9}$$

   In Equation 2.9 the acute brackets $\langle\rangle_D$ denote a modulo function with modulus D. The integer n is a random number and L is also an integer drawn from $[1, D]$. Hence, picking a random starting point, the target vector is generated by copying the trial vector from $L = 0$ to D, if the random number $rand_j[0, 1)$ less than or equal to crossover rate (CR). The remaining parameters are copied to the target vector.

2.2.5 Gradient local search

Gradient based search used in this study utilizes sequential quadratic programming (SQP). In the proposed algorithm, a local search is invoked from the best individual and a budget of 10% of the total function evaluation is allocated. In the event, such a search fails to locate any better solution using D function evaluations, a local search is initiated from the next best solution and the process continues till the function budget is exhausted.

2.2.6 Constraint-handling

In this proposed algorithm, both equality and inequality constraints have been considered. The equality constraints are transformed into inequality constraints by subtracting a tolerance value $\epsilon = 1e^{-04}$ suggested in [14]. The violation can be measured by sum of all constraint violations. The violation is calculated as follows:

$$v = \sum_{i=1}^{p} max(g_i, 0) + \sum_{i=1}^{q} max(|h_i - \epsilon|, 0) \tag{2.10}$$

Where, p and q are the number of inequality and equality constraints. An epsilon level comparison proposed in [21] is used to order the individuals according to their fitness.

2.2.7 Numerical experiments

In order to access the performance of the proposed strategies, 40 recent scalable benchmark problems were used. The results obtained using the proposed algorithm (worst, best, mean, median and standard deviation) are compared against top algorithms on the selected benchmark problems. Problems C01-C18 belong to the suite of CEC-2010 benchmark [14] while problems G01-G24 belong to the suite of CEC-2006 benchmark problems [13]. Features of the problems C01-C18 and G01-G24 are listed in Table 2.2 and Table 2.3. Table 2.1 shows the parameter settings require for AH-DEa and its variants.

To evaluate the performance of the proposed algorithm, a benchmarking is conducted using the results of 4 other algorithms. The comparison is made with the best, mean and standard deviation for all the problems. To evaluate the statistical significance among the results, a Wilcoxon signed-rank test at α=0.05 in every cases

Table 2.1: Parameter settings

Control Parameter	Actual values
Archive size (M)	if (D > 40) M=2D else M=100D
Active Population size (N)	if (D > 40) M=1D else M=4D
Scaling factor (F_0)	0.5
Crossover rate (CR)	{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1} Adaptive

Table 2.2: Summary of test problems for C01-C18

Problem	D	Search Range	f type
C01	10/30	$[0, 10]^D$	Non Separable
C02	10/30	$[-5.12, 5.12]^D$	Separable
C03	10/30	$[-1000, 1000]^D$	Non Separable
C04	10/30	$[-50, 50]^D$	Separable
C05	10/30	$[-600, 600]^D$	Separable
C06	10/30	$[-600, 600]^D$	Non Separable
C07	10/30	$[-140, 140]^D$	Non Separable
C08	10/30	$[-140, 140]^D$	Non Separable
C09	10/30	$[-500, 500]^D$	Non Separable
C10	10/30	$[-500, 500]^D$	Non Separable
C11	10/30	$[-100, 100]^D$	Rotated
C12	10/30	$[-1000, 1000]^D$	Separable
C13	10/30	$[-500, 500]^D$	Separable
C14	10/30	$[-1000, 1000]^D$	Non Separable
C15	10/30	$[-1000, 1000]^D$	Non Separable
C16	10/30	$[-10, 10]^D$	Non Separable
C17	10/30	$[-10, 10]^D$	Non Separable
C18	10/30	$[-50, 50]^D$	Non Separable

(i.e., for best, mean and Std) is used. Using a Wilcoxon rank sum test, the sampling distribution of the difference between two samples is symmetric with zero median. At the default 5% significance level, the test fails to reject to a null hypothesis of zero median in the difference. A "+" is indicated to mark the cases when a null hypothesis is rejected at the 5% significance level, if the result is superior. A mark of " − " is used when a null hypothesis is rejected at the 5% significance level, if the results

Table 2.3: Summary of test problems for G01-G24

Problem	D	Search Range	f type
G01	13	$[0,1]^9[0,100]^4[0,1]^{(D-13)}$	quadratic
G02	20	$[0,10]^D$	nonlinear
G03	10	$[0,1]^D$	polynomial
G04	5	$[78,102][33,45][27,45]^{(D-2)}$	quadratic
G05	4	$[0,1200]^2[-0.55,0.55]^{(D-2)}$	cubic
G06	2	$[13,100][0,100]$	cubic
G07	10	$[-10,10]^D$	quadratic
G08	2	$[0,10]^10$	nonlinear
G09	7	$[-10,10]^D$	polynomial
G10	8	$-$	linear
G11	2	$[-1,1]^D$	quadratic
G12	3	$[0,10]^D$	quadratic
G13	5	$[-2.3,2.3]^2[-3.2,3.2]^{(D-2)}$	nonlinear
G14	10	$[0,10]^D$	nonlinear
G15	3	$[0,10]^D$	quadratic
G16	5	$-$	nonlinear
G17	6	$-$	nonlinear
G18	9	$[-10,10]^8[0,20]^{(D-8)}$	quadratic
G19	15	$[0,10]^D$	nonlinear
G21	7	$-$	linear
G23	9	$-$	linear
G24	2	$[0,3][0,4]$	linear

*The search range of some functions is marked as '$-$' for brevity, reader should find in [13]

exhibit inferior performance and with " = ", when the performance is not statistically significant (i.e., fails to reject a null hypothesis at the 5% significance level).

1. **Statistical comparison on CEC-2010 test problems (C01-C18):** Here, the results of the proposed algorithm are compared with other forms. In this comparison, the selected algorithms have all performed well.

- The performance of (AH-DEa) is compared with SAMO-GA [77], which is a self-adaptive multi-operator genetic algorithm. SAMO-GA is an algorithm where population is divided into four sub-populations with individual crossover and mutation rates. The algorithm uses Gaussian numbers to adaptively select F and CR. In the algorithm, the population is logically divided into two sub-populations

Table 2.4: Comparison of AH-DEa with SAMO-GA, SMOA-DE, e-DEag and IEMA for CEC-2010 in 10 dimension.

Criterion		AH-DEa	SAMO-GA	SMOA-DE	e-DEag	IEMA
	B	3	0	0	0	0
Better	M	5	1	4	3	1
	S	6	1	3	5	2
	B	14	11	15	13	3
Equal	M	6	0	4	6	0
	S	3	0	0	4	0
	B	1	7	3	4	14
Worst	M	6	17	10	9	17
	S	9	17	15	9	16

Table 2.5: Comparison of AH-DEa with SAMO-GA, SMOA-DE, e-DEag and IEMA for CEC-2010 in 30 dimension.

Criterion		AH-DEa	SAMO-GA	SMOA-DE	e-DEag	IEMA
	B	8	1	2	0	0
Better	M	10	0	4	3	0
	S	12	0	4	3	0
	B	6	2	7	2	2
Equal	M	2	0	2	0	0
	S	0	0	0	1	0
	B	4	15	9	16	12
Worst	M	6	18	12	15	14
	S	6	18	14	14	14

and CR and F values are selected adaptively. The results of 10D problems (i.e., benchmark problems of CEC-2010) are presented in Table 2.4 (details are provided in Tables A.1- A.6). Using best value as the measure, AH-DEa obtained 3 better, 14 equal and 1 worst solutions, henceforth represented as a score of 3/14/1. The corresponding score of SAMO-GA based on the best value is 0/11/7. Based on mean value measure, AH-DEa has the score of 5/6/6 (i.e., 5 better, 6 equal and 6 worst solutions), whereas, SAMO-GA has the corresponding score of 1/0/17. For standard deviation, AH-DEa has the score of 6/3/9, whereas, SAMO-GA has the corresponding score of 1/0/17. In 30D problems (i.e., benchmark problems of CEC-2010), again AH-DEa has the score of 8/6/4 on best value, 10/2/6 on mean value and 12/0/6 on standard deviation value, whereas, SAMO-GA has the corresponding score of 1/2/15, 0/0/18 and 0/0/18 (Table 2.5).

- The performance of AH-DEa is compared with SAMO-DE [77], a self-adaptive algorithm with multi-operator strategy. In this algorithm, DE variant rand-to-best and current/2/best is used for mutation and Gaussian numbers have been used to find the values of F and CR. The results of 10D problems (i.e., benchmark problems of CEC-2010) are presented in Table 2.4. In 10D problems, AH-DEa has the score of 3/14/1 on best value, 5/6/6 on mean value and 6/3/9 on standard deviation value, whereas, SAMO-DE has the corresponding score of 0/15/3, 4/4/10 and 3/0/15 (Table 2.4). In 30D problems, again AH-DEa has the score of 8/6/4 on best value, 10/2/6 on mean value and 12/0/6 on standard deviation value, whereas, SAMO-DE has the corresponding score of 2/7/9, 4/2/12 and 4/0/14 (Table 2.5).

- The performance of the algorithm is further compared with e-DEag [78], an algorithm with sub-populations and efficient constraint handling. The results of 10D problems (i.e., benchmark problems of CEC-2010) are presented in Table 2.4. In 10D problems, AH-DEa has the score of 3/14/1 on best value, 5/6/6 on mean value and 6/3/9 on standard deviation value, whereas, e-DEag has the corresponding score of 0/13/4, 3/6/9 and 5/4/9 (Table 2.4). In 30D problems, again AH-DEa has the score of 8/6/4 on best value, 10/2/6 on mean value and 12/0/6 on standard deviation value, whereas, e-DEag has the corresponding score of 0/2/16, 3/0/15 and 3/1/14 (Table 2.5).

- Lastly, the results are compared with IEMA [79], which is infeasible empowered memetic algorithm without any adaptive strategy. In 10D problems, AH-DEa has the score of 3/14/1 on best value, 5/6/6 on mean value and 6/3/9 on standard deviation value, whereas, IEMA has the corresponding score of 0/3/14, 1/0/17 and 2/0/16 (Table 2.4). In 30D problems, again AH-DEa has the score of 8/6/4 on best value, 10/2/6 on mean value and 12/0/6 on standard deviation value, whereas, IEMA has the corresponding score of 0/2/12, 0/0/14 and 0/0/14 (Table 2.5).

The Wilcoxon statistical analysis has been made to find the significance between two samples by checking the normality distribution. This test is associated with a p-value measurement, which represents the dissimilarity between two samples with respect to normal shape. Tables 2.6 and 2.7 show the outcome of a statistical analysis among AH-DEa with other algorithms.

Table 2.6: The Wilcoxon sign rank test results for AH-DEa with SAMO-GA, SMOA-DE, e-DEag and IEMA for CEC-2010 in 10 dimension.

Algorithm			$W-$	$W+$	*Decision*
AH-DEa	SAMO-GA	*Best*	121	79	=
		Mean	131	52	+
	SMOA-DE	*Best*	48	67	=
		Mean	77	90	=
	e-DEag	*Best*	101	54	+
		Mean	87	64	=
	IEMA	*Best*	126.5	46.5	+
		Mean	142	39	+

The statistical comparison for 10D problems of CEC-2010 benchmark shows equal for best results and better for mean results. For all the results, the negative rank

Table 2.7: The Wilcoxon sign rank test results for AH-DEa with SAMO-GA, SMOA-DE, e-DEag and IEMA for CEC-2010 in 30 dimension.

Algorithm			$W-$	$W+$	*Decision*
AH-DEa	SAMO-GA	*Best*	143	33	+
		Mean	128	34	+
	SMOA-DE	*Best*	100	70	=
		Mean	107	64	+
	e-DEag	*Best*	158	15	+
		Mean	141	37	=
	IEMA	*Best*	134	46	+
		Mean	152	19	+

is higher than the positive for AH-DEa. Performance of SAMO-DE is the same as AH-DEa, whereas AH-DEa is better than e-DEag based on best and better than IEMA based on both mean and std. In 30D problems of CEC-2010 benchmark, AH-DEa shows better in best and mean over SAMO-GA. The mean results of AH-DEa are also better when compared with SAMO-DE and IEMA. Comparing the best results, performance of AH-DEa is the same as SAMO-DE and way better than e-DEag and IEMA.

Among all these algorithms, the t-test analysis reassert that AH-DEa ranks 1st (42/108, 42 better out of 108), SAMO-DE ranks 2nd, (17/ 108), e-DEag ranks 3rd (14/ 108), SAMO-GA ranks 4th (3/108) and IEMA ranks the last (3/108, all solutions are not available) based on best, mean and std marked as '+' from Table A.1-Table A.2.

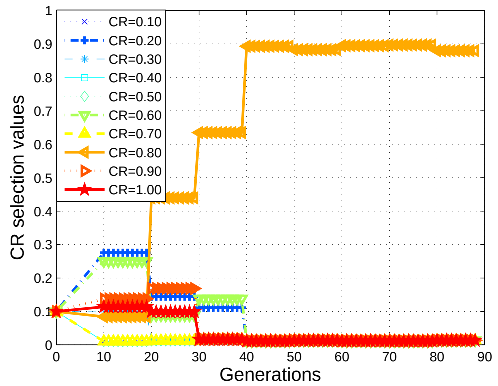
2. **Comparison on CEC-2006 test problems (G01-G24):** The results of AH-DEa are presented in Table A.7-Table A.9 based on 240,000 FEs (same as used in other algorithms to find the optimal solutions). The tables show the known *optimal* solutions for each problem with the statistics of best, mean, and std obtained from 30 runs. The problems are categorized into different classes as indicated in Table 2.12. Among those problems, G02 is multi-modal, which contain many local optima near the global optimum [21]. For most of the problems, AH-DEa was

able to find the optimal solution in around 100,000 FEs. For problems G01, G12, G13 and G24, the optimal solution is found in all 30 runs with the same global optima. In other problems, G02-G11 and G14-G23, the optimal results were found consistently in all runs with a very little variation. The results from AH-DEa are highlighted in boldface, which are better for problems G02, G07, G09, G010, G013, G14 and G18. The results obtained using AH-DEa are significantly better for problems G02, G10, G14 (in three decimal points) and marginally better for problems G07, G09, G13 and G18 or at least comparative in all the test problems.

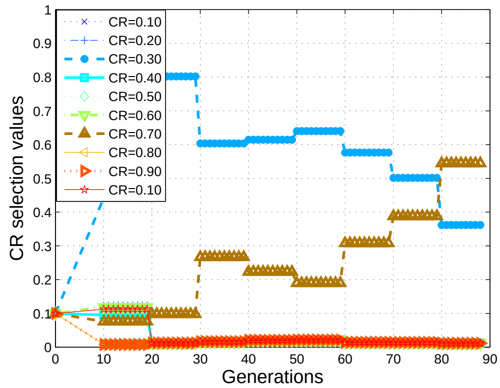
2.2.8 Investigation of CR choice in the adaptive strategy

In this proposed algorithm, an adaptive strategy is incorporated to determine a suitable crossover rate at different stages of the evolution process. In order to investigate the behavior of adaptive strategy used in AH-DEa, the evolution of CR values is plotted for the selected functions in Figure 2.1. In this experiment, 10 different crossover rates (CRs) are used. The figure also indicates the evolution counts (i.e., generations up to 90 has been shown in the figure) in X-axis. According to the results shown in Figures 2.1(a)- 2.1(f), one can see that CR selection value changes in different stages of evolution.

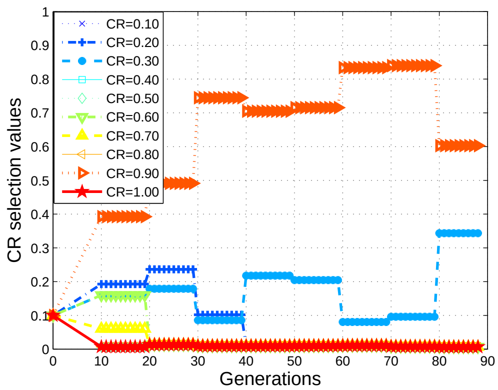
For example, in the 10D C01 function (non-separable), the CR selection value initially set to 0.1 quickly moves up for CR = 0.8 within a few generations. However, for 30D C01, the CR selection value shoots up for 0.7. For 10D and 30D C02 function (non-separable), one can observe that CR selection values of 0.3 and 0.4 are still present as illustrated in Figures 2.1(b) and 2.1(e). C01 is a non-separable function and the best result is achieved in CR = 0.8, which can be adaptively selected by this strategy in Figure 2.1(a). The CR selection probability is low for non-separable functions and high for separable functions. Function C02 is a separable function and once again one can notice the presence of CR = 0.3 or CR = 0.4. This clearly highlights the efficiency of the adaptive strategy to select appropriate crossover rates for various problem classes.



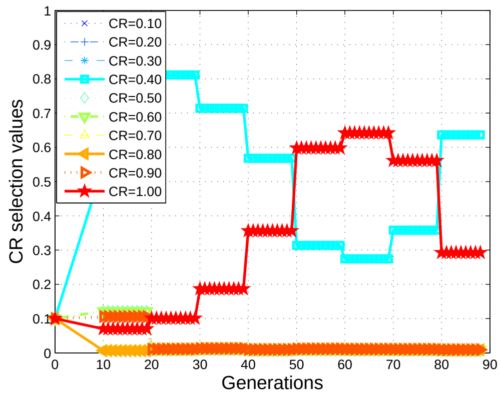
(a) C01 ($D = 10$)



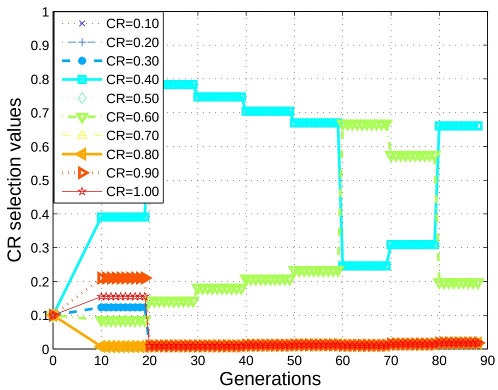
(b) C01 ($D = 30$)



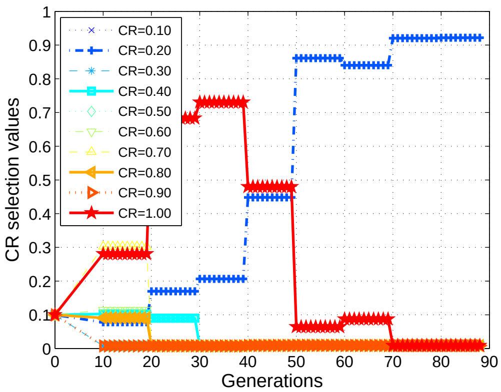
(c) C02 ($D = 10$)



(d) C02 ($D = 30$)



(e) C10 ($D = 10$)



(f) C10 ($D = 30$)

Figure 2.1: Adaptive characteristics of CR on the selected functions (a) function, C01 ($D = 10$) (b) function, C01 ($D = 30$) (c) function, C02 ($D = 10$) (d) function, C02 ($D = 30$) (e) function, C10 ($D = 10$) (f) function, C10($D = 30$).

2.3 DE with Constraint Sequencing

In real life, optimization problems involve a number of constraints arising out of user requirements, physical laws, statutory requirements, resource limitations etc. Such constraints are often evaluated using computationally expensive analysis i.e., solvers relying on finite element methods, computational fluid dynamics, computational electro magnetics etc. Existing optimization approaches spend a lot of computational effort to find a feasible solution i.e., all the constraints corresponding to a solution are evaluated throughout the course of search. An important question is, *why do we spend computational resources to evaluate constraints of a solution, when it has already violated a constraint?*. Assuming that one is only interested in a feasible solution (preferably optimum) at the end of the search process, it is important to investigate the worth of evaluating infeasible solutions i.e., the cost of evaluation versus the knowledge gained to steer the search. Other followup questions include *what is the best sequence to evaluate the constraints?* and *is there a benefit in using different sequence of constraints?*. This algorithm attempts to understand the cost-benefits of *partial evaluation policy* i.e., aborting evaluation of constraints if the solution has already violated one constraint. Above discussion becomes more relevant in the context of optimization problems involving computationally expensive constraint evaluations. This study assumes that the constraints can be evaluated independently of one another.

The proposed algorithm, DE-CS, is aimed to improve the computational efficiency based on a partial evaluation policy at the same time offers the potential to reach different regions of the search space. The pseudo code for the proposed algorithm is presented in the following.

Algorithm description:

A population of N individuals and an archive of $2N$ are initialized. For each individual in the population the parameters regarding the crossover rate (CR) and mutation factor

Algorithm 2.2 DE-CS

SET: Total number of constraints and objective function evaluation NT_{max}, Size of elite archive M, Size of population N, $FeasibleSet = \{\}$, $Evalcount = 0$

$Pop = initialize(); Archive = initialize()$
Assign a random CR and F values from $[0, 1]$ for all the individuals
Distribute individuals to $(p + q + 1)$ *Subpops*
Assign constraint sequences to *Subpops*
Evaluate solutions in the *Subpops*; $Update(Evalcount)$
Rank solutions in each *Subpop* using sequence sort
Migrate feasible solutions from all the *Subpops* to $FeasibleSet$
while $Evalcount \leq NT_{max}$ **do**
  for *i=1:p+q+1* **do**
    if $!isempty(Subpop_i)$ **then**
      for *j=1:size(*$Subpop_i$*)* **do**
        $p_1 = i, p_2 = rand(M), p_3 = rand(N), p_1 \neq p_2 \neq p_3$
        $O = DE_{evolve}(p_1, p_2, p_3)$
        $Evaluate(O); Update(Evalcount)$
        $Temp = Merge(O, Subpop_{i_j})$
        $Rank = SequenceSort(Temp)$
        Select best individual from $Temp$ and replace $Subpop_{i_j}$
      end for
    end if
  end for
  Migrate feasible solutions from all the *Subpops* to $FeasibleSet$
end while

*$Evalcount$ denotes the sum of objective and all individual constraint evaluations

(F) are assigned randomly bounded between [0,1]. The population is divided into (p+q) subpopulations with a prescribed constraint sequence for each subpopulation. In each subpopulation the solutions are ranked using a sequence sort (see Example 1). In order to create a new candidate solution, first parent is selected from the subpopulation itself, the second parent is selected from the entire population and the third parent is selected from the archive.

In the DE evolve process, a binomial crossover [80] has been used with the crossover and mutation parameters (i.e., CR and F) from the first parent (i.e., base parent). The new candidate solution is then evaluated and compared with the base parent via sequence sort. If the candidate solution replaces the parent solution, the crossover and mutation parameters of the base parent are retained by the candidate solution else new parameters are randomly assigned to the base parent and candidate solution is moved to archive. In the event a feasible solution is uncovered in any subpopulation, it migrates to the feasible subpopulation. In the event all the infeasible subpopulations are empty, three parents are chosen from the feasible subpopulation.

2.3.1 Illustrative examples

- **Example 1:** To demonstrate the proposed constraint handling method, a simple two variable optimization problem ($T1$) involving three constraints is presented below. The problem has a feasible region bounded by three linear constraints. Since the focus is on handling constraints, it is interesting to present the trajectory of solutions in different subpopulations.

$$
\begin{aligned}
&\text{Minimize } f_1(\vec{x}) = x_1^2 + x_2^2 + 2x_1x_2 \\
&\text{Subject to} \\
&\qquad g_1(\vec{x}) \equiv x_1 + 2x_2 \geq 0, \\
&\qquad g_2(\vec{x}) \equiv 10x_1 - 8x_2 - 15 \geq 0, \\
&\qquad g_3(\vec{x}) \equiv -10x_1 + 2x_2 - 2 \geq 0,
\end{aligned}
\tag{2.11}
$$

In the proposed approach, three constraint sequences have been considered each of which is assigned to a subpopulation i.e., constraints (g_1, g_2, g_3), (g_2, g_3, g_1) and (g_3, g_1, g_2) are the prescribed sequences for subpopulation, 1, 2 and 3. As an example, let us consider subpopulation 1, containing 4 solutions (S1, S2, S3, S4). The constraint violation matrix would assume a form illustrated in Table 2.8. Since the prescribed sequence for this subpopulation is (g_1, g_2, g_3), the solutions are sequentially sorted to yield (S3, S2, S4, S1) where S3 is deemed the best and S1 the worst.

A small population size of 30 is used to illustrate the behavior. Presented in Figure 2.2 is the trajectory followed by the infeasible solutions, if a CV based

Table 2.8: An example of sequence sorting

Initial order				Final order			
	g1	g2	g3	g1	g2	g3	
$S1$	5	–	–	0	0	1	S3
$S2$	0	3	–	0	3	–	S2
$S3$	0	0	1	2	–	–	S4
$S4$	2	–	–	5	–	–	S1

scheme is employed. A CV based scheme is often used for population based stochastic algorithms. It is interesting to observe that the solutions tend to be located in a small region of the space satisfying g2 and g3 constraints. With such infeasible solutions, the CV based approach will be effective if the optimal feasible solution is close to such infeasible solutions and will be ineffective if the feasible solution is away from such regions. Presented in Figure 2.3, is the same trajectory of the solutions in subpopulations 1, 2 and 3. One can clearly observe that the infeasible solutions in subpopulations clearly are in different regions of the search space and solutions in each subpopulation tend to approach the feasible region from different directions.

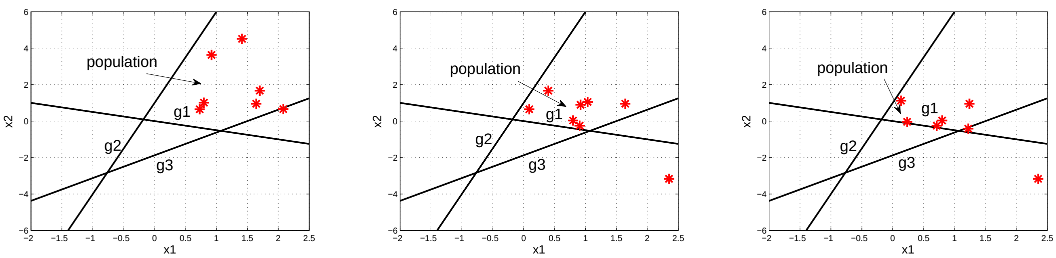
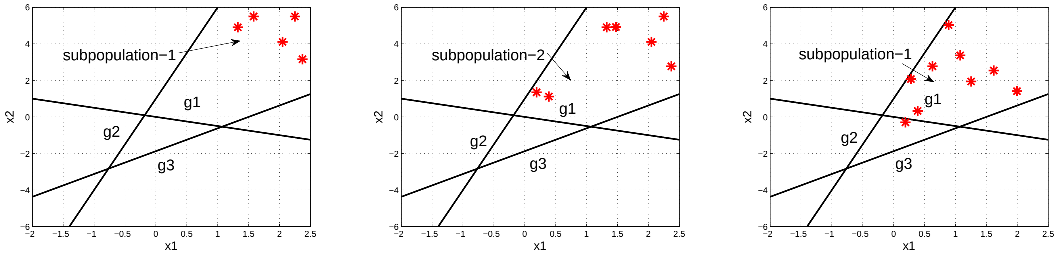
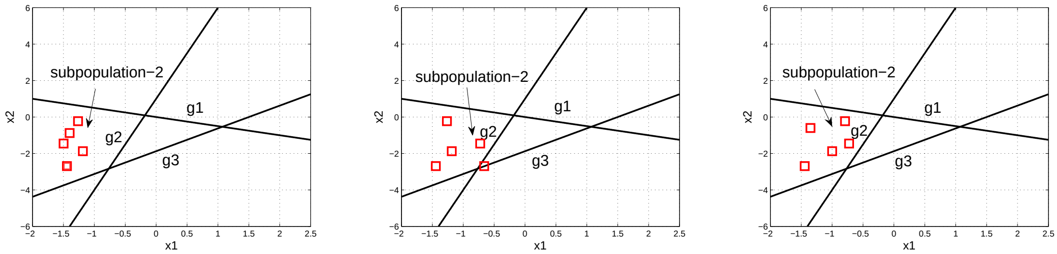


Figure 2.2: Progress plots for test problem $T1$ using DE algorithm with constraint violation (DE-CV) at generation 5, 10 and 20

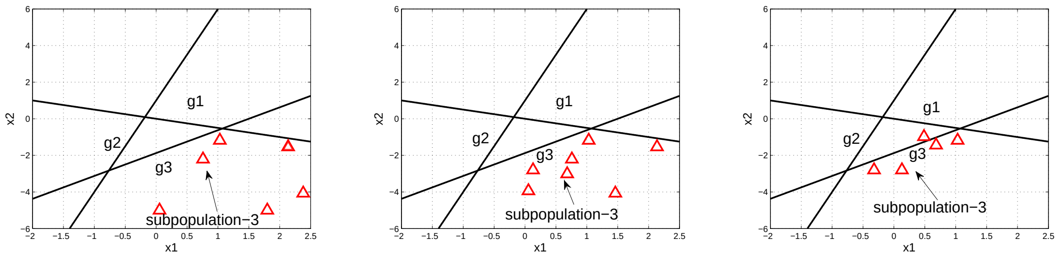
Such an approach is thus likely to locate optimal solutions as they attempt to enter the feasible search space from different directions.



(a) Progress of subpopulation-1



(b) Progress of subpopulation-2



(c) Progress of subpopulation-3

Figure 2.3: Progress plots for test problem $T1$ of multiple subpopulations using DE algorithm with constraint sequencing (DE-CS) at generation 5, 10 and 20

2.3.2 Improvements in DE-CS

The DE-CS discussed above still has the following limitations.

– A predefined set of constraint sequences are used and no new constraint sequence can emerge.

– The search often lost diversity when encountered with constraints that are difficult to satisfy.

In the modified formulation of DE-CS, these limitations have been eliminated. The pseudo code for the modified algorithm is presented below.

Algorithm 2.3 Modified DE-CS

SET: Total number of function evaluation NT_{max}, Size of population N, Crossover rate CR, A Mutation scale factor F, $Evalcount = 0$

1: Initialize the population of individuals and assign a random constraint sequence to each individual;
2: Evaluate the solutions following the above assigned sequence of constraints; $Update(Evalcount)$;
3: **while** $Evalcount \leq NT_{max}$ **do**
4:     **for** i=*1:N* **do**
5:         Select $p_1 = i$ i.e., the i^{th} parent and two other parents p_2 and p_3 randomly s.t. $p_1 \neq p_2 \neq p_3$;
6:         Generate an offspring using recombination;
7:         Evaluate the offspring using the sequence of p_1; $Update(Evalcount)$;
8:         The offspring is compared with solutions in the population for replacement based on fitness;
9:     **end for**
10: **end while**

**Evalcount* denotes the sum of objective and all individual constraint evaluations

Algorithm description:

A population of N individuals is initialized. The variables of i^{th} individual are initialized as follows:

$$x_{j,i} = x_{j,min} + rand_{i,j}[0,1).(x_{j,max} - x_{j,min}) \tag{2.12}$$

where $j = 1, 2, ...D$ is the number of variables; $x_{j,max}$ and $x_{j,min}$ are the upper and the lower bounds of j^{th} variable. For a problem with m constraints, each individual is assigned a random sequence of constraints for evaluation.

Each individual of the population is evaluated using its prescribed constraint sequence. Whenever a constraint is violated, the evaluation is aborted. The term *number of function evaluations* referred here is the sum of the number of evaluated constraints and objective function evaluations. An objective function evaluation is only undertaken if the solution is found feasible i.e., all constraints have been satisfied. In order to generate an offspring solution, the first parent is selected sequentially, the second and third parents are selected randomly from the entire population. In the recombination process, a binomial crossover [16] has been used to generate the offspring solution.

2.3.3 Constraint sequencing

The fitness (ξ) of i^{th} solution consist the objective function value $f_i(\mathbf{x})$, $\mathbf{x} \in \Re^D$ and the value of constraints c_i, where c_i is the constraint violation measure based on the equality and inequality constraints and D represents the number of variables. The equality constraints are transformed into a set of inequalities as $|h_k(\mathbf{x}) - \delta| \leq 0$ (assuming δ is small positive quantity). For every solution in the population, one can compute the number of satisfied constraints (NS) and the amount of violation (V). With the number of constraints satisfied taking precedence over the violation value, a sorting would yield the ranks of the individual solutions.

For example assume a population, containing 4 solutions (S1, S2, S3, S4). The constraint violation matrix would assume a form illustrated in Table 2.9 with S3 identified as the best and S1 the worst.

Table 2.9: Ranking of 4 individuals in the population in presence of 3 constraints

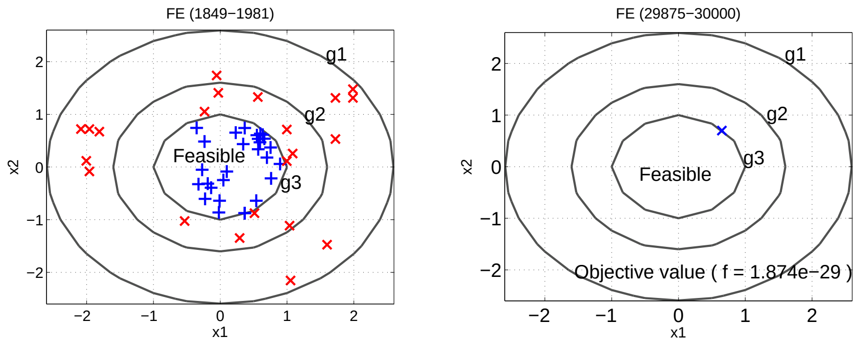
Initial order				NS	V	Final order
$S1$ (g_1, g_2, g_3)	5	–	–	0	5	S3
$S2$ (g_2, g_3, g_1)	0	3	–	1	3	S2
$S3$ (g_1, g_3, g_2)	0	0	1	2	1	S4
$S4$ (g_2, g_1, g_3)	2	–	–	0	2	S1

- **Example 1:** While the details of the algorithm and its components are described in the above section, it is important to identify if there are significant differences in the underlying search process. To observe the underlying search behavior, two example problems have been constructed with 3 constraints. In the first problem, the feasible space of the problem is the feasible space dictated by constraint g_3 while in the next example the feasible space of the problem is the intersection of the feasible spaces of the individual constraints i.e., $(g_1 \cap g_2 \cap g_3)$. For the first example, the constraints and the feasible spaces corresponding to each constraint is depicted in Figure 2.4 with the optimum located at $\mathbf{x}^*$={0.65, 0.70}. The mathematical formulation of the problem ($Example_1$) is presented below:

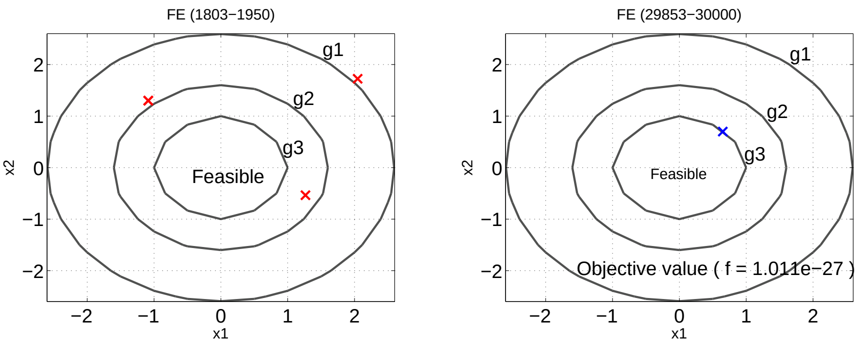
$$\begin{aligned}
&\text{Minimize } f_1(\mathbf{x}) = (x_1 - 0.65)^2 + (x_2 - 0.7)^2 \\
&\text{Subject to} \\
&\qquad g_1(\mathbf{x}) \equiv (x_1/2.6)^2 + (x_2/2.6)^2 - 1 \leq 0, \\
&\qquad g_2(\mathbf{x}) \equiv x_1^2/2.6 + x_2^2/2.6 - 1 \leq 0, \\
&\qquad g_3(\mathbf{x}) \equiv x_1^2 + x_2^2 - 1 \leq 0,
\end{aligned} \tag{2.13}$$

To understand the differences in the dynamics of the search between two strategies, (i.e., constraint sequencing adopting a *partial evaluation policy* (CS) and constraint violation (CV) adopting a *full evaluation policy*), both the strategies have been implemented within the same DE based optimization framework. A population size of 50 individuals were allowed to evolve using a crossover rate of 0.9 and mutation scale factor of 0.5 with the maximum number of function evaluations (i.e., the sum of constraints and objective function) capped at 30000. It is important to highlight that a same starting population has been used for both the strategies for a consistent comparison.

One can see from Figure 2.4, that DE-CS identifies the feasible region of the problem earlier than DE-CV in around 1849-1981 function evaluations. With the known optimum for this example, the *feasibility ratio* (Equation 2.14) and the *distance* (Equation 2.15) from the known optimum is plotted against total number of function evaluations in Figure 2.5. One can clearly observe that DE-CS identifies solutions close to the optimum better than DE-CV and at the same time offers a better feasibility ratio. The above plots are based on the median run



(a) DE-CS



(b) DE-CV

Figure 2.4: Progress plots of test problem *Example*$_1$ using CS and CV in various stages of evolution

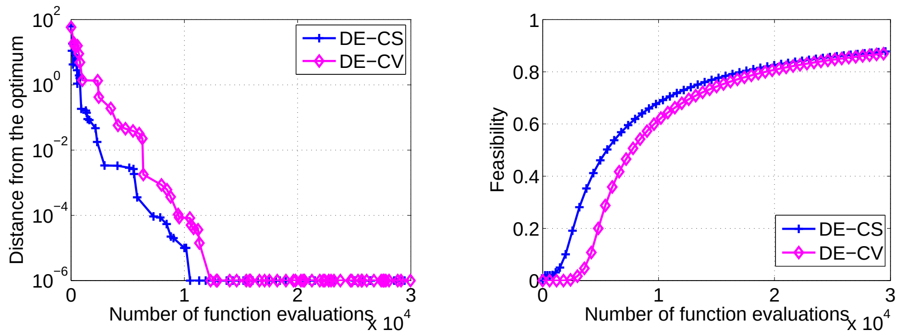
among 50 independent runs.

$$feasibility\ ratio = \frac{\text{Number of feasible solutions}}{\text{Total number of solutions evaluated}} \tag{2.14}$$

$$distance = ||\mathbf{x}_b - \mathbf{x}^*|| \tag{2.15}$$

where $||.||$ is the Euclidean distance between the best solution found so far $\mathbf{x}_b$ and the optimum $\mathbf{x}^*$.

- **Example 2:** In the second example, the feasible space for the problem lies at the intersection of the feasible spaces of the individual constraints i.e., $g_1 \cap g_2 \cap g_3$. The mathematical formulation of the problem ($Example_2$) is presented below:



(a) (b)

Figure 2.5: Progress plots (a) distance of a best feasible solution from the optimum (b) feasibility of the solutions

$$
\begin{aligned}
&\text{Minimize } f_1(\mathbf{x}) = (x_1 - 2.35)^2 + (x_2 + 0.05)^2 \\
&\text{Subject to} \\
&\qquad g_1(\mathbf{x}) \equiv (x_1/2.6 - 0.96)^2 + (x_2/2.6 - 0.96)^2 \\
&\qquad - 1 \leq 0, \\
&\qquad g_2(\mathbf{x}) \equiv (x_1/2.4)^2 + (x_2/2.4)^2 - 1 \leq 0, \\
&\qquad g_3(\mathbf{x}) \equiv (x_1 - 3.1)^2 + (x_2 + 0.6)^2 - 1 \leq 0,
\end{aligned} \tag{2.16}
$$

The problem has a small feasible region with the optimum solution of $\mathbf{x}^* = \{2.35, -0.05\}$. The same set of parameters as listed above has been used for this example. One can observe from Figure 2.6, that DE-CS once again identifies more solutions close the optimum over DE-CV with a better distance and feasibility ratio. It is important to highlight that using multiple sequences DE-CS is able to reach different regions of the feasible space as opposed to DE-CV which tend to have limited diversity of solutions spanning the feasible space. The same behavior of DE-CS can also be observed in this example as depicted in Figure 2.7.

Since the primary contribution of the proposed scheme is to identify feasible solutions faster i.e., with less computational effort, the number of function evaluations required to obtain the first feasible solution and the distance of the first feasible solution from the optima is presented in Table 2.10 and Table 2.11. The results

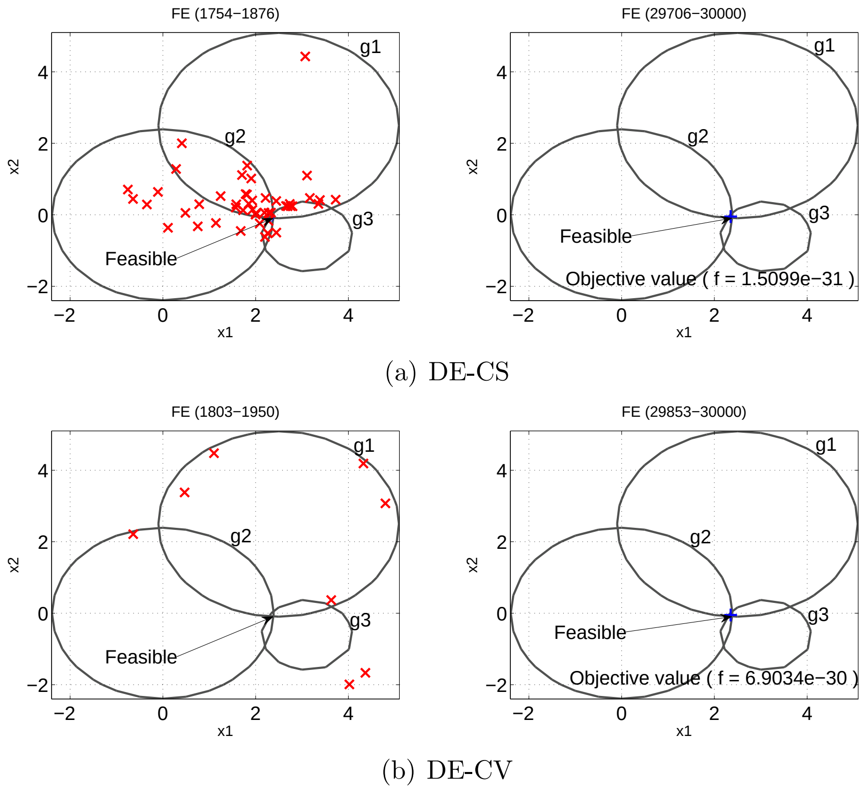


Figure 2.6: Progress plots of test problem $Example_2$ using CS and CV in various stages of evolution

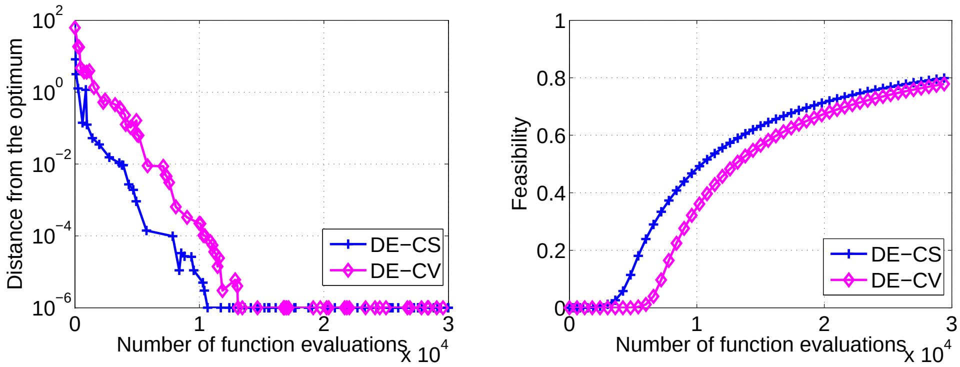


Figure 2.7: The progress of best solution and the feasibility of the solutions (a) distance of a best feasible solution from the optimum (b) feasibility of the solutions

clearly indicate the superiority of DE-CS over DE-CV.

In both the examples, g_3 is the most difficult constraint to satisfy i.e., one which has the smallest feasible area. Since the solutions are ranked using the precedence of number of satisfied constraints, individuals attempting to use the sequence i.e., (g_3,g_2,g_1) or (g_3,g_1,g_2) tend to be less successful. The dominant sequences across

Table 2.10: Comparison of the number of function evaluations required to achieve the first feasible solution using DE-CS and DE-CV

Problem	Strategy	Mean (NFEs)
$Example_1$	DE-CS	661
	DE-CV	1770
$Example_2$	DE-CS	2226
	DE-CV	4148

Table 2.11: Distance of the first feasible solution from the optimum for DE-CS and DE-CV

Problem	Strategy	Mean (Distance)
$Example_1$	DE-CS	0.9805
	DE-CV	0.9837
$Example_2$	DE-CS	0.0592
	DE-CV	0.0600

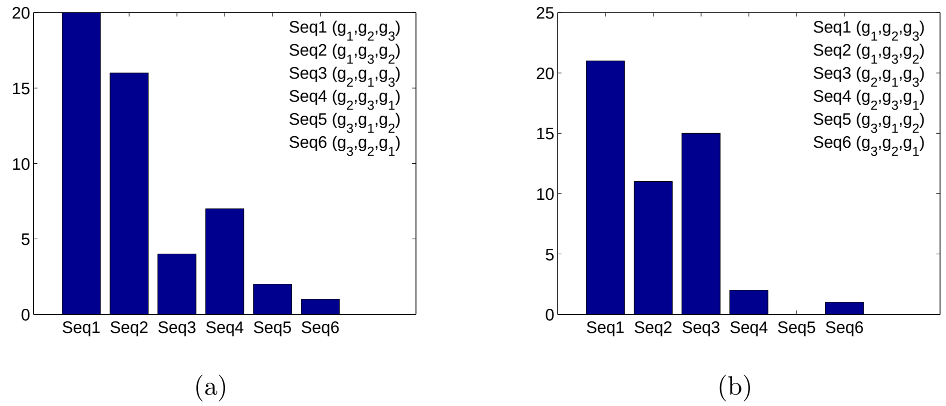


Figure 2.8: The dominant constraint sequences across 50 runs for (a) $Example_1$ and (b) $Example_2$

50 runs are shown in Figure 2.8.

2.3.4 Numerical experiments

The above section illustrated the principles of constraint sequencing and partial evaluation. In this section the performance of the algorithm is objectively analysed using

11 test problems from *CEC-2006* [13] and *CEC-2010* [14] benchmarks involving only inequality constraints. The selected test problems as listed in (Table 2.12) involve quadratic, nonlinear, cubic, polynomial, separable and non-separable objective functions with a number of inequality constraints. The results obtained by using stochastic ranking (SR) [81], self-adaptive penalty (SP) [82], superiority of feasibility (SF) [50] and epsilon constraint (EC) [78] schemes are presented using the same framework based on DE. Results based on performance profiles are included for a more objective comparison.

Table 2.12: Summary of test problems

	Prob.	D	f type	No. of Ineq.	Feasibility (ρ)
CEC-2006	G01	13	quadratic	9	0.0111%
	G02	20	nonlinear	2	99.9710%
	G04	5	quadratic	6	52.1230%
	G06	2	cubic	2	0.0066%
	G07	10	quadratic	8	0.0003%
	G08	2	nonlinear	2	0.8560%
	G09	7	polynomial	4	0.5121%
	G10	8	linear	6	0.0010%
	G12	3	quadratic	1	4.7713%
	G18	9	quadratic	13	0.0000%
	G24	2	linear	2	79.6556%
CEC-2010	C01	10/30	Non Separable	2	99.7689%
	C08	10/30	Non Separable	1	37.9512%
	C13	10/30	Separable	3	0.0000%
	C14	10/30	Non Separable	3	0.3112 %
	C15	10/30	Non Separable	3	0.3210 %

A population size of 50 is used for all the problems and the results are computed based on 30 independent runs. A fixed value of $CR = 0.9$ and $F = 0.5$ [16] have been set for all the cases (i.e., SR [81], SF [50], SP [82], EC [78]) and the maximum number of generations is set to 4800 resulting the number of function evolutions (i.e., *NFEs*) of $4800*(N*m)$, where N is the size of the population and m is the number of constraints.

- **Comparison with other constraint handling methods:** In this experiment, one can observe how quickly a feasible solution appears in the population and how close is it to the known optimum solution.

1. **Comparison on function evaluations:** The average number of function evaluations ($NFEs$) required to identify the first feasible solution and the distance of the same from the known optimum is tabulated in Table B.1. One can observe that in 8 out of 11 problems, the average number of function evaluations ($NFEs$) required by DE-CS is lower than others. A performance profile is computed for a more objective comparison between the strategies in Figure 2.9.

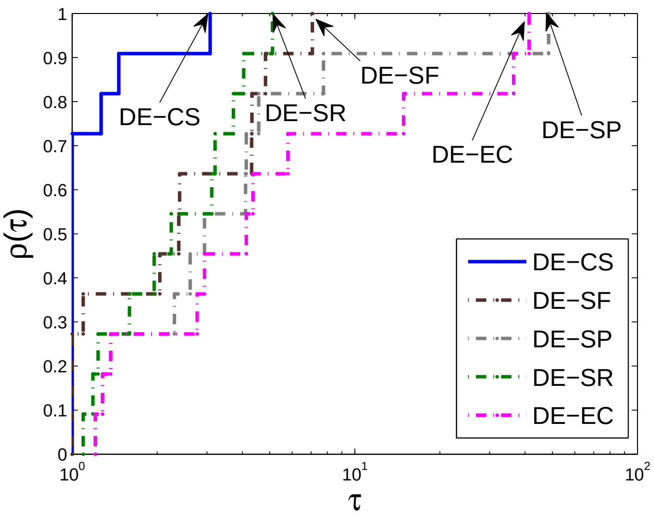


Figure 2.9: Performance profiles of DE-CS and others

The $NFEs$ required to achieve a feasible solution is computed for all the methods. The results are then normalized by the minimum values of $NFEs$ as follows

$$r_{p,s} = \frac{NFEs_{p,s}}{min\{NFEs_{p,s} : s \in S\}} \tag{2.17}$$

where p is the number of problems and s is the number of solvers. The performance of solver s on any given problem is calculated as follows [83]

$$\rho_s(\tau) = \frac{1}{n_p} size\{p \in P : r_{p,s} \leq \tau\} \tag{2.18}$$

The results clearly indicate the superiority of DE-CS over other strategies in terms of obtaining the first feasible solution.

2. **Comparison on the distance measures from the optimum:** The above section clearly highlighted the benefits of DE-CS i.e., its ability to obtain feasible solutions faster. While obtaining a feasible solution first does not guarantee that it would be close to optimum, such solutions are useful when one is interested in finding a feasible solution of a problem. Tables 2.13 and 2.14 present the distance measure of the first feasible solution from the optimum value as obtained using different constraint handling schemes. In terms of the best result, DE-CS is better in 7 problems out of 11 and 8 out of 11 based on the mean result. It is important to highlight that in some examples, DE-CV delivers the first feasible solution close to optimum.

3. **Comparison on time complexity:** The computational complexities of the strategies are analysed next. In this comparison, performance of 5 solvers is studied using 11 problems. Computational time is used as the performance measure. For each problem p and solver s, $t_{p,s}$ = average computing time required to reach a feasible solution is computed. Table B.2 shows the value of $t_{p,s}$ for all the solvers. Matlab R2011b was used to implement the algorithms with the system configuration as follows: Intel(R) Core(TM)2 Duo 3.00 GHZ, 3.49 GB of RAM, Windows XP Professional Version 2002.

   The performance on problem p by solver s is defined by the performance ratio calculated as follows:

$$r_{p,s} = \frac{t_{p,s}}{min\{t_{p,s} : s \in S\}} \tag{2.19}$$

   Figure 2.10 shows the value of $\rho(\tau)$ for $r_{p,s} \leq \tau$ of the normalized performance ratio. One can observe from the figure that DE-CS outperforms other strategies.

4. **Comparison with other state-of-the-art algorithms on *CEC-2006*:**

   The results from the proposed algorithm is further investigated on the selected benchmark problems and compared against other state-of-the-art algorithms. The best entries in CEC-2006 competition are selected here for a comparison. Each

Table 2.13: Distance of the first feasible solution from the optimum

Prob.	Algorithms	Best	Mean	Std
G01	DE-CS	**2.999**	**4.283**	0.590
	DE-SF	4.759	5.435	0.242
	DE-SP	4.581	5.408	0.322
	DE-SR	4.888	5.554	0.202
	DE-EC	4.962	5.506	**0.201**
G02	DE-CS	**0.043**	**0.050**	0.020
	DE-SF	0.051	0.055	0.012
	DE-SP	0.044	0.056	0.012
	DE-SR	0.057	0.058	**0.011**
	DE-EC	0.051	0.052	0.012
G04	DE-CS	5.543	**5.543**	**0.000**
	DE-SF	6.994	14.592	4.109
	DE-SP	5.770	12.706	5.338
	DE-SR	6.282	14.147	4.232
	DE-EC	**4.324**	13.760	5.254
G06	DE-CS	1.430	4.365	**1.241**
	DE-SF	0.801	4.563	2.235
	DE-SP	0.451	3.790	1.712
	DE-SR	0.291	7.119	5.764
	DE-EC	**0.066**	**4.005**	3.490
G07	DE-CS	**4.039**	**5.613**	**0.108**
	DE-SF	9.465	16.508	3.892
	DE-SP	7.532	12.968	3.204
	DE-SR	4.508	8.672	2.601
	DE-EC	4.127	9.756	2.538
G08	DE-CS	**0.037**	**0.317**	**0.184**
	DE-SF	0.042	0.453	0.230
	DE-SP	0.116	5.352	6.500
	DE-SR	0.039	0.359	0.255
	DE-EC	0.102	0.506	0.225

of these algorithms is specialized in different components of an optimization algorithm. Some algorithms are specialized in constraint-handling strategy such as ϵ-DE [78] and DMS-PSO [84], some are in mutation strategy such as MDE [51], PCX [85] and some are in adaptive strategy such as jDE [54], SaDE [80], and GDE [86]. Since one is interested to observe the performance of constraint-handling schemes and adaptive strategies in terms of $TotalFEs$, the results of DE-CS are compared against the results of those algorithms in CEC-2006 benchmark problems with the known optimum.

Table 2.14: Distance of the first feasible solution from the optimum

Prob.	Algorithms	Best	Mean	Std
G09	DE-CS	**3.667**	**5.514**	**0.413**
	DE-SF	6.644	12.533	2.580
	DE-SP	6.063	11.248	2.236
	DE-SR	5.143	10.530	2.064
	DE-EC	6.946	11.759	2.583
G10	DE-CS	**5.633**	**5.633**	**0.000**
	DE-SF	5.977	6.213	2.587
	DE-SP	6.123	6.542	1.900
	DE-SR	5.926	6.123	1.440
	DE-EC	5.868	5.988	1.920
G12	DE-CS	**0.532**	**0.548**	0.012
	DE-SF	0.592	0.551	0.019
	DE-SP	0.595	0.611	0.112
	DE-SR	0.575	0.599	**0.011**
	DE-EC	0.543	0.562	0.013
G18	DE-CS	0.963	1.367	0.224
	DE-SF	0.777	2.131	0.479
	DE-SP	0.809	1.991	0.505
	DE-SR	1.177	2.117	0.513
	DE-EC	**0.206**	**0.878**	**0.437**
G24	DE-CS	0.181	1.420	0.445
	DE-SF	**0.018**	**0.676**	0.433
	DE-SP	0.313	0.821	**0.297**
	DE-SR	0.206	0.878	0.437
	DE-EC	0.189	0.710	0.394

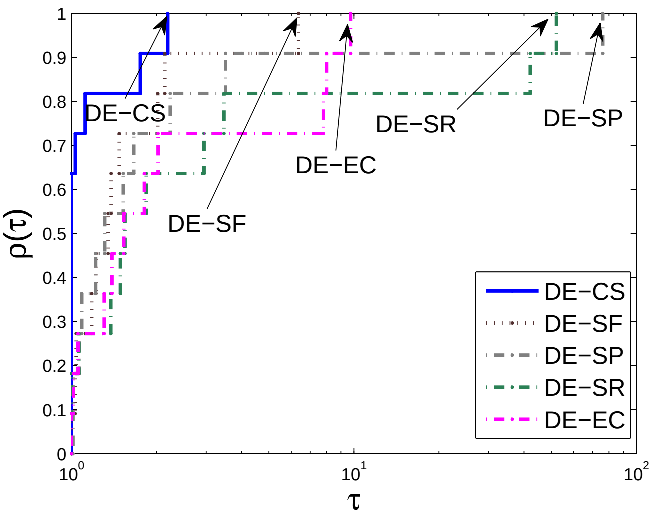


Figure 2.10: Performance profiles based on computational time

Tables B.3 and B.4 show the results achieved by each algorithm. Among 7 other algorithms, DE-CS shows better in 8 problems out of 11 problems when comparing with the best results. In median results, DE-CS shows better in 5 problems out of 11. The success performance of DE-CS shows significantly better in 4 problems. Figure 2.11, the performance profile of success performance is shown for all problems. One can see from the figure that DE-CS requires less number of total function evaluations to solve the problems.

Figure 2.12 presents the success rate of performance profile. One can see from the figure that DE-CS is marginally better with $\epsilon-$DE and way better than others.

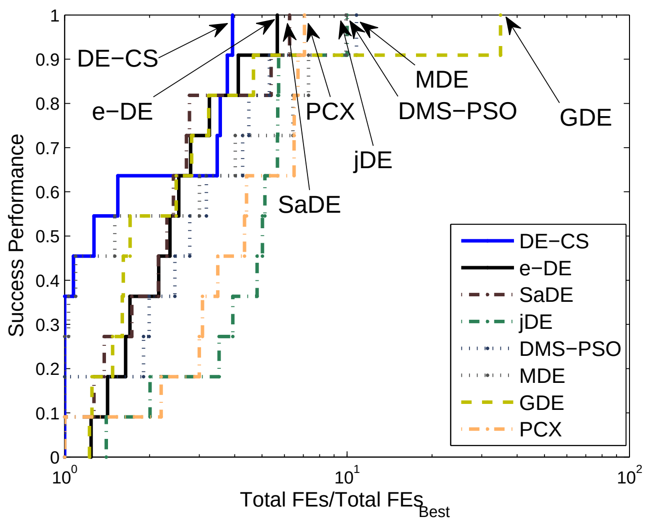


Figure 2.11: Performance profiles comparing DE-CS with other state-of-the-art algorithms in CEC-2006 benchmarks

5. **Performance on CEC-2010 problems:** The selected CEC-2010 test problems are mostly non-separable with a single exception. These problems contain a small number of constraints unlike the CEC-2006 test problems. The non-separability of the problem poses difficulty for the optimizer. In these problems, DE-CS is able to show competitive results with other recent algorithms.

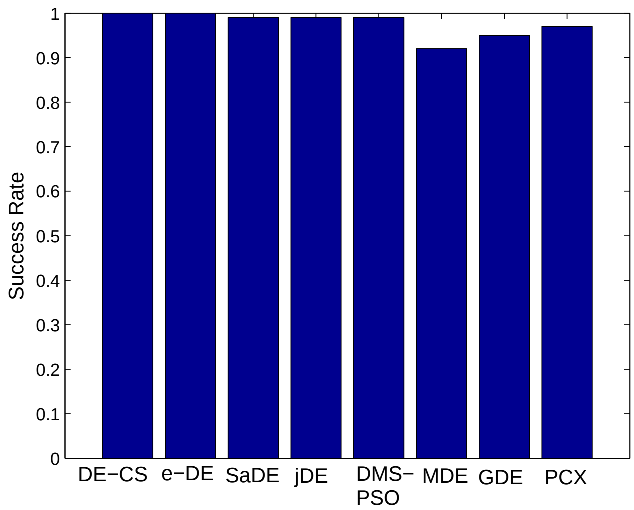


Figure 2.12: Success rate performance profiles (normalized) comparing other state-of-the-arts algorithms

2.4 MOEA/D with ACH

It is evident from a number of previous studies [72, 73, 74, 75] that, there is always a need for adaptive constraint handling method as its performance is largely dependent on its parameters. Here, an adaptive constraint handling approach is introduced that can be used within the class of evolutionary multi-objective optimization (EMO) algorithms. The proposed constraint handling approach is presented within the framework of one of the most successful algorithms i.e., multi-objective evolutionary algorithm based on decomposition (MOEA/D) [9]. The constraint handling mechanism adaptively decides on the *violation threshold* for comparison. The violation threshold is based on the type of constraints, size of the feasible space and the search outcome. Such a process intrinsically treats constraint violation and objective function values separately and adds a selection pressure, wherein infeasible solutions with violations less than the identified threshold are considered at par with feasible solutions. The pseudo code for the proposed algorithm is presented in the following.

Algorithm 2.4 MOEA/D-ACH

Input: Gen_{max} maximum number of generations, λ uniformly distributed weight vectors

1: Generate the uniform weight vectors and identify the neighborhood $B(i)$ of T (see Table 2.15) closest subproblems
2: Evaluate the initial population and update the ideal reference point z_i^* i.e., $z_i^* = min\{f_i(x)|x \in \Re^n\}$.
3: **while** $(gen \leq Gen_{max})$ **do**
4:   **for** *i=1:*λ **do**
5:     Create a mating pool P of two parents by selecting the parents from the neighborhood $B(i)$ with a probability of δ; otherwise, from a randomly shuffled population list with the probability of $(1-\delta)$.
6:     Select three parents for the recombination, one from the randomly shuffled population list and two from the mating pool P and generate offspring y with a DE operator along with a polynomial mutation with the probability of ρ_m.
7:     Evaluate the offspring and calculate the CV using equation 2.22 and use local search with a probability of γ.
8:     Update the ideal reference point z^*.
9:     Update the current individual with the generated offspring if it satisfies equation 2.28.
10:   **end for**
11: **end while**

Algorithm description:

MOEA/D with Adaptive Constraint Handling (ACH) method utilizes a decomposition approach to deal with subproblems and employs a neighborhood mating strategy to accelerate the rate of search. Uniformly distributed weight vectors (λ) are used; where, each subproblem is associated with a weight vector λ_i. The Euclidean distance among the weight vectors are used to constitute the neighborhood $B(i)$ where, $i = \{i_1, i_2, i_3.....i_T\}$ is the set of T closest subproblems. For the recombination process, the mating pool P is selected either of the two ways i.e., (a) one parent is selected from a randomly shuffled population list and two parents from the neighborhood $B(i)$ and (b) all three parents are selected from the randomly shuffled population list. The probability of selecting the former one is δ and the other one is $(1-\delta)$. The pseudocode of the modified MOEA/D to deal with constraints is presented in Algorithm 2.4.

The DE operator used in step 6 integrates two strategies i.e., a binomial crossover and a polynomial mutation for the recombination process [9]. The offspring (y) is generated with a binomial crossover as follows:

$$y_k = \begin{cases} x^{p_1} + F \times (x^{p_2} - x^{p_3}) & \text{if } rand[0,1) \leq CR, \\ x^{p_1}, & \text{otherwise} \end{cases} \tag{2.20}$$

where CR is the crossover rate between 0.1 to 1.0 [87]. While a single value of CR is commonly used in previous studies, a set of CRs (see table 2.15) is used in this study where the choice of the particular CR is decided based on the success ratio [88].

The generated offspring from the recombination process is then evaluated and the constraint violation ($CV^{'}$) is calculated by Equation 2.22. To repair the infeasible solution having the constraint violation greater than zero, a local search is used with a probability of γ (see Table 2.15). The newly generated offspring replaces the current individual of the population list, if it satisfies the Equation 2.28.

2.4.1 Adaptive Constraint Handling (ACH)

The proposed constraint handling and modified selection process have been incorporated within MOEA/D. The details of the proposed schemes are discussed below:

- **Constraint Violation (CV) measure:** Constraint violation (CV) is a scalar value derived through the summation of total violations of the inequality and equality constraints. A CV of 0 indicates the solution is feasible. Mathematically, CV is expressed as:

$$CV = \sum_{i=1}^{p} max(g_i, 0) + \sum_{i=1}^{q} max(|h_i - \epsilon|, 0) \tag{2.21}$$

where g_i and h_i are the inequality and equality constraints. In this study, apart from the violation value, the number of violated constraints is also considered. The modified formulation of the constraint violation measure is given below:

$$CV' = m_1 * \sum_{i=1}^{p} max(g_i, 0) + m_2 * \sum_{i=1}^{q} max(|h_i - \epsilon|, 0) \tag{2.22}$$

where m_1 and m_2 are the number of inequality and equality constraints actively present in the solution.

To demonstrate the working of the procedure, a two variable bi-objective constrained optimization problem with three inequality and one equality constraint is designed (Equation 2.23).

$$\begin{aligned}
&\text{Minimize } [f_1(\vec{x}), f_2(\vec{x})] \\
&\text{Subject to} \\
&\qquad g_1(\vec{x}) \equiv x_1 - 3.x_2 - 2 \leq 0, \\
&\qquad g_2(\vec{x}) \equiv x_1 + x_2 - 6 \leq 0, \\
&\qquad g_3(\vec{x}) \equiv x_2 - x_1 + 2 \leq 0, \\
&\text{Where,} \\
&\qquad f_1(\vec{x}) = -(25.(x_1 - 2)^2 + (x_2 - 2)^2), \\
&\qquad f_2(\vec{x}) = x_1^2 + x_2^2
\end{aligned} \tag{2.23}$$

A graphical representation of the feasible region and the constraint boundary for the problem (Equation 2.23) is shown in Figure 2.13. Because of a small feasible region and the presence of an equality constraint, the solutions in the initial population originated in the infeasible region with a constraint violation value greater than zero. A progressive plot of the population is presented in Figure 2.14 which indicates a decrease in the number of infeasible solutions over the generations. A comparison of such a progress plot for 30 independent runs for both the constraint violation measures (i.e., Equation 2.21 and Equation 2.22) are presented in Figure 2.14. One can observe from Figure 2.15 (median run) that using the modified measure of constraint violation, the individuals in the population identifies feasible solutions faster.

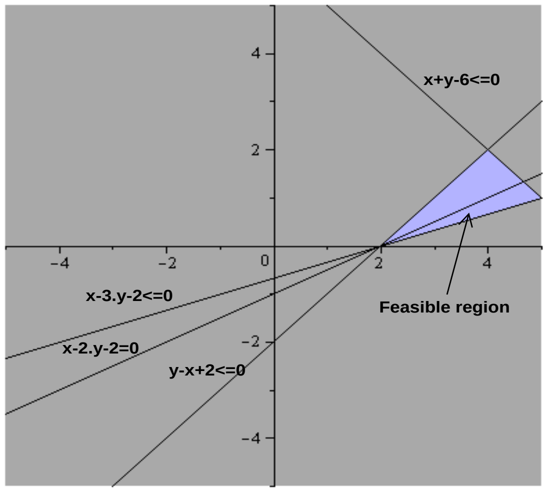


Figure 2.13: A graphical representation of the formulated problem.

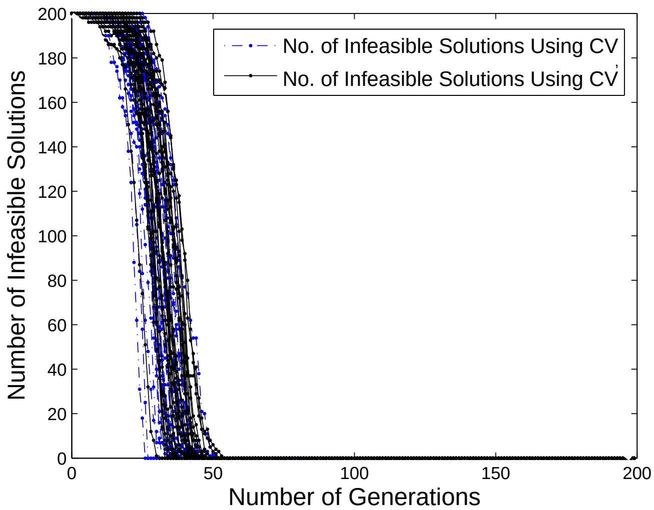


Figure 2.14: Progressive decrease of number of infeasible solutions over the generations using Equation 2.21 and 2.22 for 30 independent runs.

- **Feasibility ratio:** The feasibility ratio of a population refers to the ratio of the number of feasible solutions in the population to the number of solutions in the population. The mathematical expression of Feasibility Ratio (FR) is stated earlier in Equation 2.14.

- **Violation threshold:** The violation threshold is an allowable violation level. The infeasible solutions with violations less than the violation threshold are considered at par with feasible solutions and are compared based on objective

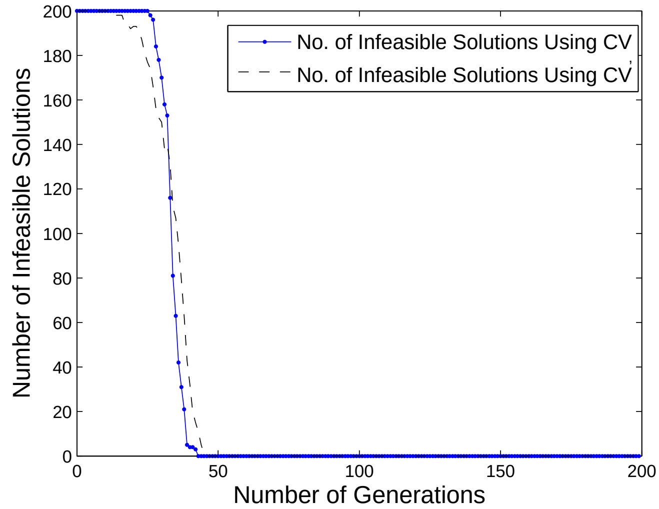


Figure 2.15: Progressive decrease of number of infeasible solutions over the generations using Equation 2.21 and 2.22 in a median run.

function value. The violation threshold is an adaptive measure and is computed as follows:

$$CV'_{mean} = \frac{1}{M}\sum_{j=1}^{M}(CV'_j) \tag{2.24}$$

$$\text{Allowable violation}(\tau) = CV'_{mean} * FR \tag{2.25}$$

where M refers the size of the population .

- **Modified selection procedure:** The fitness of a solution is represented as follows:

$$fitness(\xi) = \begin{cases} \varphi^{te}(\mathbf{x}), & \mathbf{x} \in \Re^n \\ CV', & \end{cases} \tag{2.26}$$

where φ^{te} is the objective value and CV' is the constraint violation.

Infeasible solutions with violations less than the violation threshold are compared based on their objective function values. Feasible solutions are compared against each other based on objective function value while infeasible solutions with violations more than the violation threshold are compared based on their constraint violation values. The objective function value is calculated using the weighted Tchebycheff approach [9], which is formulated as follows:

$$\varphi^{te}(x|\lambda, z^*) = max_{i=1}^{m}\{\lambda_i|f_i(x) - z_i^*|\} \tag{2.27}$$

where λ_i is the uniform weight vector and z_i^* is a ideal reference point i.e., $z_i^* = min\{f_i(x)|x \in \Re^n\}$ for a minimization problem [9]. A lexicographic ordering is used as a measure to compare solutions $\{\varphi_1^{te}, CV_1^{'}\}$ and $\{\varphi_2^{te}, CV_2^{'}\}$ as follows:

$$(\varphi_1, CV_1^{'}) <_\tau (\varphi_2, CV_2^{'}) \Leftrightarrow \begin{cases} \varphi_1 < \varphi_2, & \text{if} \quad CV_1^{'}, CV_2^{'} < \tau \\ \varphi_1 < \varphi_2, & \text{if} \quad CV_1^{'} = CV_2^{'} \\ CV_1^{'} < CV_2^{'}, & \text{otherwise} \end{cases} \tag{2.28}$$

2.4.2 Gradient local search

To further accelerate the rate of convergence, a gradient based local search is used to repair the infeasible solutions periodically. The probability of invoking the repair method is set to 5% as adopted in [89].

2.4.3 Performance evaluation

The performance of the algorithm is evaluated using 10 widely used benchmark problems and a real-world constrained optimization problem. The parameters of the algorithm are listed in Table 2.15. Features of the selected test problems are listed in Table 2.16. The experiments are conducted using a population size of 200 for all the problems.

The problems (SRN, OSY, CTP6, Toysub) have continuous Pareto-front while the remaining ones (CTP2-CTP5, CTP7, CTP8 and TNK) have disjoint Pareto-front. The feasibility ratio for all these problems indicates the level of difficulty to achieve the

Table 2.15: Parameter settings

Population size (M)	200
Neighborhood size(T)	$0.1M$
Crossover rate(CR)	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}
Mutation rate (F)	0.5
Probability of selecting the parents from the neighborhood(δ)	0.95
Polynomial mutation rate(ρ_m)	$1/D$
Probability of using the local search(γ)	0.05
FE_{max}	40000

Table 2.16: Summary of test problems

Prob.	Obj./ D	Search Range	FR	Constraints	
				Ineq.	Eq.
CTP2	2/2	$[0,1]^D$	78.65%	1	0
CTP3	2/2	$[0,1]^D$	76.85%	1	0
CTP4	2/2	$[0,1]^D$	58.17%	1	0
CTP5	2/2	$[0,1]^D$	77.54%	1	0
CTP6	2/2	$[0,1]^D$	0.40%	1	0
CTP7	2/2	$[0,1]^D$	36.68%	1	0
CTP8	2/2	$[0,1]^D$	17.83%	1	0
SRN	2/2	$[-20,20]^D$	16.18%	2	0
TNK	2/2	$[0,\pi]^D$	5.09%	2	0
OSY	2/6	$[0,10]^2$[1,5][0,6] [1,5][0,10]	3.25%	6	0
Toysub	2/8	$[0,300]^4$[35,50] [80,150][1.5,3][45,100]	0.0003%	3	0

Pareto-front. This feasibility ratio has been determined experimentally by calculating the percentage of feasible solutions among 1000000 randomly generated individuals [90].

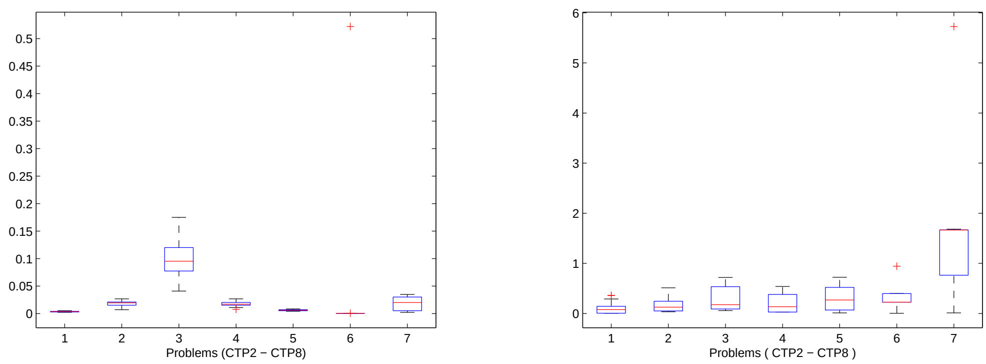
- **Comparative study for the problems (CTP2-CTP8, SRN, TNK, and OSY):** In this study, inverse generational distance (IGD)[8] is used to compute the performance of the proposed algorithm with NSGA-II [91] using *feasibility first* [50] constraint handling mechanism. The true Pareto-front for all the selected test problems have been determined by running the well-known algorithm NSGA-II up to 1000 generations. For a fair comparison, same parameter setup has been used for both the algorithms. The *convergence* metric indicate the quality of the solutions achieved after a certain generation. Thus, a lower value indicates a better performance.

  The results of all the problems are shown in Table 2.17. For problems CTP2-CTP8 the mean and standard deviation for MOEA/D-ACH is better than NSGA-II.

Table 2.17: Comparison on *Convergence* metric

Problem	MOEA/D-ACH		NSGA-II	
	Mean	Std	Mean	Std
CTP2	**0.0035**	**0.0007**	0.0948	0.1071
CTP3	**0.0178**	**0.0052**	0.1647	0.1217
CTP4	**0.0995**	**0.0304**	0.3063	0.2583
CTP5	**0.0178**	**0.0046**	0.2044	0.1864
CTP6	**0.0062**	**0.0012**	0.3033	0.2616
CTP7	**0.0180**	**0.0952**	0.2705	0.2169
CTP8	**0.0180**	**0.0129**	1.4573	0.9874
SRN	**0.4876**	**0.0109**	0.5546	0.0136
TNK	**0.1683**	**0.3750**	4.0592	0.5827
OSY	0.6705	0.9355	**0.0025**	**0.0002**

The Box plots indicate consistent performance across 30 independent runs (Figure 2.16).



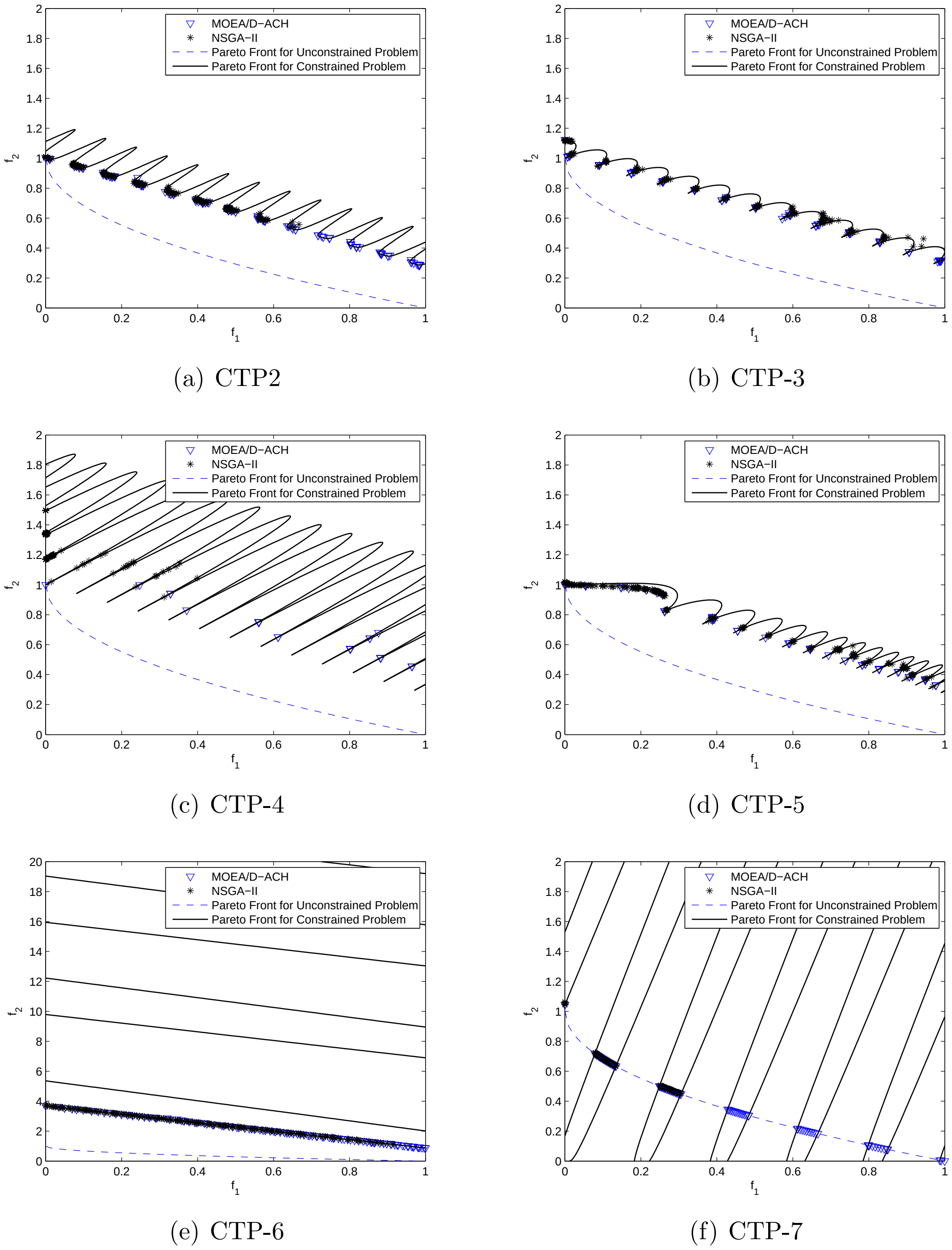
(a) MOEA/D-ACH (b) NSGA-II

Figure 2.16: Box plots of *convergence metric* using 30 independent runs.

In Figures 2.17 and 2.18 for problems CTP2-CTP8, it can be seen that the Pareto-front achieved by NSGA-II is partial in 200 generations whereas MOEA/D-ACH is able to achieve the complete Pareto-front.

The algorithm is then used to solve SRN, TNK and OSY test problems (Figure 2.18). The proposed MOEA/D-ACH performs better than NSGA-II on SRN, TNK and has shown competitive performance for OSY.

The algorithm has also been used to solve a real-world constraint optimization



(a) CTP2 (b) CTP-3

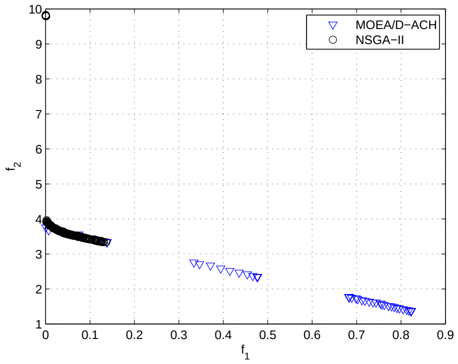
(c) CTP-4 (d) CTP-5

(e) CTP-6 (f) CTP-7

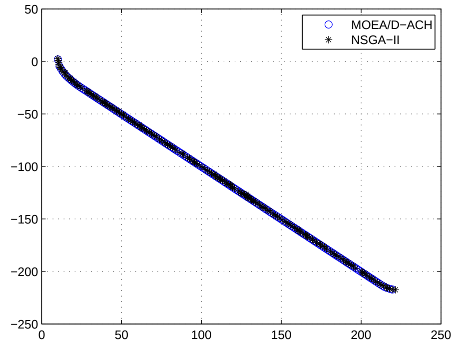
Figure 2.17: Final non-dominated fronts plots for the problems CTP2-CTP7 from MOEA/D-ACH and NSGA-II of median run.

problem referred as the toy submarine design problem [92] by formulating a bi-objective problem targeted to minimize the drag and maximize the lever arm. The problem has described in the following subsection.

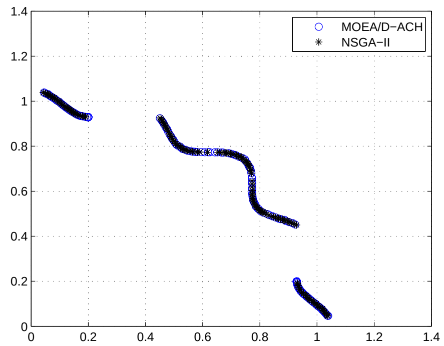
- **Toy submarine design problem (Toysub):** The toy submarine design is a eight-variable constrained optimization problem, the single objective version of



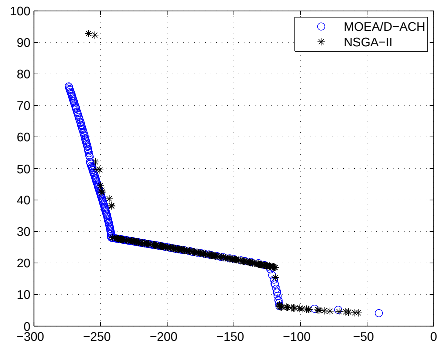
(a) CTP-8



(b) SRN



(c) TNK



(d) OSY

Figure 2.18: Final non-dominated fronts for the problems CTP8, SRN, TNK and OSY from MOEA/D-ACH and NSGA-II of median run.

which is presented by Alam *et al.* [92]. A bi-objective formulation of the problem has been studied in the present work which seeks to minimize the drag (D) and maximize one of the lever arms (LA) subject to the constraints on length (L) and weight (W) of the vehicle, and centre of gravity (CG) and centre of buoyancy (CB) separation. The first (LA_1) and second (LA_2) lever arms are the longitudinal distances of the propellers from the centre of buoyancy respectively. The higher value of lever arm produces higher pitching and turning moments that lead to better diving and heading changes. The lower the value of CG/CB separation (S), the closer the position of the CG and CB that leads to better stability of the vehicle. Constraints on overall length and weight of the vehicle are also important to meet the basic design requirements. Minimization of drag is important because minimum drag leads to least power consumption for propulsion, and corresponding savings in the operating costs.

The design variables of the problem as illustrated in Figure 2.19 are: the position of the internal components along the Z axis, such as position of the controller (Z_C), position of the propeller unit for pitch (Z_V) and yaw (Z_L) movements, position of the battery compartment (Z_B), smaller diameter (d_t) and length (l_t) of the tail and shape variation coefficient (n_n) and length (l_n) of the nose. The optimization problem is stated in Equation 2.30.

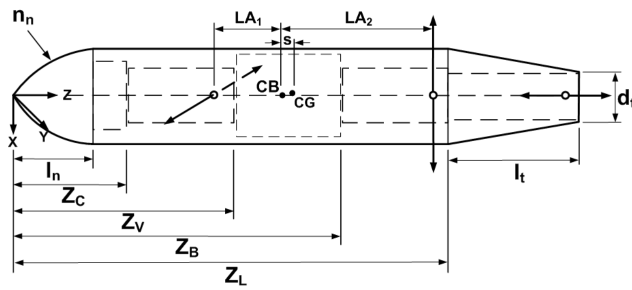


Figure 2.19: Illumination of the constraints and design variables for problem formulation of the toy submarine design problem

$$
\begin{aligned}
&\text{Minimize:}\\
&f(1) = D\\
&f(2) = -LA; \quad where\ LA = min(LA_1, LA_2)\\
&\\
&\text{Subject to:}\\
&g(1) = L \leq 400\ mm; \quad g(2) = W \leq 450\ g\\
&g(3) = S \leq 4\ mm
\end{aligned}
\tag{2.29}
$$

$$
\begin{aligned}
&\text{Variable bounds:}\\
&0 \leq Z_C \leq 300\ mm; \quad 0 \leq Z_V \leq 300\ mm\\
&0 \leq Z_B \leq 300\ mm; \quad 0 \leq Z_L \leq 300\ mm\\
&35 \leq d_t \leq 50\ mm; \quad 80 \leq l_t \leq 150\ mm\\
&1.5 \leq n_n \leq 3; \quad 45 \leq l_n \leq 100\ mm
\end{aligned}
\tag{2.30}
$$

To maximize one of the lever arms as formulated in Equation 2.30, a negative sign is placed, thereby formulating both the objectives as a minimization problem.

- **Multi-objective optimization results:** A population size of 200 solutions was allowed to evolve over 300 generations. The same parameters were used for NSGA-II. Figure 2.20 shows the final Pareto-front obtained in the median run. The comparison is based on *hypervolume* [93] which measures the volume (or area) dominated by the given set of solutions given a reference point. Larger this dominated area, better is the set of solutions.

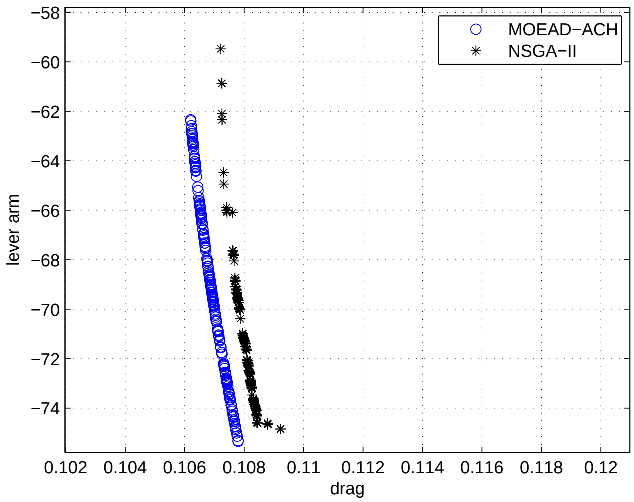


Figure 2.20: Final non-dominated fronts for median run of the toy submarine design problem using MOEA/D-ACH and NSGA-II

In this study, a reference point of [0.1319, -0.3242] is used, which is calculated by taking the maximum value of each objective across all the runs. It is seen from the results reported in Table 2.18 that MOEA/D-ACH is able to achieve marginally higher (nearly same) values of hypervolume as compared to NSGA-II.

Table 2.18: Comparison on *hypervolume* metric

Prob.	MOEA/D-ACH			NSGA-II		
	Best	Mean	Std	Best	Mean	Std
Toysub	**4.6812**	**1.8878**	**0.8931**	4.6641	1.8679	1.0774

- **Results of optimum toy submarine design:** Three solutions (i.e., considering the lowest drag value, lowest lever arm and the intermediate solution considering both) obtained from the best run of MOEA/D-ACH and NSGA-II are presented. Figure 2.21 shows the results obtained from the best runs.

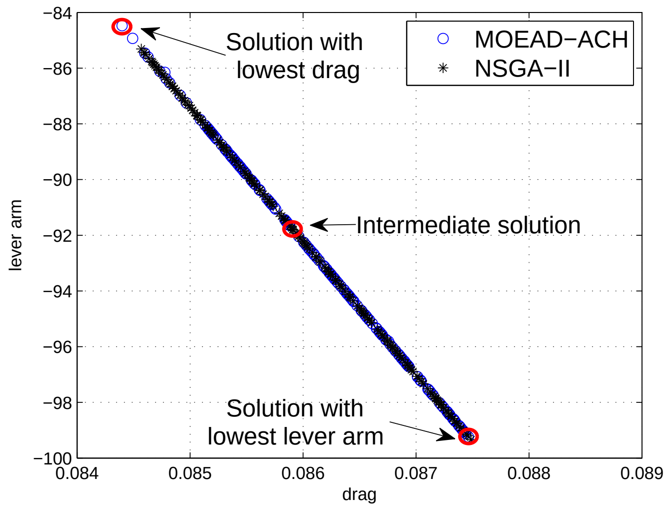


Figure 2.21: Final non-dominated fronts for best run of the toy submarine design problem using MOEA/D-ACH and NSGA-II

Presented in Table 2.19 is the comparison of the resulting performance criteria of the optimized toy submarines using both MOEA/D-ACH and NSGA-II. The results are comparable indicating that MOEA/D-ACH is capable of solving constrained optimization problems as good as NSGA-II, even when the feasibility ratio is extremely low. Shown in Figures 2.22 and 2.23 are the internal configurations of the chosen intermediate solutions for MOEA/D-ACH and NSGA-II respectively.

The constraint handling approach is embedded within the framework of multi-objective evolutionary algorithm based on decomposition (MOEA/D) [9] to equip it to deal with constrained optimization problems. To assess the performance of proposed approach, 10 well known benchmark multiobjective constrained optimization problems and a real-world toy submarine design problem were solved. The

Table 2.19: Performance criteria of the optimized toy submarines using MOEA/D-ACH and NSGA-II for solution-1 (considering the lowest drag), solution-2 (the intermediate solution), solution-3 (considering the maximum lever arm)

	MOEA/D-ACH			NSGA-II		
Vehicle particulars	Solution-1	Solution-2	Solution-3	Solution-1	Solution-2	Solution-3
Nose length	45 mm	45 mm	45 mm	45 mm	45 mm	45 mm
Parallel middle body length	250 mm	261 mm	275 mm	252 mm	263 mm	275 mm
Tail length	80 mm	80 mm	80 mm	80 mm	80 mm	80 mm
Length overall	375 mm	386 mm	399 mm	377 mm	388 mm	399 mm
Maximum diameter	58 mm	58 mm	58 mm	58 mm	58 mm	58 mm
Length to diameter ratio	6.5	6.7	6.9	6.5	6.7	6.9
Maximum dimension of the inner square	38 mm	38 mm	38 mm	38 mm	38 mm	38 mm
Wetted surface area	0.086410 m^2	0.088848 m^2	0.092030 m^2	0.086727 m^2	0.089279 m^2	0.092086 m^2
Displacement volume	0.000456 m^3	0.000473 m^3	0.000494 m^3	0.000458 m^3	0.000476 m^3	0.000495 m^3
Mass of the displaced water	456 g	473 g	494 g	458 g	476 g	495 g
Total mass of the vehicle	436.219 g	441.854 g	449.201 g	436.954 g	442.848 g	449.328 g
Length of the first lever arm	99.3352 mm	103.519 mm	108.981 mm	**99.8721 mm**	**104.258 mm**	**109.071 mm**
Length of the second lever arm	84.4732 mm	90.8139 mm	99.0644 mm	**85.3094 mm**	**91.9321 mm**	**99.200 mm**
X-coordinate of CG	-0.962819 mm	-0.95054 mm	-0.934993 mm	-0.961200 mm	-0.948407 mm	-0.934729 mm
Y-coordinate of CG	-0.206318 mm	-0.203687 mm	-0.200356 mm	-0.205971 mm	-0.20323 mm	-0.200299 mm
Z-coordinate of CG	182.41 mm	186.5 mm	196.042 mm	183.306 mm	189.029 mm	195.206 mm
X-coordinate of CB	0	0	0	0	0	0
Y-coordinate of CB	0	0	0	0	0	0
Z-coordinate of CB	178.473 mm	184.814 mm	193.066 mm	179.309 mm	185.932 mm	193.2 mm
Longitudinal distance between CB and CG	3.937 mm	1.686 mm	2.976 mm	3.997 mm	3.097 mm	2.006 mm
Nominal speed	0.5 m/s	0.5 m/s	0.5 m/s	0.5 m/s	0.5 m/s	0.5 m/s
Drag (VT method)	**0.0814820 N**	**0.0830058 N**	**0.0849836 N**	0.0816812 N	0.0832745 N	0.0850195 N
Drag (G&J method)	**0.0822890 N**	**0.0837647 N**	**0.0856861 N**	0.0824815 N	0.0840254 N	0.0857211 N
Drag (MIT method)	**0.0843982 N**	**0.0857070 N**	**0.0874267 N**	0.0845684 N	0.0859393 N	0.0874587 N

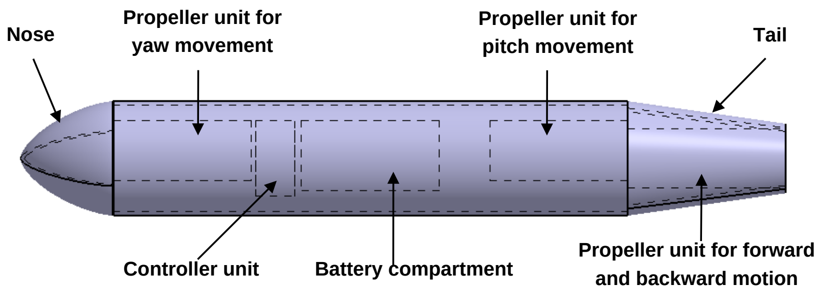


Figure 2.22: Configuration of the resulting optimized toy submarine concerning drag and lever arms simultaneously obtained from MOEA/D-ACH.

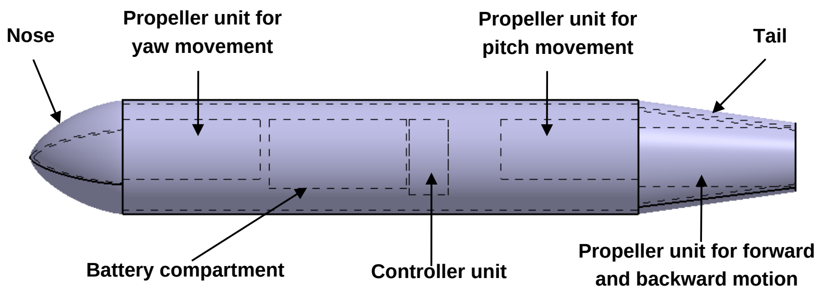


Figure 2.23: Configuration of the resulting optimized toy submarine concerning drag and lever arms simultaneously obtained from NSGA-II.

results are compared with those obtained using NSGA-II. The preliminary results of this study indicate that the constraint handling approach is effective and MOEA/D-ACH is able to deal with constrained optimization problems better or at par with NSGA-II. Since the constraint handling scheme is generic, it can be used in other forms of population based stochastic algorithms.

2.5 Summary

In this chapter, three key contributions are made related to the enhancement of a DE algorithm for the solution of single and multi-objective optimization problems. The benefits are summarized below.

- AH-DEa: The proposed algorithm incorporates an adaptive crossover rate control mechanism, a combination of crossover types and a local search strategy. Binomial and exponential crossover mechanisms have been used in various stages of evolution to exploit their strengths in exploration and exploitation.

- DE-CS: A novel constraint handling scheme has been introduced within the framework of differential evolution utilizing the concepts of partial evaluation and constraint sequencing. The utility of using multiple constraint sequences is highlighted using three illustrative examples. The approach is likely to provide significant computational benefits for problems involving computationally expensive constraints. Furthermore, since DE-CS attempts to reach the feasible space from different search directions, it is less likely to be trapped in local optima. Since the efficiency of the algorithm stems from handling constraints, the approach is likely to be less useful for problems with high feasibility ratio (such as G02 and G24). While the approach presented in the paper inherently assumes that the constraints can be evaluated independently, the method can be extended further to deal with blocks or sets of constraints as encountered in more realistic multidisciplinary optimization problems.

- MOEA/D-ACH: An adaptive constraint handling approach has been presented. The constraint handling approach is embedded within the framework of multi-objective evolutionary algorithm based on decomposition (MOEA/D) to equip it to deal with constrained optimization problems. Since the constraint handling scheme is generic, it can be used in other forms of population based stochastic algorithms.

The performance of these algorithms is rigorously assessed using different benchmark functions. The results assessed using various measures clearly indicate that the proposed developments offer competitive advantages over existing schemes.

Chapter 3

Decomposition based Many-objective Optimization

Part of this work has previously appeared in **Asafuddoula, M.**, Ray, T. and Sarker, R., "A decomposition based evolutionary algorithm for many-objective optimization with systematic sampling and adaptive epsilon control", in Proceedings of the Seventh International Conference on Evolutionary Multi-Criterion Optimization, vol. 7811 Lecture Notes in Computer Science, pp. 417-427, Springer, 2013.

Ray, T., **Asafuddoula, M.**, and Isaacs, A., "A steady state decomposition based quantum genetic algorithm for many-objective optimization", in Proceedings of the IEEE Congress on Evolutionary Computation, pp. 2817-2824, Cancun, Mexico, 2013.

Asafuddoula, M., Ray, T. and Sarker, "A decomposition based evolutionary algorithm for many-objective optimization", IEEE Transactions on Evolutionary Computation. (Under review).

Chapter Overview

This chapter introduces a decomposition based evolutionary algorithm (DBEA) for many-objective optimization. There are three key components of the algorithm i.e., uniformly distributed reference points are generated via systematic sampling, balance between con-

vergence and diversity is maintained using two independent distance measures and a simple preemptive distance comparison scheme is used for the association of solutions to the reference directions. Comparisons with other published state-of-the-art algorithms on various real-world benchmark problems as well as the problems involving redundant objectives and disconnected Pareto fronts are included in order to illustrate the performance of the algorithm.

3.1 Literature Review

Many-objective optimization relate to optimization problems where the number of objectives is large, in general greater than four [94]. There is significant amount of literature discussing the challenges involved in solving them and interested readers may refer to [94] for further details. The main difficulty arises from the inability of the non-dominance-based schemes to generate sufficient selection pressure to drive the solutions to the Pareto front. Therefore, the commonly used dominance based methods for multi-objective optimization, such as NSGA-II [91], SPEA2 [95] do not offer satisfactory results. There has been a number of attempts to modify the underlying selection pressure through the use of secondary metrics such as substitute distance measures [96][97], average rank domination [98], fuzzy dominance [99], ϵ-dominance [100][101], adaptive ϵ-ranking [102] etc. without great success. In all the above approaches, while the diversity and the convergence of the population improved during the course of evolution, there is no guarantee that the final non-dominated set spans the entire Pareto surface uniformly.

There are also radically different approaches to deal with many-objective optimization, such as attempts to identify the reduced set of objectives [103] or corners of the Pareto front [104] and subsequently solving the problem using these reduced set of objectives. Other attempts include interactive use of decision makers preferences [105], use of reference points [106][107] or solution of the problem as a hypervolume maximization problem [93]. While some progress has been made along these lines, the limiting factors include the inability to obtain solutions close to Pareto set for an accurate identification of redundant objectives, decision making burden associated with preference elicitation and the computational complexity of hypervolume computation.

Decomposition based evolutionary algorithms are yet another class of algorithms originally introduced as MOEA/D [9], wherein the multi-objective optimization problem is decomposed into a series of scalar optimization problems. MOEA/D has been quite successful in solving optimization problems involving two and three objectives and there is significant interest in developing it further to deal with many-objective optimization problems. Notable works in the area include the development of surface evolutionary algorithm (SEA) [108], many-objective evolutionary algorithm based on generalized decomposition (MAEA-gD) [109], approximation model guided selection (AMS) [110], M-NSGA-II [111] and recent works of the authors in decomposition based EAs [112] and quantum inspired many- objective algorithm [113].

Fundamentally, in all such approaches one needs to generate a set of uniformly distributed reference directions and adopt a method of scalarization. In the context of many- objective optimization, the first issue relates to the design of a computationally efficient scheme to generate W uniform reference directions for a M objective optimization problem, where M is typically more than four and W is often chosen to be the same as the population size. The second issue relates to scalarization, which essentially assigns the $fittest$ individual to each reference direction. The notion of $fittest$ is essentially derived using a trade-off between convergence and diversity measured with respect to any given reference direction. One of the early attempts to generate uniformly distributed reference directions appear in the works of Hughes [107] . The method was not computationally efficient for problems with more than six objectives and often resulted in a large number of reference directions that in turn required a huge population size. More recently, computationally efficient and scalable sampling schemes have been used in the context of many-objective optimization. A systematic sampling [4] scheme was used in M-NSGA-II [111] while an uniform sampling scheme was used in MOEA/D [114].

The second issue related to scalarization has been addressed via two fundamental means i.e., through a systematic association and niche preservation mechanism as in M-NSGA-II [111] or through the use of a penalty function (i.e., an aggregation of the projected distance along a reference direction and the perpendicular distance from a point to a given reference direction) within the framework of MOEA/D. The performance of the penalty function based approach is dependent on the penalty parameter, while the association and the niche preservation process require a careful implementation to address a number of possibilities.

This chapter presents a decomposition based evolutionary algorithm for many-objective optimization. The reference directions are generated using systematic sampling, wherein the points are systematically generated on a hyperplane with unit intercepts in each objective axis. The process of reference point generation is the same as adopted in M-NSGA-II [111]. The association of solutions to reference directions are based on two independent distance measures. The distance along the reference direction controls convergence while the perpendicular distance from the solution to the reference direction controls the diversity. The proposed algorithm utilizes a simple prioritized distance comparison to maintain this balance and control association. In order to improve the efficiency of the algorithm, a steady state form is adopted in contrast to a generational model used in M-NSGA-II [111] . Furthermore, to deal with constraints, an adaptive epsilon level based scheme is adopted which had outstanding performance on recent constrained optimization benchmarks [115, 116].

The rest of the chapter is organized as follows. The details of the proposed algorithm are presented in Section 3.2. The performance of the proposed algorithm on benchmark problems (DTLZ1-DTLZ4 for 3, 5, 8, 10 and 15 objectives) and (WFG1-WFG9 for 3, 5, 10 and 15) are presented and compared with MOEA/D-PBI and M-NSGA-II in Section 3.3. The performance on degenerate problems DTLZ5 and WFG3 are also presented in Section 3.3. In addition to the above set of mathematical benchmarks, the performance of the algorithm is also presented and compared using a number of engineering design problems (car side impact, water resource management and the constrained ten-objective general aviation aircraft (GAA) design). Another variant of decomposition based method is presented with quantum genetic algorithm and its performance and benefits are presented in the subsequent sections. The final section summarizes the contributions and future directions for further improvement.

3.2 Proposed Decomposition Based Evolutionary Algorithm

A many-objective optimization problem can be defined as follows:

$$\begin{aligned} min.\ [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), ...f_M(\mathbf{x})], \mathbf{x} \in \Omega \\ S.t. \qquad g_j(\mathbf{x}) \leq 0, j = 1, 2, ...p \\ h_k(\mathbf{x}) = 0, k = 1, 2, ...q \end{aligned} \tag{3.1}$$

where $f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), ...f_M(\mathbf{x})$ are the M objective functions, p is the number of inequalities and q is the number of equalities.

The proposed improved decomposition based evolutionary algorithm (I-DBEA) presented in this chapter is an extension of the authors previous work on DBEA-Eps [112]. While DBEA-Eps [112] was successful in solving a range of many-objective optimization problems, the performance was dependent on the choice of a number of parameters and several adaptive rules. This extension is clearly focused on elimination of such parameters and adaptive rules. The differences are summarized below for a greater clarity.

- In DBEA-Eps, the original concept of neighborhood as in MOEA/D was used to select parents for recombination with a given probability τ. In the proposed algorithm, both these parameters have been eliminated through the use of a random parent selection scheme with a first encounter replacement strategy.

- Every solution in DBEA-Eps, had two distance measures associated with it i.e., distance along a reference direction d_1 and distance perpendicular to the reference direction d_2. Comparisons between solutions were based on an adaptive epsilon level of d_2. In the proposed algorithm, a simple precedence rule is used, where d_2 has precedence over d_1.

- Scaling is an important aspect in any decomposition based scheme. In DBEA-Eps, a hyperplane was constructed using M extreme non-dominated solutions which in turn provided the axis intercept lengths. In the proposed algorithm, such solutions are identified using corner sort [104].

The algorithm is presented below and the individual components related to (a) generation of reference points (b) normalization and computation of distances (c) method of recombination (d) selection/replacement (e) means of constraint handling are discussed in subsequent subsections.

3.2.1 Generation of reference points

A structured set of reference points γ is generated spanning a hyperplane with unit intercepts in each objective axis using normal boundary intersection method (NBI) [4]. The approach generates W points on the hyperplane with a uniform spacing of $\delta = 1/s$ for any number of objectives M and there are s sampling locations along each objective axis. The process of generation of the reference points is illustrated using a 3-objective optimization problem ($M = 3$) with an assumed spacing of $\delta = 0.2$ ($s = 5$) in Figure 3.1. The process results in the generation of 21 reference points.

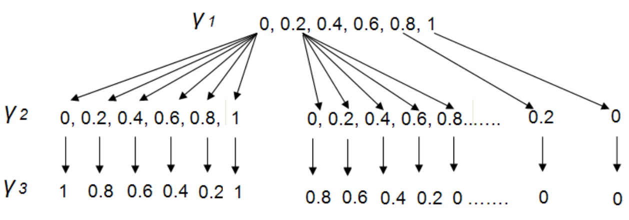
$$W = {}^{(M+s-1)}C_s \tag{3.2}$$

The distribution of the reference points are presented in Figure 3.2. The reference directions are formed by constructing a straight line from the origin to each of these

Algorithm 3.1 I-DBEA

Input: Gen_{max} maximum number of generations, W the number of reference points

Generate the reference points using NBI
Initialize the population P; $|P|$ = W and assign each individual of P to an unique reference direction randomly.
Evaluate the initial population and compute the ideal point $\bar{z}_j = (f_1^{min}, f_2^{min}, ..., f_M^{min})$, identify the corners and compute intercepts a_j's for $j = 1$ to M
Scale the individuals of the population
Assign the $2M$ corner solutions to a corner set S.
while $(gen \leq Gen_{max})$ **do**
  for *i=1:W* **do**
    Select P_i as the base parent
    I=Select its partner randomly from W
    Create a child via recombination as C_i
    Evaluate C_i and compute the distances (d_1 and d_2) using all reference directions
    Update the corner solution set S using *corner-sort*
    Replace the parent P_l with C_i using *single-first encounter strategy*, where l denotes the index of the first parent satisfying the condition of replacement
    Update the ideal point $(\bar{z})$, the intercepts and re-scale the population
  end for
end while



(a)

γ_1	γ_2	$(1-\gamma_1-\gamma_2)$
0	0	1.0000
0	0.2000	0.8000
0	0.4000	0.6000
0	0.6000	0.4000
0	0.8000	0.2000
0	1.0000	0
0.2000	0	0.8000
0.2000	0.2000	0.6000
0.2000	0.4000	0.4000
0.2000	0.6000	0.2000
0.2000	0.8000	0
0.4000	0	0.6000
0.4000	0.2000	0.4000
0.4000	0.4000	0.2000
0.4000	0.6000	0
0.6000	0	0.4000
0.6000	0.2000	0.2000
0.6000	0.4000	0
0.8000	0	0.2000
0.8000	0.2000	0
1.0000	0	0

(b)

Figure 3.1: (a) the reference points are generated computing γs recursively (b) the table shows the combination of all γs in each column

reference points. The population size of the algorithm is set to the number of reference points. The initial population consists of W individuals generated randomly within the

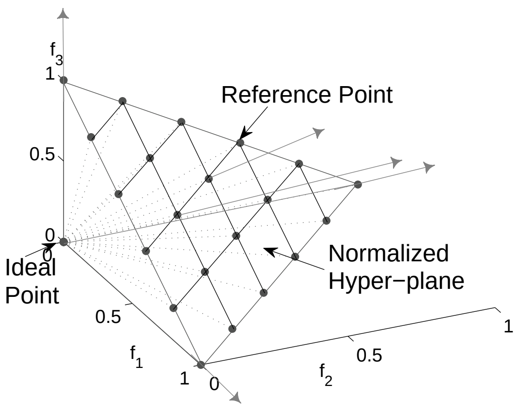


Figure 3.2: A set of reference points in a normalized hyperplane for number of objectives, $M = 3$ and $p = 5$.

variable bounds. Such solutions are thereafter assigned randomly to reference directions during the phase of initialization.

3.2.2 Normalization and computation of distances

Decomposition based algorithms rely heavily on d_1 and d_2 distances and normalization is necessary in the event the objectives are in different orders of magnitude. In DBEA-Eps and M-NSGA-II, the normalization is based on intercepts calculated using M extreme points of the non-dominated set. In I-DBEA, M solutions are identified using a *corner-sort* ranking [104] procedure. In corner sort, the top M solutions are the minimum in each objective, while the following M solutions are the minimum based on L_2 norm of all but one objectives. From the set of $2M$ solutions, the maximum in each objective is identified and corresponding solutions which have led to the maximum value is selected and referred as extreme points $\mathbf{z}^e$. Such extreme points are used to create the hyperplane and compute the intercepts. In the event the number of such extreme points are less than M, the maximum value of the objective is used as the intercept value (a_j's). The ideal point of a population is denoted by $\mathbf{z}_j = (f_1^{min}, f_2^{min}, ..., f_M^{min})$. The intercepts of the hyperplane along the objective axes are denoted by $a_1, a_2, ..., a_M$. The generic equation of a plane through these points can be represented using the following equation

$$C_1 f_1 + C_2 f_2 + ... + C_M f_M = 1 \tag{3.3}$$

where, C_1, C_2,....,C_M are the unit normal of the plane. The intercepts of the plane with the axis are given by $a_1 = 1/C_1$, $a_2 = 1/C_2$,....., and $a_M = 1/C_M$.

In the event, the number of such solutions are less than M or any of the a_j's are negative, a_j's are set to f_j^{max}. Every solution in the population is subsequently scaled as follows:

$$f'_j(\mathbf{x}) = \frac{f_j(\mathbf{x}) - \mathbf{z}_j}{a_j - \mathbf{z}_j}, \quad \forall j = 1, 2, ...M \tag{3.4}$$

For any given reference direction, the performance of a solution can be judged using two measures d_1 and d_2 as depicted in Equations 3.5 and 3.6. The first measure d_1 is the Euclidean distance between origin and the foot of the normal drawn from the solution to the reference direction, while the second measure d_2 is the length of the normal. Mathematically, d_1 and d_2 are computed as follows:

$$d_1 = \mathbf{w}^T f'_j(\mathbf{x}) \tag{3.5}$$

$$d_2 = \|f'_j(\mathbf{x}) - \mathbf{w}^T f'_j(\mathbf{x})\mathbf{w}\| \tag{3.6}$$

where $\mathbf{w}$ is a unit vector along any given reference direction. It is clear that a value of $d_2 = 0$ ensures the solutions are perfectly aligned along the required reference direction ensuring perfect diversity, while a smaller value of d_1 indicates superior convergence. These two measures are subsequently used to control diversity and convergence of the algorithm.

3.2.3 Method of recombination

In the recombination process, two child solutions are generated using simulated binary crossover (SBX) operator [117] and polynomial mutation. The first child is considered as an individual attempting to replace any parent in the population.

3.2.4 Selection/replacement

In the steady state form, if a child solution is non-dominated with respect to the individuals in the population, it attempts to enter the population via a replacement. The child solution competes with all solutions in the population in a random order until it makes a successful replacement or has competed with all individuals. If $\{d_{1_r}, d_{2_r}\}$ denotes the distances for the r^{th} solution in the population and $\{d_{1_c}, d_{2_c}\}$ denotes the distances for the child solution along r^{th} reference direction, a child is considered winner if d_{2_c} is less than d_{2_r}. In the event the d_{2_c} is equal to d_{2_r}, the child is considered a winner if d_{1_c} is less than d_{1_r}. The simple precedence of d_2 over d_1 eliminates the need for a complex epsilon based scheme.

3.2.5 Constraint Handling

The constraint handling approach used in this work is based on epsilon level comparison and has been reported earlier in [115]. The feasibility ratio (FR) of a population refers to the ratio of the number of feasible solutions in the population to the number of solutions (W). The allowable violation is calculated as follows:

$$CV = \sum_{i=1}^{p} max(g_i, 0) + \sum_{i=1}^{q} max(|h_i - \epsilon|, 0) \tag{3.7}$$

$$CV_{mean} = \frac{1}{W} \sum_{j=1}^{W} (CV_j) \tag{3.8}$$

$$\text{Allowable violation}(\epsilon_{CV}) = CV_{mean} * FR \tag{3.9}$$

An epsilon level comparison using this allowable violation measure is used to compare two solutions. If two solutions have their constraint violation value less than this epsilon level, the solutions are compared based on their objective values i.e., via d_1 and d_2 measures. Such a constraint handling scheme has performed better than *feasibility first* schemes on recent constrained optimization benchmarks [115].

3.3 Experimental Results

In this section, the results of improved decomposition based evolutionary algorithm (I-DBEA) are presented and compared with DBEA-Eps [112], M-NSGA-II and MOEA/D -PBI [118] for problems DTLZ1-DTLZ4 with 3, 5, 8, 10 and 15 objectives, WFG with 3, 5, 10 and 15 objectives and three other constrained engineering design problems.

The population sizes used in this study are the same as those adopted in [118]. The reference points are generated following Equation 3.2. For M=3, s is chosen as 12 resulting in 91 reference points, while for M=5, s is set to 6 resulting in 210 reference points (Table 3.1). For M greater than 8, the reference points are generated via a two-layer sampling scheme with two values of s i.e., one for each layer as outlined in [118] (Table 3.1). These settings have been used to make consistent comparisons between I-DBEA and the recently proposed reference direction based NSGA-III [118].

Table 3.1: Number of reference points/directions/population used in the study.

No. of Obj. (M)	Sampling size (s) in each axis	Popsize/Ref. dirn. (W)
3	s=12	91
5	s=6	210
8	s=3, s=2	156
10	s=3, s=2	275
15	s=2, s=1	135

It is important to highlight that the two-layer sampling scheme [118] results in redundant extreme points (i.e., along each objective axis). While such a scheme may provide benefit to NSGA-III, it is not required for I-DBEA as it intrinsically identifies extreme points via *corner-sort*.

Parameters for I-DBEA include a probability of crossover is set to 1 and the probability of mutation is set to $p_m = 1/D$, where D is the dimensionality of the problem. Parameters for DBEA-Eps include a neighborhood size of 20 and the probability of selecting a parent from its neighborhood (T) is set to 0.9. The distribution index of crossover is set to η_c=30 and the distribution index of mutation is set to η_m=20 as in [118]. Parameters for MOEA/D-PBI include a neighborhood size of 20, probability of selecting a parent from its neighborhood (T) is set to 0.9, the distribution index of crossover is set to η_c=20, the distribution index of mutation is set to η_m=20, maximum number of solutions replaced by a child solution $\eta_r = T$, and a penalty parameter $\theta = 5$ as in [118]. The performance of MOEA/D-PBI could have been affected by the choice of the above parameters.

To assess the performance, we have selected Inverted generational distance (IGD) [8][9] and Hypervolume (HV) [6] as the performance metric. The IGD metric is calculated with respect to the given reference directions normalized with the theoretical ideal and nadir points for the DTLZ problems. For other problems (i.e., WFG), the targeted Pareto-optimal points are generated from the non-dominated solutions obtained from all runs of the algorithms. The exact hypervolume is computed for 3 to 8 objective problems while an approximated hypervolume is computed for 10 to 15 objective problems.

3.3.1 Performance on Unconstrained DTLZ Problems

In this comparison, the best, median and worst IGD results obtained using 30 independent runs for DTLZ1-DTLZ4 are presented. The test instances of DTLZ problems are briefly discussed in below:

- DTLZ1

$$
\begin{aligned}
&Minimize\\
&f_1(x) = \frac{1}{2}x_1x_2..................x_{M-1}(1+g(x_M))\\
&f_2(x) = \frac{1}{2}x_1x_2.........(1-x_{M-1})(1+g(x_M))\\
&\vdots\\
&f_{M-1}(x) = \frac{1}{2}x_1(1-x_2)(1+g(x_M))\\
&f_M(x) = \frac{1}{2}(1-x_1)(1+g(x_M))
\end{aligned}
\tag{3.10}
$$

where, $g(x_M) = 100(|K| + \sum_{i=M}^{M+K-1}(x_i - 0.5)^2 - cos(20\pi(x_i - 0.5)))$ and $x = (x_1, x_2, ..., x_n)^T \in [0,1]^{M+K-1}$.

The Pareto-optimal solution corresponds to x_i^*=0.5 ($x_i^* \in x_M$) and the objective function values lie on the linear hyperplane: $\sum_{i=1}^{M} f_i^*$=0.5. A value of $K = 5$ is used in this study.

- DTLZ2 - DTLZ4

$$
\begin{aligned}
&Minimize\\
&f_1(x) = cos(\frac{x_1^\alpha\pi}{2})...cos(\frac{x_{M-1}^\alpha\pi}{2})(1+g(x_M))\\
&f_2(x) = cos(\frac{x_1^\alpha\pi}{2})...sin(\frac{x_{M-1}^\alpha\pi}{2})(1+g(x_M))\\
&\vdots\\
&f_M(x) = sin(\frac{x_1^\alpha\pi}{2})(1+g(x_M))
\end{aligned}
\tag{3.11}
$$

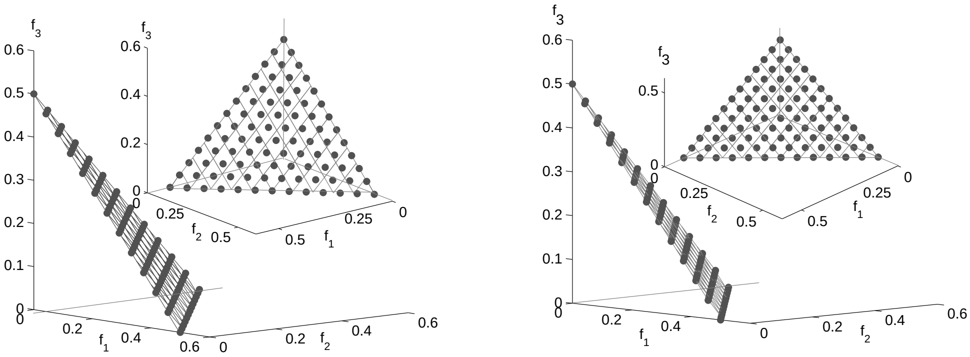
where,

$g(x_M) = \sum_{i=1}^{M+K-1}(x_i - 0.5)^2$

and $x = (x_1, x_2, ..., x_n)^T \in [0,1]^{M+K-1}$.

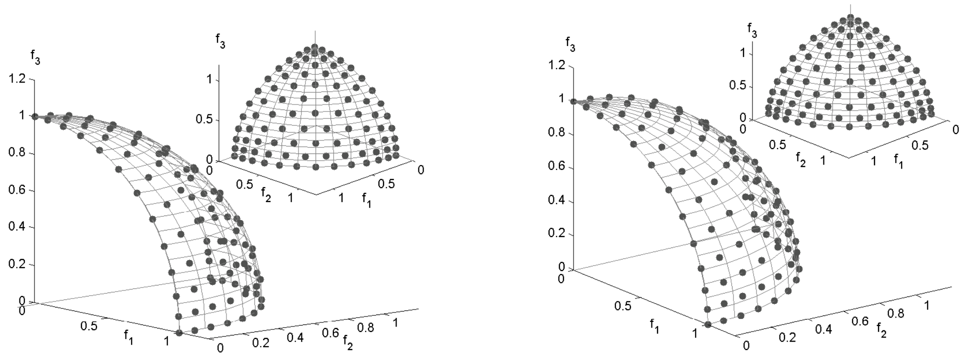
The Pareto-optimal solution corresponds to x_i^*=0.5 ($x_i^* \in x_M$) and the objective function values lie on the linear hyperplane: $\sum_{i=1}^{M}(f_i^*)^2$=1. A value of $K = 10$ and $\alpha = 1$ are suggested for DTLZ2, DTLZ3 and $\alpha = 100$ is suggested for DTLZ4 test problem. In the above problem, the total number of variables is $n = M + K - 1$.

The results of I-DBEA, DBEA-Eps, M-NSGA-II and MOEA/D-PBI are presented in Tables 3.2 and 3.3. In Figure 3.3 and Figure 3.4, the final Pareto front is shown for three-objective DTLZ1 and DTLZ2 problems using I-DBEA and DBEA-Eps. While schematically the results look similar, one can notice that I-DBEA obtained the best IGD values in 15 instances out of 20 (see Tables 3.2 and 3.3).



(a) (b)

Figure 3.3: Obtained solutions by (a) DBEA-Eps (b) I-DBEA for DTLZ1.



(a) (b)

Figure 3.4: Obtained solutions by (a) DBEA-Eps (b) I-DBEA for DTLZ2.

In order to observe the process of evolution, the average performance of the population is computed i.e., average of the d_1 and d_2 values of the individuals for DTLZ1 (3

Table 3.2: Best, Median and worst IGD values obtained for I-DBEA and DBEA-Eps on M-objective DTLZ1 and DTLZ2 problems. Best performance is shown in bold.

Test Prob.	Obj.	MaxGen	I-DBEA	DBEA-Eps	M-NSGA-II	MOEA/D-PBI
DTLZ1	3	400	1.075e-3	**8.771e-5**	4.880e-4	4.095e-4
			1.043e-2	9.521e-3	**1.308e-3**	1.495e-3
			6.502e-1	5.854e-1	4.880e-3	**4.743e-3**
	5	600	9.433e-4	**1.771e-5**	5.116e-4	3.179e-4
			5.993e-4	5.116e-4	9.799e-4	6.372e-4
			5.481e-1	5.854e-1	1.979e-3	**1.635e-3**
	8	750	8.570e-4	**4.387e-5**	2.044e-3	3.914e-3
			2.421e-4	3.581e-4	3.979e-3	6.106e-3
			1.864e-3	1.981e-3	8.721e-3	8.537e-3
	10	1000	**3.510e-4**	7.691e-4	2.215e-3	3.872e-3
			5.178e-3	**1.504e-3**	3.462e-3	5.073e-3
			1.112e-2	**2.700e-3**	6.869e-3	6.130e-3
	15	1500	**1.325e-3**	1.696e-3	2.649e-3	1.236e-2
			2.329e-3	2.606e-3	5.063e-3	1.431e-2
			3.356e-3	**2.686e-3**	1.123e-2	1.692e-2
DTLZ2	3	400	6.372e-4	2.040e-2	1.262e-3	**5.432e-4**
			1.243e-3	4.138e-2	1.357e-3	**6.406e-4**
			8.402e-3	6.417e-2	2.114e-3	**8.008e-4**
	5	600	**1.118e-3**	1.199e-3	4.254e-3	1.219e-3
			2.097e-3	3.024e-3	4.982e-3	1.437e-3
			6.165e-3	2.272e-2	5.862e-3	1.727e-3
	8	750	**2.218e-3**	1.172e-3	1.371e-2	3.097e-3
			3.185e-3	**2.899e-3**	1.571e-2	3.763e-3
			9.694e-3	6.915e-3	1.811e-2	**5.198e-3**
	10	1000	**2.173e-3**	3.656e-3	1.350e-2	2.474e-3
			3.025e-3	3.657e-3	1.528e-2	**2.778e-3**
			3.122e-3	3.657e-3	1.697e-2	3.235e-3
	15	1500	**4.238e-3**	5.160e-3	1.360e-2	5.254e-3
			4.251e-3	5.960e-3	1.726e-2	6.005e-3
			4.267e-3	5.960e-3	2.114e-2	9.409e-3

objectives). One can observe from Figure 3.5, that the average d_2 converges to near zero (i.e., near perfect alignment to the reference directions), while the average d_1 measure stabilizes at around 0.5 indicating convergence to the Pareto front.

The association mechanism (i.e., association of the solutions to each reference direction) for a 3-objective DTLZ1 problem is presented in Figure 3.6. The figure shows

Table 3.3: Best, Median and worst IGD values obtained for I-DBEA and DBEA-Eps on M-objective DTLZ3 and DTLZ4 problems. Best performance is shown in bold.

Test Prob.	Obj.	MaxGen	I-DBEA	DBEA-Eps	M-NSGA-II	MOEA/D-PBI
DTLZ3	3	400	**1.420e-4**	4.171e-4	9.751e-4	9.773e-4
			5.701e-4	**4.278e-4**	4.007e-3	3.426e-3
			3.590e-2	4.753e-1	6.665e-3	9.113e-3
	5	600	**4.984e-4**	1.102e-3	3.086e-3	1.129e-3
			4.076e-3	9.171e-2	5.960e-3	2.213e-3
			1.075e-1	5.713e-1	1.196e-2	6.147e-3
	8	750	**1.126e-3**	5.523e-2	1.244e-2	6.459e-3
			2.666e-3	7.821e-2	2.375e-2	1.948e-2
			6.983e-2	5.951e-1	9.649e-2	11.23e-1
	10	1000	**4.438e-3**	5.773e-3	8.849e-3	2.791e-3
			5.320e-3	**4.137e-3**	1.188e-2	4.319e-3
			5.396e-3	7.853e-1	2.083e-2	10.10e-1
	15	1500	8.612e-3	8.785e-3	1.401e-2	**4.360e-3**
			8.681e-3	9.135e-3	2.145e-2	1.664e-2
			8.724e-3	5.137e-1	4.195e-2	12.60e-1
DTLZ4	3	400	**9.858e-5**	2.175e-4	2.915e-4	2.929e-1
			2.300e-4	3.578e-3	5.970e-4	4.280e-1
			8.700e-1	9.154e-1	**4.286e-1**	5.234e-1
	5	600	**1.354e-4**	2.753e-4	9.849e-4	1.080e-1
			2.857e-4	2.121e-3	1.255e-3	5.787e-1
			9.894e-3	5.157e-1	1.721e-3	7.348e-1
	8	750	**1.761e-4**	3.771e-3	5.079e-3	5.298e-1
			3.319e-4	9.155e-3	7.054e-3	8.816e-1
			1.764e-2	5.951e-1	6.051e-1	9.723e-1
	10	1000	**1.716e-4**	3.771e-3	5.694e-3	3.966e-1
			1.716e-4	4.125e-3	6.337e-3	9.203e-1
			1.716e-4	5.754e-1	1.076e-1	10.77e-1
	15	1500	**1.716e-4**	6.173e-3	7.110e-3	5.890e-1
			1.716e-4	7.323e-3	3.431e-1	11.33e-1
			2.796e-4	5.753e-1	10.73e-1	12.49e-1

the associations in generation 1, 500 and 1000 using 15 reference points. One can observe that although initially the association is random, the solutions automatically get associated to the closest reference directions during the course of evolution via the pressure induced by d_2. This alleviates the need of an extensive niching and association operation as encountered in M-NSGA-II [111].

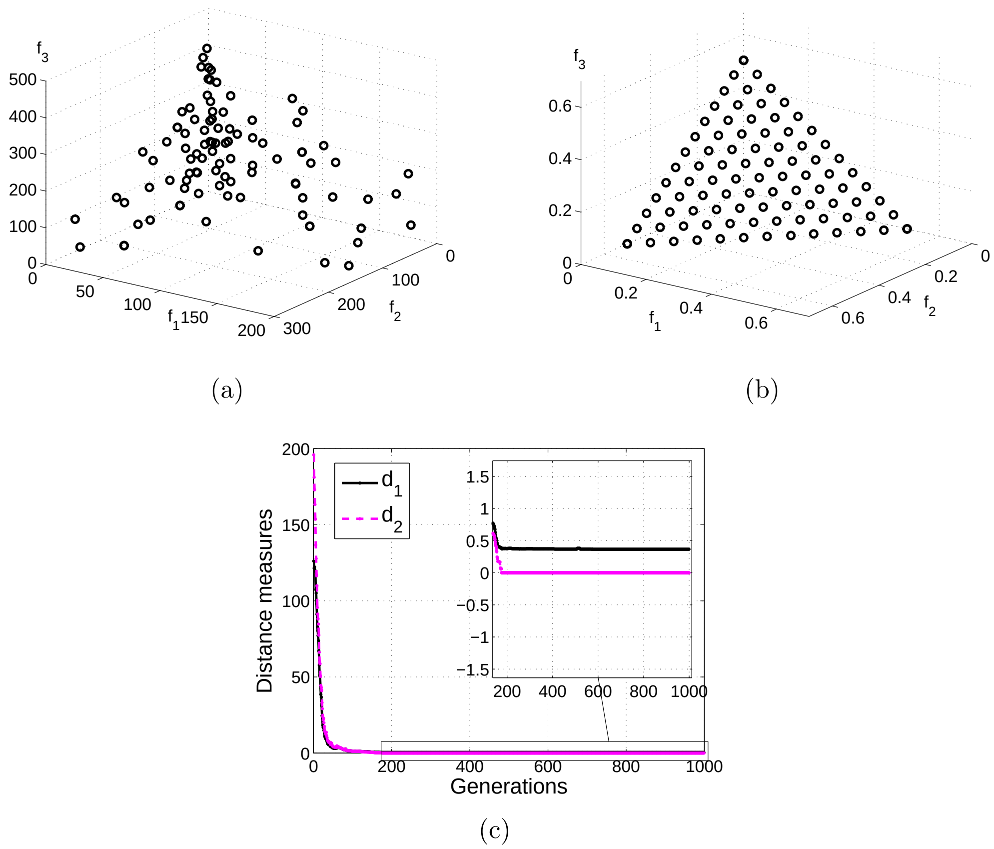


Figure 3.5: (a) the initial population of DTLZ1 test problem for number of objectives 3 (b) the final Pareto-front of DTLZ1 test problem for number of objectives 3 (c) the convergence of distance measure over the generations

3.3.2 Performance on Unconstrained WFG Problems

Next, the WFG test problems are considered which involve non-separable variables. These problems have different features i.e., disconnected, convex, concave, degenerate and linear Pareto-optimal front. The first test problem named WFG1 has a mixed Pareto-optimal front. For WFG2, the Pareto-optimal front is convex and disconnected and for WFG3 the Pareto front is linear and degenerate. The rest of the problems WFG4-WFG9 have concave Pareto front. In this study, the toolkit [6] was used to observe the performance of the algorithm. Similar to the DTLZ problem, the number of decision variables is set to 24 and other problem details are listed in Table 3.4.

Tables 3.5 and 3.6 show the best and average hypervolume measure for all numbers of objectives. For WFG1, 3, 5, 10 and 15, I-DBEA outperforms MOEA/D-PBI. For

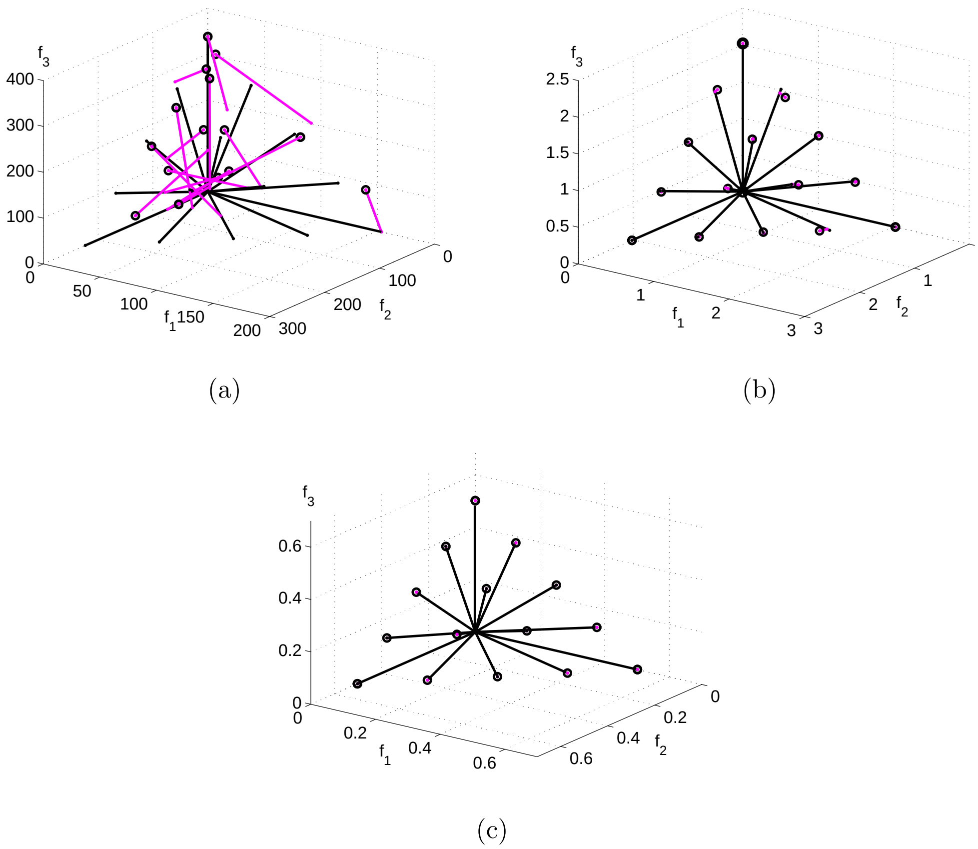


Figure 3.6: (a) the initial population of DTLZ1 test problem for number of objectives 3 with 15 reference points (b) at generation 500 (c) at final generation 1000

Table 3.4: Number of distance parameters and the position parameters used to combine the decision variables in the WFG test functions depending on the number of objectives.

	Number of Objectives			
	3	5	10	15
Distance Parameter	20	20	17	6
Position Parameters	4	4	7	18
Decision variables	24	24	24	24

WFG2 MOEA/D-PBI is better than I-DBEA for 3 and 5 objectives. For objectives 10 and 15, I-DBEA has the highest hypervolume. For the remaining 6 test problems, I-DBEA reaches the best performance with best and mean results for objectives 3, 5, 10 and 15.

Figures 3.7, 3.8 and 3.9 show the final non-dominated solutions obtained from both the algorithms. On WFG1, both I-DBEA and MOEA/D-PBI exhibit poor convergence

Table 3.5: Best and mean hypervolume statistics for problems WFG1-WFG4 using 30 independent runs

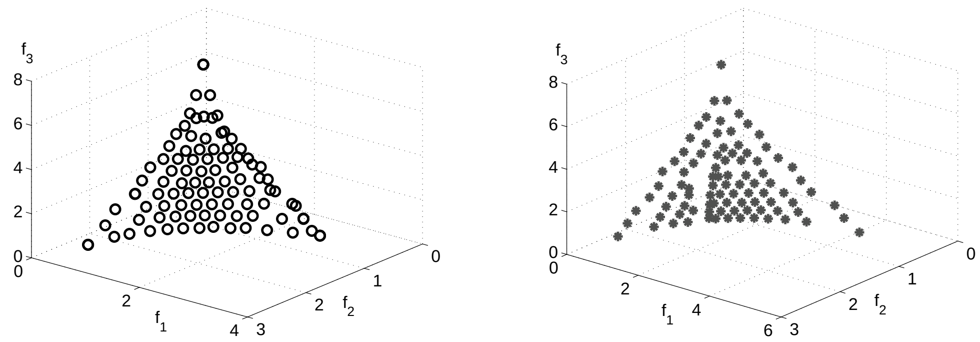
Test Problem	Obj.	MaxGen	I-DBEA	MOEA/D-PBI
WFG1	3	5000	**4.392e+01** **4.325e+01**	3.963e+01 3.711e+01
	5	5500	**2.473e+03** **2.441e+03**	2.373e+03 2.128e+03
	10	6000	**7.199e-01** **7.019e-01**	2.091e-01 1.923e-01
	15	6500	**9.982e-01** **9.203e-01**	9.904e-01 9.028e-01
WFG2	3	5000	4.331e+01 4.190e+01	**4.342e+01** **4.180e+01**
	5	5500	3.604e+03 3.149e+03	**3.617e+03** **3.151e+03**
	10	6000	**9.969e-01** **9.921e-01**	9.931e-01 9.231e-01
	15	6500	**9.982e-01** **9.203e-01**	9.904e-01 9.028e-01
WFG4	3	2000	**1.953e+01** **1.934e+01**	1.673e+01 1.660e+01
	5	3000	**2.556e+03** **2.539e+03**	2.167e+03 2.152e+03
	10	4000	**8.251e-01** **8.125e-01**	7.581e-01 7.438e-01
	15	4500	**9.616e-01** **8.329e-01**	9.606e-01 8.201e-01
WFG5	3	2000	**1.984e+01** **1.976e+01**	1.932e+01 1.917e+01
	5	3000	**2.576e+03** **2.569e+03**	2.449e+03 2.422e+03
	10	4000	**8.875e-01** **8.521e-01**	8.136e-01 7.943e-01
	15	4500	**8.871e-01** **8.177e-01**	8.842e-01 8.015e-01

due to the flat polynomial bias. However, the hypervolume measure using I-DBEA confirm a slight improvement than the result found using MOEA/D-PBI. For WFG2, the convergence has marginally improved, although the Pareto front is convex. It can be seen that both algorithms have the same difficulty to maintain the diversity as the

Table 3.6: Beast and mean hypervolume statistics for problems WFG6-WFG9 using 30 independent runs

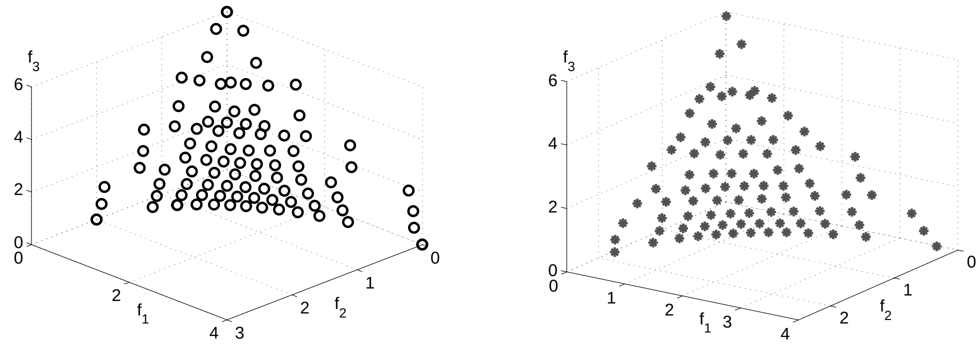
Test Problem	Obj.	MaxGen	I-DBEA	MOEA/D-PBI
WFG6	3	2000	**1.948e+01** **1.861e+01**	1.935e+01 1.724e+01
	5	3000	**2.568e+03** **2.540e+03**	2.418e+03 2.280e+03
	10	4000	**8.011e-01** **8.001e-01**	7.621e-01 7.596e-01
	15	4500	**8.832e-01** **9.236e-01**	8.393e-01 8.587e-01
WFG7	3	2000	**1.949e+01** **1.935e+01**	1.936e+01 1.933e+01
	5	3000	2.565e+03 2.559e+03	**2.569e+03** **2.568e+03**
	10	4000	**7.913e-01** **7.821e-01**	7.432e-01 7.273e-01
	15	4500	**9.617e-01** **9.283e-01**	9.604e-01 9.176e-01
WFG8	3	2000	1.969e+01 1.920e+01	**1.988e+01** **1.921e+01**
	5	3000	2.728e+03 **2.707e+03**	**2.774e+03** 2.079e+03
	10	4000	**8.175e-01** **7.713e-01**	6.950e-01 6.657e-01
	15	4500	**8.301e-01** **7.206e-01**	7.857e-01 6.662e-01
WFG9	3	2000	**1.849e+01** **1.750e+01**	1.557e+01 1.556e+01
	5	3000	**2.709e+03** **2.651e+03**	2.526e+03 2.477e+03
	10	4000	**8.102e-01** **7.027e-01**	7.100e-01 6.927e-01
	15	4500	**9.244e-01** **6.664e-01**	8.543e-01 5.889e-01

Pareto front is disconnected. By inspecting the hypervolume values of WFG2 problems (Table 3.5), one can observe that I-DBEA tend to perform well for problems with higher number of objectives.



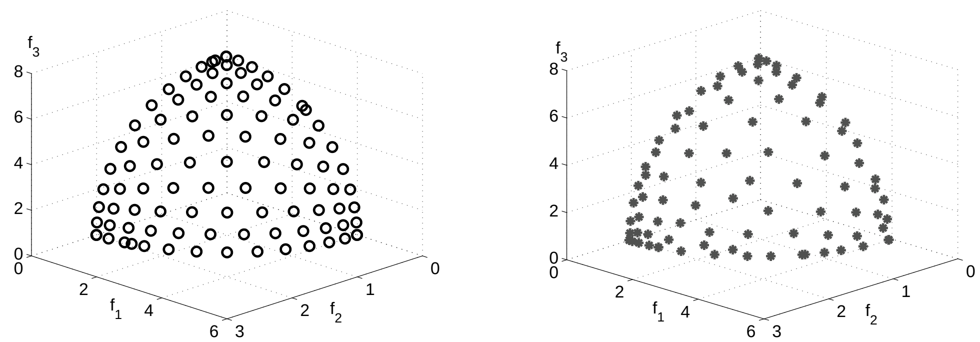
(a) (b)

Figure 3.7: Obtained solutions by (a) I-DBEA (b) MOEA/D-PBI for WFG1.



(a) (b)

Figure 3.8: Obtained solutions by (a) I-DBEA (b) MOEA/D-PBI for WFG2.



(a) (b)

Figure 3.9: Obtained solutions by (a) I-DBEA (b) MOEA/D-PBI for WFG4.

3.3.3 Performance on degenerate problems

DTLZ5-(I, M)

After demonstrating the performance of the proposed I-DBEA on commonly studied benchmark problems, its performance is investigated on degenerate test functions. For these problems the dimensionality of the Pareto front is less than the original number of objectives [17]. These test problems are referred as DTLZ5-(I, M) problems, where the problem can be formulated with different combinations of I and M. Here, I denotes the actual dimensionality of the Pareto front and M denotes the original number of objectives for the problem.

In this study, the dimensionality analysis is carried out using the final population obtained by I-DBEA along the lines suggested in [104]. If an objective is redundant, its omission from the reference set (F_R) should not result in a significant change in the number of non-dominated solutions. For quantifying the change in the number of non-dominated solutions, a parameter R is defined as a ratio of number of non-dominated solutions in the reference set F_R to the number of non-dominated solutions in F_R after discarding objective f_m (i.e., $(F_R \setminus f_m)$). The high value of R represents the omitted objective is redundant, whereas the low value of R represents the objective is relevant.

Here, the test problems DTLZ-(2, M) are attempted with various values of M. In the first case, DTLZ-(2, 3) is solved and the results are listed in Table 3.7.

Table 3.7: Dimensionality Reduction Analysis for DTLZ5-(2, 3) Problem

f_m	Objectives considered $(F_R \setminus f_m)$	R	Discard f_m
f_1	f_2, f_3	1.0	Yes
f_2	f_3	0.1240	No
f_3	f_2	0.0202	No

From the table, one can see that omitting f_1 does not affect the number of non-dominated solutions, implying redundancy of the objective. Therefore, f_1 is removed from the set of relevant objectives. In the remaining set of objectives (f_2 and f_3), it is observed that removing either objective effects the number of non-dominated solutions significantly, implying that both of them are non-redundant. Hence, the set of relevant objectives is identified as f_2, f_3 (see Figure 3.10).

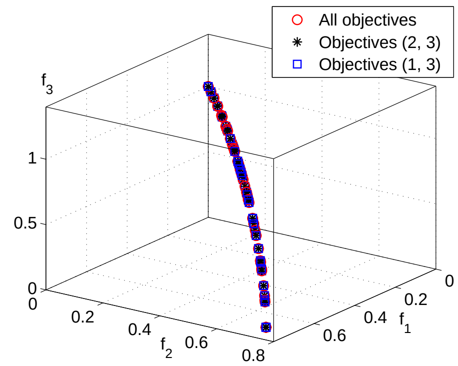


Figure 3.10: Approximation of the Pareto front obtained for DTLZ5 using all objectives, compared with the approximations obtained using two relevant objectives

The dimensionality analysis for another case, DTLZ-(2, 5) is shown in Table 3.8.

Table 3.8: Dimensionality Reduction Analysis for DTLZ5-(2, 5) Problem

f_m	Objectives considered ($F_R \setminus f_m$)	R	Discard f_m
f_1	f_2, f_3, f_4, f_5	1.0	Yes
f_2	f_3, f_4, f_5	0.9909	Yes
f_3	f_4, f_5	0.9636	Yes
f_4	f_5	0.6727	No
f_5	f_4	0.3727	No

One can see that omitting the objectives does not substantially affect the number of non-dominated solutions until all the redundant objectives are dropped, which is reflected in high R value for these objectives. However, reducing the objective set any further results in a substantial decrease in the number of non-dominated solutions, as seen from the low value of R for these cases f_4, f_5. Hence, the reduced objective set is identified as (f_4, f_5). The results for the rest of the test problems are summarized in Table 3.9. Similar to the DTLZ5-(2, 3) and DTLZ5-(2, 5) test problems, it is seen that the algorithm is able to identify the reduced set of objectives consistently across multiple runs. The success rate mentioned in the last column indicates the number of times the algorithm was able to identify these objectives out of 30 runs.

Table 3.9: Results obtained for DTLZ-(I, M) test problems

Test problem	Reduced set of objectives	Success rate
DTLZ5-(2,3)	f_2, f_3	30/30
DTLZ5-(2,5)	f_4, f_5	30/30
DTLZ5-(2,8)	f_7, f_8	30/30
DTLZ5-(2,10)	f_9, f_{10}	30/30
DTLZ5-(2,15)	f_{14}, f_{15}	30/30
DTLZ5-(3,3)	f_1, f_2, f_3	30/30
DTLZ5-(3,5)	f_3, f_4, f_5	30/30
DTLZ5-(3,8)	f_6, f_7, f_8	30/30
DTLZ5-(3,10)	f_8, f_9, f_{10}	30/30
DTLZ5-(3,15)	f_{13}, f_{14}, f_{15}	30/30

One can see from Figure 3.11 that the algorithm successfully identified the reduced set of objectives for DTLZ5-(3, 5) problem and converged to the Pareto front.

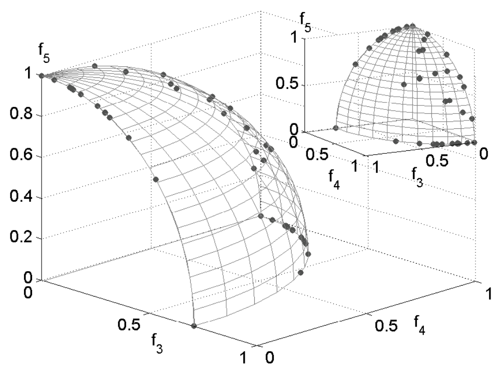


Figure 3.11: Approximation of the Pareto front obtained for DTLZ5-(3,5) using three relevant objectives

For a problem involving redundant objectives, the number of unique solutions in the corner set is less than M. In DBEA-Eps, the extreme solutions of the Pareto front could be lost due to distance comparisons which in turn would affect the normalization process as f_j^{max}'s would change. In I-DBEA, the extreme solutions are preserved in the corner set. A typical benefit of such an approach is presented for DTLZ5-(2, 3) in

Figures 3.12(a) and 3.12(b). One can observe from 3.12(a), that I-DBEA maintains the extreme solutions unlike DBEA-Eps.

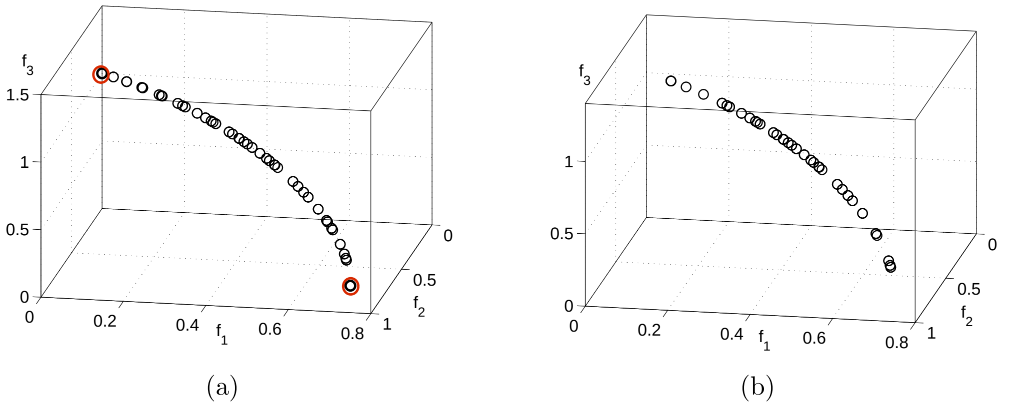


Figure 3.12: Approximation of the Pareto front using the proposed algorithm (a) with corner set i.e., I-DBEA (b) without corner set as in DBEA-eps for DTLZ5-(2,3)

WFG3

The performance of the proposed approach has also been studied with a modified version of a degenerate problem, WFG3. This problem is similar in formulation as problem WFG3 defined in [6], but instead of a linear shape function, a convex shape function has been used ($h_{m=1:M} = convex_m$). In addition, the value of $A_1 = 1$ and $A_2 = 0$ have been used, resulting in a degenerate Pareto front (dimensionality reduced by 1). For convenience, this problem have been referred to as $WFG3_{conv}$. The Pareto front for this problem, is shown in Figure 3.13. For details on construction of problems using the toolkit, the readers are referred to [6].

The final population (of size 91, evolved over 2000 generations) for $WFG3_{conv}$ obtained using I-DBEA is shown in Figure 3.13. The dimensionality analysis of the obtained solutions is shown in Table 3.10.

As mentioned before, the choice of parameters A1 = 1 and A2 = 0 makes the Pareto front of the problem degenerate. In this problem only two (f_2 and f_3) out of the three objectives are identified as relevant. With a different sequence, f_1 and f_3 can be identified as relevant. To verify the dimensionality analysis, the results using

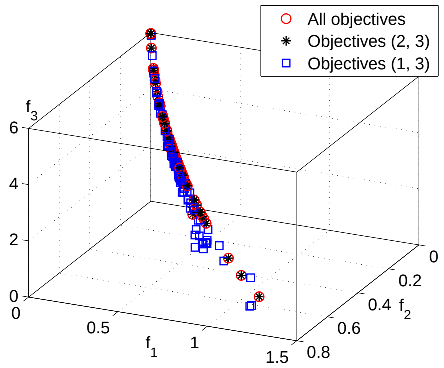


Figure 3.13: Approximation of the Pareto front obtained for $WFG3_{conv}$ using all objectives, compared with the approximations obtained using two relevant objectives

Table 3.10: Dimensionality Reduction Analysis for $WFG3_{conv}$ Problem

f_m	Objectives considered $(F_R \setminus f_m)$	R	Discard f_m
f_1	f_2, f_3	0.98	Yes
f_2	f_3	0.1010	No
f_3	f_2	0.0244	No

the original and the reduced set of objectives are shown in Figure 3.13. For better visualization, a small population size (91) is used. Since the population size is small, the algorithm is run for larger number of generations (2000) to get to the Pareto front. The value of R used in this case has been set to 0.9. The choice of threshold with a higher value (say 0.9) can successfully identify the redundant objectives.

3.3.4 Constrained Engineering Design Problems

Since the performance of the proposed algorithm was competitive on unconstrained test problems,its performance was further investigated using three constrained engineering design optimization problems i.e., the three-objective car-side-impact problem [119] with ten inequality constraints, five-objective water resource management problem [120] with seven inequality constraints and finally the ten objective general aviation aircraft (GAA) design problem [101] having a single inequality constraint.

Car side impact problem

The problem aims to minimize the weight of a car, the pubic force experienced by a passenger and the average velocity of the V-Pillar responsible for bearing the impact load subject to the constraints involving limiting values of abdomen load, pubic force, velocity of V-Pillar, rib deflection etc [119].

The problem is solved using I-DBEA and DBEA-Eps. The algorithms are run for 500 generations and the final non-dominated front is shown in Figure 3.14. It is important to note that the results of I-DBEA are derived with the same setup. One can see from Figure 3.14 that I-DBEA has achieved better alignment than DBEA-Eps.

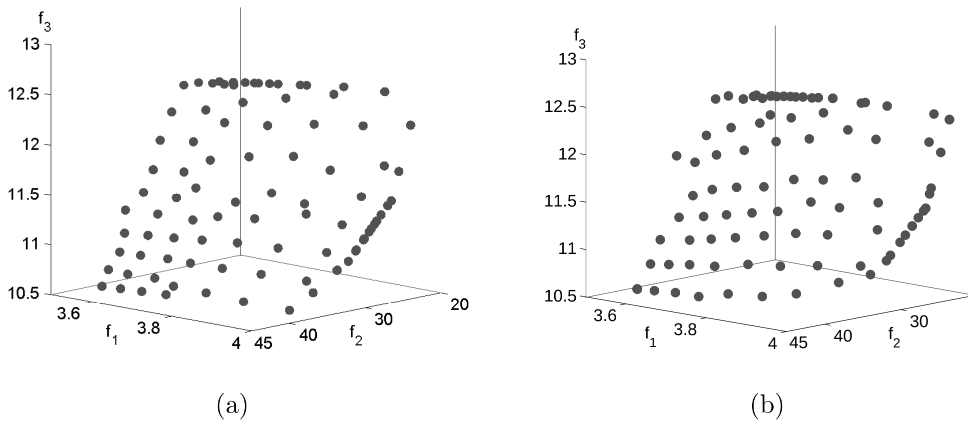


Figure 3.14: Solutions obtained using (a) I-DBEA (b) DBEA-Eps on three-objective car side impact problem

We have computed the IGD using the targeted reference set of 11900 non-dominated solutions found from all the runs considering all the algorithms and HV is computed by normalizing the solutions using the ideal point of (i.e.,[23.586, 3.5852, 10.611]) and the extreme point of (i.e.,[42.768, 4, 12.453]) the reference set. I-DBEA also outperforms DBEA-Eps on both IGD and HV metrics based on all three aspects i.e., best, median and worst (Table 3.11).

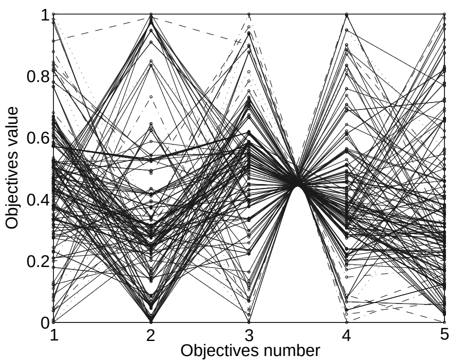
Table 3.11: IGD and HV (Best, Median and worst) values obtained using I-DBEA and DBEA-Eps for the car side impact problem

Algo.	FE	IGD
I-DBEA	45500	(**1.493e-01**, **1.501e-01**, **5.732e-01**)
DBEA-Eps		(1.529e-01, 1.805e-01, 5.956e-01)
Algo.	FE	Hypervolume
I-DBEA	45500	(**7.091e+00**, **6.781e+00**, **3.573e+00**)
DBEA-Eps		(7.014e+00, 6.780e+00, 2.013e+00)

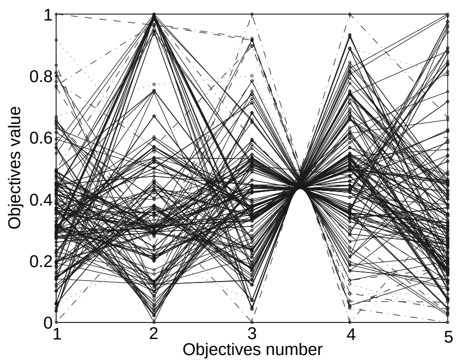
Water resource management problem

This is a five objective problem having seven constraints taken from the literature [120]. The parallel coordinate plot generated using the proposed algorithm (I-DBEA) is presented in Figure 3.15. The best IGD value of I-DBEA across 30 runs is $3.312e-2$ and the best IGD computed using the algorithm DBEA-Eps is $3.291e-2$ with the reference set of 2429 solutions [121]. A population of 210 solutions has been used for both the algorithms and evolved over 1000 generations.

In Figure 3.15, both the results from I-DBEA and DBEA-Eps have been presented using parallel coordinate plots. The result from I-DBEA is shown in the left most plot.



(a)



(b)

Figure 3.15: Solutions obtained using (a) I-DBEA (b) DBEA-Eps on five-objective water resource management problem

While the results appear similar, I-DBEA obtained a better distribution for objectives 1, 2 and 3. The IGD and HV metrics obtained from both the algorithms are given in Table 3.12. One can see in terms of IGD, DBEA-Eps has a better result based on best only, while I-DBEA outperforms DBEA-Eps in all other measures (IGD and HV).

Table 3.12: IGD and HV (Best, Median and worst) values obtained using I-DBEA and DBEA-Eps for the water resource management problem

Algo.	FE	IGD
I-DBEA DBEA-Eps	210000	(3.312e-02, **3.339e-02**, **1.123e-01**) (**3.291e-02**, 3.375e-02, 1.986e-01)
Algo.	FE	Hypervolume
I-DBEA DBEA-Eps	210000	(**2.554e-01**, **2.467e-01**, **1.962e-02**) (2.513e-01, 2.397e-01, 1.678e-02)

General aviation aircraft (GAA) design problem

This problem was first introduced by Simpson et al. [122] and has been recently solved using an evolutionary algorithm [101]. The problem involves 9 design variables i.e., cruise speed, aspect ratio, sweep angle, propeller diameter, wing loading, engine activity factor, seat width, tail length/ diameter ratio and taper ratio and the aim is to minimize the takeoff noise, empty weight, direct operating cost, ride roughness, fuel weight, purchase price, product family dissimilarity and maximize the flight range, lift/ drag ratio and cruise speed. Previous studies encountered difficulties in obtaining feasible solutions due to tight constraints [122].

In this example, 100 reference points were used and the population was allowed to evolve over 5000 generations. A reference set of 412 non-dominated solutions obtained from ϵ-MOEA and Borg-MOEA is used to compute the IGD metric. The results of the proposed algorithm are compared with five other algorithms i.e., DBEA-Eps, ϵ-MOEA, Borg-MOEA, MOEA/D and ϵ-NSGA-II [101]. The hypervolume was computed using the ideal point of (i.e., [73.251, 1881.5, 59.114, 1.7977, 359.92, 41879, -2580.2, -16.823, -204.02, 0.26847]) and the extreme point of (i.e., [74.036, 2011.5, 79.993, 2, 483.13, 44590, -2000, -14.408, -189.3, 1.9844]) obtained from the reference set. The performance of the algorithms are compared using the hypervolume in Table 3.13 and IGD in

Table 3.14. One can observe that the proposed algorithm performs marginally better than others for this problem.

Table 3.13: Performance metric value of product family design problem using 50 independent runs

Algorithm	FE	Hypervolume			
		Best	Mean	Worst	Std
I-DBEA	50,000	0.02995	0.01726	0.00699	0.05121
DBEA-Eps		0.02899	0.01715	0.00689	0.04561
ϵ-MOEA		0.02032	0.01032	0.00259	0.04125
Borg-MOEA		0.02245	0.01013	0.00424	0.02327
MOEA/D		0.00092	0.00087	0.00045	0.00145
ϵ-NSGA-II		0.01636	0.01005	0.00236	0.05232

Table 3.14: Performance metric value of product family design problem using 50 independent runs

Algorithm	FE	IGD			
		Best	Mean	Worst	Std
I-DBEA	50,000	0.63150	0.80217	0.83101	0.09613
DBEA-Eps		0.62070	0.80123	0.82430	0.09210
ϵ-MOEA		0.98312	0.99123	0.99678	0.10312
Borg-MOEA		0.98211	0.99113	0.99337	0.02321
MOEA/D		0.99117	0.99587	0.99723	0.02145
ϵ-NSGA-II		0.98571	0.98872	0.99131	0.72123

Figure 3.16 shows the parallel coordinate plot. The figure clearly shows that I-DBEA is able to find a widely distributed set of non-dominated points for 10-objective GAA design problem.

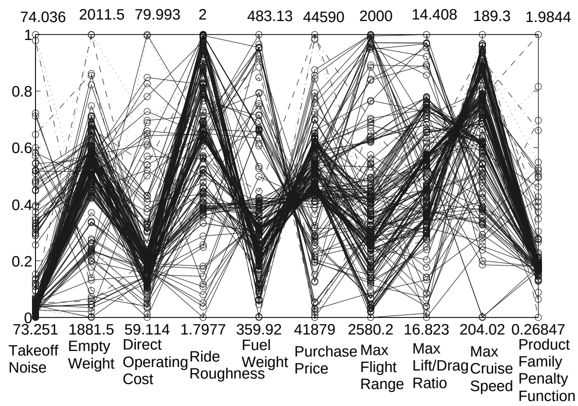


Figure 3.16: Parallel coordinate plot of the approximation of Pareto set produced by I-DBEA.

3.4 Decomposition Based Quantum Genetic Algorithm

While conventional decomposition based EAs have been applied to different many -objective optimization problems, their performance is largely dependent on the size of population. In most of the EAs, the size of the population is the same as the number of reference directions. Since the number of reference directions grows rapidly with increase in the number of objectives (assuming even a constant spacing), it becomes computationally impractical to evolve a large population of solutions. Although maintaining a large external archive and a small active population is a possibility, one needs to resolve two fundamental problems – (a) how to select parents and where to select them from, and (b) what recombination operators to use such that solutions can be created spanning the entire Pareto set.

In an attempt to resolve the aforementioned problems, a quantum representation of solutions is used. Quantum inspired genetic/evolutionary algorithms have long been in existence and have been applied for the solution of single and bi-objective discrete optimization problems [123, 124]. Unlike GAs or EAs, where a solution is represented using binary or real variables, solutions in quantum inspired models are represented using a string of Q-bits. A Q-bit is defined by a pair of variables α, β where α^2 denotes the probability of the bit to be found in state “1” and β^2 denotes the probability of the bit in its state of "0". Since a Q-bit individual represents the linear superposition of all possible states probabilistically, diverse individuals can be generated during the evolutionary process [125, 126]. A variation operator is commonly used to update α and β values of the Q-bits and solutions are generated via observations of these Q-bits. In the context of many-objective optimization, to the best of knowledge there are no reports on the use of quantum models to deliver solutions spanning the entire Pareto front. In [125], a quantum inspired algorithm was used to solve many-objective DTLZ test problems via preference articulation, wherein solutions along a preferred direction were identified. There is significant difference in the hypervolume values obtained for the DTLZ problems using the quantum approach [125] and theoretical values (computed using continuous variables).

3.4.1 Proposed Decomposition based Quantum Genetic Algorithm

The pseudo code for quantum GA is presented in Algorithm 3.2 and the components of the algorithm are discussed in the subsequent subsections except the generation of reference directions which has been described earlier in 3.2.1. The rest of the components i.e., quantum representation of solutions, diversity and convergence control via distance measures, variation operator, and adaptive epsilon scheme to deal with the constraints of the problem are described next.

3.4.2 Quantum representation of solutions

A solution is represented using a string of Q-bits. Each Q-bit is represented as follows.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3.12}$$

where α and β are complex numbers. $|\alpha|^2$ and $|\beta|^2$ are the probabilities of the Q-bit found in the state of "1" and "0" respectively. The j^{th} individual of the population

Algorithm 3.2 DQGA

Require: Gen_{max}: maximum number of generations
W: number of reference points
P: the number of Q-bit individuals

Generate W reference points and W reference directions.
Initialize the quantum population
$P_j = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_{\text{NQB}} \\ \beta_1 & \beta_2 & \cdots & \beta_{\text{NQB}} \end{array}\right]$
NQB represents the number of Q-bits (chromosome length) for $j = 1, 2, \ldots, P$ individuals
Observe P individuals resulting in P solutions that become the members of the initial population.
Evaluate these initial solutions, compute the ideal point $\mathbf{z} = (f_1^{\text{min}}, f_2^{\text{min}}, \ldots, f_M^{\text{min}})$, intercepts a_i for $i = 1$ to M and scale the individuals of the population.
Assign P individuals to the first W reference directions as members of the reference set.
while $gen \leq \text{Gen}_{\text{max}}$ **do**
  for $i = 1 : P$ **do**
    Generate a random order of W solutions.
    Observe i^{th} quantum individual resulting in a child solution.
    Check if the child solution is non-dominated with respect to the reference set.
    Identify if this child solution can replace a solution in the generated order of reference directions using d_1 and d_2 measures.
    If the above replacement is successful, update the Q-bits of the i^{th} solution using the variation operator.
    Update the ideal point ($\mathbf{z}$), intercepts and re-scale the individuals of the reference set.
  end for
end while

represented using Q-bits will assume a form:

$$Q_j = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_{\text{NQB}} \\ \\ \beta_1 & \beta_2 & \cdots & \beta_{\text{NQB}} \end{array}\right] \tag{3.13}$$

where $|\alpha_i|^2 + |\beta_i|^2 = 1, i = 1, 2, \ldots, \text{NQB}$. NQB denotes the total number of Q-bits required to represent the solution.

The above Q-bit representation can represent 2^{NQB} discrete solutions in the search space. The process of observing a quantum solution results in a binary solution. During the process of observation, a uniform random number (r_i) lying between 0 and 1 is generated for every Q-bit. If (r_i) ¡ $|\alpha_i|^2$, the corresponding binary bit is set to 1, else it is set to 0. For a variable represented using m bits, the corresponding binary bits of the variable are decoded to a discrete value using the quantum levels.

$$ql_i = 0.5^i \tag{3.14}$$

$$Ql_i = \frac{ql_i}{\sum ql_i} \tag{3.15}$$

where $i = 1, 2, \ldots, m$. The decoded discrete variable is calculated as follows

$$x = \mathbf{b}^T \mathbf{Ql} \tag{3.16}$$

where $\mathbf{b}$ denotes the vector of binary bits representing the variable. The same procedure is repeated for all other variables.

A child solution is created by observing the Q-bits of the parent solution (i.e., an individual in the population). In the steady state form, if a child solution is non-dominated with respect to the reference set, it enters the reference set via a replacement. The child solution competes with all solutions in the reference set in a random order until it makes a successful replacement or have competed with all individuals in the reference set. If the distances of the r^{th} solution are denoted as $\{d_{1_r}, d_{2_r}\}$ and $\{d_{1_c}, d_{2_c}\}$ denotes the distances for the child solution along r^{th} reference direction, the replacement rule is as follows:

Algorithm 3.3 Comparison of two solutions

```
if d_{2_c} = d_{2_r} then
    if d_{1_c} < d_{1_r} then
        Child solution wins
    end if
else
    if d_{2_c} < d_{2_r} then
        Child solution wins
    end if
end if
```

The above replacement scheme does not need explicit assignment/association operations or neighborhood parameters.

3.4.3 Epsilon level comparison

The constraint handling approach used in this work is based on epsilon level comparison. An epsilon level comparison is conducted using the allowable violation measure. If two solutions have their constraint violation value less than this epsilon level, the solutions are compared based on their objective values. Such a constraint handling scheme has been demonstrated to be more efficient than *feasibility first* schemes.

3.4.4 Variation operator

The variation operator [123] is used to update α and β values of the Q-bits of the individuals in the population P. Individuals in the population are selected in order and observed. The process of observing a quantum individual results in a child solution (C) represented using a binary vector of size NQB. The child solution (C) is compared with individuals in the reference set for possible replacement dictated by the distance measures d_1 and d_2. If the child replaces a solution (S) in the reference set, the Q-bits of the individual solution (i.e., the individual in the population that resulted in this child solution via observation) undergoes an update via the rotation gate.

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} cos(\Delta\theta) & -sin(\Delta\theta) \\ sin(\Delta\theta) & cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{3.17}$$

The value of $\Delta\theta$ is based on the following lookup table. In the table, $S_{B_{rj}}$ denote the j^{th} binary bit of the r^{th} solution in the reference set and $C_{B_{rj}}$ denotes j^{th} binary bit of a child solution being evaluated along r^{th} reference direction. The objective function $f(C)$ and $f(S)$ refer to the value (either of d_1 or d_2 distances) which has led to the child winning the competition over the existing solution along the r^{th} reference direction.

In this study $\theta_1 = 0, \theta_2 = 0, \theta_3 = 0.01\pi, \theta_4 = 0, \theta_5 = -0.01\pi, \theta_6 = 0, \theta_7 = 0, \theta_8 = 0$ have been used.

Table 3.15: Lookup table

$S_{B_{rj}}$	$C_{B_{rj}}$	$f(S) < f(C)$	$\Delta\theta_i$
0	0	false	θ_1
0	0	true	θ_2
0	1	false	θ_3
0	1	true	θ_4
1	0	false	θ_5
1	0	true	θ_6
1	1	false	θ_7
1	1	true	θ_8

3.5 Experimental Results

In this section, the results of proposed decomposition based quantum genetic algorithm is presented for discrete formulations of the DTLZ2 [127] problem with 2, 3, 5 and 8 objectives and number of variables set to 10. The reference directions were computed using s values of 20, 12, 5 and 4 resulting in 21, 91, 126 and 330 reference directions for the 2, 3, 5 and 8 objective formulations of the problem respectively. The α and β of the Q-bits were initialized to $1/\sqrt{2}$. In order to investigate the effects of fidelity on the performance, the problems were solved using 7 and 14-bit quantum representation. The effect of population size is also studied using a fixed number of function evaluations. Two population sizes have been used in the study i.e., one with 5 individuals and the other with W individuals (where W is the number of reference directions) for all the problems. The results are based on 30 independent runs and the hypervolume metric [128] is used as a measure of performance. The theoretical values of hypervolumes (based on real valued DTLZ2 test problems) are listed to aid a more objective comparison.

3.5.1 Performance on Unconstrained DTLZ2 Problem

The best, median and worst hypervolume measures using 7-bit and 14-bit representation is presented in Tables 3.16 and 3.17. A population of 5 individuals was evolved and the maximum number of function evaluations were set to 5000, 15000, 20000 and 25000 for 2, 3, 5 and 8 objective formulations respectively.

Table 3.16: Hypervolume statistics for DTLZ2 problem using 7-bit representation by DQGA

Objectives	Gen_{max}	Popsize	Best	Median	Worst
2	1000	5	0.1928	0.1925	0.1915
3	3000	5	0.3865	0.3763	0.3614
5	4000	5	0.5474	0.5353	0.5121
8	5000	5	0.6573	0.6456	0.6136

Table 3.17: Hypervolume statistics for DTLZ2 problem using 14-bit representation by DQGA

Objectives	Gen_{max}	Popsize	Best	Median	Worst
2	1000	5	0.1900	0.1883	0.1835
3	3000	5	0.3883	0.3789	0.3732
5	4000	5	0.5617	0.5470	0.5345
8	5000	5	0.6728	0.6525	0.6247

The theoretical hypervolumes for the continuous valued DTLZ2 problems with 2, 3, 5 and 8 objectives are 0.2146, 0.4764 and 0.8355 and 0.9841 respectively. There is marginal improvement using 14-bit representation over the 7-bit representation for the problems with larger number of objectives.

The same set of experiments were repeated with the population size of W i.e., same as the number of reference directions. The maximum number of function evaluations were held constant at 5000, 15000, 20000 and 25000 for the 2, 3, 5 and 8 objective formulations respectively. The results for the 7-bit and 14-bit representation is presented in Table 3.18 and Table 3.19. One can observe that there is no significant difference in the performance i.e., the small population of 5 individuals is able to deliver well spread and well converged set of solutions spanning the hyper surface.

The association mechanism (i.e., solutions to each reference direction) for a 3-objective DTLZ2 problem is presented in Figure 3.17. The figure shows the associations of the

Table 3.18: volume statistics for DTLZ2 problem using 7-bit representation by DQGA

Objectives	Gen_{max}	Popsize	Best	Median	Worst
2	238	21	0.1901	0.1876	0.1766
3	165	91	0.3821	0.3758	0.3626
5	159	126	0.5640	0.5472	0.5335
8	76	330	0.6728	0.6525	0.6247

Table 3.19: Hypervolume statistics for DTLZ2 problem using 14-bit representation by DQGA

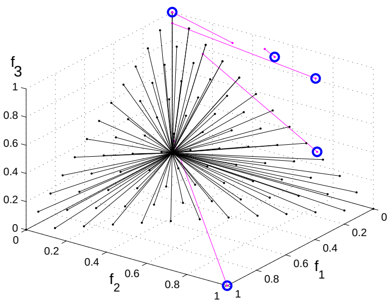
Objectives	Gen_{max}	Popsize	Best	Median	Worst
2	238	21	0.1901	0.1876	0.1766
3	165	91	0.3821	0.3758	0.3626
5	159	126	0.5640	0.5472	0.5335
8	76	330	0.6676	0.6524	0.6302

solutions in generation 1, 1500 and 3000 using 91 reference points with a population size of 5. One can observe that although initially the association is random (assigned to first 5 reference directions), the solutions automatically get associated with appropriate reference directions through the pressure induced by d_2.

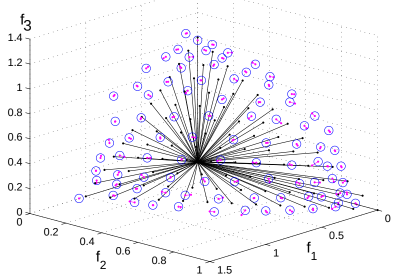
In order to observe the process of evolution, the average performance of the population is computed i.e., average of the d_1 and d_2 values for the individuals for DTLZ2 (3 objectives). One can observe from Figure 3.18, that the average d_2 converges to near zero (i.e., near perfect alignment to the reference directions) while the average d_1 measure stabilizes at around 1 indicating convergence to the theoretical Pareto front.

3.5.2 Constrained engineering design problems

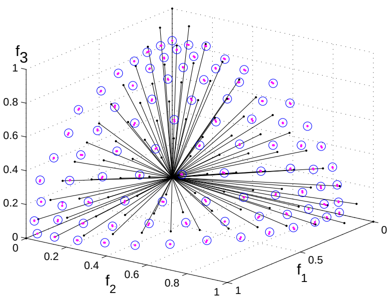
Since the performance of the proposed algorithm was reasonably good on unconstrained test problems for a small population size, its performance was further investigated using four constrained engineering design optimization problems i.e., welded beam problem [129] with four inequality constraints, speed reducer problem [130] with eleven inequality constraints, three-objective car side impact problem [119] with ten inequality



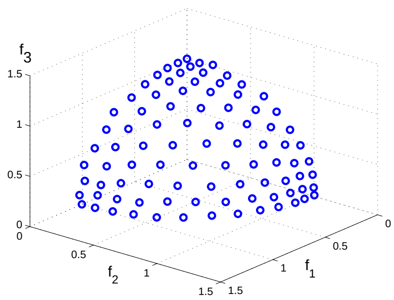
(a) normalized solutions at generation 1



(b) normalized solutions at generation 1500



(c) normalized solutions at generation 3000



(d) final non-dominated solutions

Figure 3.17: (a) the initial population of DTLZ2 test problem for number of objectives 3 normalized by ideal and intercepts (b) normalized solutions at generation 1500 (c) normalized solutions at final generation 3000 (d) final non-dominated solutions

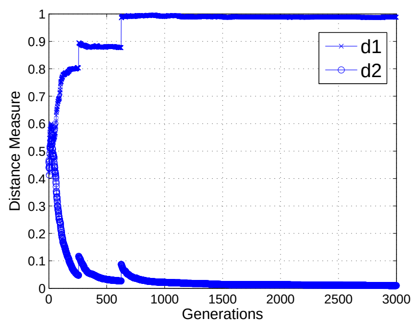


Figure 3.18: (a) the convergence of distance measure over the generations of DTLZ2 test problem for number of objectives 3

constraints and finally a five-objective water resource management problem [120] with seven inequality constraints.

Welded beam problem

The welded beam design optimization problem was originally formulated in [129]. The problem is to design a welded beam for minimum cost subject to a set of constraints. The beam is designed to support a force F = 6000 lbf and the objectives are to find the design with the minimum fabrication cost and minimum end deflection on the beam, considering four design variables i.e., thickness of the weld (x_1), length of the weld (x_2), thickness of the beam (x_3), and width of the beam (x_4) with the measurement unit in inches. These variables are represented using 9, 9, 10 and 10 quantum bits.

A small population size of 5 individuals were allowed to evolve with an aim to get solutions along 21 reference directions. The algorithms are run for 500 generations and the final non-dominated front is shown in Figure 3.19 with an approximated Pareto-front found from the earlier studies. Although the solutions of the problem are continuous the final non-dominated solutions obtain using QGA are almost close to the approximated Pareto-front with only 2500 function evaluations.

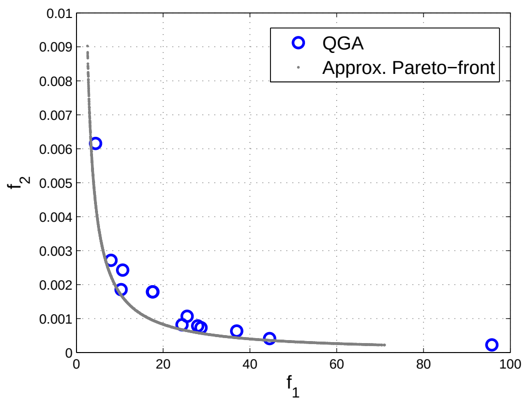


Figure 3.19: Solutions obtained using QGA on two-objective welded beam problem

Speed reducer problem

The speed reducer problem was first described in [130]. The problem involves 7 variables i.e., the face width x_1, module of teeth x_2, number of teeth on pinion x_3, length of the first shaft between bearings x_4, length of the second shaft between bearings x_5, diameter of the first shaft x_6, and diameter of the first shaft x_7 (all variables are continuous except for the integer values of variable x_3). The objectives of the problem are to minimize the total weight of the speed reducer as well as the normal stress on the first gear shaft. Since the speed reducer is to be made of the same material throughout, the first objective is the same as minimizing the total volume. The design variables are represented using 7, 4, 11, 7, 7, 7 and 6 quantum bits.

A small population size of 5 individuals were allowed to evolve with an aim to get solutions along 21 reference directions. The algorithms are run for 1500 generations and the final non-dominated front is shown in Figure 3.20 with an approximated Pareto-front found from the earlier studies. Since a fewer bits are used in the representation, the nondominated front is marginally inferior to the Pareto-front obtained using a continuous variable formulation.

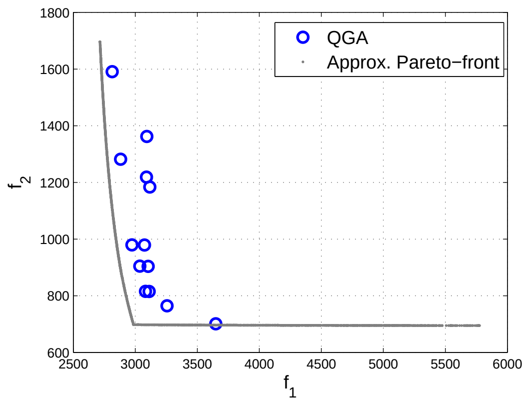


Figure 3.20: Solutions obtained using QGA on two-objective speed reducer problem

Car side impact problem

The problem involves minimization of the weight of a car, the pubic force experienced by a passenger and the average velocity of the V-Pillar responsible for bearing the impact load subject to the constraints involving limiting values of abdomen load, pubic force, velocity of V-Pillar, rib deflection etc [119]. The design variables for this problem are represented with 7, 7, 7, 7, 8, 7 and 7 quantum bits.

A small population size of 5 individuals were allowed to evolve with an aim to get solutions along 50 reference directions. The algorithms was run for 4500 generations and the final non-dominated front is shown in Figure 3.21 with an approximated Pareto-front found from the earlier studies. One can see the final non-dominated solutions are on the approximated Pareto-front. Since fewer bits were used, the solutions lie on a segment of the original front.

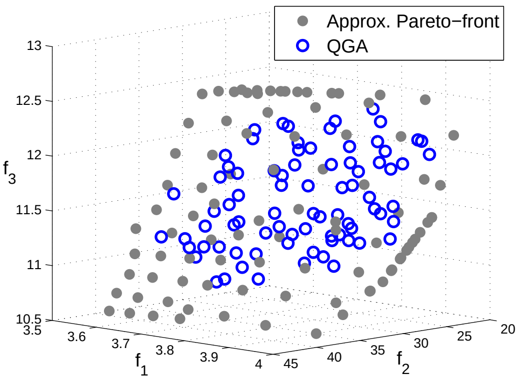


Figure 3.21: Solutions obtained using QGA on three-objective car side impact problem

Water resource management problem

This is a five objective problem having seven constraints taken from the literature [131]. The parallel coordinate plot generated using the proposed algorithm is presented in Figure 3.22. The best hypervolume value across 30 runs is $4.382e-2$. The objectives have been normalized using a reference point of [75373, 1350, 2.8535e+006, 1.2627e+007, 3.5192e+005]. A small population size of 5 individuals were allowed to evolve with an aim to solutions along 126 reference directions. Figure 3.23 shows a parallel plot with a set of known reference solutions of 2429 [121] which produce the hypervolume of 5.108e-2 after normalizing the same reference point.

3.6 Summery of Overall Performance

In this chapter, the challenges involved in dealing with many-objective problems are highlighted and, to enhance existing methods, two different approaches have been

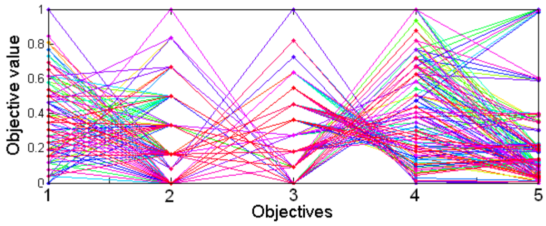


Figure 3.22: Parallel plot of the solutions obtained using DQGA

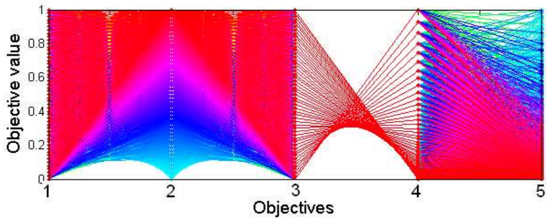


Figure 3.23: Parallel plot of the known reference set of 2429 solutions.

introduced. The first is the development of a novel decomposition based method using preemptive distance comparison while the second relates to the development of a quantum genetic algorithm for many-objective optimization. The major contributions of the chapter are summarized below.

Decomposition Based Evolutionary Algorithm

Firstly, the superior performance of decomposition based approaches over non-dominance based schemes is highlighted. The performance of I-DBEA is compared with non-dominated sorting based scheme NSGA-II for DTLZ1 problems with 3, 5, 8, 10 and 15 objectives. The mean value of g (i.e., a multi-modal function) as shown in Figure 3.24 clearly highlights the ability of decomposition based algorithms to deal with problem with 5 or more objectives.

Secondly, the performance of I-DBEA is compared with various state of the art approaches using seven widely studied benchmark problems [DTLZ1-DTLZ4, DTLZ5-(I, M), WFG, $WFG3_{conv}$], and three engineering design problems (car side impact, water resource and General aviation aircraft (GAA) design problem). The results clearly indicate the superiority of the proposed algorithm over existing forms. Among the 20 test instances of four DTLZ problems, I-DBEA obtained better results in 15 instances

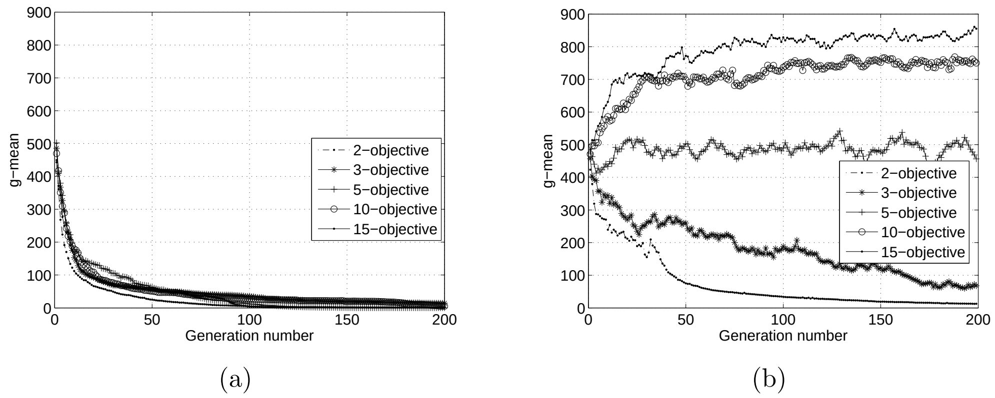


Figure 3.24: Convergence to the problem of DTLZ1 using (a) decomposition based algorithm (e.g. I-DBEA) (b) non-dominated based algorithm (e.g. NSGA2)

while DBEA-Eps was better in only 3 instances. In the context of WFG problems, once again the performance of I-DBEA is clearly better than MOEA/D-PBI. The ability to deal with concave, convex, mixed and degenerate problems with up to 15 objectives is showcased using the examples. Since the solutions obtained using I-DBEA is often of better quality, identification of redundant objectives is likely to more accurate as demonstrated using DTLZ5-(I, M) problems.

The success of the epsilon based constraint handling scheme is highlighted using the GAA problem which is known to pose problems to existing algorithms due to tight constraints.

I-DBEA attempts to identify solutions along a set of uniformly distributed reference directions. In the context of disconnected Pareto fronts, there may not be a solution along a reference direction. The behavior of the approach is illustrated using a modified bi-objective ZDT3 [9] problem. The original problem has the limits of the Pareto front with f_1 between 0 and 0.852 and f_2 between -0.773 and 1. The problem is modified as below to ensure positive limits of f_2.

- ZDT3

$$
\begin{aligned}
&Minimize\\
&f_1(x) = x_1\\
&f_2 = g(x)\left[1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)} sin(10\pi x_1)\right] + 1
\end{aligned}
\tag{3.18}
$$

where, $g(x) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)}$ and $x = (x_1, x_2, ..., x_m)^T \in [0, 1]$. Its PF is disconnected and the value of n is 10.

The performance of I-DBEA and MOEA/D-PBI is computed using 30 independent runs. A population size of 21 is evolved over 1500 generations. The performance is measured by computing the non-dominated (ND) solutions obtained by the individual algorithms. Table 3.20 shows the number of ND solutions obtained by I-DBEA and MOEA/D-PBI.

Table 3.20: Performance metric : Number of non-dominated (ND) solutions (Std)

Algorithm	Population size	ND solutions (Std)
I-DBEA	21	19 (3.6056)
MOEA/D-PBI	21	17 (4.1032)

Figure 3.25 shows the alignment of the solutions to the reference directions. It is clear that some of the solutions have d_2 values significantly greater zero indicating presence of disconnected front. Figure 3.26 shows the final non-dominated solutions achieved by both the algorithms.

Decomposition based Quantum Genetic Algorithm

In this chapter, a novel steady state decomposition based quantum genetic algorithm is proposed for the solution of unconstrained and constrained many-objective optimization

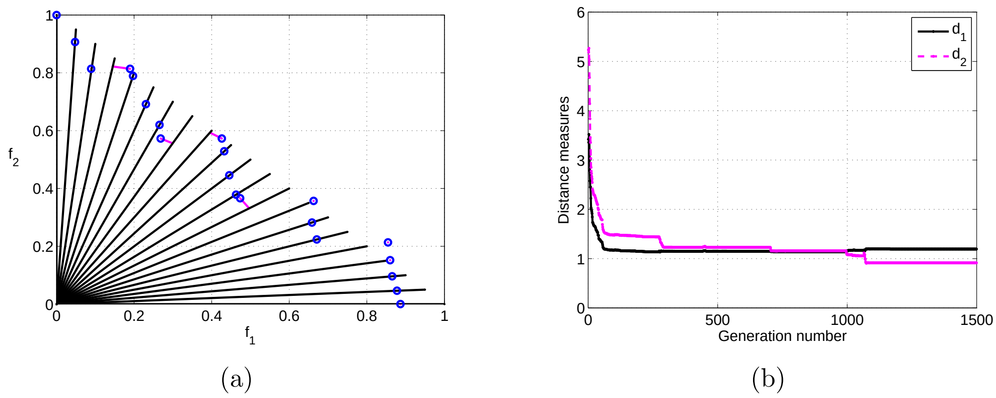


Figure 3.25: (a) final solutions associated with the reference directions in the normalized plane and (b) the convergence to the problem of ZDT3

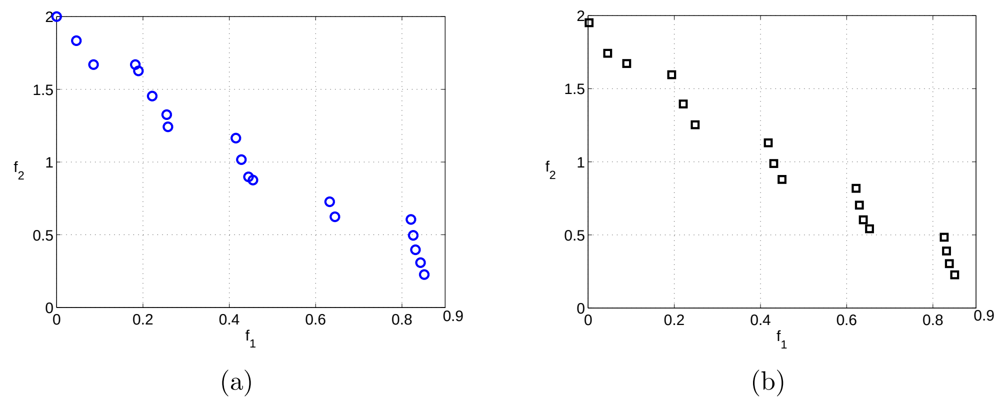


Figure 3.26: Non-dominated solutions obtained for ZDT3 using (a) I-DBEA and (b) MOEA/D-PBI

problems. While there are different approaches to solve many-objective optimization problems, the goal here is to develop a method that delivers well converged and well spread set of solutions spanning the entire Pareto surface.

In the context of the above goal, decomposition based approaches have been leading the race. Such approaches use an underlying sampling scheme to generate a set of uniformly distributed reference directions and thereafter evolve a population of solutions. Each individual in the population attempts to traverse along a particular reference direction. The number of such reference directions increase rapidly with increasing number of objectives which in turn requires larger population sizes. In order to alleviate this problem, a quantum representation of solutions is adopted. The solutions represented as strings of Q-bits offers the potential to represent multiple solutions during the course of observation. It is clear from the examples studied in this chapter, that evolution of a population of 5 quantum solutions result in the same performance (based on hypervolume) as the evolution of a population of W quantum solutions, where W is the number of reference directions. This offers now the possibility to solve many objective optimization problems with a small population size that does not scale up with increasing number of objectives.

The Q-bits of a quantum population are updated using Q-gates. The Q-bits of the solutions are updated based on the success of the child solution generated through the process of observation. While a simple rotation operator has been used in this study, other Q-gates and update rules could have been used. Furthermore, no attempts have been made to tune the $\Delta\theta_i$'s. While a single observation is used to update the Q-bits in the present study, more complex multi-observation updates can be designed.

Whenever a child solution is created via observation, it attempts to replace an existing solution in the reference set. An opportunity is offered to the child solution to replace a solution in the reference set. Multiple replacements are prohibited to ensure adequate diversity of the reference set.

The other important feature of this algorithm is its ability to automatically align solutions to the reference directions. The alignment pressure is induced via d_1 and d_2 measures the scheme does not involve any additional parameter or complex association or niching operation as in M-NSGA-II [111]. The course of evolution of the reference set for the DTLZ2-3 i.e., 3 objective optimization problem clearly illustrates this.

The effects of representing a solution using 7 and 14 bits have been studied in the context of DTLZ2 problems. There is marginal benefit in using 14-bit representation over 7-bit representation for the above problems. It is important to highlight that the hypervolumes are close to theoretical values although the gap marginally increases with increasing number of objectives. The theoretical hypervolumes are based on continuous variable representation. The choice of the number of bits essentially dictates by the number of states of the variable which in turn is dependent on the range and the step size of the variable.

The preliminary results indicate that there are advantages of quantum representation that can be exploited to alleviate some of the existing problems associated with the state of the art many-objective optimization algorithms.

Chapter 4

Robust Many-objective Optimization

Part of this work has previously appeared in **Asafuddoula, M.**, Ray, T. and Sarker, R.,"A decomposition based evolutionary algorithm for many-objective optimization with systematic sampling and adaptive epsilon control", in Proceedings of the Seventh International Conference on Evolutionary Multi-Criterion Optimization, vol. 7811 Lecture Notes in Computer Science, pp. 417-427, Springer, 2013.

Asafuddoula, M., Singh, H.K., and Ray, T., "Six-sigma robust design optimization using a many-objective decomposition based evolutionary algorithm", IEEE Transactions on Evolutionary Computation. (Under review).

Chapter Overview

This chapter addresses three key challenges facing robust design optimization i.e., correct formulation of the problem, accurate estimation of the 'robustness' measure and efficient means to identify such solutions with an affordable number of function evaluations. The proposed formulations consider robustness from two perspectives - reliability against failure and reliability against performance deterioration. The problem is posed as a many-objective optimization problem and a decomposition based evolutionary approach *is used for solving it. The performance of the proposed approach and effects of various formulations are illustrated using two numerical examples and five engineering problems.*

4.1 Literature Review

The notion of *uncertainty* is omnipresent in any real-world problem. In the context of design optimization, such uncertainties emerge from varying loading conditions, material imperfections, inaccuracies in analyses/simulations, imprecise geometries, manufacturing precision or even actual product usage. For practical implementation, designs need to be robust, i.e., less sensitive in the presence of uncertainties. While there could be different sources of uncertainties in a problem (environmental factors, design variables, performance estimates etc.), this work focuses on solving problems with uncertainties in design variables. This class of problems is frequently encountered in various engineering applications as reported in the past [132, 133, 134, 135, 136, 137, 138, 139, 140, 141].

It is evident that optimization based solely on the maximization of performance (i.e., ignoring the landscape of the objective/constraint functions in the vicinity of a solution) is incapable of identifying robust solutions. Firstly, global optimal solution(s) of a performance maximization problem may lie on a constraint boundary. With marginal deviation in the variable values, such solutions could easily violate one or more constraints. Secondly, such global optimal solutions may lie on a *very narrow peak* of the performance function, wherein a slight variation in the value of the variables could result in significant deterioration in performance.

The first aspect relates to *feasibility robustness* and is commonly dealt with using additional constraints [142, 143, 144, 145], wherein a set of solutions satisfying a prescribed level of feasibility robustness is identified. The second aspect relates to *performance robustness*, i.e., robustness in terms of performance deterioration. It is either modeled using additional constraints [146] or incorporated as an objective via variance measures [147, 142, 143, 145]. There is also a differentiation in the notion of performance used in the robust formulations. Some studies use the performance function at the given (mean) value [146, 147] while others use the expected value of the performance function [146, 142, 143, 144, 145]. The latter form of performance measure seems to be more practical and widely adopted.

In this chapter, robustness with respect to constraints (*feasibility robustness*) and objectives (*performance robustness*) are studied using four different formulations. The key differences between the formulations are discussed using two test problems. Since the proposed formulations use additional objectives to deal with feasibility robustness and performance robustness, a many-objective optimization algorithm is used to solve the problem. Unlike previous approaches which can identify a set of tradeoff solutions satisfying a prescribed feasibility robustness criteria, the proposed approach offers the complete set of tradeoff solutions i.e., solutions spanning various levels of feasibility robustness and performance robustness simultaneously in a single run. Such a set of tradeoff solutions is of practical value as one can clearly observe the performance/cost implications of delivering solutions at various robustness levels.

Identification of robust solutions have always been a problem of practical interest. In the past, conservative designs were generated by adding a factor of safety to constraints/variables [5]. In recent years, more involved research has focused on development of approaches to quantify and identify robust optimal solutions. The studies can be broadly classified into three areas which deal with (1) formulation of a robust optimization problem, (2) quantification of robustness and (3) means to deal with such problems with affordable computing resources, i.e., the search algorithms. Throughout this discussion, minimization of the objective function(s) is assumed for consistency.

Robust optimization problem formulation

In the context of problem formulation, a number of different approaches have been proposed in the literature to deal with the aspects of feasibility robustness and performance robustness. Features of various robust formulations reported in literature are summarized in Table 4.1.

Constraint modification is one of the simplest forms to deal with feasibility robustness [142, 143, 148]. A constraint $g < 0$, is translated to $\mu_g + n\sigma_g < 0$, where n defines the level of robustness in six-sigma terminology (discussed later in 4.2). The inclusion of the term $n\sigma$, essentially translates the constraint boundary inwards (i.e., reduces the feasible space), thereby ensuring that the solutions do not violate the (mean) constraint boundaries of the problem. Apart from modifying the original constraints, some other studies have suggested use of additional constraints, e.g. probability of violation of the constraints [149, 150] or reliability index [151, 145, 152, 153]. Sequential reliability assessment combined with deterministic optimization has also been reported [154]. It is important to highlight that all the aforementioned techniques deliver solutions for a prescribed level of feasibility robustness.

To deal with performance robustness, the approaches either use aggregation or include additional objectives or constraints in the formulation. For example, in [155, 156, 148], a simple aggregate function i.e., $\mu_f + \sigma_f$ (or $\mu_f + k\sigma_f$) was used. A weighted composite function was also used in [143, 157] of the form $\lambda\mu_f^2 + (1-\lambda)\sigma_f^2$ (or $\lambda\mu_f + (1-\lambda)\sigma_f$), where the factors could be varied to emphasize/deemphasize the effects of the mean and the standard deviation terms. Another different formulation was presented in [158, 159] of the form $\alpha\frac{\mu_f}{\mu_{f^*}} + (1-\alpha)\frac{\sigma_f}{\sigma_{f^*}}$. Alternate propositions of including robustness as a separate objective also appears in [160, 161, 162, 147, 163, 164, 155, 165, 142, 144] wherein a robustness measure/index was used as an additional objective during the course of search. Often to ensure performance robustness, expected performance measures are used in lieu of the original performance function [146, 144]. Another method of enforcing performance robustness appears in the works of [166, 167, 146], wherein a constraint of the form $\frac{||f^p(x)-f(x)||}{||f(x)||} < \eta$ was added to the original problem, where f^P denotes the *effective value* i.e., the worst function value among chosen neighborhood solutions and η is a user defined parameter.

It can be observed from Table 4.1 that while some of the works in the past have considered performance robustness as objective, none of the works have included both feasibility and performance robustness as objectives so as to achieve solutions with varying levels of robustness in a single run.

Quantification of robustness

Most of the approaches for robustness quantification reported in the literature utilize the *expected* value and/or variance of the performance function [156]. When the analytical function is known, the expected function value can be calculated as the integral $\int f(x)w(x)dx$, and certain analytical techniques [157, 170, 171, 172, 173, 144, 174, 175, 176] could be used to quantify the robustness. However, for most optimization problems, such expressions may not be available, and stochastic sampling based approaches have to be used instead for the estimates [177, 178, 179, 180, 137, 138, 181, 182]. The major downside lies with the mechanism for estimating the *expected* fitness as it requires a large number of samples to compute the expected value with good accuracy. For a finite number of samples, *explicit averaging* [183, 184, 138, 185, 137, 186] or *implicit averaging* [187, 188, 189] may be used for estimates. To save on function evaluations, some studies used metamodels [162, 190] for calculating the expected values instead of the original function.

In the context of reliability against failure (*feasibility robustness*), probabilistic models have been proposed, wherein the constraints of the problem are transformed to chance constraints as follows:

$$P(g_j(\mathbf{d}, \mathbf{x}) \geq 0) \geq R_j, j = 1, 2, ..., J \tag{4.1}$$

where R_j is the desired probability of constraint satisfaction of the jth constraint. Often

Table 4.1: Common formulations to identify robust solutions*

Reference Work	Robust formulation	Type of robustness	Robustness as an additional objective	Quantification of Robustness	Ability to generate tradeoff feasible robust solutions in a single run	Optimization Method	Single-/ Multi-objective
K. Deb, H. Gupta [166]	Min $[f_1, f_2..., f_M]$ Subject to $\frac{\|\|f^p(x)-f(x)\|\|}{\|\|f(x)\|\|} \leq \eta$ Or Min $[f_1^{eff}, f_2^{eff}..., f_M^{eff}]$	Robustness in performance (as additional constraint)	No	Expected measure	No	Real Parameter GA	(MO, C)
Y. Jin and B. Sendhoff [147]	Min $[f_1 = f, f_2 = \sigma_f]$ Subject to $g_j \leq 0$	Robustness in performance (as additional objective)	Yes	Variance measure	Yes, w.r.t. performance	Evolutionary Multi-objective approach	(SO, C)
W. Chen et. al. [142]	Min $[(\mu_f, \sigma_f^2)]$ Subject to $E[g_j(x,z)] + n\sigma_{g_j} \leq 0$	Feasibility and performance robustness	Yes	Expected and variance measures	Yes, w.r.t performance	Compromise DSP	(MO, C)
G. Sun et. al. [143]	Min $[(\mu_{f_1}, \sigma_{f_1}), (\mu_{f_2}, \sigma_{f_2}), ..., (\mu_{f_M}, \sigma_{f_M})]$ Subject to $\mu_{g_j} + n\sigma_{g_j} \leq 0$	Feasibility and performance robustness	Yes	Sigma level based measure	Yes, w.r.t performance	PSO	(MO, C)
S. Sundaresan et. al. [144]	Min $E[f]$ Subject to $E[g_j] \leq 0, E[h_j] = 0$	Feasibility robustness	No	Expected measure	No	Mathematical programming	(SO, C)
Z. Wang et. al. [148]	Min $[\mu_f + k\sigma_f]$ Subject to $\mu_{g_j} + n\sigma_{g_j} \leq 0$	Feasibility and performance robustness	No	Aggregation of expected and variance measure	No	Mathematical programming	(SO, C)
Z. P. Mourelatos and J. Liang [168]	Min $[(\mu_f, \Delta R_\sigma = \sigma_{R2} - \sigma_{R1})]$ Subject to $Prob\{g_j \leq 0\} \geq \alpha_i, i = 1, 2, ..., M$	Feasibility and performance robustness	Yes	Expected and variance measures	Yes, w.r.t performance	Mathematical programming	(MO, C)
B. D. Youn et. al. [169]	Min $[(\frac{\mu_H - h_t}{\mu_{H_0} - h_t})^2 + (\frac{\sigma_H}{\sigma_{H_0}})^2]$ or $[(sgn(\mu_H))(\frac{\mu_H}{\mu_{H_0}})^2 + (\frac{\sigma_H}{\sigma_{H_0}})^2]$ or $[(sgn(\mu_H))(\frac{\mu_1/H}{\mu_1/H_0})^2 + (\frac{\sigma_1/H}{\sigma_2/H_0})^2]$ Subject to $Prob\{g_j \leq 0\} \geq \alpha_i, i = 1, 2, ..., M$	Feasibility and performance robustness	No	Expected and variance measures	No	Mathematical programming	(MO, C)

*SO: single objective; MO: multi-objective; C: constrained. For complete details on the robust formulations, please refer to the cited publications.

such probabilistic constraints are substituted with deterministic constraints as follows:

$$1 - \frac{r_j}{N} \geq R_j, j = 1, 2, ..., J \quad (4.2)$$

where N is the sample size and r_j is the number of failures among N samples. Although the method is simple, the appropriate sample size required to estimate the quantity r_j often becomes computationally prohibitive. Some studies have focused on reducing the sample size via Latin hypercube sampling [191, 192], importance sampling [193], directional sampling [194], Taylor series expansion [195, 196] and polynomial chaos [197, 179]. While the above discussed methods rely on some form of sampling, there are also reports of sensitivity analysis based on the gradient information [160, 168]. Other works include computation of sensitivity via non-gradient forms, wherein, additional constraints are imposed [160] leading to the notion of Acceptable Performance Variation Region (AVPR). For such formulations, it is also necessary to compute the Sensitive Region (SR) before a solution violates AVPR. Since the sensitive region (SR) could be asymmetric, i.e., a solution could be more sensitive in one direction of variation in a variable Δx, but less sensitive in others, a Worst Case Sensitive Region (WCSR) model is often required. The process of identifying WCSR is itself a complex optimization problem. While the above methods discussed so far have their origins rooted in the field of robust optimization, there are methods in the domain of reliability optimization, where the reliability of a solution is computed by determining its distance from the closest constraint boundary. The approach is commonly known as "most probable point" (MPP) of failure [198]. MPP is known to be computationally expensive as it requires several loops of optimization. A number of variants have been proposed in recent years to reduce the computational expense involved, viz., Performance measure approach (PMA), fast performance measure approach (FastPMA), reliability index approach (RIA), fast reliability index approach (FastRIA) [149]. MPP based methods generally include the first order reliability method (FORM) and second order reliability method (SORM) [198, 199, 165]. The sensitivity analysis is achieved by simplifying the limit state function with the first order or second order Taylor expansion at the MPP or SORM. The SORM method is more accurate than FORM which requires a second order Taylor series approximation around the MPP of the limit state function.

Search algorithms

While the issues concerning the formulation of the problem and the measures of robustness have been discussed above, the outcome of a robust optimization exercise is also dependent on the efficiency of the underlying search strategy. In most of the works discussed in previous sub-section, population based stochastic techniques (e.g., [91]) have been used for optimization. When additional objectives are introduced in the formulation, the number of objectives could often be more than four. It is well reported in literature that such problems (with four or more objectives) cannot be efficiently solved using non-dominance based multi-objective optimization algorithms. Hence, popular multiobjective approaches such as NSGA-II [91], SPEA2 [95] etc. cannot be efficiently used if the formulation results in a many-objective problem. Modifying the selection pressure through the use of secondary metrics (e.g., substitute distance measures [96][97], average rank domination [98], fuzzy dominance [99, 200], ϵ-dominance [100] [101], adaptive ϵ-ranking [102] etc.) has so far exhibited only partial success in solving such problems. In all the above approaches relying on secondary measures, there is no guarantee that the final non-dominated set of solutions would span the entire Pareto surface uniformly.

The aim of this chapter is to seek enhancements in all three aspects presented above. To this effect, the following studies are presented:

1. Formulation: As clear from Table 4.1, some of the formulations used in the past to solve robust optimization problems deliver solutions for a prescribed level of robustness. Some others provide solutions with varying levels of robustness in a single run, but with respect to performance only. This chapter intends to formulate the problem so as to deliver solutions with varying levels of robustness

in a single run, both with respect to feasibility and performance. Four different formulations have been presented in order to cover some of the past formulations as well as the new one which delivers the aforementioned tradeoff set.

2. Quantification of robustness: While the concept of 6σ is well recognized as quantification of robustness for feasibility robustness, the same has been introduced for the objectives in order to quantify performance robustness. The user can prescribe the acceptable deviation from the average performance specification of a design, which is then used to calculate the sigma level in terms of performance robustness. This way of quantifying robustness helps comparing the robustness of solution with respect to each objective/constraint on a common scale (even though their raw values and standard deviations values may be of different orders). While a single value of upper specification limit(USL) can be used for single objective six-sigma robust formulations, a common USL measure cannot be used for problems involving multiple performance measures. For the case of multiple constraints and/or multiple objectives, minimum sigma level for a solution is considered as its overall sigma level, which helps in assuring this minimum level of feasibility/performance robustness on the solutions w.r.t. all objectives/constraints. The robustness measures are treated as additional criteria in the problem formulation.

3. Search algorithm: The addition of robustness criteria in order to formulate the robust optimization problem results in an increase in number of objectives, and often the resulting problem could have four or more objectives. In order to solve them efficiently, the use of *many-objective* algorithms is introduced. A steady state, decomposition based many-objective evolutionary algorithm, referred to as DBEA-r, is proposed to achieve this. The reference directions for the algorithm are generated using *systematic sampling*, and association of the solutions to these reference directions are based on two independent distance measures, one for convergence and the other for alignment/diversity. Constraints of the formulation

are handled using an adaptive epsilon level based scheme [115, 116].

4.2 Problem definition and robustness measure

A generic multi-objective optimization problem can be defined as follows:

$$
\begin{aligned}
&\underset{(\mathbf{d},\mathbf{x})}{\text{Minimize}} f_i(\mathbf{d},\mathbf{x}), i = 1, 2, ...M\\
&\text{Subject to}\\
&\qquad g_j(\mathbf{d},\mathbf{x}) \geq 0, j = 1, 2, ...p\\
&\qquad h_j(\mathbf{d}) = 0, j = p+1, p+2, ...p+q\\
&\qquad \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}\\
&\qquad \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}
\end{aligned}
\tag{4.3}
$$

where $f_1(\mathbf{d},\mathbf{x}), f_2(\mathbf{d},\mathbf{x}), f_3(\mathbf{d},\mathbf{x}), ...f_M(\mathbf{d},\mathbf{x})$ are the M objective functions. The functions involve a set of deterministic variables $\mathbf{d}$ and a set of uncertain variables $\mathbf{x}$. The number of inequality and equality constraints are denoted by p and q respectively. In practice, most engineering design optimization problems involve one or more variables of uncertain nature that is often represented using a probability distribution. In this chapter, the distribution of uncertain variables is assumed Gaussian, which reflects the behavior of majority of the variables in the context of engineering design. The variables are represented using their mean and standard deviation as $N(\mu_x, \sigma_x)$.

To convert an optimization problem to a robust optimization problem, one needs to adopt a robustness measure. In this work, the notion of six-sigma quality measure commonly used in the industry is adopted. This measure is discussed next, followed by the problem formulations.

4.2.1 Six-sigma quality measures

The notion of six-sigma refers a unit of six standard deviations between the process mean/*expected* value and the nearest specification limit as shown in Figure 4.1. It is assumed that when the ratio of *expected* value and the standard deviation is six, practically no items (2 defects per million) will fail to meet the specifications [155].

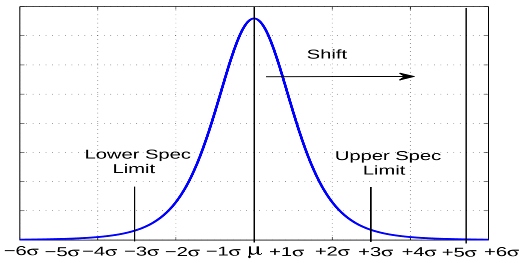


Figure 4.1: Normal distribution, 3-σ design

Table 4.2 shows the sigma level and the corresponding confidence interval that the performance of the solution will lie between the process mean and the specification limit.

Table 4.2: Sigma level on percent variation and defects per-million

Sigma Level	Confidence Interval (CI)	Defects per million
$\pm 1\sigma$	68.26	691462
$\pm 2\sigma$	95.46	308538
$\pm 3\sigma$	99.73	66807
$\pm 4\sigma$	99.9937	6210
$\pm 5\sigma$	99.999943	233
$\pm 6\sigma$	99.9999998	2

4.2.2 Problem formulation and robustness quantification

In this chapter, four different formulations for robust optimization are presented, covering aspects of both feasibility and performance robustness. Two different cases are studied, one using performance other using *expected* performance quantification measures. The formulations (*Form-1* to *Form-4*) are listed in Table 4.3. *Form-1* and *Form-3* consider feasibility robustness only, while *Form-2* and *Form-4* consider both feasibility and performance robustness. Performance value is directly used as an objective in *Form-1* and *Form-2* which, given the Gaussian variation in design variables, is equivalent to using the performance at the mean design point ($f(\mu_{\mathbf{x}})$). *Expected* values of the performance functions are used as objectives in *Form-3* and *Form-4*, which have been estimated by averaging over samples (μ_f) in the vicinity of design point with given uncertainties. Two robustness measures, namely $sigma_g$ and $sigma_f$ are introduced to quantify feasibility and performance robustness respectively.

- The term $sigma_g$ refers to the ratio of *expected* constraint value (μ_g) and standard deviation (σ_g) of constraint g. Since the constraints have been formulated as $g > 0$, the ratio $sigma_g = \mu_g/\sigma_g$ is nothing but $(\mu_g - 0)/\sigma_g$, which is a measure of how many standard deviations can be fit between the constraint boundary(0) and the given solution (see Figure 4.1). This quantity is positive for feasible solutions, and needs to be maximized for achieving high robustness.

- The term $sigma_f$ refers to the ratio of a user defined acceptable deviation σ_{f_0} and the standard deviation (σ_f) of objective f. For σ_f, unlike σ_g, the boundary is not zero but the (user prescribed) specification limit on how much deviation in the objective value is acceptable from the mean. Hence, the ratio $sigma_f = \sigma_{f_0}/\sigma_f$ denotes the number of standard deviations of the objective function that can be fit within the specification limit. Again, $sigma_f$ is intended to be maximized.

To ensure feasibility robustness, the quantity $sigma_g$ has to be positive and solutions with larger $sigma_g$ are more robust. If the value of $sigma_g$ is greater than a given value R_c, the value is truncated to R_c. It essentially means, the user is satisfied with the feasibility robustness level R_c and solutions having any higher robustness has the same preference as the one with R_c. A similar truncation strategy has been applied to $sigma_f$ (using R_f) to ensure performance robustness in *Form-2* and *Form-4*. In a problem involving multiple constraints, the *minimum* $sigma_g$ across all constraints is considered to represent the overall $sigma_g$ of the solution. This translates to measuring the sigma-level of constraint that is most likely to be violated; which is different from traditionally used six-sigma formulation where defects caused using *all* constraints together are considered. Same strategy has been adopted for $sigma_f$ for the case of multiple objectives. This way of quantifying robustness helps comparing the robustness of solution with respect to each objective/constraint on a common scale (even though their raw values and standard deviations may be of different orders), and at the same time assuring this minimum level of feasibility/performance robustness on the solutions with respect to each objective/constraint. For a six-sigma design considered in this chapter, both R_c and R_f are set to 6. *Form-1* and *Form-2* formulations are similar to those presented in [149], where objective value at mean point ($f(\mu_x)$) is considered as objective rather than the mean of the function ($\mu_{f(x)}$). These formulations have been included for completeness. However, in realistic situations, with given uncertainty in the variables, expected/average objective values for a design are more reflective of field performance compared to objective values at most likely point. It is clear from Table 4.3, that *Form-4* is the most comprehensive formulation that is capable of delivering the set of tradeoff solutions spanning the entire range of feasibility robustness and performance robustness.

4.3 DBEA for Robust optimization

Robust formulations presented above require solution of optimization problems involving additional objectives. The total number of such objectives can easily be more than four and hence a many-objective optimization algorithm is used in this study. The underlying algorithm is a decomposition based evolutionary algorithm (DBEA) described earlier in this dissertation.

Table 4.3: Different forms of robust formulation*

Robust form	Formulation Type	Robust formulation		Robust measure	TR^*
Form-1	Original objective function(s) with feasibility robustness	$\underset{(\mathbf{d},\mathbf{x})}{\text{Minimize}} f_i(\mathbf{d},\mathbf{x}), i = 1, 2, ...M$ $\underset{(\mathbf{d},\mathbf{x})}{\text{maximize}} f_{M+1}(\mathbf{d},\mathbf{x}) = \text{Min}(sigma_g, R_c)$ Subject to $sigma_g \equiv \text{Min}(\mu_{g_j(\mathbf{d},\mathbf{x})}/\sigma_{g_j(\mathbf{d},\mathbf{x})}) \geq 0$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}$	(4.4)	Sigma level based measure ($sigma_g$)	Yes
Form-2	Original objective function(s) with feasibility and performance robustness	$\underset{(\mathbf{d},\mathbf{x})}{\text{Minimize}} f_i(\mathbf{d},\mathbf{x}), i = 1, 2, ...M$ $\underset{(\mathbf{d},\mathbf{x})}{\text{maximize}} f_{M+1}(\mathbf{d},\mathbf{x}) = \text{Min}(sigma_g, R_c)$ $\underset{(\mathbf{d},\mathbf{x})}{\text{maximize}} f_{M+2}(\mathbf{d},\mathbf{x}) = \text{Min}(sigma_f, R_f)$ Subject to $sigma_g \equiv \text{Min}(\mu_{g_j(\mathbf{d},\mathbf{x})}/\sigma_{g_j(\mathbf{d},\mathbf{x})}) \geq 0$ Where $sigma_f \equiv \text{Min}(\sigma_{f_{0,i}(\mathbf{d},\mathbf{x})}/\sigma_{f_i(\mathbf{d},\mathbf{x})})$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}$	(4.5)	Sigma level based measure ($sigma_g$, $sigma_f$)	Yes
Form-3	Expected objective function(s) with feasibility robustness	$\underset{(\mathbf{d},\mathbf{x})}{\text{Minimize}}\ \mu_{f_i(\mathbf{d},\mathbf{x})}, i = 1, 2, ...M$ $\underset{(\mathbf{d},\mathbf{x})}{\text{maximize}} f_{M+1}(\mathbf{d},\mathbf{x}) = \text{Min}(sigma_g, R_c)$ Subject to $sigma_g \equiv \text{Min}(\mu_{g_j(\mathbf{d},\mathbf{x})}/\sigma_{g_j(\mathbf{d},\mathbf{x})}) \geq 0$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}$	(4.6)	Sigma level based measure ($sigma_g$)	Yes
Form-4	Expected objective function(s) with feasibility and performance robustness	$\underset{(\mathbf{d},\mathbf{x})}{\text{Minimize}}\ \mu_{f_i(\mathbf{d},\mathbf{x})}, i = 1, 2, ...M$ $\underset{(\mathbf{d},\mathbf{x})}{\text{maximize}} f_{M+1}(\mathbf{d},\mathbf{x}) = \text{Min}(sigma_g, R_c)$ $\underset{(\mathbf{d},\mathbf{x})}{\text{maximize}} f_{M+2}(\mathbf{d},\mathbf{x}) = \text{Min}(sigma_f, R_f)$ Subject to $sigma_g \equiv \text{Min}(\mu_{g_j(\mathbf{d},\mathbf{x})}/\sigma_{g_j(\mathbf{d},\mathbf{x})}) \geq 0$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}$ Where $sigma_f \equiv \text{Min}(\sigma_{f_{0,i}(\mathbf{d},\mathbf{x})}/\sigma_{f_i(\mathbf{d},\mathbf{x})})$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}$	(4.7)	Sigma level based measure ($sigma_g$, $sigma_f$)	Yes

*SO: single objective; MO: multi-objective; MaO: many-objective; C: constrained; the value of R_c and R_f is considered 6 to meet the six sigma quality; TR: is the ability to generate tradeoff feasible robust solutions in a single run.

4.4 Numerical examples

In this section, the key differences among *Form-1*, *Form-2*, *Form-3* and *Form-4* robust formulations are discussed with respect to the final set of solutions. Two simple numerical test problems are used for illustration, and subsequently the performance of the approach is further assessed using two single-objective, one multi-objective and a many-objective optimization problem.

Population sizes of 50, 91, 220, 330, 462, 462 have been used for 2, 3, 4, 5, 6 and 7 objective (including original(f)/expected(μ_f) and robustness ($sigma_g, sigma_f$) objectives) optimization problems respectively. The size of the population is set to be the same as the number of reference directions which are identified using Normal Boundary Intersection (NBI) [4]. The probability of crossover is set to 1 and the probability of mutation is set to $p_m = 1/D$, where D is the dimensionality of the problem. The distribution index of crossover and mutation are set to $\eta_c = 30$ and $\eta_m = 20$ as in [118]. The population is evolved over 100 generations. A sample size of 100 is used to compute the *expected* value and the standard deviation for given solution. Latin-hypercube Sampling with Gaussian distribution (*LHS-Gaussian*) is used to generate the samples around the solution in all studies.

4.4.1 Example-1 (robust single objective optimization)

Test Function

Function f is a one variable problem with five unequal peaks in the range $0 \leq x \leq 1$. It is defined as

$$f = \begin{cases} e^{-2ln2(\frac{x-0.1}{0.8})^2} |sin(5\pi x)|^{0.5} : & 0.4 < x \leq 0.6 \\ e^{-2ln2(\frac{x-0.1}{0.8})^2} sin^6(5\pi x) : & \text{otherwise.} \end{cases} \tag{4.8}$$

The problem is to minimize objective f subject to a constraint $(x - 0.1) \geq 0$. It can be seen from Figure 4.2 that the problem has four sharp local optima and one relatively robust optimum. The global optimum is located at $x = 0.1$ with the function value of -1.0. If the uncertain variable x is assumed to follow a Gaussian distribution with a $\sigma_x^2 = 5 \times 10^{-4}$, the robust minimum is be located at $x = 0.486$ with a function value of -0.715.

Figure 4.2 shows the solutions obtained using *Form-1* robust formulation. One can clearly observe solutions with varying levels of feasibility robustness, i.e., different values of $sigma_g$. For example, the solutions marked as A, B and C have $sigma_g$'s of 0.0012, 0.4720, 6.0 respectively. One can observe that there are no solutions between $sigma_g = 0.4720$ and 6 since solutions in this range have been dominated by solution C (which has $sigma_g = 6$, and whose objective value is only inferior to the solutions between A and B). As performance robustness has not been considered in the *Form-1* robust formulation, the values of $sigma_f$ lie within a small band i.e., between 0.3622 and 0.4309. The above listed $sigma_f$'s were computed using a user defined objective limit $\sigma_{f_{0,1}} = 0.101$ and 100 neighboring solutions using *LHS-Gaussian* sampling.

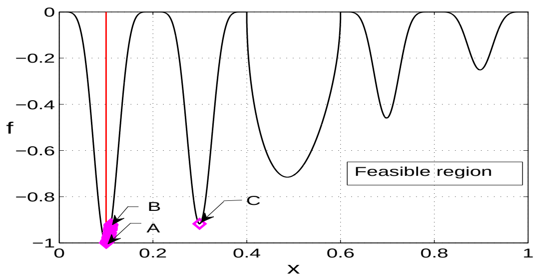


Figure 4.2: Solutions obtained using Form-1 robust formulation. The solutions are labeled as (x, f, $Sigma_g$, $Sigma_f$) i.e., A = (0.1000, -1.0000, 0.0012, 0.3952), B = (0.1104, -0.9220, 0.4720, 0.3622), C = (0.2994, -0.9172, 6, 0.4309).

For *Form-2* robust formulation, the tradeoff solutions are presented in Figure 4.3. Since $sigma_f$ is considered as an additional objective, two new solutions i.e., D and E emerge with $sigma_f = 3.3308$ and 6 respectively. One can notice the increased range of $sigma_f$ using this formulation when compared to the earlier form. Solutions C, D and E are feasibility robust solutions with $sigma_g = 6$.

In the next two formulations i.e., *Form-3* and *Form-4*, the *expected* value (μ_f) of performance function is used as objective. Once again *Form-3* considers feasibility

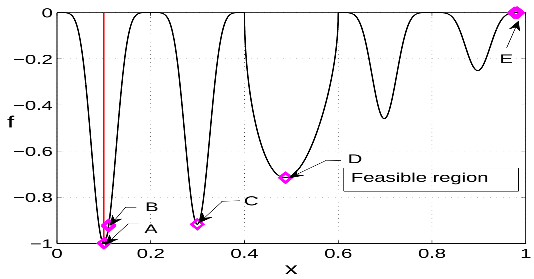


Figure 4.3: Solutions obtained using Form-2 robust formulation. The solutions are labeled as $(x,\ f,\ sigma_g,\ sigma_f)$ i.e., A = (0.1000, -1.000, 0.0012, 0.3952), B = (0.1106, -0.9187, 0.4834, 0.3625), C = (0.2994, -0.9172, 6, 0.4309), D = (0.4885, -0.7152, 6, 3.3308), E = (0.9832, -5.89e-5, 6, 6).

robustness only while *Form-4* considers both feasibility and performance robustness. As seen from Figure 4.4, the profile of the function differs from that of the *expected* function. The second minimum of the expected value function is located near the third minimum of the original function. The solution C" is now located near the third minimum of the original function since its expected value is lower than solutions around C in earlier plots (i.e., near the second minimum of the original function). Solutions around the second and the third minimum of the original function have $sigma_g = 6$.

Use of *Form-3* formulation results in a set of tradeoff solutions with feasibility robustness $sigma_g$ between 0.0021 (Solution A) and 6.000 (Solution C"). Once again, their corresponding $sigma_f$'s can be computed, which span between 0.3899 and 3.190.

When the performance robustness is considered in addition to feasibility robustness (*Form-4*), two new solutions D and E are identified as shown in Figure 4.5. Solution E has $sigma_f = 6$ while solution D has a marginally lower $sigma_f$. One can observe that the use of *Form-4* allows identification of solutions spanning a large range of feasibility robustness ($sigma_g$'s between 0.0012 and 6) and performance robustness ($sigma_f$'s between 0.3952 and 6).

The $sigma_f$ vs. $sigma_g$ plots for the solutions obtained from the four formulations are presented in Figure 4.6. One can clearly observe that the solutions obtained from formulation *Form-4* have a larger spread in $sigma_f$ and $sigma_g$ when compared with the solutions obtained from *Form-3* . The line plots on the right convey the same

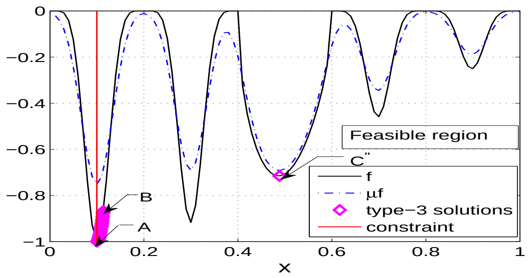


Figure 4.4: Solutions obtains using Form-3 robust formulation. The solutions are labeled as (x, f, $sigma_g$, $sigma_f$) i.e., A = (0.1000, -1.0000, 0.0021, 0.3952), B = (0.1138, -0.8668, 0.0191, 0.3899), C" = (0.4888, -0.7152, 6, 3.190).

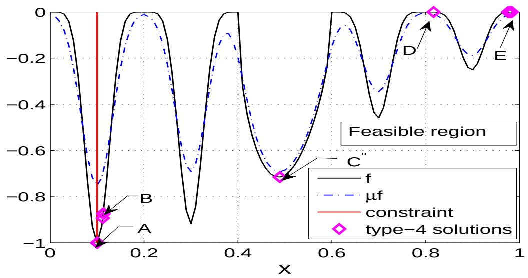
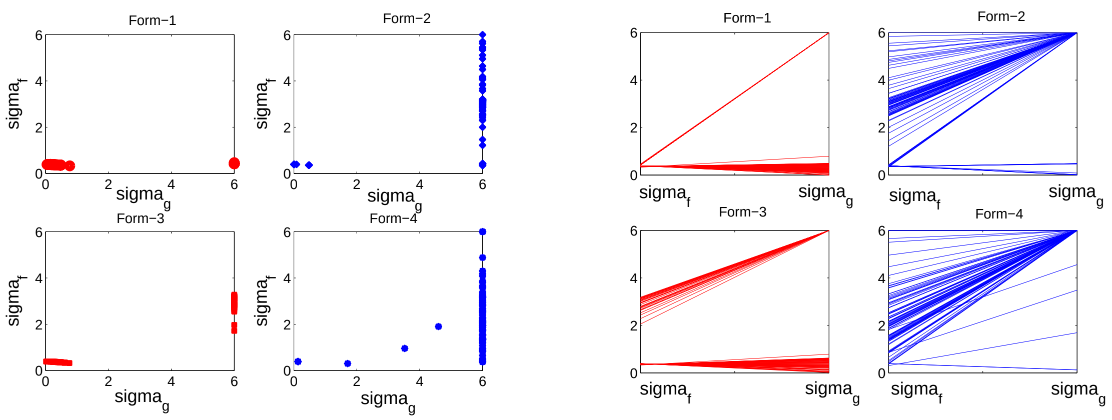


Figure 4.5: Solutions obtains using Form-4 robust formulation. The solutions are labeled as (x, f, $sigma_g$, $sigma_f$) i.e., A = (0.1000, -1.000, 0.0012, 0.3952), B = (0.1136, -0.8715, 0.3948, 0.3937), C" = (0.4892, -0.7151, 6, 3.3624), D = (0.8169, -1.057e-4, 6, 5.4359), E = (0.9832, -5.89e-5, 6, 6).

information and have been included for consistency with respect to the results presented later involving multiple constraints and/or objectives.



(a) $sigma_f$ vs. $sigma_g$ plots (b) Line plots with the value of $sigma_f$ and $sigma_g$

Figure 4.6: Obtained solutions for Example-1

The results obtained using the presented formulations can be summarized as follows:

1. Formulations *Form-1* and *Form-3* consider feasibility robustness only. Therefore the tradeoff set of solutions obtained using these formulations span feasibility robustness range, but they may be concentrated in very small range in terms of performance robustness.

2. Formulations *Form-2* and *Form-4* consider both feasibility and performance robustness. The tradeoff set of solutions obtained using such formulations span the complete possible range of feasibility robustness and performance/expected performance robustness.

3. In the event the landscape of the performance function and expected performance function is the same (or very close to each other), there would be no difference in the outcome if one chooses to use *Form-1* instead of *Form-3* or *Form-2* instead of *Form-4*. Since such an assertion cannot be made in general (and was the case in above example), especially for black-box functions, *use of Form-4 is recommended overall.* If only feasibility robustness is required, *Form-3* can be used. For the objective functions where it is a priori known that function and expected function values are very close under given uncertainties, *Form-1* or *Form-2* (as required) should be used instead, as it will save the computational effort required in calculating the expected values at each design point.

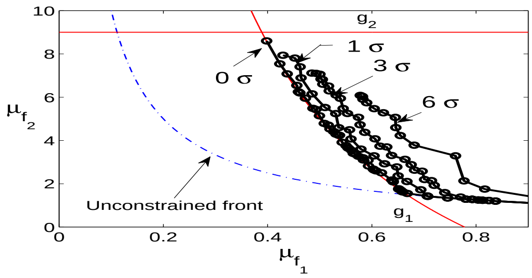
4.4.2 Example-2 (robust multi-objective optimization)

The second example is a multi-objective optimization problem [149] defined below:

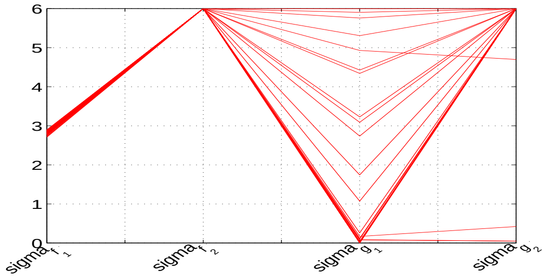
$$
\begin{aligned}
&\underset{(\mathbf{x},\mathbf{y})}{\text{Minimize}} f_1(\mathbf{x},\mathbf{y}) = x \\
&\underset{(\mathbf{x},\mathbf{y})}{\text{minimize}} f_2(\mathbf{x},\mathbf{y}) = \frac{1+y}{x} \\
&\text{Subject to} \\
&\qquad y + 9x - 6 \geq 0,
\end{aligned}
\tag{4.9}
$$

$$
\begin{aligned}
&- y + 9x - 1 \geq 0, \\
&0.1 \leq \mathbf{x} \leq 1, 0 \leq \mathbf{y} \leq 5 \qquad (4.10)
\end{aligned}
$$

The problem has two variables both of which have Gaussian uncertainty with $\sigma_x^2 = 5 \times 10^{-3}$ and $\sigma_y^2 = 5 \times 10^{-3}$. Considering a generic case where the landscape of the objective function and the expected value of the objective function may not be the same, *Form-3* and *Form-4* formulations have been used to identify robust solutions. The solutions obtained from *Form-3* robust formulation have distinct non-dominated fronts corresponding to different values of $sigma_g$. In Figure 4.7(a), the non-dominated fronts (in expected objective function space) corresponding to $sigma_g = 0$, 1, 3 and 6 are presented along with the constraint boundaries (g_1 and g_2).



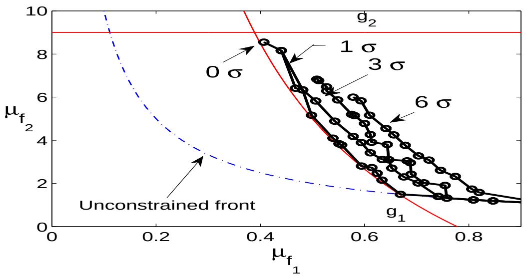
(a)



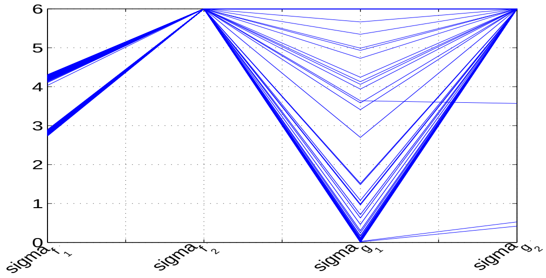
(b)

Figure 4.7: Obtained (a) the tradeoff frontiers of μ_{f_1} and μ_{f_2} for different values of $sigma_g$ (b) the performance of the robust solutions with the robust measures $sigma_f$ and $sigma_g$ for a run using Form-3.

It can be seen that for $sigma_g = 0$, the solutions lie near the active constraint boundary. The solutions progressively move away from the front as $sigma_g$ increases. The robust measures of the solutions are presented in Figure 4.7(b). A user defined limit



(a)



(b)

Figure 4.8: Obtained (a) the tradeoff frontiers of μ_{f_1} and μ_{f_2} for different values of $sigma_g$ (b) the performance of the robust solutions with the robust measures $sigma_f$ and $sigma_g$ using a run using Form-4.

of $\sigma_{f0,1} = 0.1012$ and $\sigma_{f0,2} = 1.6959$ have been used for performance robustness. Since performance robustness is not considered in *Form-3*, all the solutions have $sigma_f$'s of around 3.0. However, since both feasibility and performance robustness are considered in *Form-4*, the tradeoff set as presented in Figure 4.8 contains solutions with $sigma_f$ ranging from 3 to 4 and $sigma_g$ ranging from 0 to 6. One can also notice that, apart from the overall performance and/or feasibility robustness, the sigma levels of each constraint/objective (i.e., $sigma_{g_1}$ and $sigma_{g_2}$) in the figures also allow us to identify the constraint (or objective) that is most prone to be violated. For example, constraint g_1 has lower $sigma_g$ values and hence a larger probability of being violated compared to constraint g_2, for which $sigma_g$ values for all the solutions are close to 6.

4.5 Robust Engineering Design Problems

In this section, the results of DBEA-r on two single objective, one multi-objective and a many-objective robust engineering design problem have been presented. The problem descriptions are provided in Table 4.4, while the results are shown in Table 4.5 and 4.6

.

The results obtained for welded beam design problem using *Form-3* and *Form-4* are presented in Figures 4.9 and 4.10. One can observe that *Form-3* is capable of delivering feasibility robust solutions. However, such solutions do not have a greater diversity with respect to performance robustness (notice the small spread in $sigma_f$). *Form-4* on the other hand offers the tradeoff solutions spanning the complete range of feasibility and performance robustness.

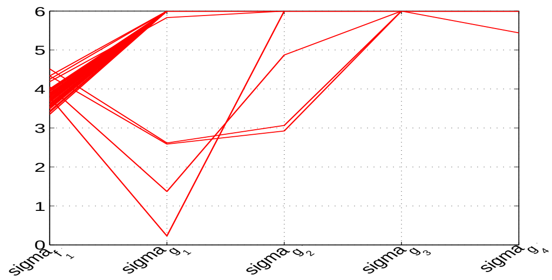


Figure 4.9: Line plot with the value of $sigma_{f_i}$ and $sigma_{g_j}$ for the solutions obtained from Form-3 for welded beam design problem.

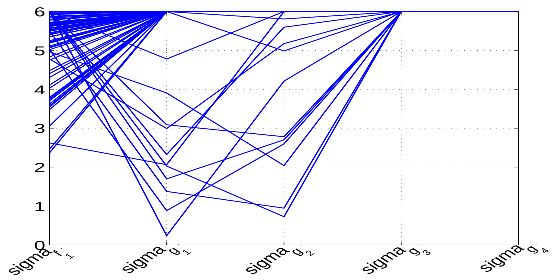


Figure 4.10: Line plot with the value of $sigma_{f_i}$ and $sigma_{g_j}$ for the solutions obtained from Form-4 for welded beam design problem.

To illustrate the benefits of proposed approach in improving robustness/providing additional solutions, a comparison is made using *Form-4* with two other experiments reported for the problem [1]. They use two formulations to solve the problem, i.e., DF-MOSS and DFSS, which are as follows : (a) DFSS: Min $[w_\mu \mu_f + w_\sigma \sigma_f^2]$ Subject to $\mu_f + n\sigma_f \leq 3$ (b) DFMOSS: Min $[\mu_f, \sigma_f]$. The uncertainty of all the variables follow a Gaussian distribution with its standard deviation (σ_x) of 0.01 and the acceptable

Table 4.4: Problem descriptions

Problem	Problem description	σ_x	σ_f	Number of Objectives
Design of a welded beam (SO-1)	The welded beam design optimization problem was originally formulated in [129]. The problem is to design a welded beam for minimum cost subject to a set of constraints. The beam is designed to support a force F = 6000 lbf and the objective is to find the design with the minimum fabrication cost, considering four design variables i.e., thickness of the weld (x_1), length of the weld (x_2), thickness of the beam (x_3), and width of the beam (x_4) with the measurement unit in inches.	The uncertainty of the variables x_1, x_2, x_3, and x_4 follow a Gaussian distribution with $\sigma^2_{x_1} = 8.33 \times 10^{-4}$, $\sigma^2_{x_2} = 8.33 \times 10^{-4}$, $\sigma^2_{x_3} = 4.2 \times 10^{-3}$ and $\sigma^2_{x_4} = 4.2 \times 10^{-3}$.	The user defined limit on the variation on the objective function is prescribed as $\sigma_{f_{0,1}} = 1.701$.	1
Design of a compression spring (SO-2)	The problem is to minimize the weight of a tension/compression spring, subject to constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter [201, 202]. The problem has three design variables: the wire diameter x_1, the mean coil diameter x_2, and the number of active coils x_3.	The uncertainty in the variables x_1, x_2 and x_3 follow a Gaussian distribution with $\sigma^2_{x_1} = 1.67 \times 10^{-5}$, $\sigma^2_{x_2} = 1.67 \times 10^{-5}$ and $\sigma^2_{x_3} = 1.67 \times 10^{-3}$.	The allowable functional variation limit set by the user is prescribed as $\sigma_{f_{0,1}} = 0.0101$.	1
Car Side Impact problem (MO-3)	The problem aims to minimize the weight of a car and the average of three rib deflections constraints i.e., $g_5(\mathbf{x})$, $g_6(\mathbf{x})$ and $g_7(\mathbf{x})$ subject to the constraints of abdomen load, pubic force, velocity of V-Pillar, rib deflection etc [119]. The problem has eleven design variables: the thickness of V-pillar inner x_1, thickness of V-pillar reinforcement x_2, thickness of floor side inner x_3, thickness of cross members x_4, thickness of door beam x_5, thickness of door belt-line reinforcement x_6, thickness of roof rail x_7, material of V-pillar inner x_8, material of floor side inner x_9, barrier height x_{10}, barrier hitting position x_{11}.	The uncertainty in the variables x_1 to x_7 follow Gaussian distribution with given standard deviations:$\sigma^2_{x_1} = 5 \times 10^{-4}$, $\sigma^2_{x_2} = 5 \times 10^{-4}$, $\sigma^2_{x_3} = 5 \times 10^{-4}$, $\sigma^2_{x_4} = 5 \times 10^{-4}$, $\sigma^2_{x_5} = 8.33 \times 10^{-4}$, $\sigma^2_{x_6} = 5 \times 10^{-4}$ and $\sigma^2_{x_7} = 5 \times 10^{-4}$.	The allowable functional variations are set as $\sigma_{f_{0,1}} = 1.080$ and $\sigma_{f_{0,2}} = 0.901$.	2
Water resource management (MaO-4)	The water resource management problem was first described in [131] . The problem has three design variables: local detention storage capacity x_1, maximum treatment rate x_2 and the maximum allowable overflow rate x_3. The objective functions to be minimized are f_1 = drainage network cost, f_2 = storage facility cost, f_3 = treatment facility cost, f_4 = expected flood damage cost, and f_5 = expected economic loss due to flood.	Three of the variables, local detention storage capacity x_1, maximum treatment rate x_2 and the maximum allowable overflow rate x_3, are considered to vary as Gaussian distribution with $(\sigma^2_{x_1}, \sigma^2_{x_2}, \sigma^2_{x_3}) = (3.33 \times 10^{-4}, 1.67 \times 10^{-4}, 1.67 \times 10^{-4})$.	The acceptable functional variations are prescribed as $\sigma_{f_{0,1}} = 3000$, $\sigma_{f_{0,2}} = 10$, $\sigma_{f_{0,3}} = 35961$, $\sigma_{f_{0,4}} = 59292$ and $\sigma_{f_{0,5}} = 27457$.	5

performance deviation $\sigma_{f_0} = 0.4045$. As reported in [1], this study uses NSGA-II [91] as a multi-objective evolutionary algorithm for DFMOSS and DFSS; while DBEA-r is used for *Form-4* to generate the results. For the robust optimization exercise, population size, sample size and maximum number of generations are set as 100, 100 and 400 respectively. In the DFSS optimization (which provides one solution in one run), seven optimization runs were performed with different weighing factors ($w_\mu : w_\sigma$=1000:1,100:1,10:1,1:1,1:10,1:100,1:1000) and $n = 6$. Figures 4.11 and 4.12 show the results obtained from all the three robust formulations.

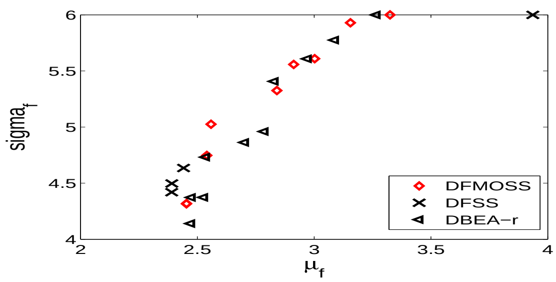


Figure 4.11: Comparison with other robust formulations DFMOSS and DFSS [1] and the solutions obtained from Form-4 for welded beam design problem.

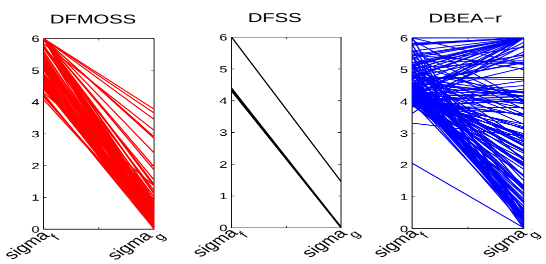


Figure 4.12: Line plot with the value of $sigma_{f_i}$ and $sigma_{g_j}$ for the solutions obtained from DFMOSS, DFSS and Form-4 for welded beam design problem.

It can be seen from Figure 4.11 that all three robust formulations delivered robust solutions with $sigma_f$ level of 6; however among these solutions (with $sigma_f = 6$), the expected performance of the robust solution using DBEA-r is better. Additionally, the proposed approach also provides solutions which are close to six-sigma robust in both performance and feasibility, as observed from Figure 4.12 which shows the line plots of $sigma_f$ and $sigma_g$ from all three formulations. It is noticeable that the robust solutions achieved from DBEA-r are much more diverse. Similar benefits can be anticipated for other engineering problems that follow, and comparison with past approaches is omitted for the sake of brevity.

The results obtained for compression design problem using *Form-3* and *Form-4* are presented in Figures 4.13 and 4.14. One can again observe that *Form-4* delivers solutions spanning a range of $sigma_f$.

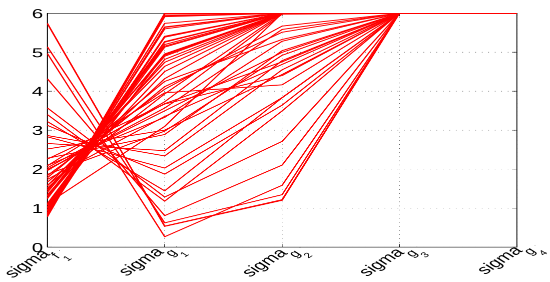


Figure 4.13: Line plot with the value of $sigma_{f_i}$ and $sigma_{g_j}$ for the solutions obtained from Form-3 for compression spring design problem.

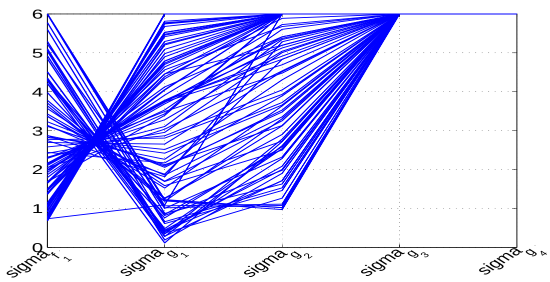
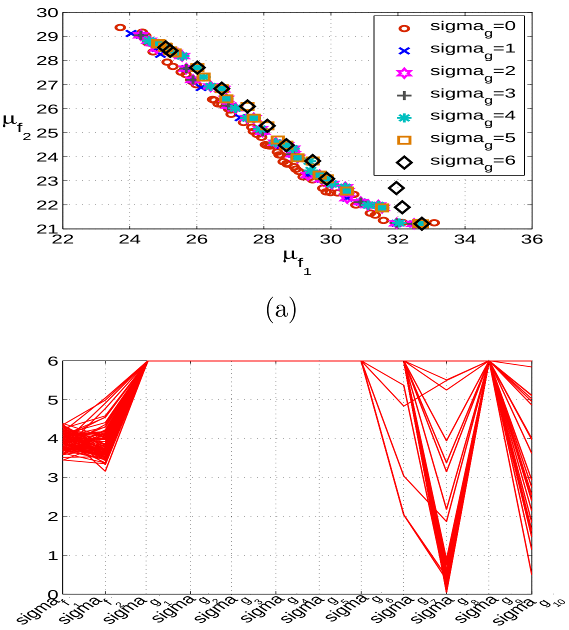


Figure 4.14: Line plot with the value of $sigma_{f_i}$ and $sigma_{g_j}$ for the solutions obtained from Form-4 for compression spring design problem.

The next problem considered is a multiobjective (car side impact) optimization problem. The robust non-dominated solutions (in expected performance function space) achieved for the problem using *Form-3* and *Form-4* are shown in Figures 4.15 and 4.16. The fronts corresponding to various levels of $sigma_g$ using *Form-3* are shown in Figure 4.15(a), while fronts corresponding to various levels of $sigma_g$ and $sigma_f$ is presented in Figure 4.16(a). Corresponding values of $sigma_g$ and $sigma_f$ are plotted in Figure 4.15(b) and Figure 4.16(b). Feasibility robustness of $sigma_g$ =6 is achieved by solutions identified using both the formulations. In the event the user requires solutions with $sigma_g$ =4 and $sigma_f$ =4 (i.e., both feasibility and performance robust solutions), the expected performance would be worse than solutions requiring only $sigma_g$ to be greater than 4 (i.e., only feasibility robustness).

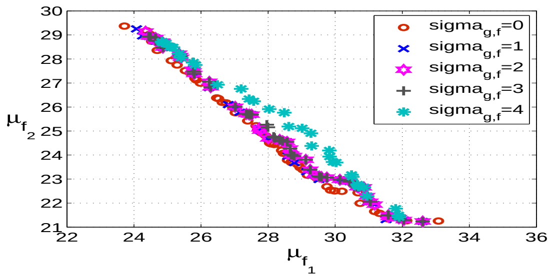


(a)

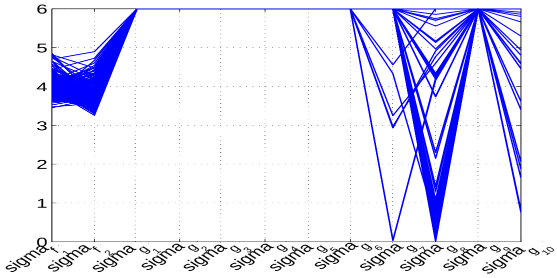
(b)

Figure 4.15: Obtained (a) non-dominated robust solutions (b) line plot with the value of $sigma_{f_i}$ and $sigma_{g_j}$ for the solutions obtained using Form-3 for car side impact problem.

The performance of the approach is further analyzed using a many-objective (water resource management) optimization problem. The parallel coordinate plots presented in Figure 4.17 and Figure 4.18 are normalized expected objective function values with the upper limits [76191, 573.15, 2762800, 1342600 7972] and lower limits [63831 295.51, 282680, 245010, 920]. These limits are computed using the non-dominated solutions obtained from the robust formulations. Figures 4.17 and 4.18 show the robust non-dominated solutions obtained using *Form-3* and *Form-4* robust formulations and their corresponding robust measures. It can be seen that solutions identified using *Form-3* robust formulation have a larger spread. But for *Form-4* robust formulation, which demands both feasibility and performance robustness, several solutions disappear. One can also note that the solutions from *Form-4* formulation have a higher range of $sigma_f$'s. While solutions with $sigma_g$=6 is achievable via both *Form-3* and *Form-4*



(a)

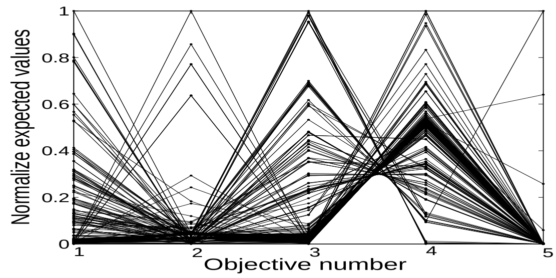


(b)

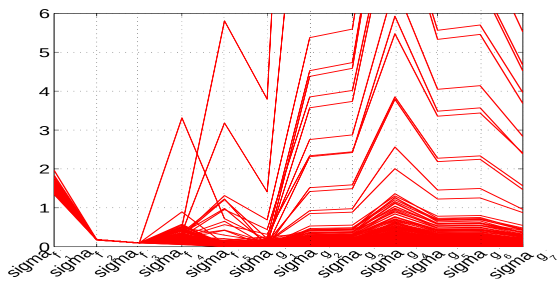
Figure 4.16: Obtained (a) non-dominated robust solutions (b) line plot with the value of $sigma_{f_i}$ and $sigma_{g_j}$ for the solutions obtained using Form-4 for car side impact problem.

formulation, $sigma_f$ limits the overall robustness of the solution. Such information assists in the identification of critical constraints/performance functions i.e., which constraints or performance function variations should be targeted to improve the overall robustness of the solution.

While the above study highlighted the effects of various formulations and the ability of DBEA-r to solve them, the next study compares the performance of DBEA-r with other many-objective optimization algorithms, SMS-EMOA [203] and MOEA/D [9]. These algorithms have been chosen for comparison as they are frequently used in the context of many-objective optimization. While these algorithms in their original form can only deal with unconstrained problems, a *feasibility first* constraint handling scheme in both algorithms has been included to deal with the constrained optimization problems studied in this chapter. Table 4.5 and 4.6 present the comparison of the results obtained using DBEA-r, SMS-EMOA, and MOEA/D on all the robust formulations for all the engineering design problems. DBEA-r offers competitive performance as compared to others based on the hypervolume measure. Three hypervolume measures have been

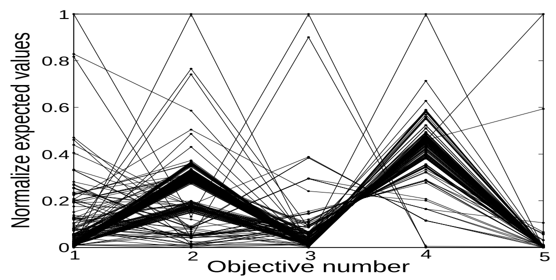


(a)

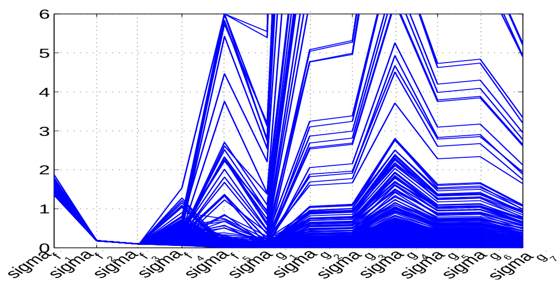


(b)

Figure 4.17: (a) Parallel coordinate plot of the approximation of Pareto set obtained using Form-3 (b) corresponding line plot with the values of $sigma_f$, $sigma_g$ for water resource management problem.



(a)



(b)

Figure 4.18: (a) Parallel coordinate plot of the approximation of Pareto set obtained using Form-4 (b) corresponding line plot with the value of $sigma_f$, $sigma_g$ for water resource management problem.

considered here: HV', which refers to the hypervolume for non-dominated solutions in original function space (f); HV^* which refers to the hypervolume in expected objective function space (μ_f), and HV which refers to hypervolume considering expected objective values (μ_f) as well as robustness measures ($sigma_f, sigma_g$). Nadir point obtained from accumulation of all solutions across all runs has been used as a reference point for the calculation of hypervolumes. The performance of DBEA-r, wherever superior to other algorithms, is shown in bold. From the Table, it can be observed that DBEA-r provides the best solution in terms of HV for most of the cases (9 out of 16). In most of the other cases, it is second only to SMS-EMOA. However, SMS-EMOA uses hypervolume-based measure during evolution, the computation time for which increases exponentially with number of objectives. Consequently, DBEA-r still has some advantage in terms of exhibiting much faster run-times compared to SMS-EMOA. The results highlight the effectiveness of DBEA-r in solving many-objective problems and the resulting advantage in solving robust optimization problems. The values for HV^* obtained using DBEA-r are lower than other two algorithms, which indicates that the solutions obtained using SMS-EMOA and MOEA/D were good in expected objective space, but may not be relatively as good in terms of robustness measures. In terms of HV', DBEA-r is again better than the other two for half of the cases, but the main takeaway from these HV' values is that the algorithm ranking is very different for HV' compared to HV^*, emphasizing that original and expected objective space landscapes could be, in general, quite different, and therefore wherever possible, the algorithms should be compared in expected space (as done in *Form-3,4*).

4.6 Summary and Future Development

Practical solutions of real life problems need to be robust. This chapter presented and analyzed four formulations of robust optimization problems involving uncertain variable values. Robustness is quantified using six-sigma measures, wherein the uncertainties associated with the variables are assumed to follow a Gaussian distribution. Robustness

Table 4.5: Comparison with other algorithms for single objective problems 1, 2 and multi-objective problems 3, 4.*

Prob.	Algorithm	Robust Form	HV'	HV^*	HV
SO-1	DBEA-r	1	-	-	**5.7089**
	MOEA/D		-	-	5.4787
	SMS-EMOA		-	-	5.2627
	DBEA-r	2	-	-	23.3325
	MOEA/D		-	-	16.0554
	SMS-EMOA		-	-	25.5567
	DBEA-r	3	-	-	3.7232
	MOEA/D		-	-	5.4829
	SMS-EMOA		-	-	5.8139
	DBEA-r	4	-	-	**38.0861**
	MOEA/D		-	-	32.6947
	SMS-EMOA		-	-	37.8455
SO-2	DBEA-r	1	-	-	**0.2105**
	MOEA/D		-	-	0.1742
	SMS-EMOA		-	-	0.1923
	DBEA-r	2	-	-	**0.2149**
	MOEA/D		-	-	0.1978
	SMS-EMOA		-	-	0.2078
	DBEA-r	3	-	-	**0.2823**
	MOEA/D		-	-	0.2713
	SMS-EMOA		-	-	0.2745
	DBEA-r	4	-	-	**0.1997**
	MOEA/D		-	-	0.1829
	SMS-EMOA		-	-	0.1895

*HV' refers the hypervolume on deterministic non-dominated objectives and HV^* refers the hypervolume on robust original function space, whereas HV refers the hypervolume on the objectives considering robustness.

Table 4.6: Comparison with other algorithms for single objective problems 1, 2 and multi-objective problems 3, 4.*

Prob.	Algorithm	Robust Form	HV'	HV^*	HV
MO-3	DBEA-r	1	69.5953	31.0741	231.8068
	MOEA/D		66.7295	32.9402	225.6509
	SMS-EMOA		67.1285	32.9354	247.7618
	DBEA-r	2	95.6037	45.9086	135.2737
	MOEA/D		102.270	47.9154	109.9465
	SMS-EMOA		104.1910	48.4418	145.1832
	DBEA-r	3	72.5755	46.4866	**290.3055**
	MOEA/D		79.6860	47.6692	259.2064
	SMS-EMOA		82.1860	48.2275	289.9788
	DBEA-r	4	91.9530	52.8294	153.2759
	MOEA/D		96.1942	53.6699	133.0404
	SMS-EMOA		98.8940	55.9082	166.1503
MaO-4	DBEA-r	1	0.9412	0.0146	0.0531
	MOEA/D		0.9373	0.0168	0.0476
	SMS-EMOA		0.9539	0.0123	0.0635
	DBEA-r	2	1.2340	0.0241	**0.0781**
	MOEA/D		1.1354	0.0253	0.0690
	SMS-EMOA		1.2015	0.0263	0.0655
	DBEA-r	3	0.9923	0.0173	0.0463
	MOEA/D		0.9813	0.0182	0.0428
	SMS-EMOA		0.9827	0.0178	0.0624
	DBEA-r	4	1.2856	0.0262	**0.0720**
	MOEA/D		1.2527	0.0266	0.0560
	SMS-EMOA		1.2766	0.0278	0.0556

*HV' refers the hypervolume on deterministic non-dominated objectives and HV^* refers the hypervolume on robust original function space, whereas HV refers the hypervolume on the objectives considering robustness.

has been studied from two perspectives, (a) feasibility robustness, i.e., robustness of solutions in terms of failure (violation of any constraint) and (b) performance robustness, i.e., robustness assuring good performance. The difference between setting the objective as performance function v/s expected value of the performance function has been discussed. The problems are formulated as many objective optimization problems and a decomposition based evolutionary algorithm has been used for the solution.

The contributions of the presented work can be summarized as follows:

- Four different robust formulations are presented and analyzed. The differences between the formulations and their capabilities are highlighted from two angles i.e., feasibility robustness and performance robustness. *Form-1* and *Form-3* formulations are capable of identifying feasibility robust solutions only, while *Form-2* and *Form-4* are both capable of identifying solutions that span the range of feasibility and performance robustness. *Form-4* is recommended over *Form-2* (and overall) as it considers *expected* performance. Treating the robustness measures as additional objectives instead of constraints delivers a set of solutions corresponding to different levels of robustness in a single run.

- Robustness measures have been quantified using $sigma_g$ and $sigma_f$ values for feasibility and performance respectively. This way of quantifying robustness helps comparing the robustness of solution with respect to each objective/constraint on a common scale. Furthermore, for multiple constraints/objectives case, minimum values of the feasibility and performance robustness (among all the constraints/objectives), are considered as the measures for overall robustness, which is equivalent to *promising* this minimum level of robustness w.r.t. all constraints and objectives. This also helps in identifying the critical constraint(s)/objective(s) that limit the overall robustness of the solution.

- A decomposition based many-objective algorithm (DBEA-r) is used to solve the robust optimization problem. To the authors' knowledge, many-objective algorithms have not been previously used to deal with robust optimization problems.

The performance of DBEA-r is compared to two other widely used algorithms, SMS-EMOA and MOEA/D. DBEA-r is able to deliver best results for most cases, and competitive results for others; thus highlighting its potential to solve single/multi/many-objective robust optimization problems.

The efficiency of the approach can be further improved through the use of more efficient sampling schemes. In our approach, the *expected* value of every solution was estimated using explicit mean of the neighboring sample points generated using Latin Hypercube Sampling (LHS) with Gaussian distribution. Such an approach requires evaluation of numerous solutions and could easily be computationally prohibitive. Use of Polynomial chaos (PC) [197, 179] based estimation is particularly attractive as it is able to estimate the fitness using far fewer points as compared to sampling only (even with efficient schemes like LHS). The results of *Form-3* robust formulation of the multi-objective problem discussed in Example 2 is presented in Table 4.7. For this illustration, a sample of 15 LHS points with explicit mean is compared with PC based estimate using 6 neighborhood samples. While both the approaches achieve nearly same level of hypervolume in original performance(objective) space (3.7046 and 3.7039) and expected performance function(objective) space (510.148 and 510.011), the PC based approach uses less than half the number of function evaluations (39,600 vs. 99,000).

Table 4.7: Performance of LHS and PC

Sampling	Estimated fitness	Sample points	Num. of FEs	HV^*	HV
LHS	Sample Mean	15	99,000	3.7046	510.148
LHS	Polynomial chaos	6	39,600	3.7039	510.011

*HV^* refers the hypervolume on expected objective function space, whereas HV refers to hypervolume considering expected objectives as well as robustness ($sigma_g$).

In the current form, the population size is set to be the same as the number of reference directions. For problems with a large number of reference directions, evolving a large population may not be efficient and there is a potential to develop archive based schemes.

Chapter 5

Conclusions

5.1 Research and outcomes

Optimization is an integral facet of any real-world problem solving scheme. Among the optimization methods, metaheuristics such as EAs and their variants are often preferred as they are easy to use and can be used to solve single and multiobjective optimization problems without any underlying assumptions on the nature of objective and constraint functions. However, in their native form, such methods require evaluation of numerous candidate solutions prior to convergence. The computational budget available to conduct an optimization exercise becomes even more stretched in the event the objectives and constraints require evaluation via computationally expensive numerical simulations such as finite element methods (FEM), computational fluid dynamics (CFD) etc. Therefore, development of efficient optimization strategies to deal with such problems is an important area of research. In particular, the work presented in this thesis is directed towards this overall goal and the key contributions are summarized below.

Efficient DE: Firstly, an efficient algorithm based on differential evolution (DE) is introduced. The proposed algorithm incorporates an adaptive crossover rate control mechanism, a combination of crossover types and a local search strategy to offer improved rate of convergence. Binomial and exponential crossover mechanisms have been used in various stages of evolution to exploit their strengths in exploration and exploitation. The performance of the approach has been compared with other state of the art approaches to illustrate the benefits.

Constraint Handling: Constraints are an integral element of any real life problem and efficient handling of such constraints is necessary. Such constraints often emerge from user requirements, physical laws, statutory requirements, resource limitations etc. Often they are evaluated using computationally expensive analysis i.e., solvers relying on finite element methods, computational fluid dynamics, computational electro magnetics etc. Existing optimization approaches adopt a *full evaluation policy*, i.e. all constraints of a candidate solution are evaluated even through the first constraint might have been violated by the solution. The study is motivated by the fundamental questions i.e., *why do we spend computational resources to evaluate constraints of a solution, when it has already violated a constraint?*, and *is there any better way?*. In this work, a novel constraint handling scheme has been introduced within the framework of differential evolution utilizing the concepts of partial evaluation and constraint sequencing. The utility of using multiple constraint sequences is highlighted using three illustrative examples. For the engineering problems studied, the proposed approach saves around 10% to 40% of the computational time. Furthermore, an adaptive constraint handling approach has been developed and embedded within the framework of multi-objective evolutionary algorithm based on decomposition (MOEA/D) to equip it to deal with constrained optimization problems. Since the constraint handling scheme is generic, it can be used in other forms of population based stochastic algorithms.

Many Objective Optimization: The third area studied in this thesis relates to development of algorithms to deal with many-objective optimization i.e. (problems with number of objectives greater or equal to four). While there are different approaches to solve many objective optimization problems, the goal here is to develop a method that delivers well converged and well spread set of solutions spanning the entire Pareto surface. In the context of the above goal, decomposition based approaches have been leading the race. In this work an algorithm based on decomposition is introduced which extends the capability of the well-known MOEA/D to deal with many objective optimization problems. The algorithm incorporates a systematic sampling scheme and the balance between convergence and diversity during the course of search is maintained via a simple preemptive distance comparison scheme. It is important to highlight that the number reference directions will increase with increasing number of objectives and in turn require use of larger population sizes. In order to alleviate this problem, a quantum representation of solutions is adopted leading to the development of a novel steady state decomposition based quantum genetic algorithm. The benefits of quantum representation is illustrated using a number of well studied benchmarks.

Robust Optimization: Solutions to practical problems need to be robust. In this work, four formulations have been presented to deal with robust optimization problems involving imprecise variable values. Robustness is quantified using the well-established six-sigma measures, wherein the uncertainty associated with the variables is assumed to follow a Gaussian distribution. The first two formulations consider the original objectives, while the remaining two consider the *expected* objective values in addition to the additional objectives arising out of robustness measures. Two goals for design have been discussed i.e., (a) feasibility robustness, i.e., robustness of solutions in terms of failure (violation of any constraint) and (b) objective robustness, i.e., solutions which limit variation in performance. The approach was able to deliver solutions with varying degrees of robustness in terms of objectives and constraints simultaneously.

5.2 Achievements

In summary, the contributions of this thesis can be grouped into three broad areas:

1. **Constraint handling**: Three algorithms are proposed to effectively deal with constrained optimization problems. Numerical experiments conducted on a number of constrained benchmark and engineering problems demonstrate significant improvements over conventional EA. These are as follows.

(a) Adaptive Hybrid Differential Evolution algorithm (AH-DEa): an adaptive algorithm incorporating crossover rate control mechanism, a combination of crossover types and a local search strategy. Binomial and exponential crossover mechanisms have been used in various stages of evolution to exploit their strengths in exploration and exploitation.

(b) Differential Evolution with Constraint Sequencing (DE-CS): a novel constraint handling scheme has been introduced within the framework of differential evolution utilizing the concepts of partial evaluation and constraint sequencing.

(c) MOEA/D with Adaptive Constraint Handling (MOEA/D-ACH): an adaptive constraint handling approach has been developed and embedded within the framework of multi-objective evolutionary algorithm based on decomposition (MOEA/D) to equip it to deal with constrained optimization problems.

2. **Many Objective Optimization**: Two improvements are proposed for handling problems with large numbers of objectives, as follows.

   (a) Decomposition Based Evolutionary Algorithm (DBEA): an algorithm based on decomposition is introduced which extends the capability of the well-known MOEA/D to deal with many-objective optimization problems. The algorithm incorporates a systematic sampling scheme and the balance between convergence and diversity during the course of search is maintained via a simple preemptive distance comparison scheme.

   (b) Decomposition based Quantum Genetic Algorithm (DQGA): a novel steady state decomposition based quantum genetic algorithm is proposed for the solution of unconstrained and constrained many objective optimization problems.

3. **Robust Many Objective Optimization**: Three relevant directions in the areas of robust design optimization are pursued:

(a) Robust optimization problem formulation: A very important step in a robust optimization exercise is to formulate a problem. Depending upon the robust formulation different optimization methods can be applied in order to get the robust solutions. In this thesis, a generic model of the robust design optimization problem is formulated which can easily be coupled with any evolutionary based optimization algorithm.

(b) Quantification of robustness: a six sigma based robust quantification with Latin hypercube sampling to approximate the *expected* value and the variance is introduced.

(c) Search algorithms: a simple decomposition based search algorithm is proposed to deal with the many-objective formulation of the robust optimization problem.

All the above algorithms and strategies are coded in Matlab. The list of the publications based on the research presented in this thesis is given at the beginning of the thesis. The algorithms developed herein are also currently being used by the Multi-disciplinary Design Optimization (MDO) group at the University of New South Wales at Australian Defence Force Academy (UNSW@ADFA) for various applications.

5.3 Future work

Although the work presented in this thesis has shown significantly better performance for some problems in the areas of constrained optimization, many-objective optimization, and robust design optimization, there are some directions (of many) that need further investigation.

In the context of many objective optimization problems, the algorithms introduced in this thesis retain all reference directions throughout the course of search. In the event, the Pareto front is disconnected, one or more reference directions may be redundant and there is a possibility of re-distributing reference directions. Such a scheme could improve resolution in regions of importance.

In particular, recent studies have demonstrated the efficacy of decomposition based approach with a very small number of quantum population by using quantum representation of the solutions. It is observed from some of the examples that evolution of a population of 5 quantum solutions result in the same performance (based on hypervolume) as the evolution of a population of W quantum solutions, where W is the number of reference directions. There is a need to study this further to ascess the utility of quantum representation to deal with problems involving large number of variables.

In the context of robust optimization, there is a serious need to use more efficient sampling schemes. In this work, the value of every solution was estimated using explicit mean of the neighboring sample points generated using Latin Hypercube Sampling (LHS) with Gaussian distribution. Use of Polynomial chaos based estimation is a potential alternative as it requires fewer samples as compared with LHS based schemes for the same level of accuracy.

Lastly, while all the above contributions are implemented in either the DE or EA paradigm, they can be extended to other metaheuristics.

References

[1] Shimoyama, K., Oyama, A., Fujii, K.: Development of multi-objective six-sigma approach for robust design optimization. Journal of Aerospace Computing, Information, and Communication **5** (2008) 215–233

[2] Goldberg, D.E., Samtani, M.P.: Engineering optimization via genetic algorithm. In: Ninth Conference on Electronic Computation, New York, N.Y., ASCE (1986) 471–82

[3] Voss, T., Beume, N., Rudolph, G., Igel, C.: Scalarization versus indicator-based selection in multi-objective cma evolution strategies. In: Proc. (IEEE World Congress Computational Intelligence). IEEE Congress Evolutionary Computation CEC 2008. (2008) 3036–3043

[4] Das., I., Dennis, J.E.: Normal-bounday intersection: A new method for generating Pareto optimal points in multicriteria optimization problems. SIAM J. Optim. **8**(3) (August 1998) 631–657

[5] Bennett, J.A., Lust, R.V.: Conservative methods for structural optimization. AIAA Journal **28**(8) (1990) 1491–1496

[6] Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**(5) (2006) 477–506

[7] Coello, C.A.C., Cortes, N.C.: Solving multiobjective optimization problems using an artificial immune system. Genetic Programming and Evolvable Machines **6**(2) (2005) 163–190

[8] Lili, Z., Wenhua, Z.: Research on performance measures of multi-objective optimization evolutionary algorithms. In: Proc. 3rd Int. Conf. Intelligent System and Knowledge Engineering ISKE 2008. Volume 1. (2008) 502–507

[9] Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation **11**(6) (2007) 712–731

[10] Coello, C., Veldhuizen, D.V., Lamont, G.: Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Publishers, New York (2002)

[11] Veldhuizen, D.A.V., Lamont, G.B.: On measuring multiobjective evolutionary algorithm performance. 2000 Congress on Evolutionary Computation, IEEE Service Center: Piscataway, New Jersey **1** (July 2000) 204211

[12] Veldhuizen, D.A.V., Lamont, G.B.: Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH (1998)

[13] Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P., Coello, C.A., Deb, K.: Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. Technical Report, Nanyang Technological University, Singapore, http://www.ntu.edu.sg/home/EPNsugan/ (Dec 2005)

[14] Mallipeddi, R., Suganthan, P.: Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Rea-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore (2010)

[15] J.Brest, S.Greiner, B.Boskovic, M.Mernik, V.Zumer: Self adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation **10**(6) (2006) 646–657

[16] Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation **13**(2) (2009) 398–417

[17] Zhang, J., Sanderson, A.C.: Jade: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation **13**(5) (2009) 945–958

[18] Mallipeddi, R., Suganthan, P.N.: Differential evolution with ensemble of constraint handling techniques for solving CEC 2010 benchmark problems. In: 2010 IEEE Congress on Evolutionary Computation (CEC'2010), Barcelona, Spain, IEEE Press (July 18–23 2010) 1907–1914

[19] Sutton, A.M., Lunacek, M., Whitley, L.D.: Differential evolution and non-separability: Using selective pressure to focus search. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation, London, England, United Kingdom, ACM Press (July 711 2007) 1428–1435

[20] Storn, R., Price, K.: Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report TR-95-012, International Computer Science Institute, Berkeley, CA (1995)

[21] Takahama, T., Sakai, S.: Efficient constrained optimization by the ϵ constrained adaptive differential evolution. In: 2010 IEEE Congress on Evolutionary Computation (CEC'2010), Barcelona, Spain, IEEE Press (July 18–23 2010) 2052–2059

[22] Takahama, T., Sakai, S.: Constrained Optimization by Applying the α Constrained Method to the Nonlinear Simplex Method with Mutations. IEEE Transactions on Evolutionary Computation **9**(5) (October 2005.) 437–451

[23] Lampinen, J., Zelinka, I.: On stagnation of the differential evolution algorithm. In: In Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing. (2000) 76–83

[24] Mallipeddi, R., Suganthan, P.N.: Ensemble of Constraint Handling Techniques. IEEE Transactions On Evolutionary Computation **14**(4) (August 2010) 561–579

[25] Mininno, E., Neri, F.: A memetic differential evolution approach in noisy optimization. Memetic Computing **2** (2010) 111–135

[26] Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.: Opposition-based differential evolution for optimization of noisy problems. In: Proc. IEEE Congress Evolutionary Computation CEC 2006. (2006) 1865–1872

[27] Gong, W., Cai, Z., Ling, C.X., Li, H.: Enhanced differential evolution with adaptive strategies for numerical optimization. IEEE Transactions on Systems, Man, and CyberneticsPart B: Cybernetics **41**(2) (2011) 397–413

[28] Mallipeddi, R., Suganthan, P., Pan, Q., Tasgetiren, M.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing **11** (2011) 1679–1696

[29] Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. IEEE Transactions on Evolutionary Computation **15**(1) (February 2011) 27–54

[30] Jeyakumar, G., Velayutham, C.S.: An empirical comparison of differential evolution variants on different classes of unconstrained global optimization problems. In: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. (2009) 866 – 871

[31] Nobakhti, A., Wang, H.: Co-evolutionary self-adaptive differential evolution with a uniform-distribution. In: Proceedings of the 2006 IEEE International Symposium on Intelligent Control, Munich, Germany (2006) 4–6

[32] Tvrdk, J., Polakova, R.: Enhanced competitive differential evolution for constrained optimization. In: Proc. Int Computer Science and Information Technology (IMCSIT) Multiconference. (2010) 909–915

[33] Storn, R.: System Design by Constraint Adaptation and Differential Evolution. IEEE Transactions on Evolutionary Computation **3**(1) (April 1999) 22–34

[34] Epitropakis, M.G., Tasoulis, D.K., Pavlidis, N.G., Plagianakos, V.P., Vrahatis, M.N.: Enhancing differential evolution utilizing proximity-based mutation operators. IEEE Transactions on Evolutionary Computation **15**(1) (2011) 99–119

[35] LaTorre, A., Muelas, S., Pefia, J.M.: Benchmarking a hybrid de-rhc algorithm on real world problems. In: Proc. IEEE Congress Evolutionary Computation (CEC). (2011) 1027–1033

[36] Kramer, O.: Iterated local search with powells method: a memetic algorithm for continuous global optimization. Memetic Computing **2** (2010) 69–83

[37] Zhangcan, H., Hao, C.: A new hybrid optimization algorithm framework to solve constrained optimization problem. In Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A., eds.: 7th International Conference Computational Science (ICCS'2007). Springer. Lecture Notes in Computer Science, Vol. 4490, Beijing, China (2007) 1005–1012

[38] Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation **4**(1) (1996) 1–32

[39] Coello, C.A.C.: Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. Computer Methods in Applied Mechanics and Engineering **191**(11-12) (January 2002) 1245–1287

[40] Hoffmeister, F., Sprave, J.: Problem-independent handling of constraints by use of metric penalty functions. In Fogel, L.J., Angeline, P.J., Bäck, T., eds.: Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP'96), San Diego, California, The MIT Press (February 1996) 289–294

[41] Joines, J., Houck, C.: On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In Fogel, D., ed.: Proceedings of the first IEEE Conference on Evolutionary Computation, Orlando, Florida, IEEE Press (1994) 579–584

[42] Kirkpatrick, S., Gelatt, J.C.D., Vecchi, M.P.: Optimization by simulated annealing. Science, 220 (1983) 671–689

[43] Bean, J.C., Hadj-Alouane, A.B.: A Dual Genetic Algorithm for Bounded Integer Programs. Technical Report TR 92-53, Department of Industrial and Operations Engineering, The University of Michigan (1992) To appear in R.A.I.R.O.-R.O. (invited submission to special issue on GAs and OR).

[44] Hadj-Alouane, A.B., Bean, J.C.: A Genetic Algorithm for the Multiple-Choice Integer Program. Operations Research **45** (1997) 92–101

[45] Powell, D., Skolnick, M.M.: Using genetic algorithms in engineering design optimization with non-linear constraints. In Forrest, S., ed.: Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93), San Mateo, California, University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers (July 1993) 424–431

[46] Sarker, R., Runarsson, T.P., Newton, C.: Genetic Algorithms for Solving a Class of Constrained Nonlinear Integer Programs. International Transaction in Operational Research **8**(2) (2001) 121–138

[47] Chootinan, P., Chen, A.: Constraint Handling In Genetic Algorithms Using A Gradient-Based Repair Method. Computers and Operations Reseach **33**(8) (August 2006) 2263–2281

[48] Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evolutionary Computation **7**(1) (1999) 19–44

[49] Takahama, T., Sakai, S.: Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 308–315

[50] Deb, K.: An efficient constraint handling Method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering **186**(2/4) (2000) 311–338

[51] Mezura-Montes, E., Velázquez-Reyes, J., Coello, C.A.C.: Modified Differential Evolution for Constrained Optimization. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 332–339

[52] Mezura-Montes, E., Velázquez-Reyes, J., Coello, C.A.C.: Promising Infeasibility and Multiple Offspring Incorporated to Differential Evolution for Constrained Optimization. In Beyer, H.G., O'Reilly, U.M., Arnold, D., Banzhaf, W., Blum, C., Bonabeau, E., Cant Paz, E., Dasgupta, D., Deb, K., Foste r, J., de Jong, E., Lipson, H., Llora, X., Mancoridis, S., Pelikan, M., Raidl, G., Soule, T., Tyrrell, A., Watson, J.P., Zitzler, E., eds.: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2005). Volume 1., New York, Washington DC, USA, ACM Press (June 2005) 225–232 ISBN 1-59593-010-8.

[53] Takahama, T., Sakai, S., Iwane, N.: Constrained optimization by the epsilon constrained hybrid algorithm of particle swarm optimization and genetic algorithm. In: AI 2005: Advances in Artificial Intelligence, Springer-Verlag (2005) 389–400 Lecture Notes in Artificial Intelligence Vol. 3809.

[54] Brest, J., Zumer, V., Maucec, M.S.: Self-Adaptative Differential Evolution Algorithm in Constrained Real-Parameter Optimization. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 919–926

[55] Zeng, S., Shi, H., Li, H., Chen, G., Ding, L., Kang, L.: A Lower-dimensional-Search Evolutionary Algorithm and Its Application in Constrained Optimization Problem. In: 2007 IEEE Congress on Evolutionary Computation (CEC'2007), Singapore, IEEE Press (September 2007) 1255–1260

[56] Zhang, Q., Zeng, S., Wang, R., Shi, H., Chen, G., Ding, L., Kang, L.: Constrained Optimization by the Evolutionary Algorithm with Lower Dimensional Crossover and Gradient-Based Mutation. In: 2008 Congress on Evolutionary Computation (CEC'2008), Hong Kong, IEEE Service Center (June 2008) 273–279

[57] Ray, T., Singh, H.K., Isaacs, A., Smith, W.: Infeasibility Driven Evolutionary Algorithm for Constrained Optimization. In Mezura-Montes, E., ed.: Constraint-Handling in Evolutionary Computation. Springer. Studies in Computational Intelligence, Volume 198, Berlin (2009) 145–165 ISBN 978-3-642-00618-0.

[58] Barbosa, H.J., Lemonge, A.C.: An adaptive penalty scheme in genetic algorithms for constrained optimization problems. In Langdon, W., E.Cantú-Paz, Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A., Miller, J.F., Burke, E., N.Jonoska, eds.: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002), San Francisco, California, Morgan Kaufmann Publishers (July 2002) 287–294

[59] Hinterding, R., Michalewicz, Z.: Your brains and my beauty: Parent matching for constrained optimisation. In: Proceedings of the 5th International Conference on Evolutionary Computation, Anchorage, Alaska (May 1998) 810–815

[60] Mezura-Montes, E., Coello, C.A.C.: A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems. Technical Report EVOCINV-04-2003, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México D.F., México (2003) Available in the Constraint Handling Techniques in Evolutionary Algorithms Repository at http://www.cs.cinvestav.mx/˜constraint/.

[61] Singh, H.K., Ray, T., Smith, W.: Performance of infeasibility empowered memetic algorithm for CEC 2010 constrained optimization problems. In: 2010 IEEE Congress on Evolutionary Computation (CEC'2010), Barcelona, Spain, IEEE Press (July 18–23 2010) 3770–3777

[62] Reynoso-Meza, G., Blasco, X., Sanchis, J., Martínez, M.: Multiobjective optimization algorithm for solving constrained single objective problems. In: 2010 IEEE Congress on Evolutionary Computation (CEC'2010), Barcelona, Spain, IEEE Press (July 18–23 2010) 3418–3424

[63] Wang, Y., Cai, Z., Guo, G., Zhou, Y.: Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. IEEE Transactions on Systems, Man and Cybernetics Part B–Cybernetics **37**(3) (June 2007) 560–575

[64] Wang, Y., Liu, H., Cai, Z., Zhou, Y.: An orthogonal design based constrained evolutionary optimization algorithm. Engineering Optimization **39**(6) (September 2007) 715–736

[65] Wang, Y., Cai, Z., Zhou, Y., Zeng, W.: An Adaptive Tradeoff Model for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation **12**(1) (February 2008) 80–92

[66] Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multi-objective optimization. Evolutionary Computation **10** (2002) 263–282

[67] Wang, Q., Gao, X.Z., Wang, C.: An Adaptive Bacterial Foraging Algorithm For Constrained Optimization. International Journal of Innovative Computing Information and Control **6**(8) (August 2010) 3585–3593

[68] Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **1** (1997) 67–82

[69] Tasgetiren, M.F., Suganthan, P.N., Pan, Q.K., Mallipeddi, R., Sarman, S.: An ensemble of differential evolution algorithms for constrained function optimization. In: 2010 IEEE Congress on Evolutionary Computation (CEC'2010), Barcelona, Spain, IEEE Press (July 18–23 2010) 967–975

[70] Mallipeddi, R., Suganthan, P.N., Qu, B.Y.: Diversity Enhanced Adaptive Evolutionary Programming for Solving Single Objective Constrained Problems. In: 2009 IEEE Congress on Evolutionary Computation (CEC'2009), Trondheim, Norway, IEEE Press (May 2009) 2106–2113

[71] Elsayed, S., Sarker, R., Essam, D.: Integrated strategies differential evolution algorithm with a local search for constrained optimization. In: 2011 IEEE Congress on Evolutionary Computation, CEC2011, New Orleans, USA, IEEE Press (2011) 2618–2625

[72] Coit, D.W., Smith, A.E., Tate, D.M.: Adaptive penalty methods for genetic optimization of constrained combinatorial problems. IN -FORMS J. Comput. **8** (1996) 173–182

[73] Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Meth. Appl. Mech. Eng. **186** (2000) 311–338

[74] Farmani, R., Wright, J.A.: Self-adaptive fitness formulation for constrained optimization. **7**(5) (2003) 445–455

[75] Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. **4**(3) (2000) 284–294

[76] Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: Proc. IEEE Congress Evolutionary Computation (CEC) 2005, Edinburgh, UK (September 2-5 2005) 1785–1791

[77] Elsayed, S.M., Sarker, R.A., Essam, D.L.: Multi-operator based evolutionary algorithms for solving constrained optimization problems. Computers & Operations Research **38**(12) (December 2011) 1877–1896

[78] Takahama, T., Sakai, S.: Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation. In: 2010 IEEE Congress on Evolutionary Computation (CEC'2010), Barcelona, Spain, IEEE Press (July 18–23 2010) 1680–1688

[79] Singh, H.K., Ray, T., Smith, W.: Performance of infeasibility empowered memetic algorithm for cec 2010 constrained optimization problems. In: Proc. IEEE Congress Evolutionary Computation (CEC). 1–8

[80] Huang, V.L., Qin, A.K., Suganthan, P.N.: Self-adaptative Differential Evolution Algorithm for Constrained Real-Parameter Optimization. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 324–331

[81] Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation **4**(3) (September 2000) 284–294

[82] Tessema, B., Yen, G.G.: A self adaptative penalty function based algorithm for constrained optimization. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE Press (July 2006) 950–957

[83] Dolan, E.D., More, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91** (2002) 201–213

[84] Liang, J.J., Suganthan, P.N.: Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constrain-Handling Mechanism. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 316–323

[85] Sinha, A., Srinivasan, A., Deb, K.: A Population-Based, Parent Centric Procedure for Constrained Real-Parametrer Optimization. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 943–949

[86] Kukkonen, S., Lampinen, J.: Constrained Real-Parameter Optimization with Generalized Differential Evolution. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 911–918

[87] Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation **15**(1) (2011) 4–31

[88] Asafuddoula, M., Ray, T., Sarker, R.: An adaptive differential evolution algorithm and its performance on real world optimization problems. In: Proc. IEEE Congress Evolutionary Computation (CEC). (2011) 1057–1062

[89] Orvosh, D., Davis, L.: Using a genetic algorithm to optimize problems with feasibility constraints. In: Proc. First IEEE Conf. Evolutionary Computation IEEE World Congress Computational Intelligence. (1994) 548–553

[90] Venkatraman, S., Yen, G.G.: A Generic Framework for Constrained Optimization Using Genetic Algorithms. **9**(4) (2005) 424–435

[91] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. **6**(2) (2002) 182–197

[92] Alam, K., Ray, T., Anavatti, S.: Design of a toy submarine using underwater vehicle design optimization framework. In: Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2011 IEEE Symposium on. (April 2011) 23 –29

[93] Bader, J., Zitzler, E.: HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation **19** (Spring 2011) 45–76

[94] Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: Proc. IEEE World Congress Computational Intelligence. (2008) 2419–2426

[95] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multi-objective optimisation. In: Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (2002) 95–100

[96] Koppen, M., Yoshida, K.: Substitute distance assignment in NSGA-II for handling many-objective optimization problems. In Obayashi, S., ed.: EMO. Volume 4403., Springer, Heidelberg, LNCS (2007) 727–741

[97] Singh, H.K., Isaacs, A., Ray, T., Smith, W.: A study on the performance of substitute distance based approaches for evolutionary many objective optimization. In Li, X., ed.: SEAL. Volume 5361., Springer-Verlag Berlin Heidelberg: LNCS (2008) 401–410

[98] Kachroudi, S., Grossard, M.: Average rank domination relation for NSGA-II and SMPSO algorithms for many-objective optimization. In: Proc. Second World Congress Nature and Biologically Inspired Computing (NaBIC). (2010) 19–24

[99] Wang, G., Wu, J.: A new fuzzy dominance GA applied to solve many-objective optimization problem. In: Proc. Second Int. Conf. Innovative Computing, Information and Control ICICIC '07. (2007) 617–621

[100] Zou, X., Chen, Y., Liu, M., Kang, L.: A new evolutionary algorithm for solving many-objective optimization problems. IEEE Transactions On Systems, Man, and Cybernetics-Part B **38**(5) (2008) 1402–1412

[101] Hadka, D., Reed, P.M., Simpson, T.W.: Diagnostic assessment of the Borg MOEA for many-objective product family design problems. In: IEEE World Congress on Computational Intelligence, Brisbane, Australia (June 2012) 10–15

[102] Aguirre, H., Tanaka, K.: Adaptive ϵ-ranking on MNK-landscapes. In: Proc. IEEE symposium Computational intelligence in miulti-criteria decision-making (MCDM) '09. (2009) 104–111

[103] Saxena, D.K., Deb, K.: Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective. In: IEEE Congress on Evolutionary Computation. (2008)

[104] Singh, H.K., Isaacs, A., Ray, T.: A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. IEEE Transactions on Evolutionary Computation **15**(4) (2011) 539–556

[105] Deb, K., Sinha, A., Korhonen, P.J., Wallenius, J.: An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. IEEE Transactions on Evolutionary Computation **14** (2010) 723–739

[106] Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation-,and indicator-based methods in many -objective optimization. In Obayashi, S., ed.: EMO. Volume 4403., Springer, Heidelberg, LNCS (2007) 742–756

[107] Hughes, E.J.: MSOPS-II: A general-purpose many-objective optimiser. In: Proc. IEEE Congress Evolutionary Computation. (2007) 3944–3951

[108] Moen, H.J., Hansen, N.B., Hovland, H., Trresen, J.: Many-objective optimization using taxi-cab surface evolutionary algorithm. In Purshouse, R., ed.: Evolutionary Multi-Criterion Optimization. Volume 7811 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (March 19-22 2013) 128–142

[109] Giagkiozis, I., Purshouse, R.C., Fleming, P.J.: Generalized decomposition. In Purshouse, R., ed.: Evolutionary Multi-Criterion Optimization. Volume 7811 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (March 19-22 2013) 428–442

[110] Zhou, A., Zhang, Q., Zhang, G.: Approximation model guided selection for evolutionary multiobjective optimization. In Purshouse, R., ed.: Evolutionary Multi-Criterion Optimization. Volume 7811 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (March 19-22 2013) 398–412

[111] Deb, K., Jain, H.: Handling many-objective problems using an improved NSGA-II procedure. In: IEEE World Congress on Computational Intelligence, Brisbane, Australia (June 2012) 10–15

[112] Asafuddoula, M., Ray, T., Sarker, R.: A decomposition based evolutionary algorithm for many objective optimization with systematic sampling and adaptive epsilon control. In Purshouse, R., ed.: Evolutionary Multi-Criterion Optimization. Volume 7811 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (March 19-22 2013) 413–427

[113] Ray, T., Asafuddoula, M., Isaacs: A steady state decomposition based quantum genetic algorithm for many objective optimization. In: Proc. IEEE Congress Evolutionary Computation. (2013) 2817–2824

[114] Tan, Y.Y., Jiao, Y.C., Lib, H., Wang, X.K.: MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. Computers & Operations Research (January 2012)

[115] Asafuddoula, M., Ray, T., Sarker, R., Alam, K.: An adaptive constraint handling approach embedded MOEA/D. In: IEEE World Congress on Computational Intelligence. (10-15 June 2012) 1–8

[116] Takahama, T., Sakai, S.: Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations. IEEE Transactions on Evolutionary Computation **9**(5) (2005) 437–451

[117] Deb, K., R. B. Agarwal: Simulated binary crossover for continuous search space. Complex Systems **9**(2) (1995) 115–148

[118] Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation (2013) Available online.

[119] Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. IEEE Transactions on Evolutionary Computation (2013) Available online.

[120] Ray, T., Tai, K., Seow, K.C.: An evolutionary algorithm for multiobjective optimization. Engineering Optimization **33**(3) (2001) 399–424

[121] Durillo, J., Nebro, A.: jMetal: a java framework for multi-objective optimization. Advances in Engineering Software **42** (2011) 760–771

[122] Simpson, T.W., Chen, W., Allen, J.K., Mistree, F.: Conceptual design of a family of products through the use of the robust concept exploration method. in 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization **2** (1996) 1535–1545

[123] Han, K.H., Kim, J.H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Transactions on Evolutionary Computation **6**(6) (December 2002) 580–593

[124] Kim, Y.H., Kim, J.H.: Multiobjective quantum-inspired evolutionary algorithm for fuzzy path planning of mobile robot. In: IEEE Congress Evolutionary Computation. (May 2009) 1185–1192

[125] Kim, J.H., Han, J.H., Kim, Y.H., Choi, S.H., Kim, E.S.: Preference-based solution selection algorithm for evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation **16**(1) (2012) 20–34

[126] Zhang, J., Wang, W., Zhao, Y., Cattani, C.: Multiobjective quantum evolutionary algorithm for the vehicle routing problem with customer satisfaction. Mathematical Problems in Engineering **2012** (2012) 1–19

[127] K. Deb, L. Thiele, M.L., Zitzler, E.: Scalable multi-objective optimization test problems. In: IEEE Congress Evolutionary Computation. (2002) 825–830

[128] Beume, N., Fonseca, C.M., Lpez-Ibez, M., Paquete, L., Vahrenhold, J.: On the complexity of computing the hypervolume indicator. IEEE Transactions on Evolutionary Computation **13**(5) (2009) 1075–1082

[129] Ragsdell, K., Phillips, D.: Optimal design of a class of welded structures using geometric programming. J. Eng. Ind. **98**(3) (1976) 1021–1025

[130] Golinski, J.: An adaptive optimization system applied to machine synthesis. Mech. Mach. Theory **8**(4) (1973) 419–436

[131] Musselman, K., Talavage, J.: A tradeoff cut approach to multiple objective optimization. Operations Research **28**(6) (1980) 1424–1435

[132] DeLaurentis, D.A., Mavris, D.N.: Uncertainty modeling and management in multidisciplinary analysis and synthesis. In: 38th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV (January 10-13, 2000)

[133] Mavris, D.N., DeLaurentis, D.A., Bandte, O., Hale, M.A.: A stochastic approach to multi-disciplinary aircraft analysis and design. In: 36th Aerospace Sciences Meeting and Exhibit, Reno, NV (January 12-15 1998)

[134] Bandte, O., Mavris, D.N., DeLaurentis, D.A.: Viable designs through a joint probabilistic estimation technique. In: 4th World Aviation Congress and Exposition, San Francisco, CA (October 19-21, 1999)

[135] Gu, X., Renaud, J., Batill, S., Brach, R., Budhiraja, A.: Worst case propagated uncertainty of multidisciplinary systems in robust design optimization. Structural and Multidisciplinary Optimization **20**(3) (2000) 190–213

[136] Anthony, D., Keane, A.: Robust-optimal design of a lightweight space structure using a genetic algorithm. AIAA Journal **41**(8) (2003) 1601–1604

[137] Greiner, H.: Robust optical coating design with evolution strategies. Applied Optics **35**(28) (1996) 5477–5483

[138] Wiesmann, D., Hammel, U., Back, T.: Robust design of multilayer optical coatings by means of evolutionary algorithms. IEEE Transactions on Evolutionary Computation **2**(4) (1998) 162–167

[139] Beyer, H.G., Sendhoff, B.: Robust optimization - a comprehensive survey. Computer Methods in Applied Mechanics and Engineering **196**(33-34) (2007) 3190–3218

[140] Arnold, D.V., Beyer, H.G.: A general noise model and its effects on evolution strategy performance. IEEE Transactions on Evolutionary Computation **10**(4) (2006) 380–391

[141] Arnold, D.V., Beyer, H.G.: Evolution strategies with cumulative step length adaptation on the noisy parabolic ridge. Natural Computing **7**(4) (2008) 555–587

[142] Chen, W., Allen, J.K., Tsui, K.L., Mistree, F.: A procedure for robust design: Minimizing variations caused by noise factors and control factors. ASME Journal of Mechanical Design **118** (1996) 478–485

[143] Sun, G., Li, G., Zhou, S., Li, H., Hou, S., Li, Q.: Crashworthiness design of vehicle by using multiobjective robust optimization. Structural and Multidisciplinary Optimization **44**(1) (2011) 99–110

[144] Sundaresan, S., Ishii, K., Houser, D.R.: A robust optimization procedure with variations on design variables and constraints. Engineering Optimization **24**(2) (1995) 101–117

[145] Wang, L., Grandhi, R.V., Hopkins, D.A.: Structural reliability optimization using an efficient safety index calculation procedure. International Journal for Numerical Methods in Engineering **38** (1995) 1721–1738

[146] Deb, K., Gupta, H.: Searching for robust pareto-optimal solutions in multi-objective optimization. Lecture Notes in Computer Science **3410** (2005) 150–164

[147] Jin, Y., Sendhoff, B.: Trade-off between performance and robustness: An evolutionary multiobjective approach. In Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., Deb, K., eds.: Evolutionary Multi-Criterion Optimization. Volume 2632 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2003) 237–251

[148] Wang, Z., Huang, H.Z., Liu, Y.: A unified framework for integrated optimization under uncertainty. ASME Journal of Mechanical Design **132** (2010) 1–8

[149] Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A.K., Padmanabhan, D.: Reliability-based optimization using evolutionary algorithms. IEEE Transactions on Evolutionary Computation **13**(5) (2009) 154–174

[150] Jung, D.H., Lee, B.C.: Development of a simple and efficient method for robust optimization. International Journal for Numerical Methods in Engineering **53** (2002) 2201–2215

[151] Frangopol, D., Moses, F.: 13. In: Reliability-based structural optimization. London: Chapman and Hall (1994) 492–570

[152] Al-Harthy, A.S., Frangopol, D.M.: Reliability-based design of prestressed concrete beams. Journal of Structural Engineering **120** (1994) 3156–3177

[153] Kleiber, M., Siemaszko, A., Stocki, R.: Interactive stability-oriented reliability-based design optimization. Computer Methods in Applied Mechanics and Engineering **168**(1-4) (1999) 243 – 253

[154] Du, X., Chen, W.: Sequential optimization and reliability assessment method for efficient probabilistic design. ASME Journal of Mechanical Design **126**(2) (2004) 225–233

[155] Koch, P., Yang, R.J., Gu, L.: Design for six sigma through robust optimization. Structural and Multidisciplinary Optimization **26**(3-4) (2004) 235–248

[156] Lei, G., Zhu, J.G., Guo, Y.G., Hu, J.F., Xu, W., Shao, K.R.: Robust design optimization of pm-smc motors for six sigma quality manufacturing. IEEE Transactions on Magnetics **49**(7) (2013) 3953–3956

[157] Lee, K.H., Eom, I.S., Park, G.J., Lee, W.I.: Robust design for unconstrained optimization problems using the taguchi method. AIAA Jouranl **34** (1996) 1059–1063

[158] Lee, K.H., Park, G.J.: Robust optimization considering tolerances of design variables. Computers & Structures **79**(1) (2001) 77 – 86

[159] Zaman, K., Mahadevan, S.: Robustness-based design optimization of multidisciplinary system under epistemic uncertainty. AIAA Jouranl **51**(5) (May 2013) 1021–1031

[160] Li, M., Azarm, S., Aute, V.: A multi-objective genetic algorithm for robust design optimization. In: GECCO '05 Proceedings of the 2005 conference on Genetic and evolutionary computation, ACM (2005) 771–778

[161] Das, I.: Robustness optimization for constrained nonlinear programming problems? Engineering Optimization **32**(5) (2000) 585–618

[162] Paenke, I., Branke, J., Jin, Y., Member, S.: Efficient search for robust solutions by means of evolutionary algorithm and fitness approximation. IEEE Transactions on Evolutionary Computation **10**(4) (2006) 405–420

[163] Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments-a survey. IEEE Transactions on Evolutionary Computation **9**(3) (2005) 303–317

[164] Farmani, R., Walters, G.A., Savic, D.A.: Trade-off between total cost and reliability for anytown water distribution network. Journal of Water Resources Planning and Management **131**(3) (2005) 161–171

[165] Du, X., Sudjianto, A.: First-order saddlepoint approximation for reliability analysis. AIAA Jouranl **42** (2004) 1199–1207

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

[167] Gupta, H., Deb, K.: Handling constraints in robust multiobjective optimization. In: IEEE Congress on Evolutionary Computation, Piscataway, NJ (2005) 450–457

[168] Mourelatos, Z.P., Liang, J.: A methodology for trading-off performance and robustness under uncertainty. ASME Journal of Mechanical Design **128** (2005) 856–863

[169] Youn, B.D., Choi, K.K., Yi, K.: Performance moment integration (PMI) method for quality assessment in reliability-based robust design optimization. Mechanics Based Design of Structures and Machines **33**(2) (2005) 185–213

[170] Lee, K.H., Eom, I.S., Park, G.J., Lee, W.I.: A study on the robust design for unconstrained optimization problems. KSME Journal of Mechanical Design **18** (1994) 2825–2836

[171] Park, G.J., Hwang, W.J., Lee, W.I.: Structural optimization post-process using taguchi method. JSME International Journal. Series A, Mechanics and Material Engineering **37**(2) (1994) 166–172

[172] Balling, R.J., Free, J.C., Parkinson, A.R.: Consideration of worst-case manufacturing tolerances in design optimization. Journal of Mechanical Transaction and Automation **108** (1986) 438–441

[173] Sundaresan, S., Ishii, K., Houser, D.R.: Design optimization for robustness using performance simulation programs. Engineering Optimization **20**(3) (1992) 163–178

[174] Zhu, J., Ting, K.L.: Performance distribution analysis and robust design. ASME Journal of Mechanical Design **123**(1) (2000) 11–17

[175] Su, J., Renaud, J.E.: Automatic differentiation in robust optimization. AIAA Journal **35**(6) (1997) 1072–179

[176] Messac, A., Yahaya, A.I.: Multiobjective robust design using physical programming. Structural and Multidisciplinary Optimization **23**(5) (2002) 357–371

[177] Phadke, M.S.: Quality engineering using robust design. Englewood Cliffs: Prentice Hall (1989)

[178] Chao, L.P., Gandhi, M.V., Thompson, B.S.: A design-for-manufacture methodology for incorporating manufacturing uncertainties in the robust design of fibrous laminated composite structures. Journal of Composite Materials **27**(2) (1993) 175–194

[179] Poles, S., Lovison, A.: A polynomial chaos approach to robust multiobjective optimization. In Deb, K., Greco, S., Miettinen, K., Zitzler, E., eds.: Hybrid and Robust Approaches to Multiobjective Optimization. Number 09041 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2009) 1862–4405

[180] Tsutsui, S., Ghosh, A., Fujimoto, Y.: A robust solution searching scheme in genetic search. In: Parallel problem solving from nature - PPSN IV, Springer-Verlag (1996) 543–552

[181] Branke, J.: Creating robust solutions by means of evolutionary algorithms. In Eiben, A.E., Bck, T., Schoenauer, M., Schwefel, H.P., eds.: Parallel Problem Solving from Nature - PPSN V. Volume 1498 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1998) 119–128

[182] Forouraghi, B.: A genetic algorithm for multiobjective robust design. Applied Intelligence **12**(3) (2000) 151–161

[183] Tsutsui, S., Ghosh, A.: Genetic algorithms with a robust solution searching scheme. IEEE Transactions on Evolutionary Computation **1**(3) (1997) 201–208

[184] Branke, J.: Reducing the sampling variance when searching for robust solutions. In Spector, L., ed.: GECCO 2001-Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann (2001) 235–242

[185] Sebald, A., Fogel, D.: Design of fault tolerant neural networks for pattern classification. In Fogel, D.B., Atmar, W., Jolla, L., eds.: First Annual Conference on Evolutionary Programming, CA:Evolutionary Programming Society (1992) 90–99

[186] Thompson, A.: On the automatic design of robust electronics through artificial evolution. In: International Conference on Evolvable System. (1998) 13–24

[187] Rattray, M., Shapiro, J.: Noisy fitness evaluation in genetic algorithms and the dynamics of learning. Foundations of Genetic Algorithms **4** (1997) 117–139

[188] Miller, B.L., Goldberg, D.E.: Genetic algorithms, selection schemes, and the varying effects of noise. Evolutionary Computation **4**(2) (1996) 113–131

[189] Fitzpatrick, J.M., Grefenstette, J.J.: Genetic algorithms in noisy environments. Machine Learning **3**(2-3) (1988) 101–120

[190] Kim, C., Choi, K.K.: Reliability-based design optimization using response surface method with prediction interval estimation. ASME Journal of Mechanical Design **130**(12) (2007) 1–12

[191] McKay, M.D., Beckmkan, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2) (1979) 239–245

[192] Loughlin, D.H., Ranjithan, S.R.: Chance-constrained genetic algorithms. In: Genetic Evolutionary Computation, San Francisco, CA, Morgan Kaufmann (1999) 369–376

[193] Harbitz, A.: An efficient sampling method for probability of failure calculation. Structural Safety **3**(2) (1986) 109–115

[194] Ditlevsen, O., Bjerager, P.: Plastic reliability analysis by directional simulation. ASCE Journal of Engineering Mechanics **115**(6) (1989) 1347–1362

[195] Chen, W., Wiecek, M.M., Zhang, J.: Quality utility – a compromise programming approach to robust design. ASME Journal of Mechanical Design **121**(2) (2007) 179–187

[196] Putko, M., Newman, P., III, A.T., Green, L.: Approach for uncertainty propagation and robust design in cfd using sensitivity derivatives. In: Design in CFD Using Sensitivity Derivatives, AIAA Paper 2001-2528, in AIAA 15 th Computational Fluid Dynamics Conference. (2001) 2001–2528

[197] Wiener, N.: The homogeneous chaos. American Journal of Mathematics **60**(4) (1938) 897–936

[198] Hasofer, A.M., Lind, N.C.: Exact and invariant second-moment code format. ASCE Journal of the Engineering Mechanics Division **100**(1) (January / February 1974) 111–121

[199] Hohenbichler, M., Gollwitzer, S., Kruse, W., Rackwitz, R.: New light on first- and second-order reliability methods. Structural Safety **4**(4) (1987) 267 – 284

[200] Sato, H., Aguirre, H., Tanaka, K.: Controlling dominance area of solutions and its impact on the performance of moeas. In Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T., eds.: Evolutionary Multi-Criterion Optimization. Volume 4403., Springer, Heidelberg, LNCS (2007) 5–20

[201] Belegundu, A.: A Study of Mathematical Programming Methods for Structural Optimization. PhD thesis, Department of Civil Environmental Engineering, University of Iowa, Iowa (1982)

[202] Arora, J.: Introduction to Optimum Design. McGraw-Hill (1989)

[203] Beume, N., Naujoks, B., Emmerich, M.: Sms-emoa: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research **181** (2007) 16531669

Appendix A

Detailed Results statistics Using AH-DEa

Table A.1: Results of the t-test for CEC-2010 in 10 dimension.

Prob.		AH-DEa	SAMO-GA	SMOA-DE	e-DEag	IEMA
	B	=	=	=	=	=
C01	M	−	−	+	+	−
	S	−	−	−	+	−
	B	=	=	=	=	=
C02	M	−	−	−	−	+
	S	−	−	−	−	+
	B	=	−	=	=	−
C03	M	+	−	+	=	−
	S	+	−	−	=	−
	B	=	=	=	−	−
C04	M	=	−	=	−	−
	S	+	−	−	−	−
	B	=	=	=	=	=
C05	M	=	−	=	=	−
	S	+	−	−	−	−
	B	=	=	=	=	=
C06	M	=	−	=	−	−
	S	+	−	−	−	−
	B	=	=	=	=	−
C07	M	=	−	−	=	−
	S	=	−	−	=	−
	B	=	=	=	−	
C08	M	−	−	+	−	−
	S	−	−	+	−	−
	B	=	=	=	=	−
C09	M	=	−	−	=	−
	S	=	−	−	=	−
	B	=	=	=	=	−
C10	M	=	−	−	=	−
	S	=	−	−	=	−
	B	+	−	−	−	−
C11	M	+	−	−	−	−
	S	−	−	−	+	−
	B	+	−	−	−	−
C12	M	+	−	−	−	−
	S	−	−	−	−	+
	B	−	−	=	=	−
C13	M	−	−	=	=	−
	S	−	−	+	+	−
	B	=	−	=	=	−
C14	M	−	−	−	+	−
	S	−	−	−	+	−
	B	=	−	=	=	−
C15	M	+	−	−	−	−
	S	+	−	−	−	−
	B	=	=	=	=	−
C16	M	−	+	−	−	−
	S	−	+	−	−	−
	B	=	=	=	=	−
C17	M	−	−	+	+	−
	S	−	−	+	+	−
	B	+	−	−	−	−
C18	M	+	−	−	−	−
	S	+	−	−	−	−

Table A.2: Results of the t-test for CEC-2010 in 30 dimension.

Prob.		AH-DEa	SAMO-GA	SMOA-DE	e-DEag	IEMA
C01	B	=	−	=	−	=
	M	−	−	−	+	−
	S	−	−	−	+	−
C02	B	−	−	+	−	−
	M	−	−	+	−	−
	S	−	−	+	−	−
C03	B	+	−	−	−	N/A
	M	+	−	−	−	N/A
	S	+	−	−	−	N/A
C04	B	+	−	−	−	N/A
	M	+	−	−	−	N/A
	S	+	−	−	−	N/A
C05	B	=	−	=	−	−
	M	=	−	=	−	−
	S	+	−	−	−	−
C06	B	=	−	=	−	−
	M	−	−	+	−	−
	S	−	−	+	−	−
C07	B	+	−	−	−	−
	M	+	−	−	−	−
	S	+	−	−	−	−
C08	+	−	−	−	−	
	M	+	−	−	−	−
	S	+	−	−	−	−
C09	B	+	−	−	−	−
	M	+	−	−	−	−
	S	+	−	−	=	−
C10	B	−	−	+	−	−
	M	+	−	−	−	−
	S	+	−	−	−	−
C11	B	=	−	=	−	N/A
	M	+	−	−	−	N/A
	S	+	−	−	−	N/A
C12	B	−	+	−	−	N/A
	M	=	−	=	−	N/A
	S	+	−	−	−	N/A
C13	B	=	−	=	−	=
	M	−	−	+	−	−
	S	−	−	+	−	−
C14	B	+	−	−	−	−
	M	−	−	−	+	−
	S	−	−	−	+	−
C15	B	+	−	−	−	−
	M	+	−	−	−	−
	S	+	−	−	−	−
C16	B	=	=	=	=	−
	M	+	−	−	−	−
	S	+	−	−	−	−
C17	B	−	=	=	=	−
	M	−	−	+	+	−
	S	−	−	+	+	−
C18	B	+	−	−	−	−
	M	+	−	−	−	−
	S	+	−	−	−	−

Table A.3: The Results of CEC-2010 10D problems averaged over 25 runs using AH-DEa and Best reported results in FEs = 2e5.

Prob.	Alg.	10D		
		Best	Mean	S.D
	SAMO-GA	**-7.4731E-01**	-7.4702E-01	1.3477E-03
	SMOA-DE	**-7.4731E-01**	**7.4704E-01**	1.3506E-03
	e-DEag	**-7.4731E-01**	**-7.4704E-01**	**1.3234E-03**
C01	IEMA	**7.4731E-01**	-7.4319E-01	4.3301E-03
	AH-DEa	**-7.4731E-01**	-7.4700E-01	3.2955E-03
	SAMO-GA	-2.2775E+00	-2.7260E+00	2.2678E-03
	SMOA-DE	-2.7771E+00	-2.2768E+00	1.1550E-03
	e-DEag	**-2.2777E+00**	-2.2695E+00	2.3898E-02
C02	IEMA	**-2.2777E+00**	**-2.2777E+00**	**1.8228E-07**
	AH-DEa	**-2.2777E+00**	-2.2640E+00	1.3334E-02
	SAMO-GA	6.4927E-22	1.1907E-10	2.0740E-10
	SMOA-DE	**0.0000E+00**	4.1730E-23	1.6405E-22
	e-DEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C03	IEMA	1.4667E-16	6.2345E-07	1.4023E-06
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	SAMO-GA	**-1.0000E-05**	-9.9343E-06	5.1183E-08
	SMOA-DE	**-1.0000E-05**	**-1.0000E-05**	1.4461E-11
	e-DEag	-9.9923E-06	-9.9185E-06	1.5467E-07
C04	IEMA	-9.9861E-06	9.9114E-06	1.4023E-06
	AH-DEa	**-1.0000E-05**	**-1.0000E-05**	**1.1545E-15**
	SAMO-GA	**-4.8361E+02**	-4.0170E+02	1.1126E+02
	SMOA-DE	**-4.8361E+02**	**-4.8361E+02**	4.1443E-06
	e-DEag	**-4.8361E+02**	**-4.8361E+02**	3.8904E-13
C05	IEMA	**-4.8361E+02**	-3.7916E+02	1.7942E+02
	AH-DEa	**-4.8361E+02**	**-4.8361E+02**	**3.4106E-13**
	SAMO-GA	**-5.7866E+02**	-5.7774E+02	3.2605E+00
	SMOA-DE	**-5.7866E+02**	**-5.7866E+02**	9.3522E-03
	e-DEag	**-5.7866E+02**	-5.7865E+02	3.6272E-03
C06	IEMA	**-5.7866E+02**	-5.5147E+02	7.3582E-01
	AH-DEa	**-5.7866E+02**	**-5.7866E+02**	**3.4502E-09**
	SAMO-GA	**0.0000E+00**	7.8566E-23	1.2081E-22
	SMOA-DE	**0.0000E+00**	7.7628E-23	3.8808E-22
	e-DEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C07	IEMA	1.7473E-10	3.2569E-09	3.3872E-07
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	SAMO-GA	3.6543E-25	3.6487E-23	5.6932E-23
	SMOA-DE	**0.0000E+00**	**2.5201E-25**	**1.2600E-24**
	e-DEag	**0.0000E+00**	6.7275E+00	5.5606E+00
C08	IEMA	1.0075E-10	4.0702E+00	6.3829E+00
	AH-DEa	**0.0000E+00**	1.1162E+00	1.7900E+00
	SAMO-GA	1.8980E-22	2.7315E+03	4.6297E+03
	SMOA-DE	**0.0000E+00**	5.0898E+00	2.4108E+01
	e-DEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C09	IEMA	1.2022E-09	1.9511E+12	5.4014E+12
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**

Table A.4: The Results of CEC-2010 30D problems averaged over 25 runs using AH-DEa and Best reported results in FEs = 6e5.

Prob.	Alg.	30D		
		Best	Mean	S.D
C01	SAMO-GA	-8.2178E-01	-8.1152E-01	7.2479E-01
	SMOA-DE	**-8.2188E-01**	-8.1437E-01	4.7660E-03
	e-DEag	-8.2186E-01	**-8.2868E-01**	**7.1030E-04**
	IEMA	**-8.2188E-01**	-8.1777E-01	4.7885E-03
	AH-DEa	**-8.2188E-01**	-8.1804E-01	4.1176E-03
C02	SAMO-GA	-2.2651E+00	-2.2522E+00	5.7473E-03
	SMOA-DE	**-2.2809E+00**	**-2.2761E+00**	**3.7060E-03**
	e-DEag	-2.1692E+00	-2.1514E+00	1.1975E-02
	IEMA	-2.2809E+00	-1.5044E+00	2.1406E+00
	AH-DEa	-2.2751E+00	-2.2453E+00	1.3150E-02
C03	SAMO-GA	5.4810E-19	2.2551E-07	8.1540E-07
	SMOA-DE	4.5996E-24	4.8265E-22	1.1460E-21
	e-DEag	2.8673E+01	2.8837E+01	8.0472E-01
	IEMA	-	-	-
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C04	SAMO-GA	-2.6443E-06	1.4709E-03	7.3594E-03
	SMOA-DE	-3.2480E-06	-2.4113E-06	4.4923E-07
	e-DEag	-4.6981E-03	-8.1629E-03	3.0678E-03
	IEMA	-	-	-
	AH-DEa	-3.3333E-06	-3.3333E-06	7.0055E-12
C05	SAMO-GA	-4.7848E+02	-4.7164E+02	3.7832E+00
	SMOA-DE	-4.8361E+02	-4.8361E+02	5.3899E-06
	e-DEag	-4.5313E+02	-4.4955E+02	2.8991E+00
	IEMA	-2.8668E+02	-2.7093E+02	1.4117E+01
	AH-DEa	**-4.8361E+02**	**-4.8361E+02**	**2.2509E-13**
C06	SAMO-GA	-5.2496E+02	-5.2081E+02	3.1696E+00
	SMOA-DE	**-5.3064E+02**	**-5.3062E+02**	**1.2885E-02**
	e-DEag	-5.2858E+02	-5.2791E+02	4.7483E-01
	IEMA	-5.2959E+02	-1.3288E+02	5.6104E+02
	AH-DEa	**-5.3064E+02**	-5.3042E+02	4.7525E-01
C07	SAMO-GA	3.2368E-28	3.2368E-28	4.5766E-44
	SMOA-DE	9.4952E-23	1.7828E-13	3.6280E-13
	e-DEag	1.1471E-15	2.6036E-15	1.2334E-15
	IEMA	4.8157E-10	8.4861E-10	4.8429E-10
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C08	SAMO-GA	3.2367E-28	1.5948E-24	6.7765E-24
	SMOA-DE	6.4258E-21	1.0329E-09	2.3733E-09
	e-DEag	2.5187E-14	7.8314E-14	4.8552E-14
	IEMA	1.1201E-09	1.7703E+01	4.0803E+01
	AH-DEa	**0.0000E+00**	**5.4322E-28**	**2.6612E-27**
C09	SAMO-GA	3.5363E+02	1.5259E+04	1.8081E+04
	SMOA-DE	1.1862E-20	6.0796E+00	1.4326E+01
	e-DEag	2.7707E-16	1.0721E+01	2.8219E+01
	IEMA	7.3142E+03	2.9879E+07	4.5001E+07
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**

Table A.5: The Results of CEC-2010 10D problems averaged over 25 runs using AH-DEa and Best reported results in FEs = 2e5.

Prob.	Alg.	10D		
		Best	Mean	S.D
	SAMO-GA	1.1240E-19	1.7321E+02	2.6369E+02
	SMOA-DE	**0.0000E+00**	4.4677E-01	1.5463E+00
	e-DEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C10	IEMA	5.4012E-09	2.5613E+12	3.9679E+12
	AH-DEa	**0.0000E+00**	3.8497E-01	1.3055E+00
	SAMO-GA	-6.2050E-04	-5.2620E-04	4.9043E-05
	SMOA-DE	-1.5227E-03	-1.5227E-03	3.6676E-09
	e-DEag	-1.5227E-03	-1.5227E-03	**6.3410E-11**
C11	IEMA	-1.5227E-03	-1.5227E-03	2.7313E-08
	AH-DEa	**-8.7342E-02**	**-8.3882E-02**	2.3282E-02
	SAMO-GA	-5.6974E+02	-5.5884E+01	1.3518E+02
	SMOA-DE	-5.7009E+02	-1.1661E+02	1.8300E+02
	e-DEag	-5.7009E+02	-3.3673E+02	1.7822E+02
C12	IEMA	-1.0974E+01	-6.4817E-01	2.1993E+00
	AH-DEa	**-8.8705E+02**	**-8.3580E+02**	**1.3893E+02**
	SAMO-GA	-6.8428E+01	-6.8377E+01	1.5585E-01
	SMOA-DE	**-6.8429E+01**	**-6.8429E+01**	**1.5429E-07**
	e-DEag	**-6.8429E+01**	**-6.8429E+01**	1.0260E-06
C13	IEMA	**-6.8429E+01**	-6.8018E+01	1.4007E+00
	AH-DEa	-6.3518E+01	-5.2651E+01	7.5558E+00
	SAMO-GA	1.8760E-22	3.8726E+02	1.2448E+03
	SMOA-DE	**0.0000E+00**	1.2064E-21	2.4359E-21
	e-DEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C14	IEMA	8.0306E-10	5.6308E+01	1.8287E+02
	AH-DEa	**0.0000E+00**	3.7140E+01	1.8103E+02
	SAMO-GA	2.8250E-20	8.5745E+02	1.7926E+03
	SMOA-DE	**0.0000E+00**	7.0538E-04	2.4414E-03
	e-DEag	**0.0000E+00**	1.7990E-01	8.8132E-01
C15	IEMA	9.3541E-10	2.6172E+01	1.5753E+08
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	SAMO-GA	**0.0000E+00**	**1.4034E-03**	**4.8800E-03**
	SMOA-DE	**0.0000E+00**	6.4696E-03	1.0870E-02
	e-DEag	**0.0000E+00**	3.7021E-01	3.7105E-01
C16	IEMA	4.4409E-16	3.3030E-02	2.2601E-02
	AH-DEa	**0.0000E+00**	2.3407E-01	1.8262E-01
	SAMO-GA	**0.0000E+00**	1.2707E-02	1.3310E-02
	SMOA-DE	**0.0000E+00**	**1.2655E-23**	**3.2180E-23**
C17	e-DEag	1.4632E-17	1.2496E-01	1.9372E-01
	IEMA	9.4797E-15	3.1509E-03	1.5755E-02
	AH-DEa	**0.0000E+00**	1.8239E-01	1.9294E-01
	SAMO-GA	4.3590E-17	1.0531E-02	1.5401E-02
	SMOA-DE	1.1740E-23	4.4480E-19	6.6367E-19
	e-DEag	3.7314E-20	9.6788E-19	1.8112E-18
C18	IEMA	2.2366E-15	1.6179E-14	3.8203E-14
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**

Table A.6: The Results of CEC-2010 30D problems averaged over 25 runs using AH-DEa and Best reported results in FEs=6e5.

Test problem	Alg.	30D		
		Best	Mean	S.D
C10	SAMO-GA	4.1644E+01	6.0316E+03	7.1774E+03
	SMOA-DE	9.7690E-21	1.9608E+01	2.2137E+01
	e-DEag	3.2520E+01	3.3262E+01	4.5456E-01
	IEMA	2.7682E+04	1.5834E+07	1.6836E+07
	AH-DEa	3.1133E-08	**1.7989E-07**	**1.1938E-07**
C11	SAMO-GA	-1.3600E-04	-1.2600E-04	4.8870E-06
	SMOA-DE	-3.9230E-04	-3.8690E-04	6.1497E-06
	e-DEag	3.2684E-04	-2.8638E-04	2.7076E-05
	IEMA	N/A	N/A	N/A
	AH-DEa	**-3.9234E-04**	**-3.9234E-04**	**1.4307E-09**
C12	SAMO-GA	-8.5360E+01	-3.4932E+00	1.7056E+01
	SMOA-DE	-1.9926E-01	-1.9926E-01	1.3219E-06
	e-DEag	-1.9915E-01	-3.5623E+02	2.8893E+02
	IEMA	N/A	N/A	N/A
	AH-DEa	**-1.9926E-01**	**-1.9926E-01**	**2.5882E-08**
C13	SAMO-GA	-6.6377E+01	-6.3098E+01	1.2860E+001
	SMOA-DE	-6.8429E+01	**-6.8192E+01**	**3.8916E-01**
	e-DEag	-6.6424E+01	-6.5353E+01	5.7330E-01
	IEMA	-6.8429E+01	-6.7487E+01	9.8366E-01
	AH-DEa	**-6.8429E+01**	-6.8181E+01	5.7303E-01
C14	SAMO-GA	4.1452E+00	1.7629E+03	5.4650E+03
	SMOA-DE	1.7470E-22	1.1969E-08	2.5694E-08
	e-DEag	5.0158E-14	3.0894E-13	5.6084E-13
	IEMA	3.2883E-09	6.1524E-02	3.0736E-01
	AH-DEa	**0.0000E+00**	9.7393E-09	4.7713E-08
C15	SAMO-GA	8.3844E-01	1.7496E+04	3.4114E+04
	SMOA-DE	5.8424E-18	2.1128E+00	4.5106E+00
	e-DEag	2.1603E+01	2.1604E+01	1.1048E-04
	IEMA	3.1187E+04	2.2949E+08	4.6404E+08
	AH-DEa	**0.0000E+00**	3.3720E-01	1.1435E+00
C16	SAMO-GA	**0.0000E+00**	3.7370E-03	6.2840E-03
	SMOA-DE	**0.0000E+00**	4.1607E-03	7.6779E-03
	e-DEag	**0.0000E+00**	2.1684E-21	1.6230E-20
	IEMA	6.1567E-12	1.6329E-03	8.1647E-03
	AH-DEa	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C17	SAMO-GA	**0.0000E+00**	1.3436E-02	1.6162E-02
	SMOA-DE	**0.0000E+00**	**1.0226E-10**	**1.4552E-10**
	e-DEag	2.1657E-01	6.3265E+00	4.9866E+00
	IEMA	9.2766E-10	8.8397E-02	1.5109E-01
	AH-DEa	1.8574E-03	7.8242E-01	1.2896E+00
C18	SAMO-GA	2.8182E-01	7.5357E+00	1.0517E+01
	SMOA-DE	9.1928E-18	2.5681E-09	6.9848E-09
	e-DEag	1.2260E+00	8.7545E+01	1.6647E+02
	IEMA	1.3753E-14	4.7384E-14	6.5735E-14
	AH-DEa	**0.0000E+00**	**1.4426E-29**	**3.2263E-29**

Table A.7: Comparison of statistical results among AH-DEa, SAMO-GA, SAMO-DE, APF-GA, ATMES, and SMES.

Prob.& Opt.	Alg.	Best	Mean	Std.
G01 -15.0000	AH-DEa	-1.500E+01	-1.500E+01	0.000E+00
	SAMO-GA	-1.500E+01	-1.500E+01	0.000E+00
	SAMO-DE	-1.500E+01	-1.500E+01	0.000E+00
	APF-GA	-1.500E+01	-1.500E+01	0.000E+00
	ATMES	-1.500E+01	-1.500E+01	1.600E-14
	SMES	-1.500E+01	-1.500E+01	0.000E+00
G02 -0.803619	AH-DEa	-8.036E-01	**-8.007E-01**	3.912E-03
	SAMO-GA	-8.036E-01	-7.960E-01	5.803E-03
	SAMO-DE	-8.036E-01	-7.987E-01	8.801E-03
	APF-GA	-8.036E-01	-8.035E-01	1.000E-04
	ATMES	-8.033E-01	-7.901E-01	1.300E-02
	SMES	-8.036E-01	-7.852E-01	1.670E-02
G03 -1.0005	AH-DEa	-1.001E+00	-1.001E+00	0.000E+00
	SAMO-GA	-1.001E+00	-1.001E+00	0.000E+00
	SAMO-DE	-1.001E+00	-1.001E+00	0.000E+00
	APF-GA	-1.001E+00	-1.001E+00	0.000E+00
	ATMES	-1.000E+00	-1.000E+00	5.900E-05
	SMES	-1.000E+00	-1.000E+00	2.090E-05
G04 -30665.5386	AH-DEa	-3.067E+04	-3.067E+04	0.000E+00
	SAMO-GA	-3.067E+04	-3.067E+04	0.000E+00
	SAMO-DE	-3.067E+04	-3.067E+04	0.000E+00
	APF-GA	-3.067E+04	-3.067E+04	1.000E-04
	ATMES	-3.067E+04	-3.067E+04	7.400E-12
	SMES	-3.067E+04	-3.067E+04	0.000E+00
G05 5126.497	AH-DEa	5.126E+03	5.126E+03	0.000E+00
	SAMO-GA	5.126E+03	5.128E+03	1.117E+00
	SAMO-DE	5.126E+03	5.126E+03	0.000E+00
	APF-GA	5.126E+03	5.128E+03	1.432E+00
	ATMES	5.126E+03	5.128E+03	1.800E+00
	SMES	5.127E+03	5.174E+03	5.174E+03
G06 -6961.813875	AH-DEa	-6.962E+03	-6.962E+03	0.000E+00
	SAMO-GA	-6.962E+03	-6.962E+03	0.000E+00
	SAMO-DE	-6.962E+03	-6.962E+03	0.000E+00
	APF-GA	-6.962E+03	-6.962E+03	0.000E+00
	ATMES	-6.962E+03	-6.962E+03	4.600E-12
	SMES	-6.962E+03	-6.961E+03	1.850E+00
G07 24.306209	AH-DEa	2.431E+01	**2.431E+01**	**0.000E+00**
	SAMO-GA	2.431E+01	2.441E+01	4.591E-02
	SAMO-DE	2.431E+01	2.431E+01	1.589E-03
	APF-GA	2.431E+01	2.431E+01	0.000E+00
	ATMES	2.431E+01	2.432E+01	1.100E-02
	SMES	2.433E+01	2.448E+01	1.320E-01

Table A.8: Comparison of statistical results among AH-DEa, SAMO-GA, SAMO-DE, APF-GA, ATMES, and SMES.

Prob.& Opt.	Alg.	Best	Mean	Std.
G08 -0.095825	AH-DEa	-9.583E-02	-9.583E-02	0.000E+00
	SAMO-GA	-9.583E-02	-9.583E-02	0.000E+00
	SAMO-DE	-9.583E-02	-9.583E-02	0.000E+00
	APF-GA	-9.583E-02	-9.583E-02	0.000E+00
	ATMES	-9.583E-02	-9.583E-02	2.800E-17
	SMES	-9.583E-02	-9.583E-02	0.000E+00
G09 680.63006	AH-DEa	6.806E+02	**6.806E+02**	**0.000E+00**
	SAMO-GA	6.806E+02	6.806E+02	1.457E-03
	SAMO-DE	6.806E+02	6.806E+02	1.157E-05
	APF-GA	6.806E+02	6.806E+02	0.000E+00
	ATMES	6.806E+02	6.806E+02	1.000E-02
	SMES	6.806E+02	6.806E+02	1.550E-02
G10 7049.2481	AH-DEa	7.049E+03	**7.049E+03**	**1.688E-09**
	SAMO-GA	7.049E+03	7.144E+03	6.786E+01
	SAMO-DE	7.049E+03	7.060E+03	7.856E+00
	APF-GA	7.049E+03	7.078E+03	5.124E+01
	ATMES	7.052E+03	7.250E+03	1.200E+02
	SMES	7.052E+03	7.253E+03	1.360E+02
G11 0.7499	AH-DEa	7.499E-01	7.499E-01	0.000E+00
	SAMO-GA	7.499E-01	7.499E-01	0.000E+00
	SAMO-DE	7.499E-01	7.499E-01	0.000E+00
	APF-GA	7.499E-01	7.499E-01	0.000E+00
	ATMES	7.500E-01	7.500E-01	3.400E+04
	SMES	7.500E-01	7.500E-01	1.520E-04
G12 -1.0000	AH-DEa	-1.000E+00	-1.000E+00	0.000E+00
	SAMO-GA	-1.000E+00	-1.000E+00	0.000E+00
	SAMO-DE	-1.000E+00	-1.000E+00	0.000E+00
	APF-GA	-1.000E+00	-1.000E+00	0.000E+00
	ATMES	-1.000E+00	-1.000E+00	1.000E-03
	SMES	-1.000E+00	-1.000E+00	0.000E+00
G13 0.0539415	AH-DEa	5.394E-02	**5.394E-02**	**0.000E+00**
	SAMO-GA	5.394E-02	5.403E-02	5.941E-05
	SAMO-DE	5.394E-02	5.394E-02	1.754E-08
	APF-GA	5.394E-02	5.394E-02	0.000E+00
	ATMES	5.395E-02	5.396E-02	1.300E-05
	SMES	5.399E-02	1.664E-01	1.770E-01
G14 -47.76488	AH-DEa	-4.776E+01	**-4.776E+01**	**3.894E-05**
	SAMO-GA	-4.719E+01	-4.647E+01	3.159E-01
	SAMO-DE	-4.776E+01	-4.768E+01	4.043E-02
	APF-GA	-4.776E+01	-4.776E+01	1.000E-04
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A

Table A.9: Comparison of statistical results among AH-DEa, SAMO-GA, SAMO-DE, APF-GA, ATMES, and SMES.

Prob.& Opt.	Alg.	Best	Mean	Std.
G15 961.71502	AH-DEa	9.617E+02	9.617E+02	0.000E+00
	SAMO-GA	9.617E+02	9.617E+02	5.524E-05
	SAMO-DE	9.617E+02	9.617E+02	0.000E+00
	APF-GA	9.617E+02	9.617E+02	0.000E+00
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A
G16 -1.905155	AH-DEa	-1.905E+00	-1.905E+00	0.000E+00
	SAMO-GA	-1.905E+00	-1.905E+00	6.952E-07
	SAMO-DE	-1.905E+00	-1.905E+00	0.000E+00
	APF-GA	-1.905E+00	-1.905E+00	0.000E+00
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A
G17 8853.5397	AH-DEa	8.854E+03	8.858E+03	1.847E+01
	SAMO-GA	8.854E+03	8.854E+03	1.740E-01
	SAMO-DE	8.854E+03	8.854E+03	1.150E-05
	APF-GA	8.854E+03	8.888E+03	2.903E+01
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A
G18 -0.866025	AH-DEa	-8.660E-01	**-8.660E-01**	0.000E+00
	SAMO-GA	-8.660E-01	-8.655E-01	4.080E-04
	SAMO-DE	-8.660E-01	-8.660E-01	7.044E-07
	APF-GA	-8.660E-01	-8.659E-01	0.000E+00
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A
G19 32.65559	AH-DEa	3.266E+01	3.457E+01	2.524E+00
	SAMO-GA	3.266E+01	3.643E+01	1.037E+00
	SAMO-DE	3.266E+01	3.276E+01	6.145E-02
	APF-GA	3.266E+01	3.266E+01	0.000E+00
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A
G21 193.72451	AH-DEa	1.937E+02	1.939E+02	6.977E-01
	SAMO-GA	1.937E+02	2.461E+02	1.492E+01
	SAMO-DE	1.937E+02	1.938E+02	1.964E-02
	APF-GA	1.966E+02	1.995E+02	3.866E+00
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A
G23 -400.0551	AH-DEa	-4.001E+02	-3.444E+02	7.782E+01
	SAMO-GA	-3.557E+02	-1.948E+02	5.328E+01
	SAMO-DE	-3.962E+02	-3.608E+02	1.962E+01
	APF-GA	-3.998E+02	-3.948E+02	3.866E+00
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A
G24 -5.508013	AH-DEa	-5.508E+00	-5.508E+00	0.000E+00
	SAMO-GA	-5.508E+00	-5.508E+00	0.000E+00
	SAMO-DE	-5.508E+00	-5.508E+00	0.000E+00
	APF-GA	-5.508E+00	-5.508E+00	0.000E+00
	ATMES	N/A	N/A	N/A
	SMES	N/A	N/A	N/A

Appendix B

Detailed Results statistics Using DE-CS

Table B.1: The number of function evaluations required to achieve the first feasible solution : DE-CS and others

Algorithms	G01 Mean(Std)	G02 Mean(Std)	G04 Mean(Std)
DE-CS	**629** (3.162E+02)	285 (2.256E+01)	**475** (9.724E+01)
DE-SF	4451 (2.064E+03)	**195** (2.013E+00)	520 (2.407E+01)
DE-SP	4874 (2.809E+03)	250 (0.000E+00)	650 (**0.000E+00**)
DE-SR	3216 (2.118E+03)	241 (1.384E+01)	520 (1.235E+00)
DE-EC	3650 (1.723E+03)	250 (0.000E+00)	650 (0.000E+00)
Algorithms	G06 Mean(Std)	G07 Mean(Std)	G08 Mean(Std)
DE-CS	**1343** (1.083E+03)	**9148** (4.978E+03)	**180** (1.047E+01)
DE-SF	2742 (8.180E+02)	21963 (5.153E+03)	429 (2.746E+02)
DE-SP	3090 (8.165E+02)	23922 (6.340E+03)	8730 (1.989E+05)
DE-SR	2142 (**2.591E+02**)	20510 (7.892E+03)	351 (1.526E+01)
DE-EC	19986 (1.010E+04)	37778 (1.703E+04)	498 (2.960E+02)
Algorithms	G09 Mean(Std)	G10 Mean(Std)	G12 Mean(Std)
DE-CS	**316** (1.893E+02)	**3450** (4.442E+03)	157 (2.537E+01)
DE-SF	1527 (1.008E+03)	14924 (3.930E+03)	**51** (1.826E-01)
DE-SP	1298 (8.893E+02)	15770 (4.236E+03)	**150** (**0.000E+00**)
DE-SR	1175 (9.242E+02)	11051 (3.165E+03)	159 (3.500E+01)
DE-EC	1378 (8.696E+02)	142780 (4.320E+04)	**150** (**0.000E+00**)
Algorithms	G18 Mean(Std)	G24 Mean(Std)	Score
DE-CS	**25150** (1.230E+04)	262 (1.466E+01)	8/11
DE-SF	108590 (1.259E+04)	**207** (6.025E+00)	3/11
DE-SP	103890 (1.215E+04)	250 (0.000E+00)	1/11
DE-SR	101761 (8.557E+03)	245 (0.000E+00)	0/11
DE-EC	917650 (7.436E+04)	250 (0.000E+00)	1/11

Table B.2: Average computational time across strategies

Algorithms	G01	G02	G04	G06
DE-CS	**0.00411**	0.00508	**0.00105**	**0.00530**
DE-SF	0.00577	0.00555	0.00121	0.01071
DE-SP	0.00445	0.00460	0.00127	0.01185
DE-SR	0.00576	**0.00454**	0.00117	0.00563
DE-EC	0.00418	0.00462	0.00124	0.04245
Algorithms	G07	G08	G09	G10
DE-CS	**0.01444**	**0.00196**	**0.00138**	**0.01482**
DE-SF	0.01703	0.00248	0.01248	0.01540
DE-SP	0.01895	0.14924	0.00486	0.01805
DE-SR	0.04253	0.00361	0.00480	0.77168
DE-EC	0.02614	0.00256	0.00280	0.14436
Algorithms	G12	G18	G24	Score
DE-CS	0.00205	**0.16782**	0.00196	8/11
DE-SF	**0.00195**	0.19780	**0.00193**	2/11
DE-SP	0.00221	0.16989	0.00217	0/11
DE-SR	0.00216	7.06680	0.00202	1/11
DE-EC	0.00201	1.30910	0.00200	0/11

Table B.3: Comparison of function evaluations is used by the proposed method, DE-CS with other best known algorithms with an accuracy level of $(f(\vec{x}) - f^*(\vec{x})) \leq 0.0001$ and feasible solution, the lowest evaluation in total of function and constraints are in bold face

Prob.	Algo.	Best		Median		Worst		ρ	γ	β
		#func.	#const.	#func.	#const.	#func.	#const.			
G01	DE-CS	**8219**	**11076**	90542	98837	117460	119660	1.00	1.00	189379
	$\epsilon-$DE	57122	57122	59308	59308	61712	61712	1.00	1.00	118616
	SaDE	25115	25115	**25115**	**25115**	**25115**	**25115**	1.00	1.00	**50230**
	jDE	46559	46559	50386	50386	56968	56968	1.00	1.00	100772
	DMS-PSO	22844	22844	47816	47816	33333	33333	1.00	1.00	95632
	MDE	63300	63300	75373	75373	90900	90900	1.00	1.00	150746
	GDE	38076	38076	40519	40519	43838	43838	1.00	1.00	81038
	PCX	32420	32420	55204	55204	62026	62026	1.00	1.00	110409
G02	DE-CS	93759	106880	108980	114490	**118260**	**119130**	1.00	0.95	223470
	$\epsilon-$DE	126152	126152	149825	149825	175206	175206	1.00	1.00	299650
	SaDE	76915	76915	188990	188990	-	-	1.00	0.84	377980
	jDE	101201	101201	123490	123490	173964	173964	1.00	0.92	246980
	DMS-PSO	53379	53379	175130	175130	500000	500000	1.00	0.84	350260
	MDE	53250	53250	96222	96222	245550	245550	1.00	0.16	192444
	GDE	93550	93550	107684	107684	124386	124386	1.00	0.72	215368
	PCX	**52900**	**52900**	**87900**	**87900**	500000	500000	1.00	0.64	**175800**
G04	DE-CS	**4011**	**5950**	**8952**	**11712**	98434	105100	1.00	1.00	**20664**
	$\epsilon-$DE	24800	24800	26216	26216	28206	28206	1.00	1.00	52432
	SaDE	25107	25107	25107	25107	25113	25113	1.00	1.00	50214
	jDE	38288	38288	40728	40728	42880	42880	1.00	1.00	81456
	DMS-PSO	24974	24974	25404	25404	25777	25777	1.00	1.00	50808
	MDE	33900	33900	41562	41562	61950	61950	1.00	1.00	83124
	GDE	13679	13679	15281	15281	**17692**	**17692**	1.00	1.00	30562
	PCX	25310	25310	30989	30989	40140	40140	1.00	1.00	61978
G06	DE-CS	**3474**	6180	**3724**	7455	5973	8451	1.00	1.00	11179
	$\epsilon-$DE	6499	6499	7381	7381	8382	8382	1.00	1.00	14762
	SaDE	12546	12546	14394	14394	18347	18347	1.00	1.00	28788
	jDE	26830	26830	29488	29488	31299	31299	1.00	1.00	58976
	DMS-PSO	26656	26656	27636	27636	28287	28287	1.00	1.00	55272
	MDE	4650	**4650**	5202	**5202**	**5250**	**5250**	1.00	1.00	**10404**
	GDE	6101	6101	6503	6503	7160	7160	1.00	1.00	13006
	PCX	32560	32560	33821	33821	36180	36180	1.00	1.00	67642
G07	DE-CS	79267	104730	82602	106460	85685	117880	1.00	1.00	189062
	$\epsilon-$DE	69506	69506	74303	74303	78963	78963	1.00	1.00	148606
	SaDE	**25195**	**25195**	143090	143090	422860	422860	1.00	1.00	286180
	jDE	114899	114899	127740	127740	141847	141847	1.00	1.00	255480
	DMS-PSO	25574	25574	**26578**	**26578**	**27416**	**27416**	1.00	1.00	**53156**
	MDE	124650	124650	194202	194202	380400	380400	1.00	1.00	388404
	GDE	87437	87437	123996	123996	412908	412908	1.00	1.00	247992
	PCX	72920	72920	117121	117121	258840	258840	1.00	1.00	234242
G08	DE-CS	**42**	**403**	2918	4305	13590	19990	1.00	1.00	7223
	$\epsilon-$DE	327	327	1139	1139	**1334**	**1334**	1.00	1.00	2278
	SaDE	782	782	1268	1268	1775	1775	1.00	1.00	2536
	jDE	1567	1567	3236	3236	4485	4485	1.00	1.00	6473
	DMS-PSO	1621	1621	4124	4124	7990	7990	1.00	1.00	8249
	MDE	900	900	**918**	**918**	1350	1350	1.00	1.00	**1836**
	GDE	1178	1178	1469	1469	1822	1822	1.00	1.00	2938
	PCX	1710	1710	2826	2826	3510	3510	1.00	1.00	5652

Table B.4: Comparison of function evaluations is used by the proposed method, DE-CS with other best known algorithms with an accuracy level of $(f(\vec{x}) - f^*(\vec{x})) \leq 0.0001$ and feasible solution, the lowest evaluation in total of function and constraints are in bold face

Prob.	Algo.	Best		Median		Worst		ρ	γ	β
		#func.	#const.	#func.	#const.	#func.	#const.			
G09	DE-CS	**5982**	**6153**	**9940**	**11519**	19642	20612	1.00	1.00	**21459**
	$\epsilon-$DE	19530	19530	23121	23121	24790	24790	1.00	1.00	46242
	SaDE	12960	12960	18560	18560	33166	33166	1.00	1.00	37120
	jDE	49118	49118	54919	54919	58230	58230	1.00	1.00	109838
	DMS-PSO	29272	29272	29782	29782	29456	29456	1.00	1.00	59564
	MDE	14850	14850	16152	16152	**19200**	**19200**	1.00	1.00	32304
	GDE	25743	25743	30230	30230	33140	33140	1.00	1.00	60460
	PCX	32100	32100	46527	46527	58700	58700	1.00	1.00	93054
G10	DE-CS	50194	80692	72943	104150	76887	117600	1.00	1.00	177093
	$\epsilon-$DE	93743	93743	105234	105234	122387	122387	1.00	1.00	210468
	SaDE	26000	26000	58760	58760	153000	153000	1.00	1.00	117520
	jDE	139095	139095	146150	146150	165498	165498	1.00	1.00	292300
	DMS-PSO	**24500**	**24500**	**25520**	**25520**	**26500**	**26500**	1.00	1.00	**51040**
	MDE	152400	152400	164160	164160	179850	179850	1.00	1.00	328320
	GDE	67344	67344	82604	82604	101487	101487	1.00	1.00	165208
	PCX	57770	57770	89028	89028	109970	109970	1.00	1.00	178056
G12	DE-CS	**350**	**356**	**1219**	1315	2162	3639	1.00	1.00	**2534**
	$\epsilon-$DE	1645	1645	4124	4124	5540	5540	1.00	1.00	8248
	SaDE	463	463	1611	1611	2576	2576	1.00	1.00	3222
	jDE	1820	1820	6356	6356	9693	9693	1.00	1.00	12711
	DMS-PSO	812	812	5409	5409	9192	9192	1.00	1.00	10818
	MDE	1200	1200	1308	**1308**	**1650**	**1650**	1.00	1.00	2616
	GDE	2419	2419	3149	3149	4422	4422	1.00	1.00	6298
	PCX	3710	3710	8960	8960	11940	11940	1.00	1.00	17920
G18	DE-CS	**4994**	**11375**	**6601**	**14285**	**12573**	**22756**	1.00	1.00	**20886**
	$\epsilon-$DE	51035	51035	59153	59153	72112	72112	1.00	1.00	118306
	SaDE	26000	26000	65400	65400	-	-	1.00	1.00	130800
	jDE	91049	91049	104460	104460	142674	142674	1.00	1.00	208920
	DMS-PSO	27500	27500	33180	33180	90500	90500	1.00	1.00	66360
	MDE	54000	54000	103482	103482	133800	133800	1.00	1.00	206964
	GDE	169424	169424	364861	364861	499909	499909	0.84	0.76	729722
	PCX	48350	48350	70027	70027	96180	96180	1.00	1.00	140054
G24	DE-CS	**1036**	**1345**	2267	3273	5019	8836	1.00	1.00	5540
	$\epsilon-$DE	2661	2661	2952	2952	3474	3474	1.00	1.00	5904
	SaDE	4280	4280	4847	4847	5657	5657	1.00	1.00	9694
	jDE	7587	7587	10196	10196	11550	11550	1.00	1.00	20392
	DMS-PSO	13721	13721	19376	19376	26096	26096	1.00	1.00	38752
	MDE	1650	1650	**1794**	**1794**	**1950**	**1950**	1.00	1.00	**3588**
	GDE	2656	2656	3059	3059	3408	3408	1.00	1.00	6118
	PCX	4800	4800	11646	11646	13690	13690	1.00	1.00	23292

Table B.5: The Results of C01-C11 10D and 30D problems averaged over 25 runs.

Prob.	Alg.	10D			30D		
		Best	Mean	S.D	Best	Mean	S.D
C01	e-DEag	**-7.4731E-01**	**-7.4704E-01**	**1.3233E-03**	-8.2186E-01	**-8.2868E-01**	**7.1030E-04**
	ECHT	-7.4730E-01	-7.4700E-01	1.4000E-03	-8.2170E-01	-7.9940E-01	1.7900E-02
	DCDE	**-7.4731E-01**	-6.2082E-01	1.0256E-01	-8.0979E-01	-6.0201E-01	1.2123E-01
	JDEsoco	**-7.4731E-01**	-7.3836E-01	1.6006E-02	**-8.2188E-01**	-8.1238E-01	1.3187E-02
	DE-CS	**-7.4731E-01**	-7.22096E-01	4.6006E-02	-7.9289E-01	-6.92065E-01	7.3187E-02
C08	e-DEag	**0.0000E+00**	6.7275E+00	5.5606E+00	2.5187E-14	**7.8314E-14**	**4.8552E-14**
	ECHT	**0.0000E+00**	6.1566E+00	6.4527E+00	**0.0000E+00**	3.3585E+01	1.1072E+02
	DCDE	1.9700E-26	2.3345E+01	6.3160E+01	1.8800E-26	8.7627E+00	3.0233E+01
	JDEsoco	**0.0000E+00**	3.7421E+00	1.0330E+01	7.2311E-26	8.2585E+01	2.4395E+02
	DE-CS	**0.0000E+00**	**1.1162E+00**	**1.7900E+00**	**0.0000E+00**	2.4322E+01	4.6702E+02
C13	e-DEag	-6.8429E+01	**-6.8429E+01**	**1.0260E-06**	-6.6424E+01	-6.5353E+01	5.7330E-01
	ECHT	-6.8429E+01	-6.5121E+01	2.3750E+00	-6.8429E+01	-6.4583E+01	1.6690E+00
	DCDE	-6.3525E+01	-5.6169E+01	6.5330E+00	-6.7654E+01	-6.3613E+01	3.1226E+00
	JDEsoco	-6.8429E+01	-6.8315E+01	5.7018E-01	**-6.8429E+01**	**-6.7537E+01**	**5.0553E-01**
	DE-CS	**-6.8429E+01**	-6.4129E+01	7.5558E+00	**-6.8429E+01**	-6.1181E+01	1.9203E+01
C14	e-DEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	5.0158E-14	**3.0894E-13**	**5.6084E-13**
	ECHT	0.0000E+00	7.0242E+05	3.1937E+06	0.0000E+00	1.2368E+05	6.7736E+05
	DCDE	0.0000E+00	1.5946E-01	7.9732E-01	0.0000E+00	6.3786E-01	1.4917E+00
	JDEsoco	0.0000E+00	9.1221E-01	2.4538E+00	5.7102E-26	1.5946E-01	7.9732E-01
	DE-CS	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	9.7393E-09	4.7713E-08
C15	e-DEag	**0.0000E+00**	1.7990E-01	8.8132E-01	2.1603E+01	2.1604E+01	**1.1048E-04**
	ECHT	**0.0000E+00**	2.3392E+13	5.2988E+13	1.9922E+09	1.9409E+11	4.3524E+11
	DCDE	**0.0000E+00**	**2.5700E-24**	**1.2900E-23**	**0.0000E+00**	**5.0580E-01**	1.3980E+00
	JDEsoco	2.0258E-26	1.2452E+09	3.8127E+09	9.6993E-16	1.5357E+09	1.6045E+09
	DE-CS	**0.0000E+00**	1.8883E+013	2.7863E+013	**0.0000E+00**	8.6399e+013	7.9073e+013

Appendix C

The reference points used in DBEA

Table C.1: Reference point used for hyper-volume calculation of WFG problems

Test Problem	Obj.	Reference point
WFG1	3	[2.6765 , 4.6893 , 6.7021]
	5	[2.5659 , 4.4052 , 6.4816 , 8.5614 , 8.7174]
	10	[2.6955, 4.6709, 4.5965, 6.2138, 7.6552, 9.7142, 12.6239, 15.2759, 18.6304, 20.8089]
	15	[2.7322, 4.6650, 2.0958, 8.5328, 10.6251, 12.1879, 14.4389, 16.7062, 18.6898, 20.7244, 22.7111, 24.7145, 26.7299, 28.7244, 30.7421]
WFG2	3	[2.0002 , 3.9953 , 5.9927]
	5	[1.9747 , 3.9441 , 5.9695 , 7.8723 , 9.9405]
	10	[1.6819, 3.4599, 4.7745, 6.9021, 8.2840, 8.5491, 11.6828, 12.9936, 14.3702, 19.5332]
	15	[1.4033, 1.011, 5.9989, 7.9998, 9.9894, 12,13.797, 15.946, 18.001, 19.981, 21.982, 23.969, 25.998, 28, 29.999]
WFG4	3	[2.0033 , 4.0030 , 6.0052]
	5	[2.0182 , 4.0172 , 6.0099 , 8.0152 , 10.018]
	10	[2.1136, 4.0950, 6.0876, 8.1385, 10.1373, 12.1196, 14.1150, 16.1095, 18.1099, 20.0927]
	15	[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
WFG5	3	[2.0500 , 4.0502 , 6.0501]
	5	[2.0508 , 4.0509 , 6.0509 , 8.0509 , 10.051]
	10	[2.3387, 4.1593, 6.0954, 8.0870, 10.1113, 12.0202, 14.0653, 16.0654, 18.0691, 20.0500]
	15	[2.05, 4.05, 6.05, 8.05, 10.05, 12.05, 14.05, 16.05, 18.05, 20.05, 22.05, 24.05, 26.05, 28.05, 30.05]
WFG6	3	[2.0274 , 4.0274 , 6.0273]
	5	[2.0235 , 4.0246 , 6.0219 , 8.0222 , 10.022]
	10	[2.1250, 4.1250, 6.1250, 8.1250, 10.1250, 12.1250, 14.1250, 16.1250, 18.0300, 20.1250]
	15	[2.0363, 4.0495, 6.0495, 8.0388, 10.05, 12.01, 14.005, 16.004, 18.004, 20.002, 22.002, 24.007, 26.001, 28, 30.002]
WFG7	3	[2.0017, 4.0018, 6.0016]
	5	[2.0035 , 4.0034 , 6.0037 , 8.0036 , 10.003]
	10	[2.1222, 4.0978, 6.1236, 8.1505, 10.1239, 12.1146, 14.1325, 16.1385, 18.0839, 20.0747]
	15	[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
WFG8	3	[2.2070 , 4.0114 , 6.0130]
	5	[2.204 , 4.1084 , 6.0227 , 8.0217 , 10.023]
	10	[2.4587, 4.3840, 6.4188, 8.2408, 10.1233, 12.0873, 14.0761, 16.0573, 18.0470, 20.0399]
	15	[2.8006, 4.8003, 6.801, 8.7923, 10.602, 12.4, 14.201, 16.001, 18, 20, 22, 24, 26, 28, 30]
WFG9	3	[2.0107 , 4.0072 , 6.0085]
	5	[2.1136 , 4.0936 , 6.1016 , 8.0977 , 10.071]
	10	[2.1762, 4.1437, 6.1694, 8.1460, 10.1525, 12.1521, 14.1659, 16.1448, 18.1400, 20.1421]
	15	[2.0604, 4.0911, 6.0555, 8.0515, 10.049, 12.051, 14.042, 16.009, 18.044, 20.01, 22.013, 24.012, 26.009, 28.009, 30.005]