INDIAN INSTITUTE OF TECHNOLOGY KANPUR

# Modeling Pareto-Optimal Set Using B-Spline Basis Functions

by

Piyush Bhardwaj
Y7027283

A thesis submitted in partial fulfillment for the degree of Master of Technology

in the
Department of Mechanical Engineering

May 14, 2012

# Certificate

It is certified that the work contained in the thesis entitled, **Modeling Pareto-optimal set using B-Spline basis functions**, by **Piyush Bhardwaj** (**Y7027283**), has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Bhaskar Dasgupta
Department of Mechanical Engineering,
Indian Institute of Technology,
Kanpur.

Date:

Dr. Kalyanmoy Deb
Department of Mechanical Engineering,
Indian Institute of Technology,
Kanpur.

Date:

# Contents

# List of Figures

# Abstract

Multi-Objective Optimization (MOO) finds widespread use in several fields including engineering design problems. Most of the Multi-Objective Optimization Algorithms (MOOA) return a set of non-dominated points spread on a non-dominated front. To solve the Multi-Objective Optimization Problem (MOOP) completely, one needs a continuous parameterization of the Pareto-optimal front. In this paper, we present a data fitting approach for continuous parameterization of the Pareto-optimal front. Cubic B-spline basis functions are used for fitting the data returned by some MOOA in the variable space. No prior knowledge about the order in the data is assumed. An automatic procedure for detecting gaps in the Pareto-optimal front is also implemented. The algorithm takes points returned by the MOOA as input and returns the control points of the B-spline manifold representing the Pareto-optimal set. Results are mapped to the objective space, thereby achieving a continuous parameterization in the objective space. Results for several standard bi-objective and tri-objective optimization problems are shown.

# Acknowledgement

I would like to express my gratitude toward my supervisors Prof. Dasgupta and Prof. Deb for their excellent help and invaluable suggestions at all stages of my research work. I thank Prof. Deb for introducing me to the field of optimization and encouraging me throughout my stay at IIT Kanpur. I thank Prof. Dasgupta for providing generous help and motivation throughout the thesis work. It was a pleasure working in their guidance and they have been a great influence on my personal and professional life.

I'll like to thank members of Kanpur Genetic Algorithm Laboratory (KANGAL), Sunith, Abhinav, Faez, Himanshu and Rituparna, for the fruitful discussions and wonderful time spent there. I'll like to thank the members of the robotics group for providing a conducive environment to conduct research and holding regular meetings to discuss a variety of research problems. I thank Vivek for proof-reading my thesis work and providing valuable insights, at all stages of my work.

I'll like to express my gratitude towards all my friends at hostel especially, the phatta group, for making my stay a memorable one and for providing a perennial source of rejuvenation. Last but not the least, I'll thank my parents and my sister for providing support and motivation when times were tough.

Piyush Bhardwaj

Chapter 1

# Introduction

We are in a constant search of finding optimum solutions. The necessity of making every task and process efficient with respect to a certain goal gives rise to the field of optimization. Whether it is science, engineering, finance, management or business, optimization is indispensable. This is the reason that several interrelated research fields like mathematical programming, operations research, optimal control and decision-making have come up, all working for similar goals but on different kinds of problems with different approaches. Optimization helps us find efficient and competitive solutions. The applications of optimization are diverse. In engineering alone, all the major branches find applications of optimization. For example, space craft trajectory optimization in aerospace engineering, process optimization in chemical engineering, maximizing component life in mechanical engineering, structure optimization in civil engineering, throughput maximization in electrical engineering, etc. It is helping designers to find a small factor of safety leading to economical designs. It plays a major role in cost minimization and profit maximization in businesses. In short, it is everywhere.

Optimization is not a new field. The need has been felt since 300 BC when Euclid found the minimal distance between a point and a line, and proved that a square has the greatest area among the rectangles with given total length of edges. However, most of the development in the field occurred in the twentieth century. Once an optimization problem has been formulated mathematically, optimization algorithms can be used to solve the problem. Thanks to the affordable high-speed computing power of personal computers, optimization algorithms have become increasingly popular.

Optimization problems can be divided into two classes, viz. single objective optimization problems (SOOPs) and multi-objective optimization problems (MOOPs). As the name suggests SOOPs have a single objective, contrary to the MOOPs which have more than one objectives. Although most of the research in the field of optimization has concerned itself with single objective optimization, majority of the real-life problems have multiple conflicting objectives. To solve MOOPs, several algorithms have been proposed in the last decade. This thesis focuses on improving the results of MOOAs and to study common characteristics of the solutions of a MOOP.

This chapter provides an introduction to MOO and the objectives of the thesis. In Section 1.1 fundamentals of MOO are discussed, followed by a discussion on MOOAs in

Section 1.2. The objectives of the thesis are discussed in Section 1.3. At last, the structure of the thesis is laid out in Section 1.4.

## 1.1 Multi-Objective Optimization

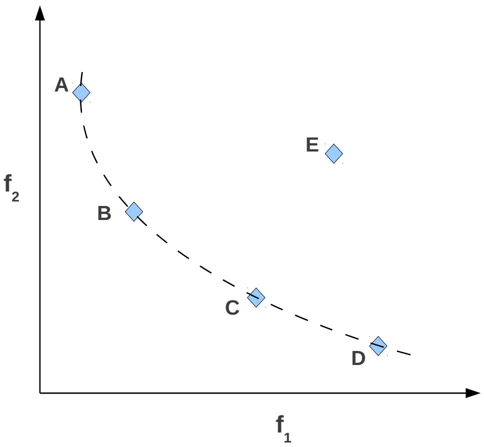


Figure 1.1: Solutions in the objective space

Multi-objective optimization entails multiple conflicting objectives for which no unique single solution can be found. The exact solution of a multi-objective optimization problem is often a set of trade-off solutions. To understand this, let us consider a bi-objective optimization problem. Both the objectives, $f_1$ and $f_2$, are to be minimized. Figure 1.1 shows some points in the objective space. Points A, B, C and D belong to the solution set. These points can't be compared among themselves as they are better than each other with respect to some of the objectives. Thus, these are the trade-off solutions. Point E is worse than point C and point B with respect to all the objectives and is therefore not part of the solution set. The solution of the MOO problem is the trade-off solutions represented by the dashed line in Figure 1.1.

MOOP in its general form is stated as:

$$\begin{aligned} \text{Minimize/Maximize} \quad & f_m(\mathbf{x}) && m = 1, 2, \ldots M; && (1.1) \\ \text{subject to} \quad & g_j(\mathbf{x}) \geq 0 && j = 1, 2, \ldots J; && (1.2) \\ & h_k(\mathbf{x}) = 0, && k = 1, 2, \ldots K; && (1.3) \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, && i = 1, 2, \ldots N && (1.4) \end{aligned}$$

where $f_m$, $g_j$, $h_k$, $x_i^{(L)}$ and $x_i^{(U)}$ represent the objective functions, the inequality constraints, the equality constraints and the lower and upper bounds on the variables respectively. The components of vector **x** represent the variables of the problem. The variables collectively constitute the variable or design space and the objectives collectively constitute the objective space. Any point in the variable space can be mapped to the objective space by using the function definition. As a reverse mapping may not necessarily exist from the objective to the variable space, finding the variable values given the function values is not a trivial task. An illustration of the variable and the objective space are shown in Figure 1.2. The shaded part shows the feasible region in both the variable and the objective spaces. Squares and circles represent the solutions found by an optimizer in the variable space and the corresponding objective space points, respectively while the curve represents the true solution of the problem. Only a finite number of points are returned by the algorithm on the solution curve.

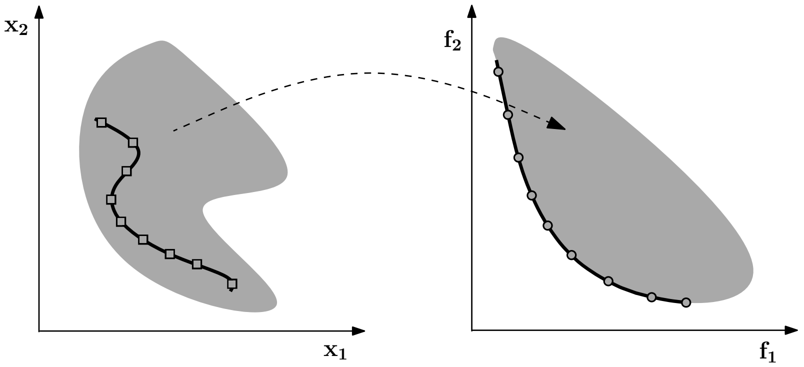


Figure 1.2: Figure showing the variable space and the corresponding objective space. The Pareto-optimal front and the Pareto-optimal set are both shown.

Points on the Pareto-optimal front are the trade-off solutions which means no point on the Pareto-optimal front can be improved in one objective without compromising it in another objective. The concept of dominance is used to compare two solutions in MOO.

**Definition 1.** ***Concept of domination***
*A solution 'X' is said to dominate another solution 'Y' ($X \succeq Y$) if X is not worse than Y in all of the objectives and X is better than Y in at least one of the objectives [1].*

In Figure 1.1, point E is dominated by point B and point C. Points A, B, C and D are not dominated by any other point. The variable values corresponding to these points constitute the non-dominated set.

**Definition 2.** ***Non-dominated set***
*In a set, the points not dominated by any other set member are called the non-dominated points of that set.*

**Definition 3.** ***Pareto-optimal set***
*Non-dominated set of the entire search space constitutes the Pareto-optimal set.*

**Definition 4.** ***Pareto-optimal front***
*The image of the Pareto-optimal set in the objective space is called the Pareto-optimal front.*

In Figure 1.2, the points on the curve passing through the squares in the variable space represent the Pareto-optimal set while the curve passing through the circles in the objective space represent the Pareto-optimal front.

## 1.2 Principles of Multi-objective Optimization Algorithms

An intuitive way to solve a MOOP is to convert it into a SOOP. Weighted sum approach [2] is based on taking a weighted sum of the objectives. Another conversion, as done in the $\varepsilon$-constraint method [3], is by choosing one of the objectives as the only objective and converting rest of the objectives into constraints for the SOOP. By varying the constraints bounds different points on the Pareto-optimal front are generated. Other classical methods like NBI [4], weighted min-max approach [5], goal programming [6, 7] and physical programming [8] are also based on similar conversions. Most of the classical optimization algorithms give a single point on the Pareto-optimal front in each run. Evolutionary methods differ from classical methods as they work on a population of points. As a result evolutionary Multi-objective optimization (EMO) algorithms give multiple Pareto-optimal solutions in a single run. A survey of methods to solve MOOP in engineering is done by Marler et al. [5]. As evolutionary methods are used in this thesis to obtain initial data points, a brief introduction to evolutionary algorithms is provided.

### 1.2.1 Evolutionary algorithms

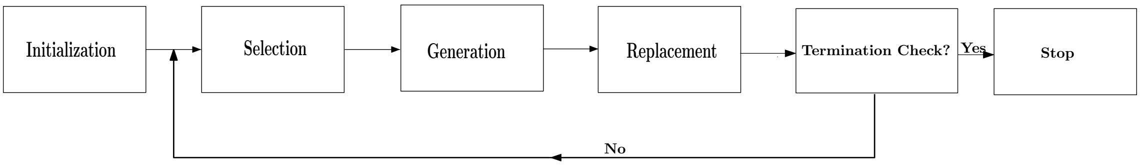


Figure 1.3: Major steps in an EOS

Evolutionary algorithms are based on Darwinian evolutionary system. The basic steps of all evolutionary optimization systems (EOS) are shown in Figure 1.3. The key steps of EOS are as follows: (1) *Initialization* of the population, (2) *Selection* of the member(s) to act as parent(s), (3) *Generation* of offspring from selected parent(s) and (4) *Replacement* of members to create population for the next generation. After initialization; selection, generation and replacement are carried out iteratively till a termination criteria is met. Although detailed description on each step are required before EOS can be simulated, but just a few steps procedure as above is sufficient to represent major EA paradigms for optimization. Several evolutionary MOOAs like vector-evaluated genetic algorithm (VEGA) [9], Multi-Objective Genetic Algorithm (MOGA) [10], Niched-Pareto Genetic Algorithm (NPGA) [11], Non-dominated Sorting Genetic Algorithm (NSGA) [12], Strength Pareto Evolutionary Algorithm (SPEA) [13] and Pareto Archived Evolution Strategy (PAES) [14] have been proposed to solve MOOPs. A survey of EMO algorithms is carried out by Guliashki et al. [15] and Zhou et al. [16].

## 1.3 Objective and Scope

Most of the algorithms discussed in the previous section return either a single solution or a set of solutions on the Pareto-optimal front, which is a continuous or a piecewise continuous entity. Though one or only a few solutions may be implemented, a user may like to see solutions well-spread on the front to aid the process of decision making. To give a user, the flexibility of choosing any point that lies on the Pareto-optimal front, one needs a continuous parameterization of the front. A continuous parameterization of the front allows the user to produce points on the front at his own discretion by varying the parameter values. Hence, well-spaced points can be produced on the front to aid the process of decision making. Missing solutions can also be generated by this procedure. Moreover, if this parameterization is achieved in the variable space rather than the objective space, it will allow us to map the parameterized entity in the variable space to the objective space by using definition of the objective function and thereby get a continuous parameterization in both the spaces. This will also facilitate an easy transition from the objective to the variable space as the parameter value associated with a solution in the variable space is the same as its image in the objective space.

A continuous parameterization allows one to investigate the variation of variables with parameters along the Pareto-optimal front. This helps the user in finding interesting common properties of the points in the Pareto-optimal set. For example some variables may not change their values at all along the Pareto-optimal front. The objective of the present study is to develop an algorithm to find a continuous parameterization of the Pareto-optimal set. This parameterization can be used

1. to produce points on the Pareto-optimal front with discretion,
2. to facilitate transition from the objective to the variable space, and
3. to investigate the variation of variables along the Pareto-optimal front and deduce common principles.

In this thesis, a novel data fitting method is proposed to give the continuous parameterization of the non-dominated set returned by the MOOA. Thus, a continuous picture is obtained from the discrete data given by the MOOA. The data fitting procedure is carried out in the variable space. Major contributions of the thesis are:

- An error reduction method based on KKT-error reduction is implemented to improve the results of a MOOA.
- A clustering procedure to find piecewise continuous Pareto-optimal set is implemented.
- A tree based data fitting procedure using B-spline basis function is proposed and implemented, to find the parameterization.
- No order in input data is assumed. Principal component analysis (PCA) is used to find the order in the data.
- Several standard and real-life optimization problems are used to test the method.
- Innovative design principles are found and analyzed for bi-objective problems using the procedure.

## 1.4 Structure of the thesis

The thesis is structured in the following manner.

**Chapter 2** covers the literature review of the methods that try to find a continuous parameterization of the Pareto-optimal front. We also look at some previous methods which don't parameterize the front but try to model the front as an intermediate step of the optimization process. In this chapter a brief literature review of the data fitting procedure is also provided.

**Chapter 3** covers the mathematical formulation of the problem and provides a brief background of several topics necessary for the understanding of the thesis. Regularity and connectedness property of the Pareto-optimal front are described followed by a brief discussion on B-spline data fitting procedure.

**Chapter 4** provide details of the error reduction and the clustering steps of the algorithm respectively. The procedure is outlined and examples are shown for explaining the nuances of the procedure.

**Chapter 5** gives a detailed explanation of the data fitting approach. Use of PCA to implement the tree based data division procedure is described here. Other details regarding the parameterization procedure and maintaining continuity are discussed in this chapter.

In **Chapter 6** results for bi-objective and tri-objective standard and real-life optimization problems are presented. Common properties of solutions are identified and their significance is discussed.

In **Chapter 7** limitations of the algorithm are presented and conclusions are discussed.

Chapter 2

# Literature Review

In this chapter a brief overview of the past research relevant to the thesis is provided. Existing algorithms that try to provide complete description of the Pareto-optimal front are discussed and their difference from the algorithm proposed in this thesis is pointed out. Methods that work on finding common principles in solutions of MOOP are described. Research done in the field of data fitting using B-spline basis functions is also presented in this chapter.

## 2.1 Approaches to find a continuous Pareto-optimal front

A few algorithms attempt to give a continuous representation of the Pareto-optimal front as the result of the optimization process. Steuer et al. [17] proposed continuous parametric curves and surfaces for bi-objective and tri-objective problems using parametric quadratic programming but their algorithm is restricted to Markovitz portfolio selection problem. Markovitz portfolio selection problem [18] is a bi-objective optimization problem and has one quadratic objective function and one linear objective function with linear constraints. Their algorithm is fast and can be used for problems with large number of variables. It works considerably faster than the traditionally used $\varepsilon$-constraint method in the field of portfolio selection for producing points, one at a time, on the Pareto-optimal front. Their method uses a parameter which is varied to trace out the full equation structure of the front. The equations of all the curves (hyperbolas, and typically many) that contribute segments to the Pareto-optimal front are derived for a precise mathematical specification of the front. The output of algorithm proposed in this thesis is similar to one proposed by Steuer [17] in giving closed form mathematical equations for Pareto-optimal front but it need not be restricted to Markovitz problem only. For Steuer's algorithm the parameter is varied as part of the optimization process while our algorithm takes points as input from some optimizer. Hence, if the optimizer is able to solve non-convex problems, the algorithm will give closed-form equations, representing the Pareto-optimal set.

Grosan [19] proposed a Pareto Evolutionary Continuous Region Algorithm (PECRA) to handle Pareto continuous regions. Dumitrescu et al. [20] proposed an evolutionary approach called Continuous Pareto Set (CPS) algorithm to find continuous Pareto-optimal set (and the corresponding Pareto-optimal front). In their algorithm, individuals in the population are closed intervals or points. Mutation and crossover are used to generate new solutions. However, both algorithms are restricted to single-variable MOOPs only.

Finding the true Pareto-optimal front by using computational grids spread out over the variable space has been attempted by Veldhuizen [21]. By using the computational grids, all the solutions at grid intersection are checked for Pareto-optimality. Binary coding is used to obtain the computational grid. Binary coding restricts the search space and hence the results are still discrete but are the best possible at a given computational resolution. As all the points in the search space are checked for Pareto-optimality, this is a computationally expensive method. As the number of grids are increased, computation required also increases. Finding true Pareto-optimal front of a general MOOP is an NP-hard problem [22]. Algorithm presented in this work can give a continuous representation but doesn't guarantee true Pareto-optimality.

Response surface approximation (RSA) was used by Goel et.al. [23] to approximate Pareto-optimal front. Their method gives one of the objectives as a function of other objectives. The equation developed for the Pareto optimal front by RSA is valid only in a limited region of the objective space. Moreover, their algorithm was tested on a problem with continuous Pareto-optimal front. Both RSA and the algorithm proposed in this thesis operate on points returned by some optimizer.

Dellnitz et.al. [24] use multilevel subdivision techniques to approximate a Pareto-optimal set. They use a set-oriented approach to create tight boxes covering the Pareto-optimal set. Their algorithm is developed as a stand-alone global optimization method in contrast to the algorithm presented here, which is based on post-processing of points returned by some MOOA. However, their method does not give parameterization of the Pareto-optimal set, closed form equations of the Pareto-optimal set and variations of the design variables along the Pareto-optimal front. Similar to our approach it also works in the variable space rather than objective space.

Several evolutionary algorithms make use of the statistical properties of the solutions of the current generation to create solutions for the next generations [25, 26, 27, 28]. These algorithms are known as estimation of distribution algorithms (EDAs) and have been claimed as paradigm shift in the field of evolutionary computation. Though they don't try to obtain a continuous representation of the Pareto-optimal front, they try to model it using some stochastic model. EDAs build a stochastic model using the promising members of the parent population and create an offspring population based on the model. Thus, traditional mutation and crossover operators of evolutionary algorithms get replaced by the modeling step in EDAs. In Voronoi-based estimation of distribution algorithm (VEDA) [25], Voronoi diagrams are used to create a stochastic model in the variable space. Here clustering techniques are used to divided data in groups and one model is built for each group. Principal component analysis (PCA) is used to reduce the dimensionality of data in each group. Some EDAs try to make use of the mathematical properties of the Pareto-optimal front. RM-MEDA models the promising region of the search space using a probability distribution whose centroid is an $M-1$-dimensional manifold where $M$ are the number of objectives in the problem [29]. Local PCA is used to build this model. Solutions are sampled from the sample and the non-dominated sorting of NSGA-II is used as the replacement step of the algorithm. Some EDAs use the hybrid strategy of using the model in combination with the crossover and mutation operator [28]. However, these algorithms focus on producing points on the front rather than giving a continuous picture.

## 2.2 Finding common design principles

Finding innovative design principles has been investigated in detail by Deb and his collaborators in a series of papers [30, 31, 32]. In a MOOP, only a few out of numerous possible combination of variables are optimum. 'Innovization', proposed by Deb, tries to find answer to the question of what makes these points optimal. It is a post optimization analysis of solution. The process of 'innovization' produces relationships between objectives, variables and constraints. These common principles can be applicable to all the solutions or a subset of solution. Such analysis provides new in-depth knowledge about the solutions which can be beneficial in many ways. Optimization techniques are used to find the relationships between variables, constraints, objectives or any other function by using them as basis. This allows the process to find mathematical relationships that exist between these functions. Clustering is used to identify the subsets that satisfy a common principle.

In this thesis we use cubic B-spline functions as basis to approximate the Pareto-optimal set. As a result we get parametric equations for the entire Pareto-optimal set. Because we use B-splines, total number of such parametric equations depend on the number of control points that model the B-spline. For a complicated Pareto-optimal set, many such cubic parametric equations are generated, which makes finding exact mathematical relationships between variables a cumbersome task. However, parametric equations for all the variables can be easily used to find qualitative relationships between the variables or verify known relationships. We observe that most of the design principles we discover are simple and can easily be concluded by using our procedure. This work focuses on design changes as the designer progresses along the Pareto-optimal front rather than finding exact mathematical relationships between variables. The innovization process does not give a continuous parameterization of the Pareto-optimal set.

Monotonicity analysis was done by Papalambros et al. [33] to bring out important properties among variables for optimal solution. But the objective functions and the constraints need to be monotonic to the decision variables or the objective functions must be free from one or more decision variables. Data mining and knowledge discovering techniques have also been used to find interesting pattern in the data. Multivariate adaptive regression splines (MARS) was used by Deb et al. in [30] in comparison with the innovization process. Parmee et al. [34] proposed a novel method called Cluster Oriented Genetic Algorithms (COGA) to identify high performing regions in the search space. COGA works in the design space to rapidly restrict the domain of interest to regions of high performance. Multi-objective COGA explores the high performing regions independently for each objective and then project them on a hyperplane to find the common high performance region as well as relationships and interaction between objectives. Rather than giving a Pareto-optimal front it focuses on finding design space of high performance. A similar process called activity analysis [35] uses qualitative Kuhn-Tucker conditions to strategically cut away subspaces that it can quickly rule out as suboptimal thus allowing a numerical optimization method to focus on the reduced search space.

## 2.3 Regularity and Connectedness

In this paper, a data fitting method is proposed to give the continuous parameterization of the non-dominated set returned by the MOOA. Thus, a continuous picture is obtained from the discrete data given by the MOOA. The data fitting procedure is carried out in the variable space. The variable space can have high dimension, but the dimension of the manifold to be fitted to the data depends on the number of objectives. It can be shown using the Karush-Kuhn-Tucker (KKT) conditions that for smooth objective and constraint functions, under some mild regularity conditions, the Pareto-optimal set of an $M$-objective MOOP is locally at most an $(M-1)$-dimensional entity (termed as *Hyposurface* in this paper). This property comments on the topology of the Pareto-optimal set which is locally connected. Also, the order of the function describing the Pareto-optimal set in the variable space is lower than or equal to that of the function describing the Pareto-optimal front in the objective space. This has been shown empirically for several standard optimization problems [36]. The regularity and connectedness property has been used by several researchers to design MOOA [37, 38, 39, 40]. Continuation methods to find Pareto-optimal front are based on this property. Starting with a given set of KKT points, continuation methods find new KKT points in the neighborhood and the process is continued till the whole Pareto-optimal set is discovered. Hence, the connected set of KKT points are found by this procedure. Predictor-corrector algorithm, which is a continuation method was used by Shutze et al. [41] and Ringkamp et al. [42] to find the Pareto-optimal set. However, these methods are local in nature and can't find disconnected Pareto-optimal set. Dellnitz et al. [24] also made use of this property to approximate a Pareto-optimal set using multilevel subdivision techniques.

We can have confidence in the data fitting approach because a Pareto-optimal set often consists of connected or piecewise connected points [36]. This means that, between two close points of the Pareto-optimal set returned by the MOOA, we can find more points by data fitting, which lie in the Pareto-optimal set but are not included in the output of the MOOA. This property of connectedness has been studied by several researchers, for example [43, 36, 44]. Also, as the dimension of the fitted entity depends on the number of objective function, under mild regularity conditions, we can pre-determine the number of independent parameters required to model the entity. For example, for a two objective problem, the Pareto-optimal set can be modeled by a curve using one independent parameter only. To detect disconnected parts of the Pareto-optimal set for piecewise connected Pareto-optimal set, we use a clustering procedure. To our knowledge, no study based on regularity and connectedness has tried to find disconnected part of the Pareto-optimal set.

## 2.4 Data fitting using B-spline

Data fitting is a fundamental problem in graphics, data mining and many other application areas. For data fitting we use cubic B-spline basis function with uniform knot vector [45]. Many methods exist for interpolation and approximation of the data points. However, most of the methods assume some initial ordering of the data [46, 47, 48, 49, 50]. The data that is returned by MOOAs is unorganized and can have non-uniform distribution with noise. In an ordered data, where it is known which data point precedes the other, several methods like chord length parameterization and the centripetal method [51, 52, 53] exist to parameterize the data. The regularity and connectedness properties ensure that the Pareto-optimal set lies in an ordered way on a curve, surface or a higher dimensional entity depending on the number of objectives in the problem. The initial data is noisy but the noise can be reduced by using some local search procedure to ensure that the updated points are in the Pareto-optimal set. However, sometimes local search may not be effective and updated data may still be noisy. Moreover, the points returned by the MOOA population may not be in any particular sequence in the array. Thus, the data fitting algorithm should be able to deal with unorganized noisy data. The present algorithm assumes no order in the data and can find an approximation to the noisy data. It can, in principle be extended to an $M$-dimensional manifold in an N-dimensional space.

A common way of parameterization is to choose the parameter value in such a way that the Euclidean distance between the generated entity and the input data points is minimized. Using an iterative procedure the whole entity can be modeled. This parameterization and its simple variants have been used by several researchers for curve fitting [54, 55, 56, 57]. The idea extended to surface fitting has also been used by several researchers [58, 59, 60, 61, 62, 63, 64]. Initial guess value of parameter plays an important role in the fitting process. For highly non-linear fitting process a random guess may not lead to a low fitting error. Most of the fitting procedure require the end-user to supply a good initial parameter values. If an initial estimate is not available, it becomes important to generate it by ordering the data. A number of approach have been developed to solve this issue. Fang et al. [65] proposed a novel method which simulates the deformation of a perfectly elastic beam under the application of spring force for reconstructing a smooth curve from a set of unordered and error filled data points. Their method doesn't need an initial parameter estimate but required end points of the data as input. Yan [66] proposed a fuzzy curve tracing (FCT) algorithm to extract smooth curve from unordered noisy data. Their method is based on finding cluster centers and reordering them using relational graphs. Liu et al. [67] developed a novel method based on stretching curves along the tangential direction. A simple curve is placed somewhere along the point cloud and then is made to stretch and crawl along the point cloud. Their method is restricted to planar curves only. Goshtasby [68] proposed a novel method to fit irregularly spaced set of points. The idea is to divide points into subsets, order points in each subset and fit a curve through them. A potential function is constructed based on the mapped points in an image to generate a new gray image and compute ridge contours. Then the mapped points are ordered by their projections on the ridge contours and a fitting curve is reconstructed from the ordered points.

Principal curves, defined by Hatsie and Stuetzle [69] are non-linear generalization of principal components. They are defined as self-consistent smooth curves that pass through the middle of a data cloud. They have been used in several fields including data fitting. Furferi et al. [70] also used PCA to order the data set. Clusters are found using PCA and polyline is constructed joining cluster centroids. B-spline is constructed for points lying on each polyline. The points not lying on the polyline are projected on the B-spline and their order is used for initial parameterization. Principal components are also used in this thesis to find an initial parameterization of the data. If the parameterization is not good enough and the fit is poor, the data is divided into several nodes recursively, thus creating a tree like structure. Quadtree-like data structures called strip tree has been used by Gregorski et al. [71] in reconstruction of B-spline surface from scattered data points. In their method, a bounding box is created around the data in each node and control points are found using the data. The data fitting algorithm presented in this thesis is inspired from their work on surface fitting, however it is more general, uses iterative fitting procedure and a different strategy of joining nodes to create smooth curves, surfaces or higher dimensional entities.

Chapter 3

# Mathematical Background and Formulations

This chapter gives the mathematical formulations and a brief background necessary for the understanding of the thesis. Basics of MOO were discussed in Chapter 1. In this chapter, KKT conditions are discussed, followed by discussion of regularity and connectedness properties of the Pareto-optimal set. After a brief description of B-spline basis functions, we formulate the data fitting problem. In the end, the structure of the algorithm is described.

## 3.1 Karush Kuhn Tucker conditions

MOOP in its general form was stated in Chapter 1, Equation 1.1. Here, it is stated again for easy reference.

$$\begin{aligned} \text{Minimize/Maximize} \quad & f_m(\mathbf{x}) & m = 1, 2, \ldots M; \quad (3.1) \\ \text{subject to} \quad & g_j(\mathbf{x}) \geq 0 & j = 1, 2, \ldots J; \quad (3.2) \\ & h_t(\mathbf{x}) = 0, & t = 1, 2, \ldots T; \quad (3.3) \\ & x_k^{(L)} \leq x_k \leq x_k^{(U)}, & k = 1, 2, \ldots K. \quad (3.4) \end{aligned}$$

where $f_m$, $g_j$, $h_t$, $x_k^{(L)}$ and $x_k^{(U)}$ represent the objective functions, the inequality constraints, the equality constraints and the lower and upper bounds on the variables, respectively. In this thesis, we will restrict ourselves to optimization problems with smooth objectives and constraints. The MOOP has $K$ variables, $M$ objectives, $J$ inequality constraints, and $T$ equality constraints.

As explained in Chapter 1, the solutions of a MOOP with conflicting objectives is a set of points referred to as the Pareto-optimal set. The image of the Pareto-optimal set in the objective space is called the Pareto-optimal front. For a MOOP with differentiable objectives and constraints a point $\mathbf{x}^*$ in the Pareto-optimal set will necessarily satisfy the Karush Kuhn Tucker (KKT) conditions [72, 73]. The KKT conditions are stated as following:

$$\begin{aligned}
\sum_{m=1}^{M} \lambda_m \, \nabla f_m(\mathbf{x}) \, - \, \sum_{j=1}^{J} \mu_j \, \nabla g_j(\mathbf{x}) - \sum_{t=1}^{T} \nu_t \, \nabla h_t(\mathbf{x}) &= \mathbf{0} && (3.5)\\
\sum_{m=1}^{M} \lambda_m &= 1; &&\\
\mu_j g_j(\mathbf{x}) &= 0, && \forall j \in [1, \ldots J];\\
\lambda_m \geq 0, & && \forall m \in [1, \ldots M];\\
\mu_j \geq 0, & && \forall j \in [1, \ldots J];\\
g_j(\mathbf{x}) \geq 0, & && \forall j \in [1, \ldots J];\\
h_t(\mathbf{x}) = 0, & && \forall t \in [1, \ldots T].
\end{aligned}$$

where $\lambda$ and $\mu$ represent the lagrange multipliers for the objectives and the constraints, respectively.

A point satisfying the KKT conditions is a candidate solution for optimality. Above conditions are necessary (under some regularity assumptions) but not sufficient conditions for Pareto-optimality. All points on the Pareto-optimal front (and hence in the Pareto-optimal set) will satisfy the KKT conditions. However, as the KKT conditions are not sufficient conditions for Pareto-optimality, not all KKT points are optimal points.

For an unconstrained single objective problem, it can be verified that KKT conditions reduce to $\nabla f = 0$ which are first order necessary conditions for optimality. There are certain regularity assumptions on constraints for KKT conditions to hold true at the optimal point. These are called constraint qualifications. If the constraints satisfy the constraint qualification the KKT conditions are valid. Various authors have given several constraint qualifications [74, 75]. Linear independence of constraints qualification (LICQ) is the most popular constraint qualification. Let $I = \{j | g_j(\mathbf{x}) = 0\}$ denote the set of active inequality constraints. If $\nabla g_j(\mathbf{x})$, $\forall j \in I$ and $\nabla h_t(\mathbf{x})$, $\forall t \in [1, \ldots T]$ are linearly independent then LICQ are met.

## 3.2 Regularity and Connectedness

It has been observed that there is a regularity in the distribution of points in the Pareto-optimal set. Using the KKT conditions, regularity in the Pareto-optimal front solutions can be inferred under certain smoothness conditions. Regularity implies that for an $M$-objective problem, under certain conditions [76], the Pareto-optimal front is at most an $(M-1)$-dimensional piecewise continuous manifold in both the objective and the variable space. This is an important property which has mostly gone unnoticed in the EMO field [29]. Here, this property is used to ensure that for an $M$-objective problem we need $M-1$ parameters to parameterize the Pareto-optimal set.

A closer look at the KKT conditions (Equation 3.5) reveals that there are $K+J+T+1$ equations in $K+J+T+M$ variables. This can be written in form of a vector function $\mathbf{G}(\mathbf{x},\lambda,\mu,\nu)$ as shown below.

$$\mathbf{G}(\mathbf{x},\lambda,\mu,\nu)=\begin{pmatrix} \sum_{m=1}^{M}\lambda_m\,\nabla f_m(\mathbf{x})\;-\;\sum_{j=1}^{J}\mu_j\,\nabla g_j(\mathbf{x})-\;\sum_{t=1}^{T}\nu_k\,\nabla h_t(\mathbf{x}) \\ \mu_j g_j(\mathbf{x}) \\ h_t(\mathbf{x}) \\ \sum_{m=1}^{M}\lambda_m-1 \end{pmatrix} \tag{3.6}$$

For a point satisfying KKT conditions, $\mathbf{G}(\mathbf{x},\lambda,\mu,\nu)\;=\;\mathbf{0}$. By implicit function theorem, under some smoothness conditions the level set of $\mathbf{G}(\mathbf{x},\lambda,\mu,\nu)$ is a $(M-1)$-dimensional manifold. A detailed discussion on regularity conditions is done by Hillermeier in [76].

Connectedness implies that the members of the Pareto-optimal set and front are connected in a topological sense. If the Pareto-optimal front or the Pareto-optimal set consists of a finite number of disconnected regions, it is called loosely connected set [36]. It has been shown that Pareto-optimal set is connected if all the objective functions are convex [44]. The convexity condition can be relaxed for several kinds of problems. For a detailed description of connectedness, see [44, 43]. To find the disconnected Pareto regions, a clustering process is suggested. Connectedness is important as it provides confidence that the data fitting procedure will be able to find members of the Pareto-optimal set in the neighborhood of already known Pareto-optimal set members.

An $M$-dimensional manifold in a $K$-dimensional space represents a curve for $M=1$, a surface for $M=2$ and a hypersurface for $M=K-1$. 'Manifold' is used as a generalized term for the intermediate dimensions. The term 'manifold' has a specific definition in mathematics and topology, and can not be used casually. For the lack of any other well-established term, we define a term *Hyposurface* as a substitute to the term 'manifold'.

**Definition 5.** *An M-dimensional hyposurface in an $N$-dimensional space $(1\leq\ M\ <N)$ is an entity which requires M independent parameters for its complete description.*

Thus, an 1-d hyposurface is a curve and a 2-d hyposurface is a surface. From here on the use of the term 'manifold' will be avoided and instead of referring data fitting as curve or surface fitting, general term *hyposurface fitting* will be used.

## 3.3 Hyposurface Fitting

Cubic B-spline basis functions with uniform knot vector are used for modeling the hyposurface. They provide a convenient matrix notation for hyposurface fitting. Control points of B-spline hyposurface provide a good global and local control. The number of control points ($N_{cp}$) required to model the B-spline hyposurface depend on the dimension of the hyposurface and the order of the B-spline basis functions used for the modeling. For an $M$-dimensional hyposurface, modeled by B-splines, in a $K$-dimensional space, any point on the surface can be given by

$$x_j \;=\; F(u_1,\, u_2,\, \ldots, u_M,\, \mathbf{P_j}), \qquad\qquad \forall\, j \in [1,\, \ldots,\, K], \tag{3.7}$$

where $F$ is the B-spline basis function which takes parameters ($\mathbf{u}$) and control points ($\mathbf{P}$) as the arguments, $\mathbf{x}$ is a K-dimensional vector representing a point on the hyposurface, $u_1, u_2, \ldots, u_M$ represent the M parameters corresponding the point $\mathbf{x}$ that model an M-dimensional hyposurface, $\mathbf{P_j}$ represent the $j^{th}$ column of an $N_{cp}{\times}K$ matrix of control points modeling the hyposurface. Detailed theory of B-spline curves and surfaces is covered in Mortenson [45].

Here, the data fitting problem is defined. Given a set of data points, $\mathbf{x_1}$, $\mathbf{x_2}$,. . . ,$\mathbf{x_N}$, where each data point is a $K$-dimensional vector ($\mathbf{x_i} = [x_{i1}\ x_{i2}\ \ldots x_{iK}]^T$), and a function $f(u_{i1},\ u_{i2},\ \ldots,\ u_{iM},\ \mathbf{P_k})$: $\mathbb{R}^{M+N_{cp}} \to \mathbb{R}$, where $u_{ij} \in \mathbb{R}\ \forall\ j \in [1, \ldots M]$ are the M parameters representing the $i^{th}$ point on the $M-$dimensional hyposurface in space and $\mathbf{P_k}$ is the k-th column of an $N_{cp}\mathbb{X}K$ dimensional matrix of control points that models the hyposurface, the problem of hyposurface fitting is to find the appropriate values of $\mathbf{P}$ and parameters $\mathbf{u_i} \in \mathbb{R}^M$ corresponding to each data point, $i$ such that $f(\mathbf{u_i},\ \mathbf{P_j}) = x_{ij}$, $\forall i \in [1, \ldots, N]$, and $\forall j \in [1, \ldots, K]$. The problem of hyposurface fitting inherently leads to a least square problem which is mathematically stated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{N} \left(f(\mathbf{u_i}, \mathbf{P_j}) - x_{i,j}\right)^2, \qquad\qquad \forall j \in [1, \ldots, K]\,. \tag{3.8}$$

The problem, stated in other words is the minimization of the distance between the input points $\mathbf{X}$ and the hyposurface modeled by the function $f$, where the parameters assigned to each data point ($\mathbf{u_i}$) and the control points ($\mathbf{P}$) that model the hyposurface are the unknowns.

The complete problem of parameterizing the Pareto-optimal set can be stated as following:

**Problem 1.** *Given the non-dominated set of points of an $M$-objective MOOP in $N$ variables, returned by some MOOA, find the control points and the parameterization which model the Pareto-optimal set of the given MOOP.*

## 3.4 Structure of the algorithm

We propose an algorithm to solve the above problem. The proposed algorithm can be divided into two major parts. First part is a data pre-processing step where the given input data ($\mathbf{X}$) undergoes an error reduction step and a clustering step. In the error reduction step, KKT-error of the data is reduced. Clustering is used to find the disconnected parts of the Pareto-optimal set. The second part of the algorithm deals with the hyposurface fitting. Figure 3.1 shows the major steps of the algorithm pictorially.

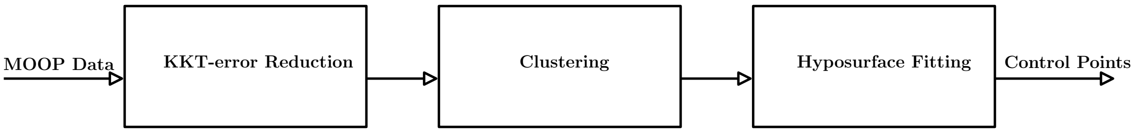


Figure 3.1: Main steps of the algorithm

In Chapter 4, data pre-processing steps are explained followed by description of hyposurface fitting in Chapter 5.

Chapter 4

# Data Pre-processing: KKT-error reduction and Clustering algorithms

In this chapter two important parts of the algorithm, KKT-error reduction and Clustering are discussed. These are termed as pre-processing steps as the next step of the algorithm is the hyposurface fitting where actual data fitting takes place.

## 4.1 KKT-error Reduction

### 4.1.1 Motivation

As statistical methods for approximating the Pareto-optimal set are used in this study, it is important that the input data has a low error. Lower the error, better will be the modeled Pareto-optimal set. Non-dominated points returned by certain MOOAs like the EMO algorithms, may not necessarily be Pareto-optimal. In NSGA-II [12], which is a popular EMO algorithm, population quickly gets close to the Pareto-optimal front and all the members in the population become non-dominated after a few generations only. These non-dominated members may have some Pareto-optimal as well as some sub-optimal solutions. It becomes difficult for this non-dominated population to generate better solutions. As NSGA-II is dependent on crossover and mutation only, to create new solutions, the chances of creating better solutions reduce as such solutions reduce in space because of the proximity of the current solutions to the Pareto-optimal front. Also, because of diversity preserving measures like crowding distance, any Pareto-optimal solution that is obtained, may be lost. This may lead to slow convergence to the true Pareto-optimal front. This problem is called Pareto drift and has been studied by Goel et al. [23]. One possible solution suggested by them, is the use of an archive to store all the non-dominated points. Another remedy is the use of local search on the sub-optimal points to take them to the Pareto-optimal front. We use local search on the sub-optimal points returned by NSGA-II algorithm. As local search is computationally expensive, not all input data is local searched. First an error estimate is found indicating the proximity of points to the Pareto-optimal front. Local search is conducted only for members with high error estimate.

### 4.1.2 KKT-error calculation

KKT-error ($\varepsilon$) is defined to estimate the error in the input data. Using the KKT conditions defined in Equation 3.5, we define the KKT-error for an input point $\mathbf{X_i}$ as:

$$\varepsilon = \| \sum_{m=1}^{M} \lambda_m \frac{\nabla f_m(\mathbf{X_i})}{\| \nabla f_m(\mathbf{X_i}) \|} - \sum_{j=1}^{J} \mu_j \frac{\nabla g_j(\mathbf{X_i})}{\| \nabla g_j(\mathbf{X_i}) \|} - \sum_{t=1}^{T} \nu_t \frac{\nabla h_t(\mathbf{X_i})}{\| \nabla h_t(\mathbf{X_i}) \|} \|^2 \tag{4.1}$$

At a point in space satisfying the KKT conditions, the vectors $\lambda \nabla f$, $\mu \nabla g$ and $\nu \nabla h$ are in equilibrium and are linearly dependent. The KKT-error term ($\varepsilon$) is a measure of the degree of imbalance in these vectors at every input point. The degree of imbalance does not depend on the size of these vectors, therefore, unit vectors $\frac{\nabla f_m(\mathbf{X_i})}{\|\nabla f_m(\mathbf{X_i})\|}$, $\frac{\nabla g_j(\mathbf{X_i})}{\|\nabla g_j(\mathbf{X_i})\|}$ and $\frac{\nabla h_t(\mathbf{X_i})}{\|\nabla h_t(\mathbf{X_i})\|}$ can be used. This has the advantage of making $\varepsilon$ independent of the scale of the objectives, hence, permitting the use of a common threshold error value ($\varepsilon_{thresh}$) for all the problems.

At a point on the Pareto-optimal front $\varepsilon = 0$. The problem is to find the value of vectors $\lambda$, $\mu$ and $\nu$ such that $\varepsilon$ is minimized. To calculate the error, the bounds on the variables are converted into inequality constraints. Active constraints are found and their gradients are calculated. For weights, $\lambda \neq \mathbf{0}$ and each $\lambda_m \geq 0$, we do a normalization by using $\sum_{m=1}^{M} \lambda_m = 1$. The problem can be mathematically defined as:

$$\text{Minimize} \quad \varepsilon \tag{4.2}$$

$$\text{such that} \quad \sum_{m=1}^{M} \lambda_m = 1 \tag{4.3}$$

$$\lambda_m \geq 0 \qquad \forall m \in [1, \dots M] \tag{4.4}$$

$$\mu_j \geq 0 \qquad \forall j \in [1, \dots J] \tag{4.5}$$

$$\nu_t \geq 0 \qquad \forall t \in [1, \dots T] \tag{4.6}$$

Above problem is quadratic in $\lambda$, $\mu$ and $\nu$. We use SQP implemented by MATLAB to solve the above optimization problem. Minimum value of $\varepsilon$ for each input point, as returned by SQP, is stored and compared with a minimum threshold value ($\varepsilon_{thresh}$). Points for which $\varepsilon > \varepsilon_{thresh}$ are tagged as erroneous points and a local search is conducted for them, to reduce the KKT-error.

### 4.1.3 Local Search

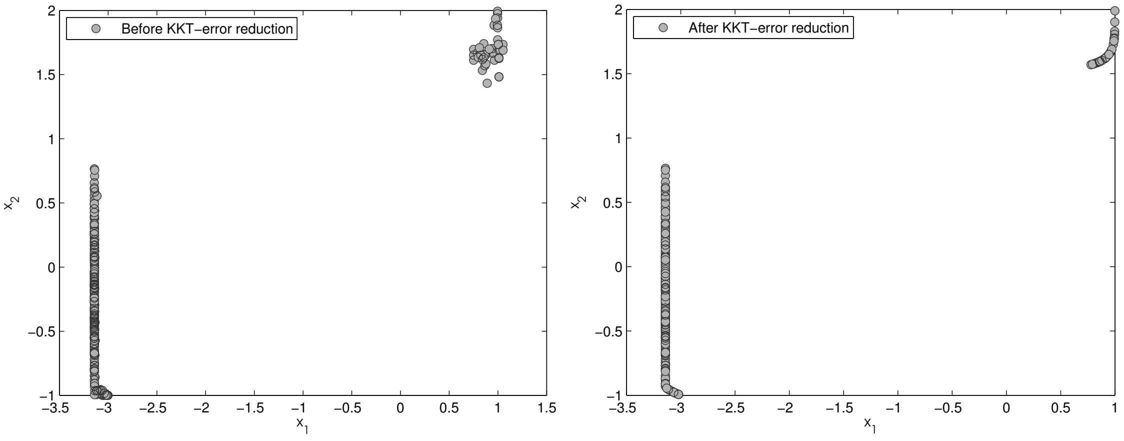
Local search based on minimizing an achievement scalarizing function (ASF) [77] is conducted. ASF has been successfully used by Deb et al. [78] to conduct the local search. ASF uses a reference point $\mathbf{z}$, which is a vector of objective function values, at the erroneous point and finds the solution on the Pareto-optimal front close to the reference point. The single-objective problem solved to get a point on the Pareto-optimal front is defined as:

$$\text{Minimize} \quad \max_{i=1}^{M} \left[w_i \left(f_i(\mathbf{x}) \; - \; z_i\right)\right], \tag{4.7}$$

$$\text{such that} \quad x \in \mathcal{S}, \tag{4.8}$$

where $\mathbf{w}$ is a vector of weights used to scalarize the function and $\mathcal{S}$ is the feasible search space. The reference point $\mathbf{z}$ helps to focus on a particular part of the Pareto-optimal front where as the weight vector provides the finer tradeoffs between the objectives leading to convergence on a particular point on the Pareto-optimal front [79]. The erroneous input point for which local search is to be conducted is taken as the reference point. The $\lambda$ vector found during the process of KKT-error calculation is used to calculate the weights for the local search procedure. $w_i = \frac{\lambda_i}{up_i^{bd} - lw_i^{bd}}$ is used as weight for each objective function where $up_i^{bd}$ and $lw_i^{bd}$ are the upper and lower bounds of the $i^{th}$ objective function. Max function in the problem definition causes the problem to be non-smooth. The problem is converted into an equivalent smooth problem and is solved using SQP routine of MATLAB. The smooth variation of the problem is shown in Equation 4.9.

$$\begin{aligned} \text{Minimize} \quad & \varepsilon, \\ \text{Subject to} \quad & w_i(f_i(\mathbf{x}) - z_i) \; \le \; \varepsilon, \quad \forall i \in [1, \ldots, M], \\ & x \in \mathcal{S}, \\ & \varepsilon \; \le \; 0. \end{aligned} \tag{4.9}$$



(a) Before KKT-error Reduction

(b) After KKT-error Reduction

Figure 4.1: KKT-error reduction leads to a better organization of points

Figure 4.1 shows the KKT-error reduction for ‘POL’ problem. Figure 4.1a shows the points before the KKT-error reduction. In Figure 4.1a, the points on the left seem to have a lower error as a curve can be visualized to fit through these points, but the points on the right are highly unordered and no curve can fit through these points with a low fitting error. After KKT-error reduction points have arranged themselves in a better order for both the clusters. A regularity in the points was expected and the arrangement of points in an orderly fashion after KKT-error reduction shows the same.

## 4.2 Clustering

The Pareto-optimal set of a MOOP may be disconnected. To take care of this, a clustering procedure is carried out to find the gaps that may exist in the Pareto-optimal set. The clustering process divides the corrected input data into $K$ clusters where K is the number of disconnected Pareto regions that exist. The procedure should give exactly the same number of clusters as the number of disconnected Pareto regions. For example, for a connected data the procedure should not divide data at all, for a problem with two disconnected Pareto regions the cluster should return exactly two sets of data.

As the number of clusters that may exist are unknown, the clustering process is carried out in three steps viz. approximate estimate of the number of clusters, K-means clustering based on the estimate and the combination of clusters which are close to each other. Figure 4.2 pictorially shows this process.

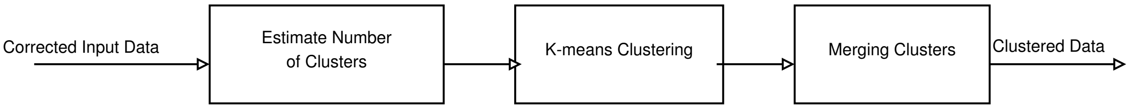


Figure 4.2: Pictorial representation of clustering process

### 4.2.1 Estimating the number of clusters

As the number of clusters are unknown subtractive clustering [80] is used, initially, to find an initial estimate of the number of clusters and cluster centers. Subtractive clustering is a fast, one-pass method to determine the number of clusters and cluster centers in a set of data [81]. Subtractive clustering process assigns a cluster potential index to each data points which measures the potential of a data point to be a cluster center. Initially the point with the highest cluster potential index is chosen as the first cluster center after which the index is updated. The process continues till a termination criteria is matched. Suppose the data set to be clustered has N data points, $X_1$, $X_2$, $\ldots$, $X_N$ in a K-dimensional space. Cluster potential for each point is calculated as

$$P_i = \sum_{n=1}^{N} \exp\left(-\frac{\|X_i - X_n\|}{(r_a/2)^2}\right) \tag{4.10}$$

where $r_a$ represent the radius of influence corresponding to the cluster center $X_i$. The points beyond the radius of influence don't significantly contribute to the potential index. The point $(X^*)$ with the highest cluster potential index $(P^*)$ is marked as cluster center and potential for all points is then updated as

$$P_i = P_i - P^* \exp\left(-\frac{\|X_i\ -\ X^*\|}{(r_b/2)^2}\right), \tag{4.11}$$

where $r_b$ is a number that define the radius of influence of $X^*$ on the potential of other data points. The process is iteratively repeated till some termination criteria is reached.

### 4.2.2 K-means Clustering

Subtractive clustering gives us only an approximate estimate of the number of clusters in the data set and doesn't associate a cluster with every data point. Moreover, the cluster center found by subtractive clustering may not not be as good as those by K-means clustering [82]. Therefore, the cluster centers and number of clusters returned by subtractive clustering are fed as initial center estimates for K-means clustering.

K-means clustering, as the name suggests, is used to partition N member data set into K clusters where K is an input argument. Given N data members, $X_1, X_2, \ldots, X_N$ and K initial cluster centers, $c_1, c_2, \ldots, c_K$, the problem is to partition data in K sets, $S_1, S_2, \ldots, S_K$, such that for each set within cluster variance is minimized. The function to be minimized is

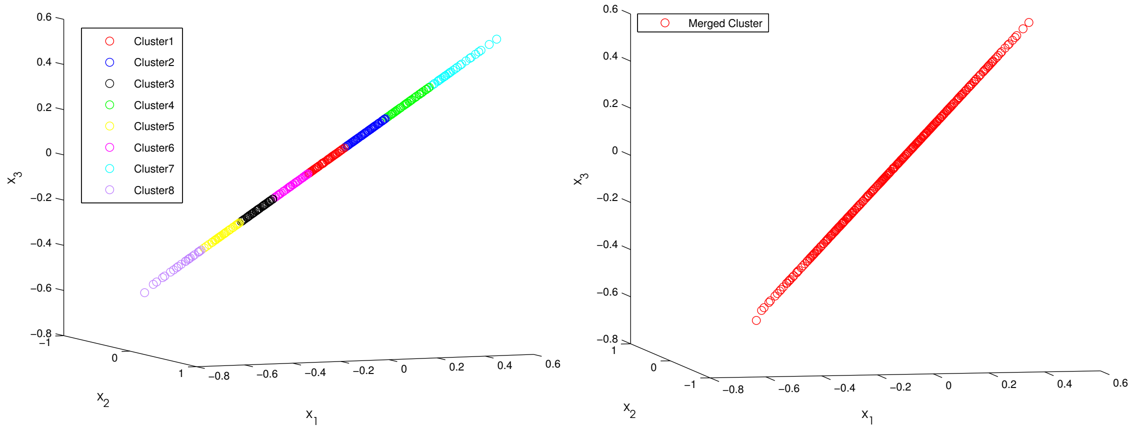
$$min \sum_{i=1}^{K} \sum_{X_j \in S_i} \|X_j - C_i\|^2 \tag{4.12}$$

where $C_i$ is the centroid of $i^{th}$ cluster. At every iteration, the centers $\mathbf{C}$ and the data members associated with each set are varied, till the function is minimized. Each data member is assigned a closest cluster center, leading to Voronoi partition of the data [83].

Hybridizing subtractive clustering with K-means clustering gives good results when initial number of clusters are unknown. Use of such hybridized clustering procedures has been advocated by several researchers [84, 82]. MATLAB routines for subtractive clustering and K-means clustering are used in this work.

### 4.2.3 Merging Clusters

The above clustering procedure can return more than the expected number of clusters. Therefore, a cluster correction step is carried out in which clusters with small minimum distance between them are combined together to form a larger cluster. The threshold for the minimum distance is decided on the basis of the bounds of the data. As the main concern here is to divide data into clusters and cluster centers are not important, taking a union of the sets to be merged solves the problem.



(a) Erroneous result of initial clustering

(b) Corrected result after merging clusters

Figure 4.3: Erroneous and corrected clustering

One such situation is shown in Figure 4.3a. Here initial clustering step gives eight clusters when only one was expected. The cluster correction step joins all the clusters to give a single large cluster (Figure 4.3b). Figure 4.4 shows the clustering results for 'POL' problem. Two clusters are identified by the clustering process here. The next step of the algorithm is the Hyposurface fitting which is discussed in the following chapter.

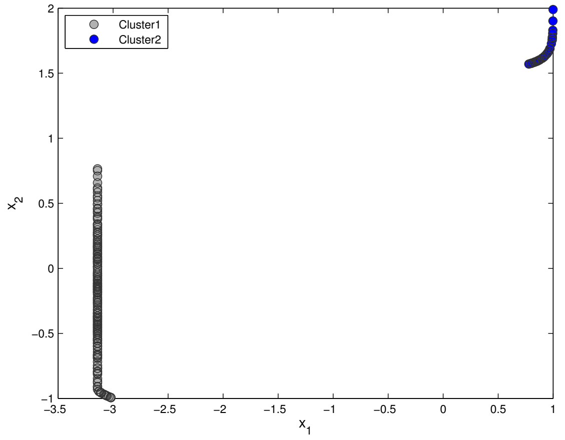


Figure 4.4: Clusters returned by clustering process for 'POL' problem

Chapter 5

# Hyposurface Fitting

Next step after KKT-error reduction and clustering is hyposurface fitting. In this chapter a detailed description of hyposurface fitting procedure is made. Cubic B-splines are used as basis functions to describe the hyposurface. The process of fitting is an iterative two-step process. The fitting process includes several steps like finding outliers in the data and assigning initial parameter values, all of which are explained in this chapter. At the end, joining of hyposurface segments together with $C^2$ continuity is discussed.

## 5.1 Uniform cubic B-spline basis function

In Section 3.3, the problem of hyposurface fitting was discussed through equation (3.8). For uniform open cubic B-spline data fitting, f$(\mathbf{u_i}, \mathbf{P_j})$ can be defined by using a convenient matrix notation [45]. For example, for 1-d hyposurface

$$f(\mathbf{u_i}, \mathbf{P_j}) = \begin{bmatrix} u_{i1}^3 & u_{i1}^2 & u_{i1} & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{6} & \frac{3}{6} & \frac{-3}{6} & \frac{1}{6} \\ \frac{3}{6} & \frac{6}{6} & \frac{3}{6} & \frac{0}{6} \\ \frac{-3}{6} & \frac{0}{6} & \frac{3}{6} & \frac{0}{6} \\ \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & \frac{0}{6} \end{bmatrix} \begin{bmatrix} p_{i-1,j} \\ p_{i,j} \\ p_{i+1,j} \\ p_{i+2,j} \end{bmatrix}, \tag{5.1}$$

where $i \in [1, \dots N]$ and $j \in [1, \dots K]$ where $N$ is the number of data members of the node and $K$ is the dimension of the data. Similar matrix notation can also be written for a 2-d hyposurface.

The advantage of using B-spline functions is that they can provide a good local and global control on the modeled hyposurface by varying the control points. A cubic B-spline modeled hyposurface consists of composite sequence of hyposurface segments connected with $C^2$ continuity. Every control point models a few segments of the B-spline hyposurface. Each segment of the B-spline is controlled by $(t + 1)$ control points, where $t$ is the degree of the polynomial defining a B-spline segment. For cubic B-spline segments, every control point has influence on four segments, and conversely every segment is influenced by four control points.

## 5.2 Hyposurface fitting procedure

Given $N$ data points, $\mathbf{u}$ and $\mathbf{P}$ are the unknowns for the minimization problem (3.8). The problem for B-spline curves can be stated as:

$$\underset{\mathbf{u},\mathbf{P}}{\operatorname{argmin}} \| \sum_{j=1}^{K} \sum_{i=1}^{N} \begin{bmatrix} u_{i1}^3 & u_{i1}^2 & u_{i1} & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{6} & \frac{3}{6} & \frac{-3}{6} & \frac{1}{6} \\ \frac{3}{6} & \frac{6}{6} & \frac{3}{6} & \frac{0}{6} \\ \frac{-3}{6} & \frac{0}{6} & \frac{3}{6} & \frac{0}{6} \\ \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & \frac{0}{6} \end{bmatrix} \begin{bmatrix} p_{i-1,j} \\ p_{i,j} \\ p_{i+1,j} \\ p_{i+2,j} \end{bmatrix} - x_{ij} \|^2, \tag{5.2}$$

$$\textit{subject to} \qquad 0 \leq u_{ij} \leq 1, \qquad \forall i \in [1, \ldots N], \quad j \in [1, \ldots K]. \tag{5.3}$$

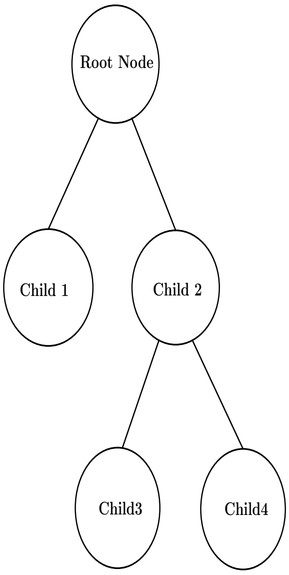
Similar problem formulation can also be stated for B-spline surfaces and higher dimensional hyposurfaces. The problem is linear in control points $\mathbf{P}$ and non-linear in parameters $\mathbf{u}$. Starting with a good guess for $\mathbf{u}$ values, an iterative procedure is used to find the unknowns that minimize the error function . The iterative procedure consists of following two steps:

1. Finding the control points: Keeping the parameters values fixed find the optimal control points that minimize the error.

2. Updating the parameters: Keeping the control points found in previous step fixed, find the parameter values that minimize the error.

Both steps are repeated till either the error is reduced below the fixed threshold or maximum number of iterations are reached. If the error is large, the data is subdivided into several nodes using a full $k$-way tree approach. A full $k$-way tree is a rooted tree in which each node other than the leaf nodes has exactly $k$ children. Hyposurface fitting is done during the process of tree creation, using the above two-step iterative process, with acceptable data fitting leading to leaf node creation. The procedure for tree building goes as follows. Initially all the data is in the root node of the tree. Hyposurface fitting takes place for the data in the node. If the fit is not acceptable, the node is further divided into $2^M$ children nodes where $M$ is the dimension of the hyposurface. For every child node the above process is recursively repeated. At the end of the recursion, hyposurface model stored in every leaf node is joined together to give a continuous or piecewise continuous hyposurface.

See Figure 5.1a for an example of a 2-way tree. Here, the initial fit for the data in the root node had an error greater than the threshold. Therefore, the data was divided into two nodes, child 1 and child 2. Fitting for the data in child 1 node was acceptable, so no further subdivision takes place and child 1 becomes the leaf node. For node child 2, fitting was still not good and subdivision of the data in further two children nodes: child 3 and child 4, takes place. Fitting for these two nodes is satisfactory and they become the leaf node. Figure 5.1a shows a flowchart of the algorithm.



(a) 1-way tree

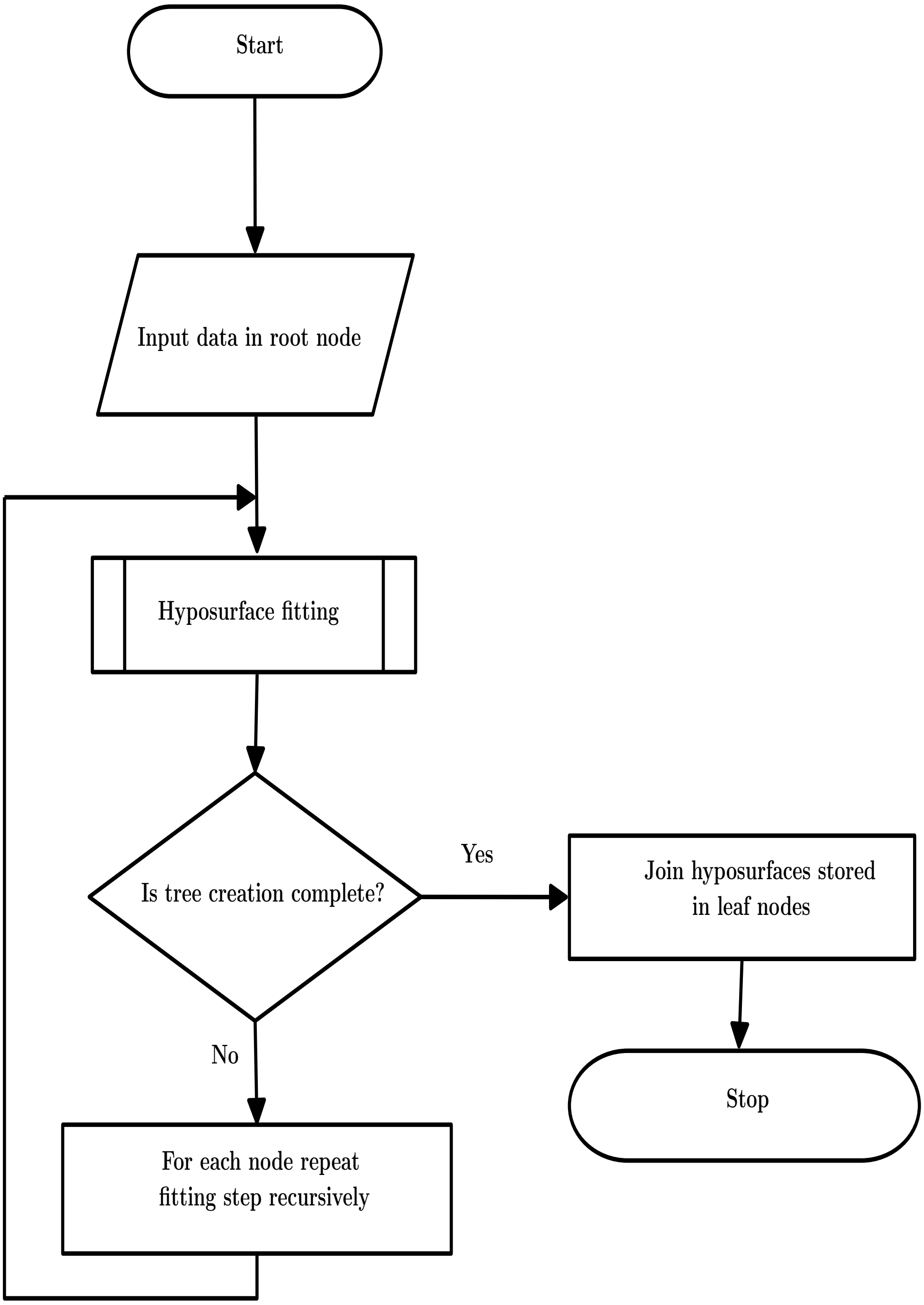
For the data in each node, several steps are carried out for hyposurface fitting. Figure 5.2 shows a flowchart of these steps. A description of these steps follows.

### 5.2.1 Finding Outliers

The purpose of this subroutine is to find those data members in the node, which are dramatically different from the rest of the data members in the same node. An archive of outliers is maintained, which are added to the members of the current node. After this, the centroid of the data is found. Data members which lie more than three times the radius distance from the centroid are deleted from the current node and are stored as the new updated archive. The data members which are outliers for one node may represent valid members for other nodes. This process ensures that no information is lost.

### 5.2.2 Principal Component Analysis

The next step is to perform the principal component analysis (PCA) of the data members in the node. The aim of doing PCA is to find an initial parameterization of the data. PCA helps in finding a new basis system in which the covariance between the variables is minimized, i.e. the covariance matrix becomes diagonal in the new basis system. The problem is to find a linear transformation, $\mathbf{L}$, such that the data points ($\mathbf{X}$) after transformation ($\mathbf{Y}$) have a diagonal covariance matrix. Find a orthonormal matrix $\mathbf{L}$ such that $\mathbf{Y} = \mathbf{LX}$ and $cov_Y = \frac{1}{N-1}\mathbf{YY^T}$, is diagonalized. It can be mathematically shown that the principal components of the data matrix $\mathbf{X}$ are the eigenvectors of matrix $\mathbf{XX^T}$ [85].


Start
Input data in root node
Hyposurface fitting
Is tree creation complete?
Yes
Join hyposurfaces stored in leaf nodes
Stop
No
For each node repeat fitting step recursively

(a) Flowchart of the algorithm

Figure 5.1

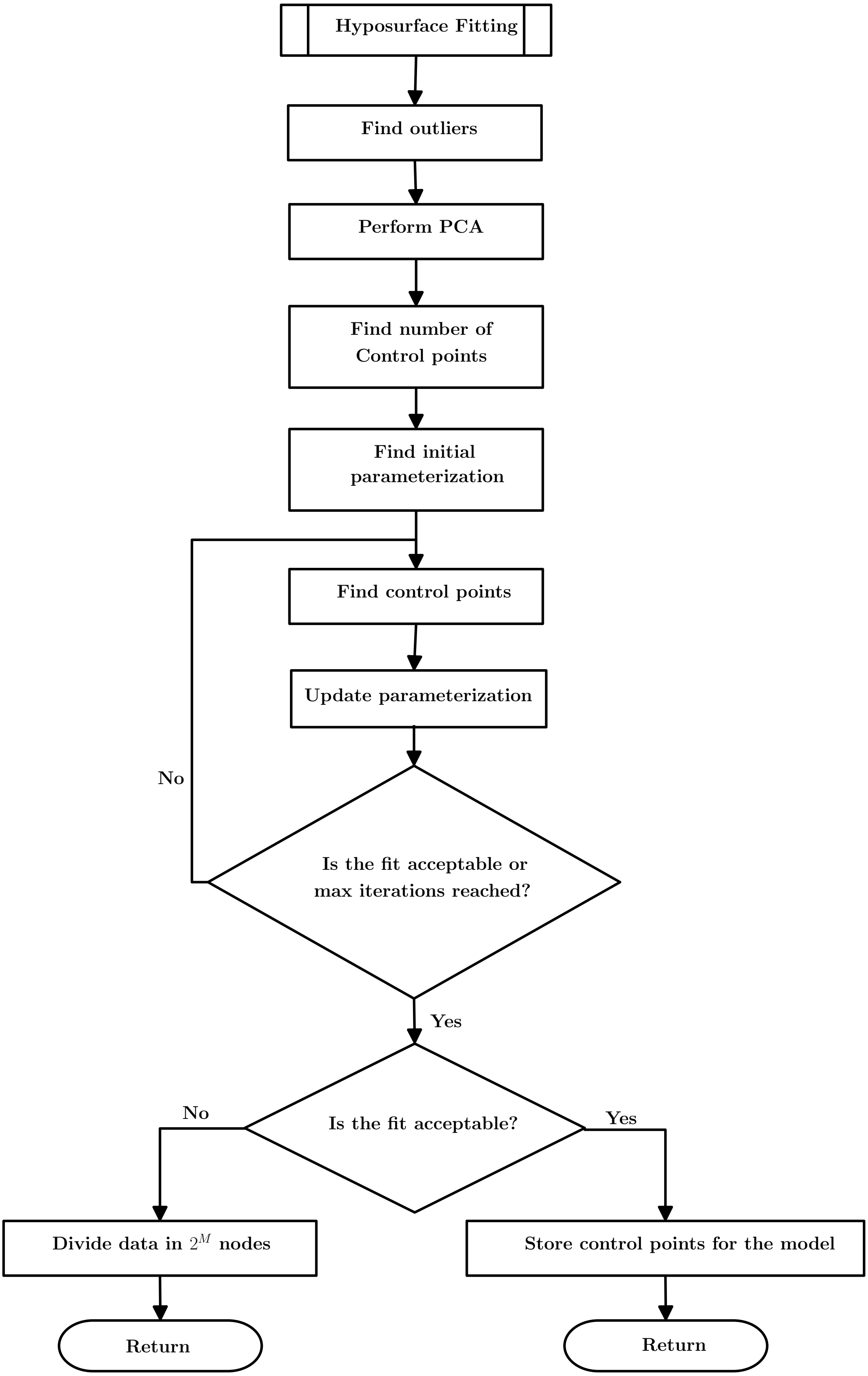

Hyposurface Fitting
Find outliers
Perform PCA
Find number of Control points
Find initial parameterization
Find control points
Update parameterization
No
Is the fit acceptable or max iterations reached?
Yes
No
Is the fit acceptable?
Yes
Divide data in $2^M$ nodes
Store control points for the model
Return
Return

Figure 5.2: Flowchart showing the hyposurface fitting procedure

The principal directions are sorted according to the corresponding eigenvalues in a descending order and the data is projected along each of the first $M$ principal directions. Scaling the projected data in $[0, 1]$ provides a good estimate of the initial parameterization.

It is important that all the neighboring nodes have similar orientation of the directions returned by the PCA, as it helps in maintaining a proper continuity in the parameterization necessary for joining the modeled segments with $\mathcal{C}^2$ continuity, at the end. To ensure this, the sign of the dot products of eigenvectors of the parent node and the child node is checked. If the sign is negative, direction for the child node is flipped. This process ensures that the parent and the children are similarly oriented.

PCA is also used to decide the number of control points $(m)$ in each direction for the B-spline hyposurface. Total number of control points are fixed as some percentage of the number of data members in each node. These control points are divided between each of the M orthogonal directions in proportion of the eigenvalue associated with the corresponding eigenvector.

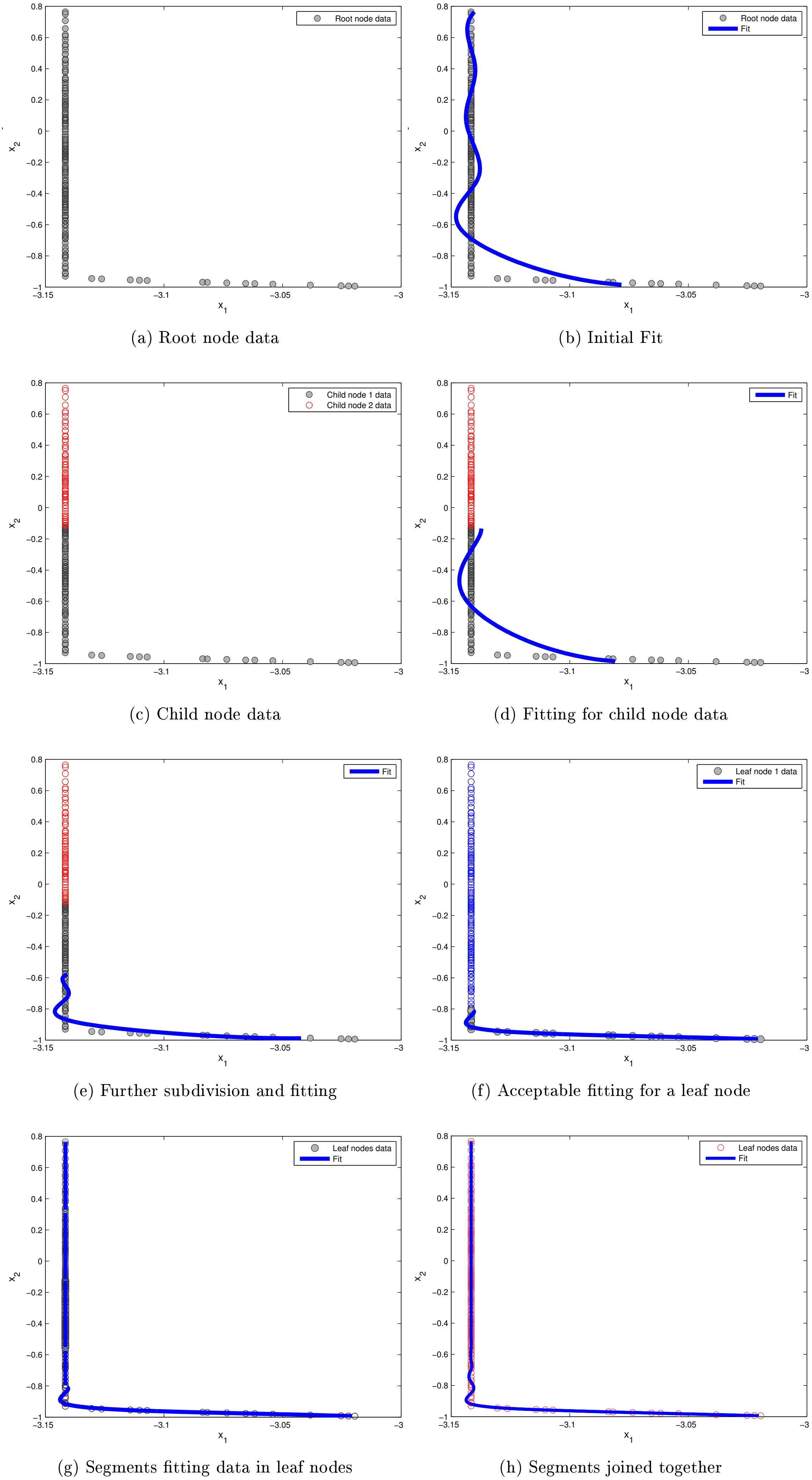
### 5.2.3 Iterative hyposurface fitting

The steps for finding the control points and parameter updation are iteratively repeated to reduce the error, as described earlier. The problem of finding the control points for fixed parameterization is a linear least square problem with linear constraints and is solved using SQP routine of MATLAB. The linear constraints represent the bounds on the control points. These bounds are decided depending upon the range of input data and ensure that the hyposurface is restricted in space. In the next step, the control points found in the previous step are fixed and the non-linear problem in unknown parameters is solved to find the updated parameter values. The iterations of these two steps end if the error value becomes small or the maximum number of iterations are reached. If the error value is small, the control points, corresponding parameterization, and the number of control points for the current node are stored, otherwise data members of the current node are subdivided into $2^M$ nodes.

### 5.2.4 Joining Hyposurface segments

The k-way tree creation process results in discontinuous segments modeled using B-spline basis functions stored in the leaf nodes of the tree. Next step is to join together these segments to create a $C^2$ continuous hyposurface. The two step, iterative hyposurface fitting method, described above, is used for modeling the whole hyposurface in one go. To model a hyposurface using this iterative procedure, a good initial parameterization and total number of control points necessary for modeling are required. A parameterization as well as the number of control points required for modeling every segment is stored in the leaf nodes of the k-way tree. The sum of these control points serves as the number of control points required to model the complete hyposurface. Scaling the parameterization stored in each node in the required range, based on the position of node in the tree an initial parameterization for the data is worked out. The iterative procedure for the error reduction is now used to find the control points, which model the whole hyposurface. The iterations are carried till the threshold error is not satisfied. At the end of this process we get a $C^2$ continuous hyposurface whose parametric equation is known to us.

Figure 5.3 shows the steps of the algorithm for the data points in the first cluster for the 'POL' problem. Figure 5.3a shows the data in the root node. No outliers were found in the data. Figure 5.3b shows the initial fitting. As the error value for this fit is greater than the threshold error value, the data is divided into two nodes. Figure 5.3c shows the subdivided data in the children nodes. Figure 5.3d shows the data fitting for child node 1. This fitting is only slightly better than the previous fit and does not satisfy the threshold error and so the node is further divided into children nodes. As a result, after several subdivisions, a binary tree is created with the level of a leaf node indicating the intricacy of the features captured by it. Subsequent Figures show the progress as the data is subdivided till an acceptable fit is found in Figure 5.3f. Figure 5.3g shows the discontinuous segments after the tree creation is complete. Finally the segments are joined together to give a $C^2$ continuous curve as shown in Figure 5.3h.



(a) Root node data

(b) Initial Fit

(c) Child node data

(d) Fitting for child node data

(e) Further subdivision and fitting

(f) Acceptable fitting for a leaf node

(g) Segments fitting data in leaf nodes

(h) Segments joined together

Figure 5.3: Fitting Algorithm

Chapter 6

# Results and discussion

To show the efficacy of the model, the algorithm is tested on several bi-objective and tri-objective problems. A problem set that covers unconstrained and constrained standard optimization problems as well as two engineering design problems is considered. Following standard MOOPs are used: FON, ZDT2, ZDT3, POL, OSY, DTLZ1, DTLZ2 and DTLZ7 [1]. Two engineering design problems, a two bar truss design problem and a cutting parameter selection problem for machining, are considered. For all of these problems both Pareto-optimal front and Pareto-optimal set are $(M-1)$-dimensional hyposurface. Some special kind of ill-conditioned problems for which this is not true are not considered in this study. A class of ill-conditioned problems is discussed by Zhou et al. [86].

## 6.1 Parameter values

The input points are returned by NSGA-II. NSGA-II is run with population size of 400 for 400 generations for the bi-objective problems and the results are passed to the algorithm as the input data. For tri-objective problems population size of thousand is run for 800 generations. Threshold value of the KKT-error is fixed at $10^{-4}$. If the KKT-error is greater than the threshold, a local search is conducted using SQP routine of MATLAB. Maximum function evaluations for SQP are fixed to thousand. For the subtractive clustering process, radii of influence is fixed to 0.15. For merging clusters, the threshold distance between the two clusters is dependent on the length of body diagonal of the hypercube enclosing the data. The number of the control points that model the hyposurface is fixed as one-tenth of the number of data members stored in the node.

The equation of the B-spline curve and surface is given in the form of a matrix equation. For a cubic B-spline curve modeled by $N_{cp}$ number of control points, there are $N_{cp}-3$ number of cubic segments. These segments are represented as

$$\mathbf{x}_i(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{6} & \frac{3}{6} & \frac{-3}{6} & \frac{1}{6} \\ \frac{3}{6} & \frac{6}{6} & \frac{3}{6} & \frac{0}{6} \\ \frac{-3}{6} & \frac{0}{6} & \frac{3}{6} & \frac{0}{6} \\ \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & \frac{0}{6} \end{bmatrix} \begin{bmatrix} \mathbf{p_i} \\ \mathbf{p_{i+1}} \\ \mathbf{p_{i+2}} \\ \mathbf{p_{i+3}} \end{bmatrix}, \quad \forall i \in [1 : N_{cp}-3].$$

Here $\mathbf{p_i}$ is a row vector that represents the co-ordinates of the $i^{th}$ control point in space. For equation in this form, for each segment the parameter $u \in [0,1]$. To show the final result the whole curve can be reparameterized to $u \in [0,1]$, so that each point of the curve is identified by a unique parameter value. Similar representation for Surface can also be written. The control points for each test problem are shown in an appendix A at the end.

## 6.2 Standard Multi-objective Optimization Problems

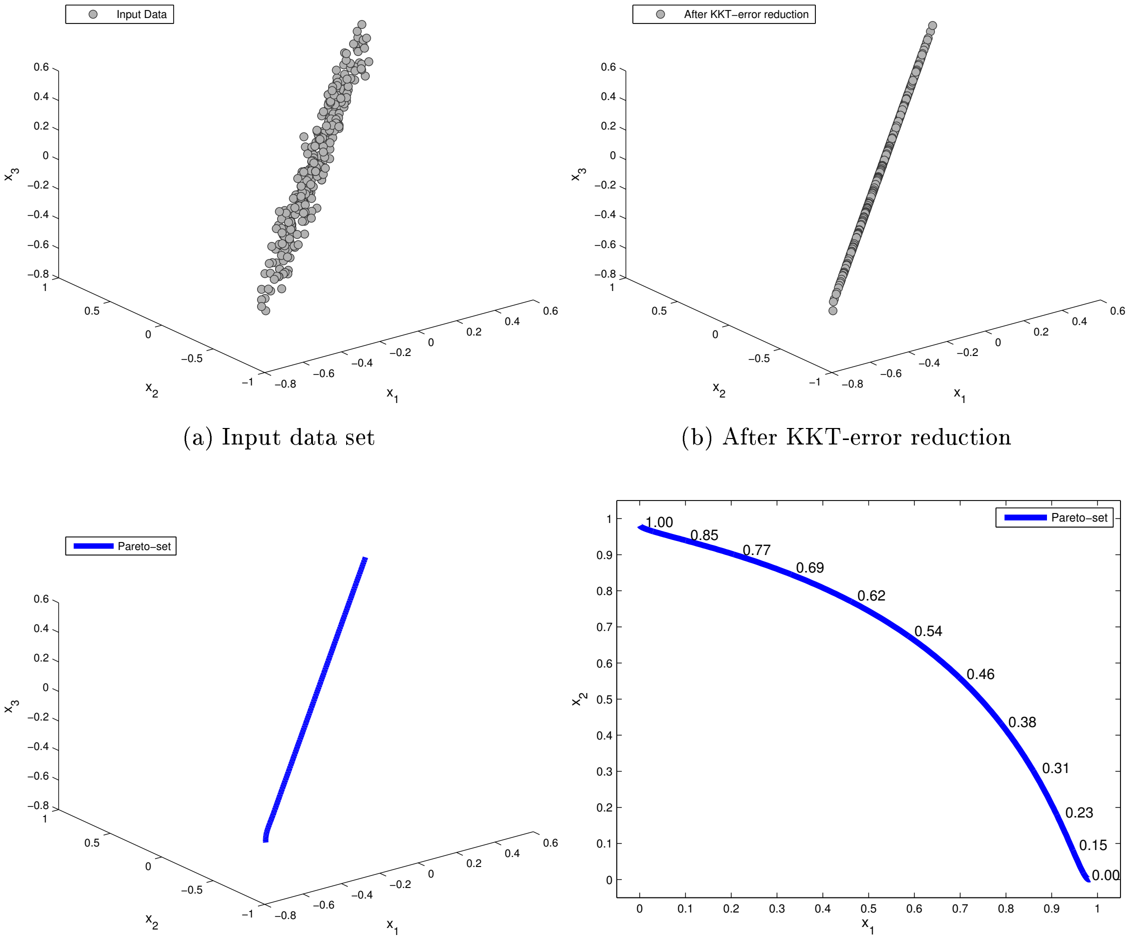
In this section results for standard optimization problems are provided. Some of the standard problems have scalable number of variables. For such problems, results with low as well as high number of variables are shown. The problems cover constrained as well as unconstrained problems. MOOPs with disconnected Pareto-optimal front are also considered.

### 6.2.1 FON problem

FON is a two objective problem with $n$-variables. We show the results for $n = 3$ for better visualization of solutions in the objective and variable space. The problem definition is given as following:

$$\begin{aligned} \text{Minimize} \quad & f_1(x) = exp(-\Sigma_{i=1}^{3}(x_i - 1/\sqrt{3})^2), \\ \text{Minimize} \quad & f_2(x) = exp(-\Sigma_{i=1}^{3}(x_i + 1/\sqrt{3})^2), \\ \text{Subject to} \quad & -4 \leq x_i \leq 4, \quad i = 1,2,3. \end{aligned} \tag{6.1}$$

Figure 6.1 shows the results for the FON problem. Figure 6.1a shows the input data set, which is updated to Figure 6.1b by the KKT-error reduction technique. The KKT-error reduction routine reduces the error in the data and now the data is better organized for curve fitting. Average KKT-error reduced from $9.4 \times 10^{-3}$ to $1.73 \times 10^{-7}$. Figures 6.1c and 6.1d show the Pareto-optimal set and the Pareto-optimal front in the variable and the objective space, respectively. The Pareto-optimal front is obtained by mapping the fitted curve to the objective space. 15 control points model the B-spline curve. Thus, there are 12 cubic segments modeling the Pareto-optimal set. Matrix of control points modeling the Pareto-optimal set and the parametric equation of the curve are given in appendix A. Parameter values of points in the Pareto-optimal set, reparameterize to $[0,1]$, are shown on their respective image in the objective space in Figure 6.1d. This makes moving from the objective space to the variable space easy. Thus, one can easily find variable values corresponding to any point on the Pareto-optimal front regardless of presence of that point in the GA population. Also, the parameter values allows the user to choose any point on the front. From the curve, it can be interpreted that there are linkages in the variables. This means that to move along the Pareto-optimal front one needs to increase all the variables simultaneously. From the parametric equation of the curve, it can be inferred that $x_1 = x_2 = x_3$ at any point in the Pareto-optimal set.



(a) Input data set

(b) After KKT-error reduction

(c) Pareto-optimal set

(d) Pareto-optimal Front with parameter values

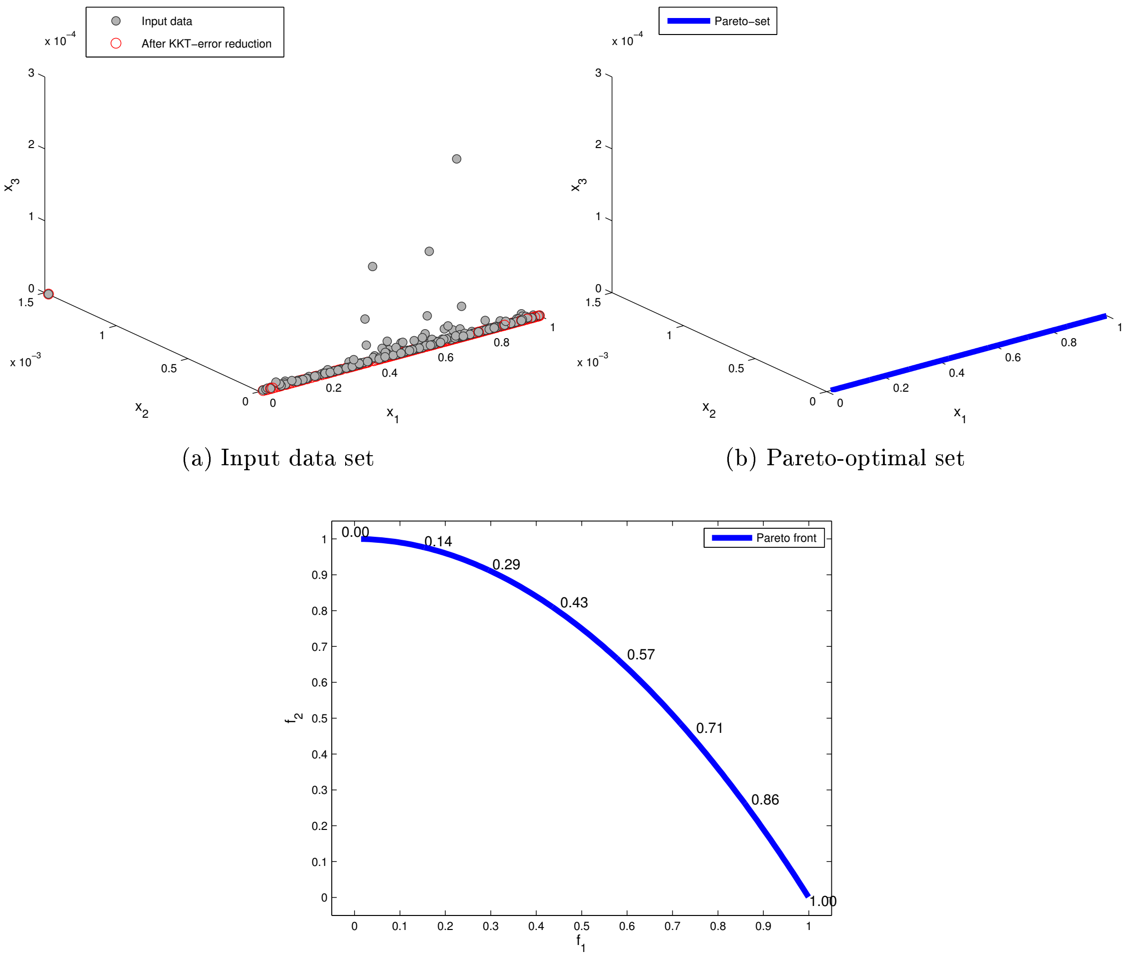
Figure 6.1: Results for 3 variable 'FON' problem

### 6.2.2 ZDT2

ZDT2 has a non-convex Pareto-optimal front. Definition of the $n$-variable ZDT2 problem is as follows:

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathbf{x}) = x_1, \\ \text{Minimize} \quad & f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})), \\ \text{Where} \quad & g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ & h(f_1, g) = 1 - (f_1/g)^2, \\ \text{Subject to} \quad & 0 \ \le x_i \ \le 1, \forall i \in [1, \ \ldots \ , n]. \end{aligned} \tag{6.2}$$

First the results of a three-variable ZDT2 problem are shown followed by those for a 30-variable problem. Figure 6.2a shows the initial input and the corrected data for the ZDT2 problem. The red points show the corrected data. We can see that the data after error correction is much more organized and lies on a straight line. Figure 6.2b shows the curve fitting for the Pareto-optimal set. All variables except $x_1$ are close to zero, while $x_1$ takes all values from 0 to 1. These values are same as those shown by Deb et al. [1]. Figure 6.2c shows the image of the Pareto-optimal set in the objective space. Figure 6.3 shows the Pareto-optimal front for a 30-variable ZDT2 problem. The Pareto-optimal set for a 30-variable ZDT2 problem can not be shown on a plot due to the high dimensions of the variable space. However, we can see that the Pareto-optimal front of a 30-variable ZDT2 problem is similar to its three-variable counterpart. Control points and parametric equation of the modeled Pareto-optimal set are shown in Appendix A.



(a) Input data set

(b) Pareto-optimal set

(c) Pareto-optimal Front with parameter values

Figure 6.2: Results for three-variable ‘ZDT2’ problem

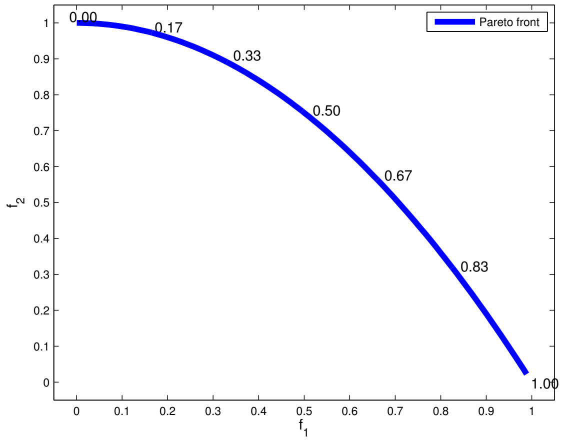


Figure 6.3: Pareto-optimal front for 30-variable ‘ZDT2’ problem

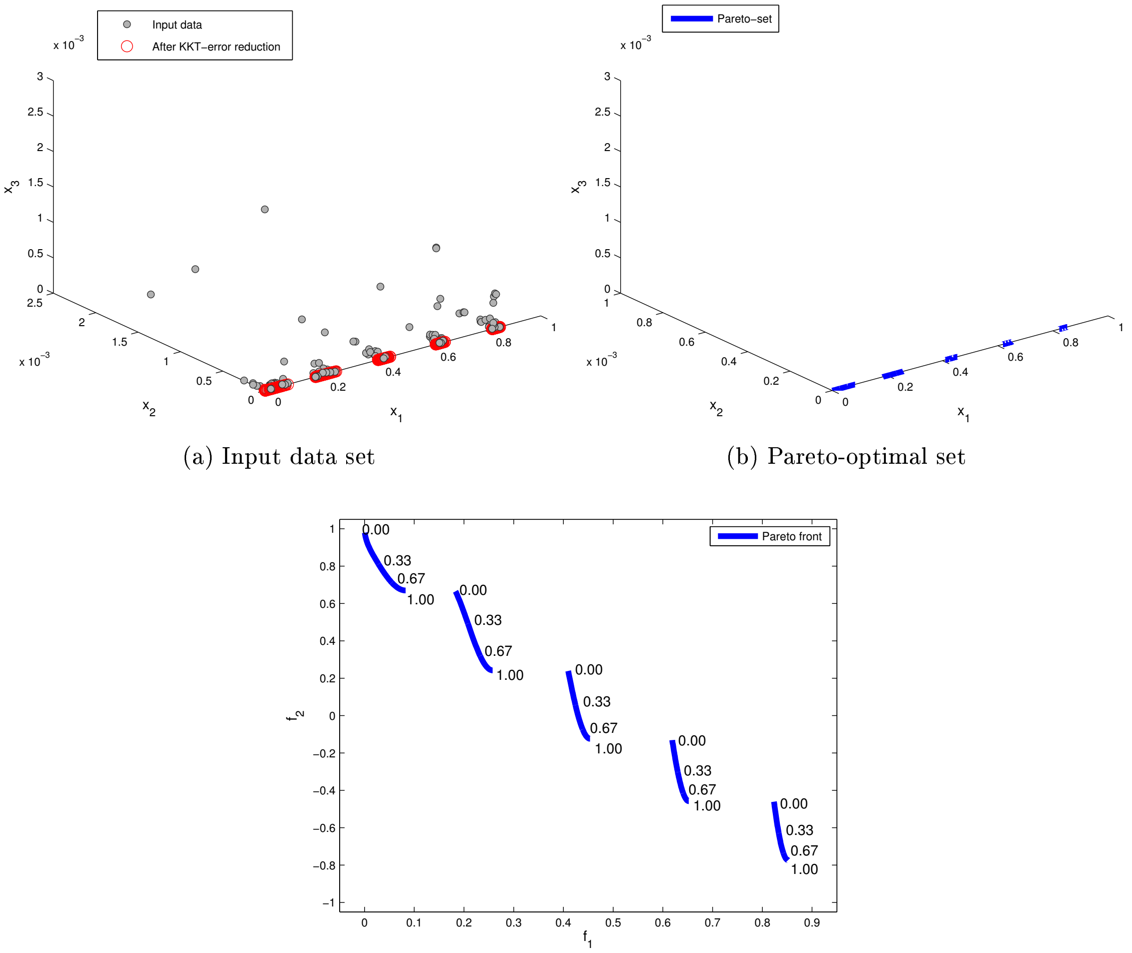
### 6.2.3 ZDT3

Definition of the $n$-variable ZDT3 problem is as follows:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\mathbf{x}) = x_1, \\
\text{Minimize} \quad & f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})), \\
\text{Where} \quad & g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\
& h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g)sin(10\pi f1), \\
\text{Subject to} \quad & 0 \ \leq x_i \ \leq 1, \forall i \in [1, \ \ldots \ , n].
\end{aligned} \tag{6.3}$$

ZDT3 has a number of disconnected Pareto-optimal fronts. First the results for a three-variable ZDT3 problem are shown followed by those for a 30-variable problem. Figure 6.4a shows the initial input and the corrected data for ZDT3 problem. Several points with high error values have been corrected and the corrected data is better organized for all the five disconnected Pareto-optimal sets. Average KKT-error reduced from $7.18 \times 10^{-4}$ to $5.24 \times 10^{-8}$. The clustering routine identified five clusters in the corrected data. Figure 6.4b shows the curve fitting for the Pareto-optimal set. Curve fitting is carried out, independently, for each of the clusters, as a result we get five different parameterizations. Figure 6.4c shows the image of the Pareto-optimal set in the objective space. From Figure 6.4b it can be inferred that all the points in the Pareto-optimal set have all variables, except $x_1$, close to zero. $x_1$ does not take all values in $[0, 1]$, it consists of five disconnected parts which leads to a disconnected Pareto-optimal front. The exact values of control points and the parametric equation of the modeled Pareto-optimal set for each cluster, refer to appendix A.

We can not show the Pareto-optimal set for a 30-variable ZDT3 problem. The Pareto-optimal front with parameter values for the 30-variable problem is shown in Figure 6.5.



(a) Input data set

(b) Pareto-optimal set

(c) Pareto-optimal Front with parameter values

Figure 6.4: Results for three-variable 'ZDT3' problem

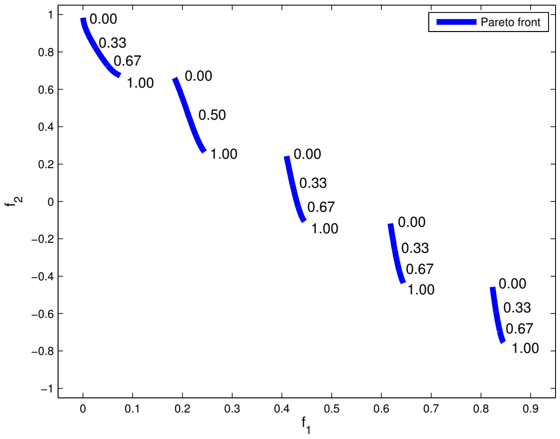


Figure 6.5: Pareto-optimal front for 30-variable 'ZDT3' problem

### 6.2.4 POL

'POL' problem has a non-convex and disconnected Pareto-optimal front. The problem is defined as following:

$$
\begin{aligned}
\text{Minimize} \quad & f_1(\mathbf{x}) = [1 + (A_1 - B_1)^2 + (A_2 - B_2)^2], \qquad (6.4)\\
\text{Minimize} \quad & f_2(\mathbf{x}) = [(x_1 + 3)^2 + (x_2 + 1)^2],\\
\text{Subject to} \quad & -\pi \ \leq x_1, x_2 \ \leq \pi,\\
\text{Where} \quad & A_1 = 0.5sin1 - 2cos1 + sin2 - 1.5cos2,\\
& A_2 = 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\cos 2,\\
& B_1 = 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\cos x_2,\\
& B_2 = 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\cos x_2,
\end{aligned}
$$

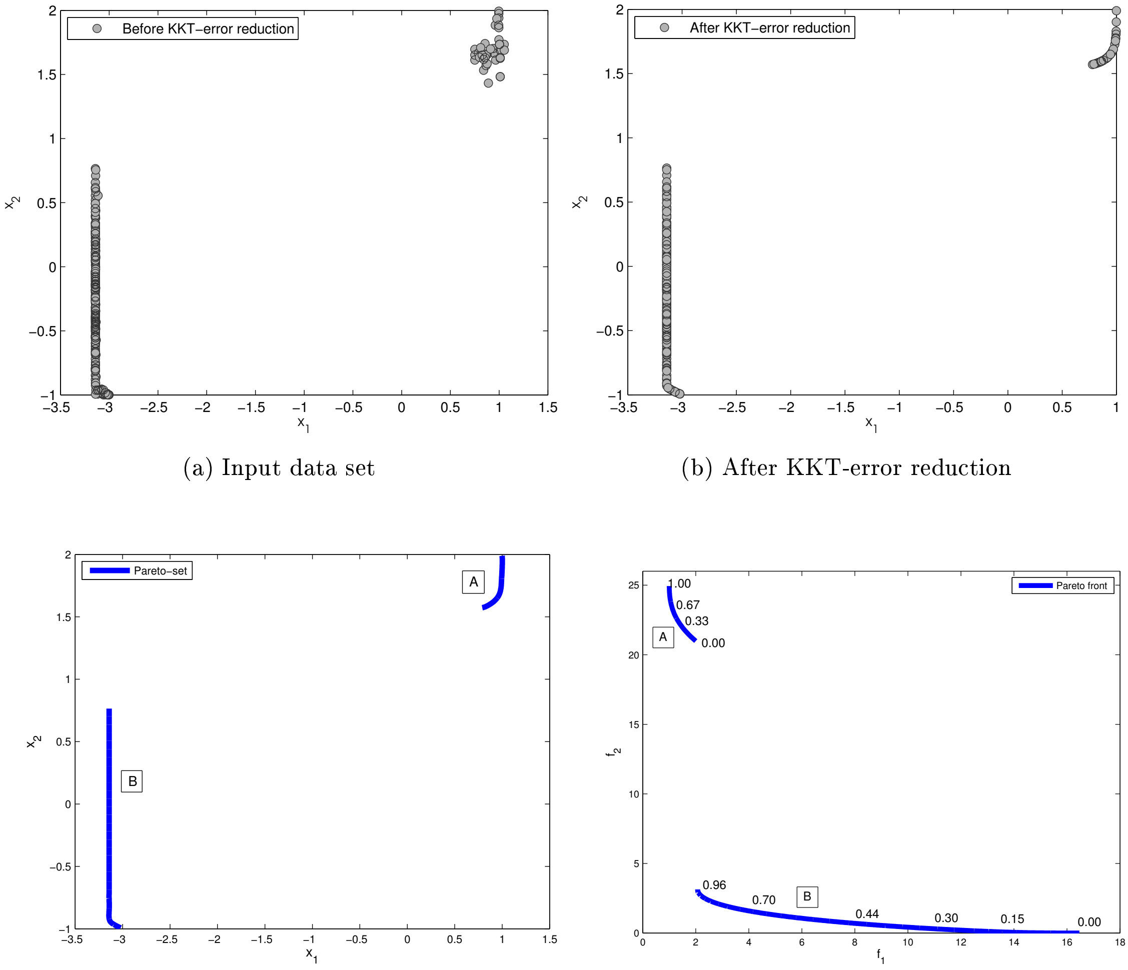
Results for this two-variable problem are shown in Figure 6.6. The KKT-error reduction routine reduces the error in the input points and provide better structure to the input data. The KKT-error reduces from $1.704\times10^{-2}$ to $3.768\times10^{-6}$. The clustering procedure detects two clusters as both Pareto-optimal set and Pareto-optimal front are disconnected. The Pareto-optimal set has two parts A and B, their mapping in the objective space along with the parameter values is shown in Figure 6.6d as A and B. Exact values of control points for both parts are shown in appendix A.

Variation of variables $x_1$ and $x_2$ with parameter u is shown in Figure 6.7. For part A of the front $x_1$ value increases initially and then becomes constant at variable value of 1. Variable $x_2$ however rises continuously from value of 1.57 to 2. For the part B of the front, $x_1$ remains constant for most part of the front at a variable value of $-3.14$, indicating that front occurs at the lower bound for $x_1$. Variable $x_2$ rises from $-1$ to 0.71. We observe that for both parts of the front, $x_2$ is greater than $x_1$. Interesting properties of the solutions revealed from the above analysis are enumerated:

1. Part B of the front occurs at the lower bound of variable $x_1$
2. Variable $x_2$ is greater than variable $x_1$ for the whole front
3. Variable $x_2$ always increases as we move along the front.

### 6.2.5 DTLZ1

DTLZ test problems are scalable test problems [87]. Results for a three-objective and three-variable DTLZ1 problem with linear Pareto-optimal front are presented here. As the number of objectives are three, the Pareto-optimal front and Pareto-optimal set are a 2-d



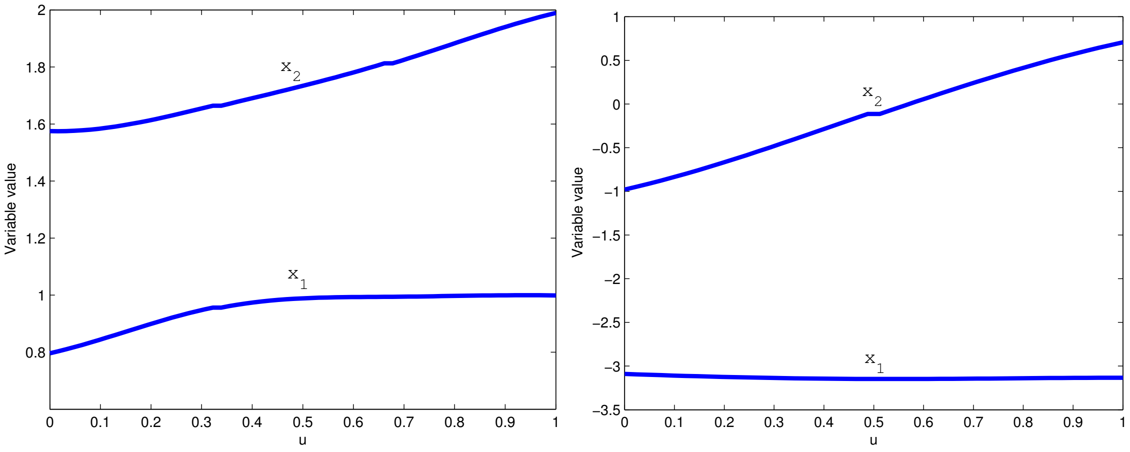
(a) Input data set

(b) After KKT-error reduction

(c) Pareto-optimal Set

(d) Pareto-optimal Front with parameter values

Figure 6.6: Results for two-variable 'POL' problem

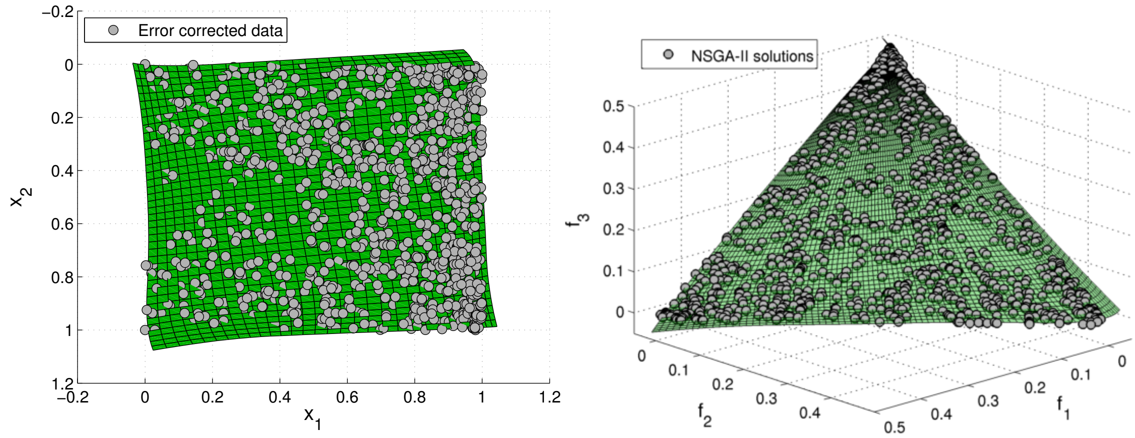


(a) Variation of variables with parameter for A (b) Variation of variables with parameter for B

Figure 6.7: Variation of variables with parameter for 'POL' problem

hyposurface i.e. a surface. A three-objective, $n$-variable DTLZ1 problem is stated below.

$$\begin{aligned}
\text{Minimize} \quad & f_1(\mathbf{x}) = 0.5\, x_1 x_2\, (1 + g(x_3, \ldots, x_n)), \qquad (6.5)\\
\text{Minimize} \quad & f_2(\mathbf{x}) = 0.5\, x_1\, (1 - x_2)\, (1 + g(x_3, \ldots, x_n)),\\
\text{Minimize} \quad & f_3(\mathbf{x}) = 0.5\, (1 - x_1)\, (1 + g(x_3, \ldots, x_n)),\\
\text{Subject to} \quad & \mathbf{0} \ \leq \mathbf{x} \ \leq \mathbf{1},\\
\text{Where} \quad & g(x_3, \ldots, x_n) = 100\, (n - 2 + \sum_{i=3}^{n} (x_i - 0.5)^2 \ - \ cos(20\pi(x_i - 0.5))).
\end{aligned}$$



(a) Pareto-optimal Set (b) Pareto-optimal Front

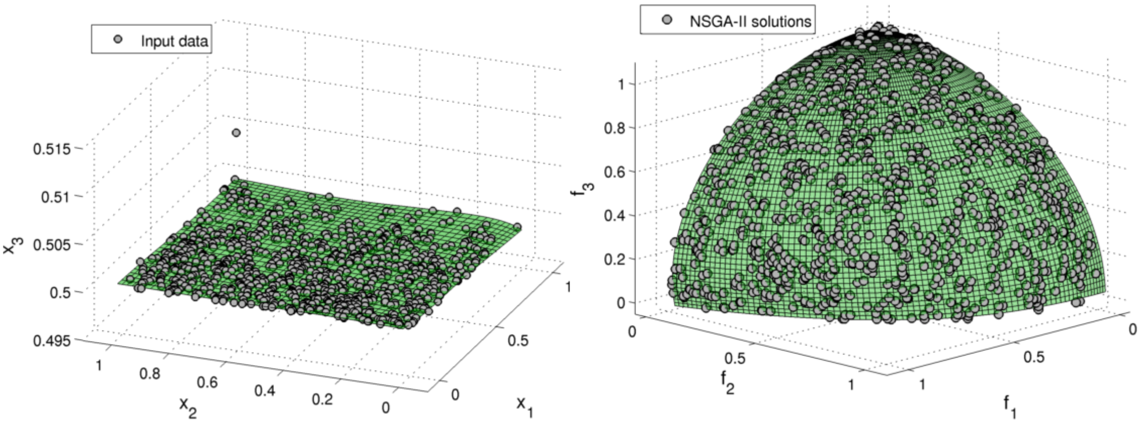
Figure 6.8: Results for 3-variable 'DTLZ1' problem

The points in the Pareto-optimal set lie on a plane. Their mapping to the objective space is also a plane. Figure 6.8 show the results for DTLZ1 problem. The average KKT-error for the input points reduced from $2.67\times10^{-1}$ to $6.7\times10^{-4}$ after the error reduction routine. Figure 6.8a shows the fitted surface in the variable space. Exact values of the control points are shown in appendix A. Pareto-optimal set is a plane in the variable space with variable $x_3 = 0.5$ throughout the front. The fitted surface covers those Pareto regions where no input data was found, thus points not found by MOOA can be found by using the parametric equation of the surface. At the same time, there are parts of the modeled surface which may not be necessarily Pareto-optimal. The mapping of Pareto-optimal set in the decision space is shown in Figure 6.8b.

### 6.2.6 DTLZ2

Problem definition of an $n$-variable three-objective DTLZ2 problem is as follows:

$$\begin{aligned}
\text{Minimize}\quad & f_1(\mathbf{x}) = (1+g(x_3,\ldots,x_n)cos(0.5\,\pi x_1)\,cos(0.5\,\pi x_2), \\
\text{Minimize}\quad & f_2(\mathbf{x}) = (1+g(x_3,\ldots,x_n)cos(0.5\,\pi x_1)\,sin(0.5\,\pi x_2), \\
\text{Minimize}\quad & f_3(\mathbf{x}) = (1+g(x_3,\ldots,x_n)cos(0.5\,\pi x_1), \\
\text{Subject to}\quad & \mathbf{0}\ \le \mathbf{x}\ \le \mathbf{1}, \\
\text{Where}\quad & g(x_3,\,\ldots,\,x_n) = \sum_{i=3}^{n}(x_i-0.5)^2,
\end{aligned} \tag{6.6}$$



(a) Pareto-optimal Set (b) Pareto-optimal Front

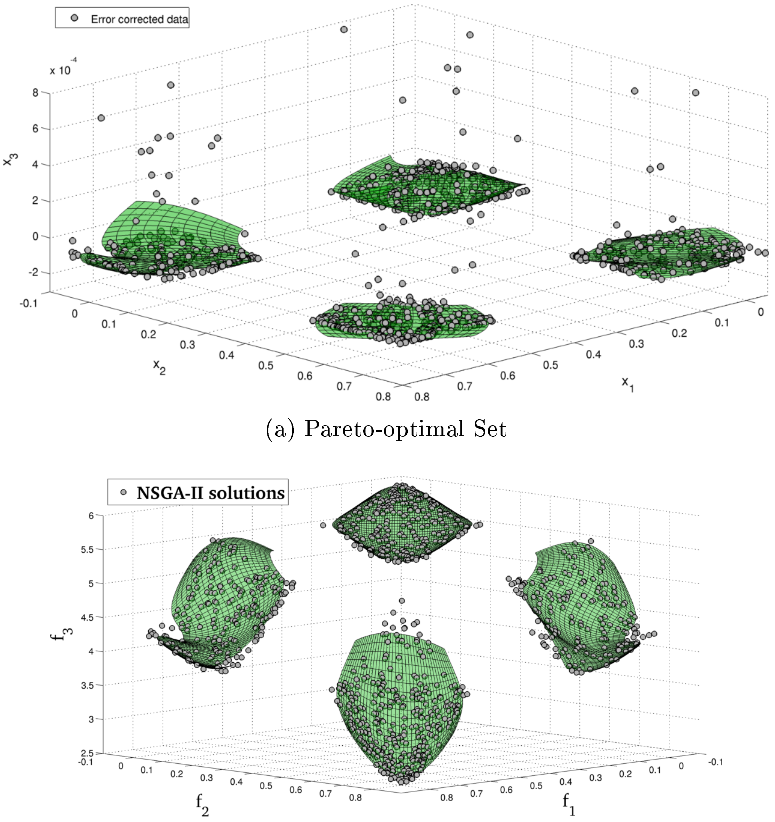
Figure 6.9: Results for 3-variable 'DTLZ2' problem

Pareto-optimal set is a plane with $x_3 = 0.5$ and Pareto-optimal front is part of a sphere with $f_1^2 + f_2^2 + f_3^2 = 1$. Results for the problem are shown in 6.9. KKT-error reduction routine reduces the error from .0011 to $8\times10^{-10}$. Figures 6.9a and 6.9b show the Pareto-optimal set and Pareto-optimal front for the problem, respectively. Exact values of control points can be found in appendix A.

### 6.2.7 DTLZ7

DTLZ7 has a disconnected set of Pareto-optimal front and Pareto-optimal set. There are $2^{M-1}$ disconnected Pareto regions in this problem. Problem definition for a three-objective, $n$-variable DTLZ7 problem is stated below.

$$\begin{aligned}
\text{Minimize} \quad & f_1(\mathbf{x}) = x_1, \\
\text{Minimize} \quad & f_2(\mathbf{x}) = x_2, \\
\text{Minimize} \quad & f_3(\mathbf{x}) = (1 + g(x_3, \ldots, x_n)\, h(f_1,\, f_2,\, g), \\
\text{Subject to} \quad & \mathbf{0} \ \leq \mathbf{x} \ \leq \mathbf{1}, \\
\text{Where} \quad & g(x_3,\, \ldots,\, x_n) \, = \, 1 + \frac{9}{n - M + 1} \sum_{i=3}^{n} x_i, \\
& h(f_1,\, f_2,\, g) \, = \, M \, - \sum_{i=1}^{M-1} (\frac{f_i}{1+g}(1 + sin(3\,\pi\, f_i)).
\end{aligned} \tag{6.7}$$



(a) Pareto-optimal Set

(b) Pareto-optimal Front

Figure 6.10: Results for 3-variable 'DTLZ7' problem

Figure 6.10 shows the results for a three-objective and three-variable DTLZ7 problem. There are four disconnected Pareto regions here which have been identified by the clustering procedure. The average KKT-error for the input points reduced from $8.1 \times 10^{-3}$ to $1.7 \times 10^{-6}$ after the error reduction routine. The fitted Pareto-optimal set and the Pareto-optimal front along with the input points are shown in Figures 6.10a and 6.10b, respectively. It can be noticed that some of the points in the Pareto-optimal set don't lie exactly on the front. Due to this the surface fitting is not exact and generated surface is not planar. Exact values of control points are given in appendix A

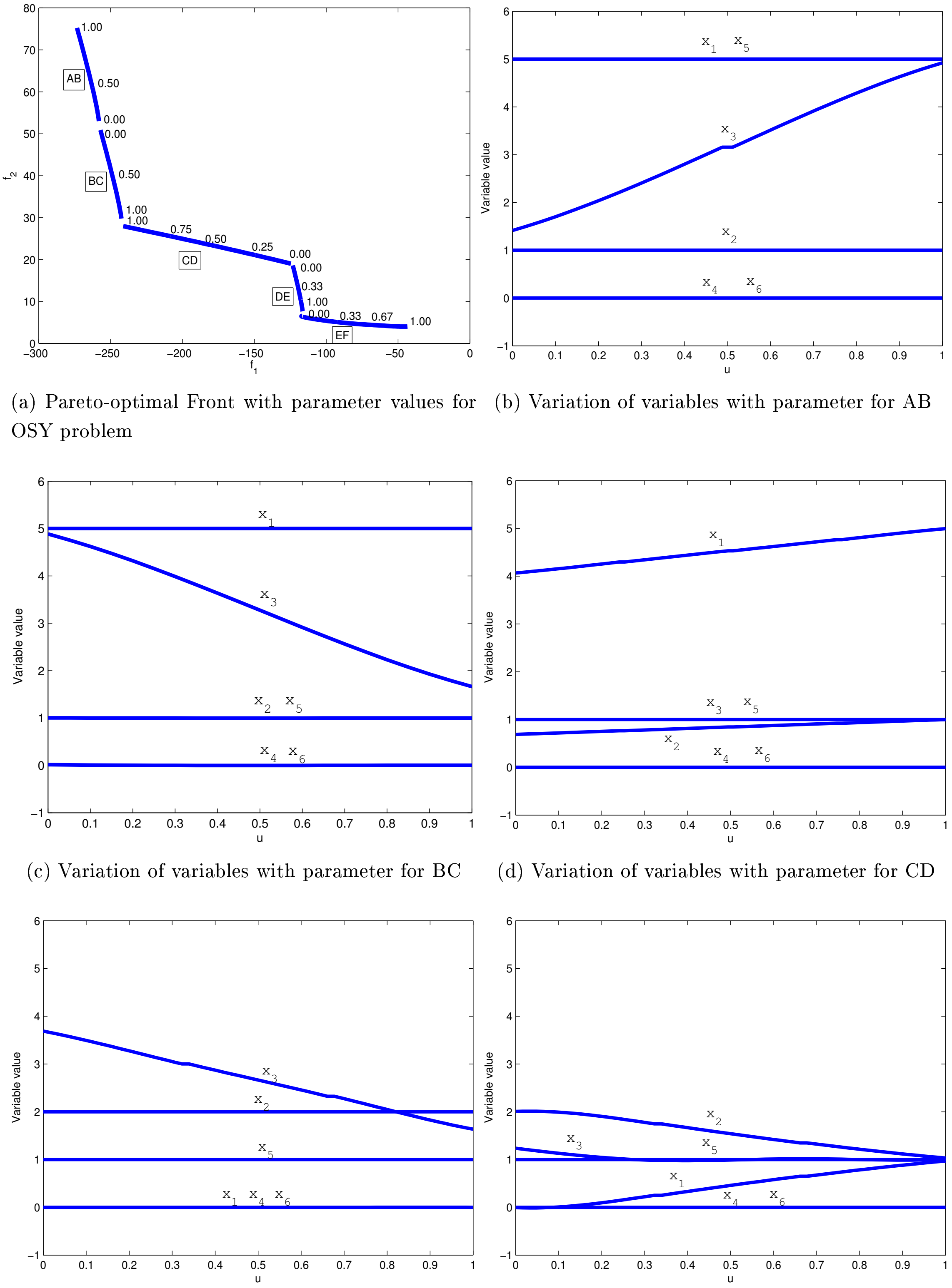
### 6.2.8 OSY

OSY is a six variable problem. The Pareto-optimal front consists of five parts which are continuously concatenated. However, in the variable space these five parts are discontinuous. The problem is defined as following:

$$\begin{aligned}
\text{Minimize}\quad & f_1(\mathbf{x}) = -[25(x_1-2)^2+(x_2-2)^2+(x_3-1)^2+(x_4-4)^2+(x_5-1)^2], \qquad (6.8)\\
\text{Minimize}\quad & f_2(\mathbf{x}) = x_1^2+x_2^2+x_3^2+x_4^2+x_5^2+x_6^2,\\
\text{Subject to}\quad & C_1(\mathbf{x}) \equiv x_1+x_2-2 \geq 0,\\
& C_2(\mathbf{x}) \equiv 6-x_1-x_2 \geq 0,\\
& C_3(\mathbf{x}) \equiv 2-x_2+x_1 \geq 0,\\
& C_4(\mathbf{x}) \equiv 2-x_1+3x_2 \geq 0,\\
& C_5(\mathbf{x}) \equiv 4-(x_3-3)^2-x_4 \geq 0,\\
& C_6(\mathbf{x}) \equiv (x_5-3)^2+x_6-4 \geq 0,\\
& 0 \ \leq x_1, x_2, x_6 \ \leq 10,\\
& 1 \ \leq x_3, x_5 \ \leq 5,\\
& 0 \ \leq x_4 \ \leq 6.
\end{aligned}$$

Figure 6.11a show the Pareto-optimal front for the OSY problem. 'AB', 'BC', 'CD', 'DE' and 'EF' show the five parts of the front. The data is divided into five parts by the clustering procedure indicating that the Pareto-optimal set is discontinuous. For each of the five parts, the B-spline curve fitting process gives several cubic splines joined with $C^2$ continuity. Exact values of control points for each part are given in appendix A.

To show the variation of variables with parameter u, graphs between variable values and parameter values are plotted for each of the 5 parts of the Pareto-optimal front. Figure 6.11 shows these plots. For AB, as we move from parameter value 0 to 1, $x_1$, $x_2$, $x_4$, $x_5$ and $x_6$ remain constant at 5, 1, 0, 5 and 0, respectively. Only $x_3$ changes along this part of the Pareto-optimal front and its value rises from 1.41 to 5. For BC also, $x_1$, $x_2$, $x_4$, $x_5$ and $x_6$ remain constant at 5, 1, 0, 1 and 0, respectively. Here also, only $x_3$ changes from 4.88 to 1.66. For CD, $x_3$ and $x_5$ are constant at variable value one while $x_4$ and $x_6$ take a value of zero. The rise in value of $x_1$ and $x_2$ is linear. A closer inspection of the expressions of



(a) Pareto-optimal Front with parameter values for OSY problem

(b) Variation of variables with parameter for AB

(c) Variation of variables with parameter for BC

(d) Variation of variables with parameter for CD

(e) Variation of variables with parameter for DE

(f) Variation of variables with parameter for EF

Figure 6.11: Variation of six variables with parameter for ‘OSY’ problem

$x_1(u)$ and $x_2(u)$ reveal that $x_2(u) = (x_1(u) - 2)/3$. Such an inspection was carried out because the relationship was known to us from a previous study by Deb et al. [1]. For DE, we notice that $x_1$, $x_2$, $x_4$, $x_5$ and $x_6$ are constant at 0, 2, 0, 4 and 0 respectively. Value of $x_3$ varies between 1.63 and 3.69. For the part EF of the front, from parameter values 0 to nearly 0.1, properties of the variables are similar to those for DE. It seems that the end of DE has been captured as EF. For the initial part of EF, $x_1$ varies from 0 to 0.01, $x_2$ varies from 2 to 1.99,$x_3$ varies from 1.24 to 1.14 and $x_4$, $x_5$ and $x_6$ are constant at 0,1 and 0 respectively. For the rest of the part of EF, $x_1$ rises with u, while $x_2$ declines with u. It appears that the plots of $x_1$ and $x_2$ are mirror image of each other about line $x = 1$. This makes $x_2 = 2 - x_1$, which was also found by Deb in [1]. $x_3$ is nearly constant and $x_5$ is constant at variable value of 1 while $x_4$ and $x_6$ have a constant value of 0.

We summarize the results in Table 6.1. These results are very close to those found by Deb in [1]. We can infer some interesting properties of optimal points by this procedure which can be useful to the decision maker. Some inferences are enumerated:

1. $x_4 = x_6 = 0$ along the front.
2. $x_5$ is constant at 5 for part AB and at 1 for the rest of the front.
3. For AB, BC and DE the change in the objective function value is caused by the change in the value of $x_3$ alone.
4. $x_3$ is non-decreasing throughout the Pareto-optimal front .
5. For part EF, $x_2 = 2 - x_1$.

| Front part | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| AB | 5 | 1 | (1.41,...,5) | 0 | 5 | 0 |
| BC | 5 | 1 | (1.66,...,4.88) | 0 | 1 | 0 |
| CD | (4.06, ...,5) | (0.69, ...,1) | 1 | 0 | 1 | 0 |
| DE | 0 | 2 | (1.63,...,3.69) | 0 | 1 | 0 |
| EF(u∈ [0, .1]) | (0,...,.01) | (2.00,...,1.99) | (1.24,...,1.14) | 0 | 1 | 0 |
| EF(u∈ (.1, 1]) | (.01,...,.97) | (1.99,...,1.03) | (1.14,...,1) | 0 | 1 | 0 |

Table 6.1: Relationships among variables inferred from the plots

## 6.3 Engineering Design problems

### 6.3.1 Two bar truss design problem

In a two bar truss design problem the goal is to minimize the volume of the structure while allowing the structure to carry a minimum load without elastic failure. Figure 6.12

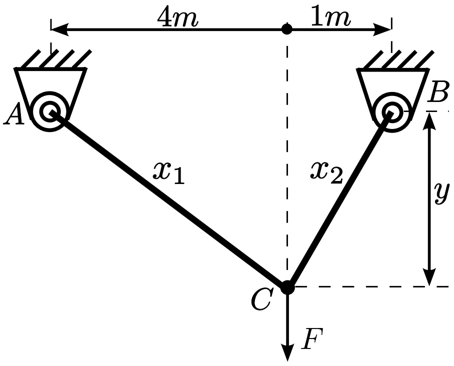


Figure 6.12: A two bar truss.

shows the diagram of a two bar truss. The stress in the two trusses, AC and BC, has to be minimized. A load $F = 10^5\,N$ acts at the junction C. The vertical distance between B and C is variable $y$ whereas the horizontal distance between the hinges is 5m. $x_1$ and $x_2$ are the cross-sectional areas of AC and BC respectively and are measured in $m^2$. The yield stress $(S_y)$ is $10^8\,Pa$. The three variable problem is defined as follows:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\mathbf{x}, y) = x_1\sqrt{16+y^2} + x_2\sqrt{(1+y^2)}, \qquad (6.9)\\
\text{Minimize} \quad & f_2(\mathbf{x}, y) = \max(\sigma_{AC}, \sigma_{BC}),\\
\text{Subject to} \quad & \max(\sigma_{AC}, \sigma_{BC}) \;\le\; 10^8,\\
& 1 \;\le y \;\le 3,\\
& \mathbf{x} \ge \mathbf{0},\\
\text{where} \quad & \sigma_{AC} = \frac{F\sqrt{16+y^2}}{5yx_1}, \qquad \sigma_{BC} = \frac{4F\sqrt{16+y^2}}{5yx_2}.
\end{aligned}$$

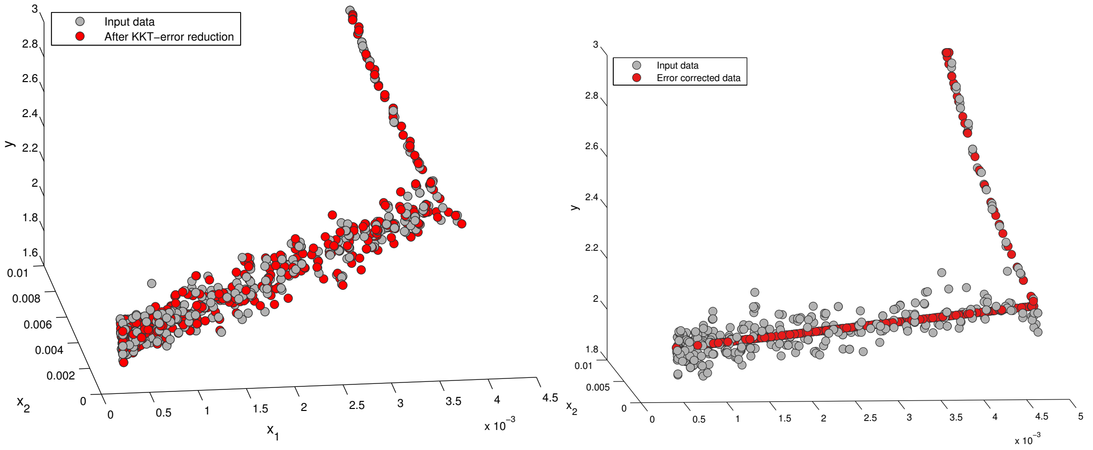
The objective $f_2$ and the constraint on the maximum stress use a 'max' function which makes them non-differentiable. This prevents an effective local search and the data points may not have a good structure to fit a curve through them. To bypass this problem, an alternate formulation is considered. The constraint can be split into two different constraints, $\sigma_{AC} \;\le\; 10^8$ and $\sigma_{BC} \;\le\; 10^8$. These constraints demand that stress in both the rods be less than the maximum permissible value of stress, which also implies that the maximum of the stress in the two rods is less than the maximum limit, thus satisfying the original constraint.

To circumvent the problem of non-differentiability of $f_2$, an alternate smooth formulation of local search procedure can be considered. The following optimization problem is solved for local search starting from a reference point $\mathbf{z}$ returned by NSGA-II:

$$
\begin{aligned}
\text{Minimize} \quad & \varepsilon_1 + \varepsilon_2, && (6.10)\\
\text{Subject to} \quad & w_1(f_1(\mathbf{x}, y) - z_1) \;\leq\; \varepsilon_1, \\
& w_2(\varepsilon_2 - z_2) \;\leq\; \varepsilon_1, \\
& \sigma_{AC} \;\leq\; \varepsilon_2, \\
& \sigma_{BC} \;\leq\; \varepsilon_2, \\
& 1 \;\leq y \;\leq\; 3, \\
& \mathbf{x} \;\geq\; \mathbf{0}, \\
& \varepsilon_1 \;\leq\; 0
\end{aligned}
$$

$$(6.11)$$

This formulation helps us to remove the max function which caused the objective function to be non-smooth. The difference between the results of the original formulation for local search and the new formulation is shown in Figure 6.13. The original formulation doesn't lead to any considerable correction in the data while the new formulation leads to substantial reduction in the error.

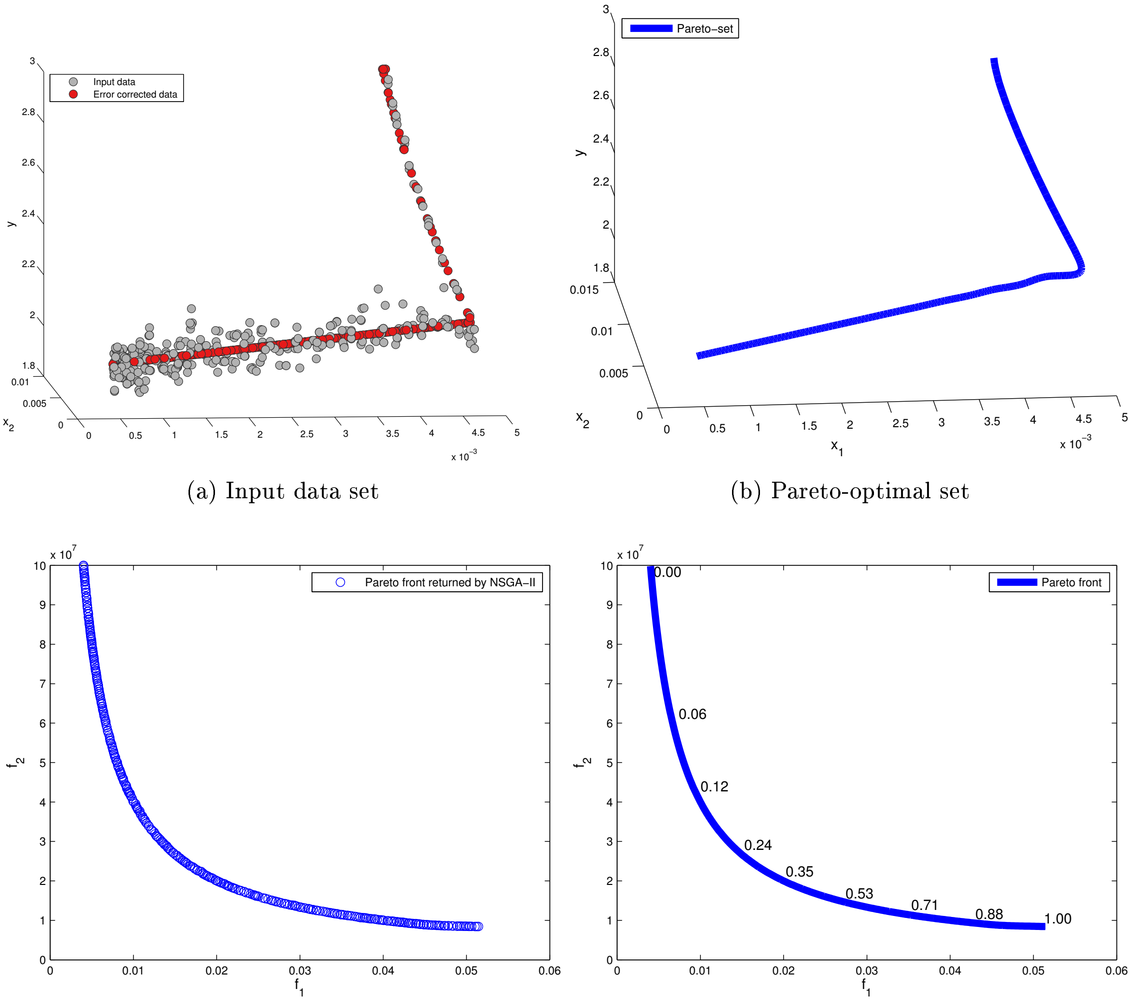


(a) Error reduction with original local search formulation

(b) Error reduction with new local search formulation

Figure 6.13: Error reduction results for two bar truss problem

In this problem, $f_1$ lies in $[0, 0.05]$ and $f_2$ lies in $[0, 10^8]$. There is a vast difference in the scales of both the objectives. Scaling in the variable space is also considerably different. This can cause problems in the the data fitting step. To circumvent this problem, the data is normalized to the interval $[0, 1]$. The data is rescaled back to show the results. Figure 6.14 shows the results for the above problem. Input data set and the data set after error correction are shown in Figure 6.14a. Figure 6.14b shows the Pareto-optimal set for this problem. The mapped Pareto-optimal front is shown in Figure 6.14d. It is very close to the non-dominated points shown in Figure 6.14c returned by the MOOA. Exact values of control points can be found in A.



(a) Input data set

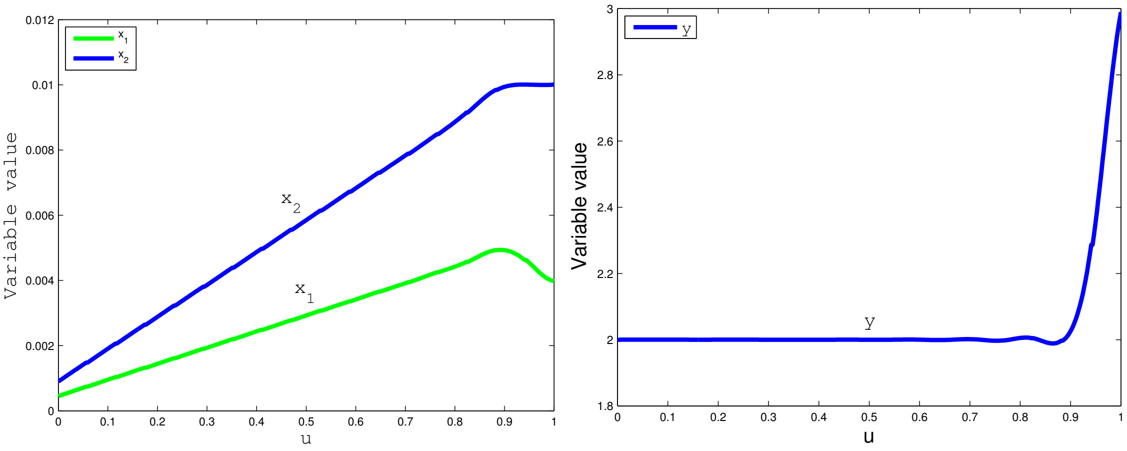
(b) Pareto-optimal set

(c) Non-dominated points returned by NSGA-II

(d) Pareto-optimal Front with parameter values

Figure 6.14: Results for 3-variable ‘truss’ problem

The variation of the variables with parameter values is shown in Figure 6.16. Notice that after parameter value of around 0.89, there is a sudden change in the behavior of all the variables. Moving from parameter value 0 to 0.89, it can be noticed that variable $y$ value does not change much, however the value of $x_1$ and $x_2$ rises steadily. It can be noticed that variation of $x_1$ and $x_2$ with parameter u is linear and a relation between the slopes of $x_1$ and $x_2$ can be found using the parametric equation of the curve. The value of cross-sectional area of BC ($x_2$) is always greater than cross-sectional area of AC ($x_1$) for all the optimal solutions. There is a little change in the value of $x_2$ for parameter values



(a) Variation of $x_1$ and $x_2$ with parameter

(b) Variation of $y$ with parameter value

Figure 6.15: Variation of 3 variables with parameter for 'truss' problem

greater than 0.89, $x_2$ reaches its maximum value of .01 and remains constant there. Thus, for variable $x_2$, the bound becomes active for $u > 0.89$. This may be the reason for sudden change in the behavior of variables at $u = 0.89$. Variable value, $x_1$ decreases slightly for $u > 0.89$. However, a drastic change is seen for the variable y in the same range. This means beyond the parameter value of 0.89, change in the optimal value of both functions is mostly caused by change in the value of y.

For further analysis, variation of constraint $g_1 \equiv 10^8 - \sigma_{AC}$ and $g_2 \equiv 10^8 - \sigma_{BC}$ with parameter $u$ is studied. Figure 6.16 shows the variation of $g_1$ and $g_2$ with parameter $u$. We observe that $g_2$ and $g_1$ have identical plots, implying that $\sigma_{AC} = \sigma_{BC}$ throughout the Pareto-optimal front.

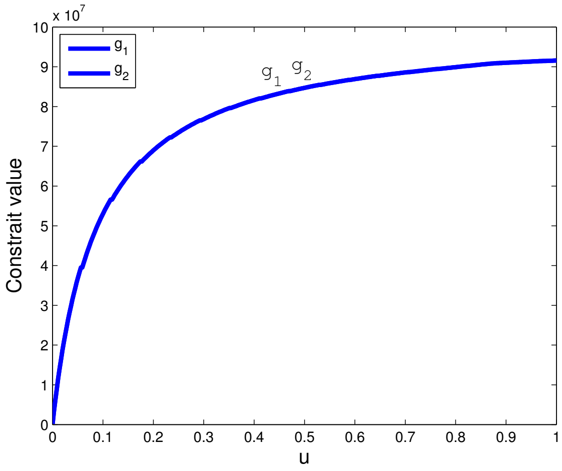


Figure 6.16: Variation of constraints $g_1$ and $g_2$ with parameter $u$ for 'truss' problem

Several interesting inferences which can be drawn from the plots are enumerated below. These observations are similar to observations made by Deb et al. in [30] but here we find them using a different approach. The relationships that exist between the variables as found by Deb et al. in [30] are $y = 2$ and $x_2 = 2x_1$.

1. $x_2$ is greater than $x_1$ throughout the Pareto-optimal front.
2. Relationship, $x_2 = 2\,x_1$, can be confirmed from the parametric equation
3. Variable $y$, remains constant at $y = 2$ for parameter $u < 0.89$ and rises rapidly for higher $u$ values
4. After the parameter value of around 0.89, $x_2$ reaches its upper bound and remains constant there.
5. $\sigma_{AC} = \sigma_{BC}$ throughout the front.
6. After the parameter value of around 0.89, the relative change in volume $(f_1)$ of the structure is much larger than change in maximum stress $(f_2)$. Also, the relative increase in $y$ value is larger than decrease in $x_2$ value. This implies that an increase in $y$ will cause much rapid rise in the optimal volume than the decrease it will cause in the maximum stress. Thus, part of the Pareto-optimal front beyond $u = 0.89$, may not be very useful to the decision maker.

### 6.3.2 Metal cutting problem

Metal cutting problem was introduced by Sardinas et al. in [88] and was discussed by Deb et al. in [89]. The problem is to minimize the production time ($T_p$) and used tool life ($\xi$) for a machining process. The variables are the cutting parameters: cutting speed ($v$), feed ($f$) and depth of cut ($a$). The problem has three constraints apart from the lower and upper bounds on all the variables. Bounds on the variable are supplied by the tool maker. The cutting force ($F_c$) should be less than a threshold ($F_{cmax}$) depending on the strength and stability of the machine and the tool. Maximum allowable cutting power ($P$) should be less than the machine motor power ($P_{MOT}$). Maximum surface roughness should be less than a specified maximum roughness ($R_{max}$). The problem is defined as follows:

$$\begin{aligned}
\text{Minimize} \quad & T_p(v,f,a), \\
\text{Minimize} \quad & \xi(v,f,a), \\
\text{Subject to} \quad & g_1(v,f,a) \equiv F_{cmax} - F_c(v,f,a) \geq 0, \\
& g_2(v,f,a) \equiv \eta\frac{P_{MOT}}{100} - P(v,f,a) \geq 0, \\
& g_3(v,f,a) \equiv R_{max} - R(v,f,a) \geq 0, \\
& v_{min} \leq v \leq v_{max}, \\
& f_{min} \leq f \leq f_{max}, \\
& a_{min} \leq a \leq a_{max}, \\
\text{where} \quad & T_p = \tau_s + \frac{V}{M(v,f,a)}(1 + \frac{\tau_{tc}}{T(v,f,a)}) + \tau_o, \\
& M(v,f,a) = 1000vfa, \\
& T(v,f,a) = C_T v^{\alpha} f^{\beta} a^{\gamma}, \\
& \xi(v,f,a) = \frac{100V}{M(v,f,a)T(v,f,a)}, \\
& F_c(v,f,a) = C_F v^{\alpha'} f^{\beta'} a^{\gamma'}, \\
& P(v,f,a) = \frac{vF_c(v,f,a)}{6\times 10^4}, \\
& R(v,f,a) = \frac{125f^2}{r_e}.
\end{aligned} \tag{6.12}$$

Parameters $\tau_s$, $\tau_{tc}$, $\tau_o$ and V are the setup time, tool change time, idle tool time and volume of the material removed, respectively. These are usually constant for a cutting process. $C_F$, $\alpha$, $\beta$, $\gamma$, $C_F$, $\alpha'$, $\beta'$ and $\gamma'$ are experimentally calculated constants. T(v,f,a) and M(v,f,a) are tool life and Material removal rate respectively and are functions of the cutting parameters. $\eta$ is the transmission efficiency of the motor and $r_e$ is the tool nose radius. Values of constants for a machining of a steel bar on a CNC machine using a P20 carbide tool as described in [88] are given in Table 6.2. Further details on the problem can be found in [88].

| Parameter | Value |
|---|---|
| $\tau_s$ | .15min |
| $\tau_{tc}$ | .20min |
| $\tau_o$ | .05min |
| V | 219,912$m^3$ |
| $C_T$ | $5.48 \times 10^9$ |
| $\alpha$ | -3.46 |
| $\beta$ | -.696 |
| $\gamma$ | -.460 |
| $C_F$ | $6.56 \times 10^3$ |
| $\alpha'$ | -.286 |
| $\beta'$ | .917 |
| $\gamma'$ | 1.10 |
| $a_{min},\ a_{max}$ | .6, 6mm |
| $f_{min},\ f_{max}$ | .15, .55mm/rev |
| $v_{min},\ v_{max}$ | 250, 400m/min |
| $r_e$ | .8mm |
| $F_{cmax}$ | 5000N |
| $P_{MOT}$ | 10KW |
| $\eta$ | .75 |

Table 6.2: Parameters for metal cutting problem

Figure 6.17 shows the results for the above problem. Points returned by NSGA-II, before and after error correction are shown in Figures 6.17a and 6.17b, respectively. NSGA-II was run for 700 generations for a population size of 400. Not much difference between the initial and the corrected data is observable from the figures. However, an investigation of the average KKT-error of the initial and corrected points reveals a reduction from the initial value of 0.2629 to 0.0311, after the error reduction. As there is a large difference in the range in which the variables lie, scaling of variables is done to achieve accurate results for hyposurface fitting. Figure 6.17c shows the continuous Pareto-set as modeled by the cubic B-splines. The results mapped to the objective space along with the parameter values are shown in Figure 6.17d. Exact values of control points are given in appendix A. The use of cubic B-spline enables accurate modeling of the linear as well as the non-linear part of the Pareto-optimal set. The sharp junction between the linear and non-linear part of the Pareto-optimal front is also captured by the B-spline.

The production time varies from 0.8545 min to 1.1258 min and used tool life varies from 2.1131 to 9.4883 along the Pareto-optimal front. From parameter value u=0 to u=0.6, the production time ($T_p$) decreases from 1.1258 min to 0.9554 min. In the same interval, the percentage used tool life ($\xi$) changes from 2.1131 to 2.5659. In this interval, the change in production time with respect to the total change in production time along the front is large compared to the same relative change for the used tool life. It implies that the solutions in this interval may not be of much interest to the user. Figure 6.18a shows the variation in the value of cutting speed ($v$) with parameter value. From parameter value $u = 0$ to $u = 0.6$, $v$ remains constant at its lower bound value of 250 m/min. In this interval, value of $f$ rises steadily to 0.55 mm/rev which is its upper bound. Value of $a$ always decreases along the front. In the range u=0 to u=0.6, $f$ and $a$ show an inverse relationship where as from u=0.6 to u=1, $v$ and $a$ show an inverse relationship. Figure 6.18d shows the variation of constraints $g_2$ and $g_3$ along with parameters. The constraint $g_1$ is inactive everywhere along the front. From figure, it is observed that the roughness constraint ($g_3$) is also inactive everywhere along the front while the maximum cutting power constraint ($g_2$) is active along the whole front. The inferences drawn are summarized below:

1. From $u = 0$ to $u = 0.6$, $v \; = \; 250$ m/min

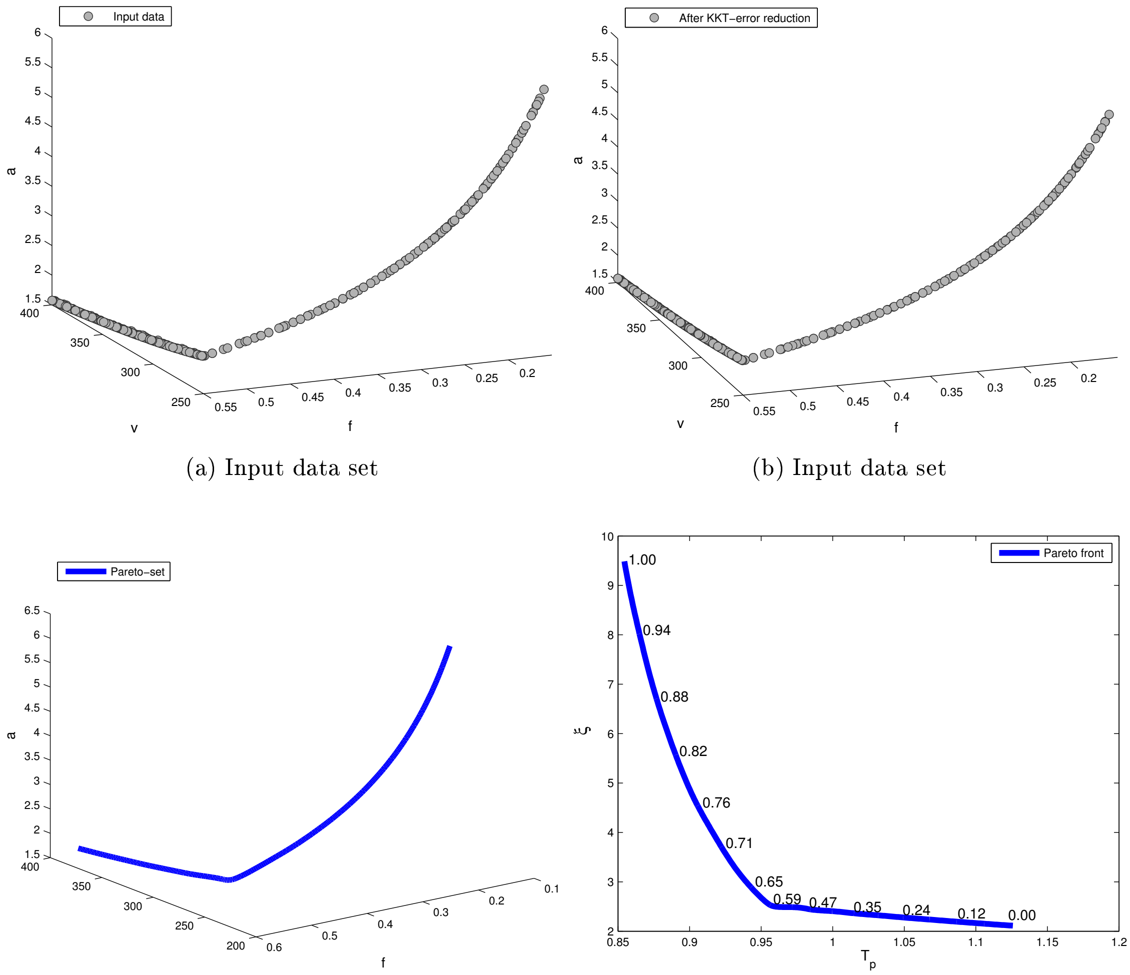
2. From $u = 0$ to $u = 0.6$, $f$ and $a$ are inversely related.

3. From $u = 0.6$ to $u = 1$, $f \; = \; .55$ mm/rev

4. From $u = 0.6$ to $u = 1$, $v$ and $a$ are inversely related.

5. $a$ decreases from u=0 to u=1.

6. Constraint $g_2$ is active everywhere along the front.



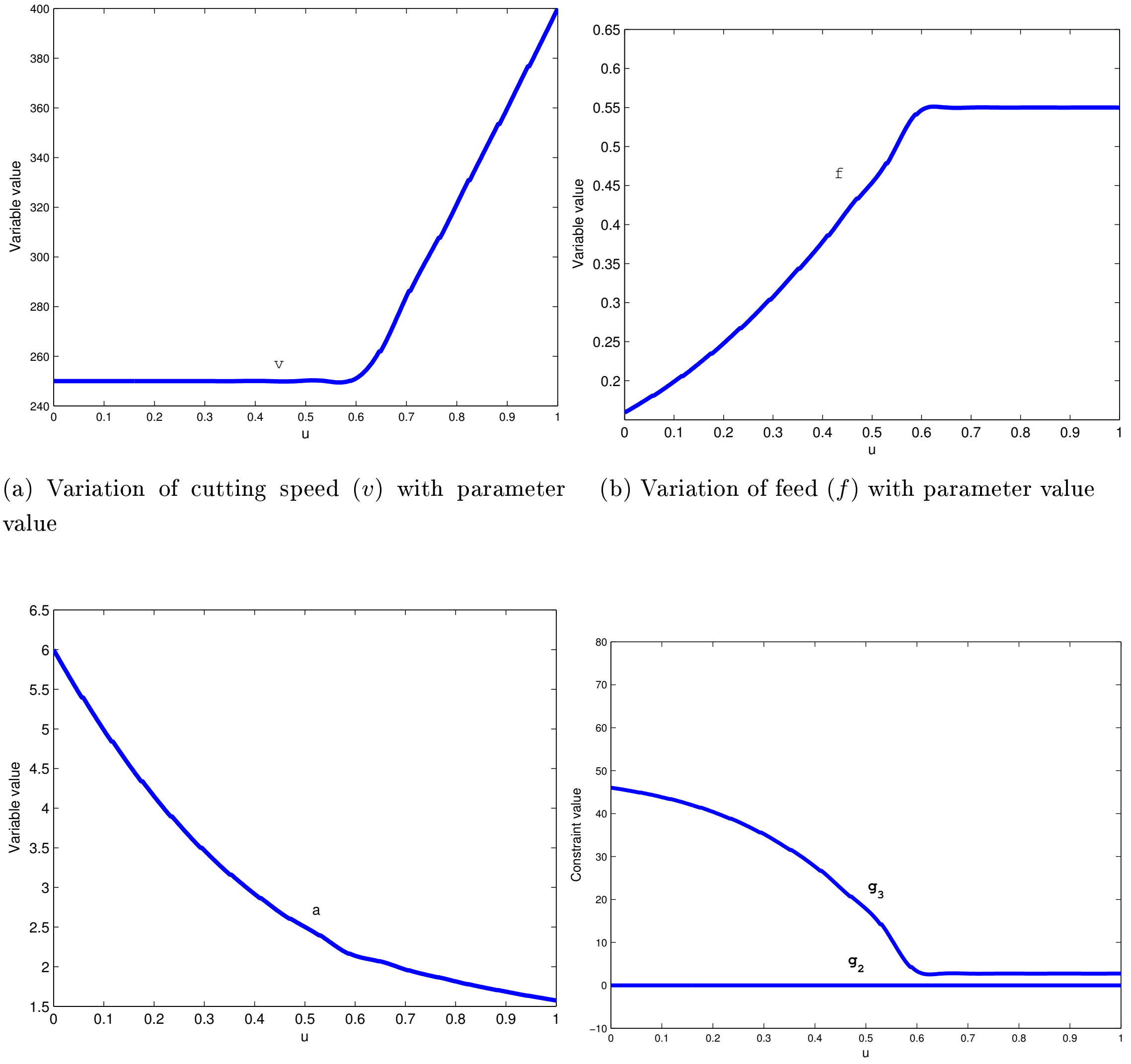
(a) Input data set

(b) Input data set

(c) Pareto-optimal set

(d) Pareto-optimal Front with parameter values

Figure 6.17: Results for three-variable metal cutting problem



(a) Variation of cutting speed ($v$) with parameter value

(b) Variation of feed ($f$) with parameter value

(c) Variation of depth of cut ($a$) with parameter value

(d) Variation of constraints with parameter value

Figure 6.18: Variation of 3 variables with parameter for metal cutting problem

Chapter 7

# Conclusions and Future Scope

In this chapter, we discuss the conclusions and future scope of this work. Before going into that we also look at some of the limitations of this algorithm.

## 7.1 Limitations of the algorithm

As a statistical model is used, it can not be said with certainty that all the points on the modeled hyposurface are truly Pareto-optimal points. This is an inherent problem of any data fitting procedure. Getting a continuous picture from a discrete image means making some assumptions about the model. If the data is dense we will have more confidence in the modeled hyposurface as we do not expect sudden changes in the Pareto-optimal front. However, with sparse data we can get an erroneous Pareto-optimal front even though the least square distance error is sufficiently low. In our model, the clustering process ensures that all the features of the input data are captured. The maximum distance between two points to be considered a continuous entity is fixed. If the distance between two points exceeds this distance, the points are considered to belong to two different clusters and a gap is thus detected.

For constrained optimization problems, the solutions may lie on the constraints boundary. In such situation the points modeled by the hyposurface may be infeasible. Often for such infeasible points, the function values may be slightly better than the true Pareto-optimal points. Thus, constraints of any point generated using this model must be checked.

The process of joining hyposurfaces together is a complicated process because the parameterization at the joints needs to be continuous. For continuity of parameterization, principal directions for all the patches need to be in synchronization. The complexity of the problem increases as the dimension of the hyposurface increases. The procedure presented here works for curves and surfaces, but a more efficient strategy is required for higher dimensional hyposurfaces.

The process of KKT-error reduction requires the objective functions to be differentiable so that local search can be conducted. If the objective function is not smooth and there are sudden kinks in the objective function or constraints then the process will have difficulties in reducing error leading to a higher unacceptable fitting error.

For higher dimensional hyposurface the process of variation of variable and constraints value along the front is difficult to analyze by using graphical methods. There are numerous directions in the parameter ($u$-$w$) space in which one can move. The graphical analysis is possible if one knows, a priori, the direction along which the properties are to be analyzed, otherwise it becomes a cumbersome task.

We consider problems where regularity conditions are met. So, for an $M$-objective problem the algorithm assumes that the Pareto-optimal front and the Pareto-optimal set are $(M-1)$-dimensional piecewise continuous entities. However, there is a class of problems for which this may not be so and the Pareto-optimal set can be a higher or lower dimensional manifold for a $(M-1)$-dimensional Pareto-optimal front. Several such problems are considered by Zhou et al. in [86].

## 7.2 Conclusions

This thesis proposes an algorithm to model the Pareto-optimal set of a MOOP using the data returned by a MOOA. A continuous picture is found from the discrete data set returned by a MOOA. The algorithm takes advantage of the regularity and connectedness property of the Pareto-optimal front to do modeling for certain class of problems. Cubic B-spline basis functions are used for the process of data fitting. Results are shown for some standard bi-objective test problems which show that the Pareto-front obtained by the model is close to the actual Pareto-optimal front.

Such a procedure helps to obtain a continuous picture of the front which gives a sense of completion in solving the optimization problem. Using the parametric hyposurface, points not returned by the MOOA can be generated. As the fitting is done in the variable space, one can easily generate points on the Pareto-optimal front at one's own discretion and retrace back to the variable space using the parameter values. This can be helpful in cases where variable values play an important role in decision making. In this paper results for several standard bi-objective and tri-objective problems and two engineering design problem were discussed. In future studies more engineering design problems can be analyzed.

The major contribution of the thesis are:

- An error reduction method based on KKT-error reduction is implemented to improve the results of a MOOA.

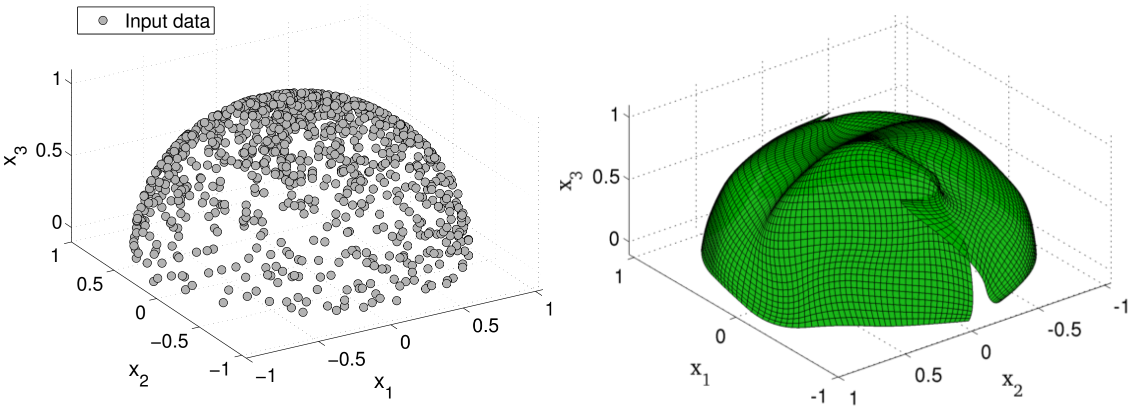
- A clustering procedure to find piecewise continuous Pareto-optimal set is implemented.

- A tree based data fitting procedure using B-spline basis function is proposed and implemented, to find the parameterization.

- No order in input data is assumed. Principal component analysis (PCA) is used to find the order in the data.

- Several standard and real-life optimization problems are used to test the method.
- Innovative design principles are found and analyzed using the procedure.

## 7.3 Future Scope



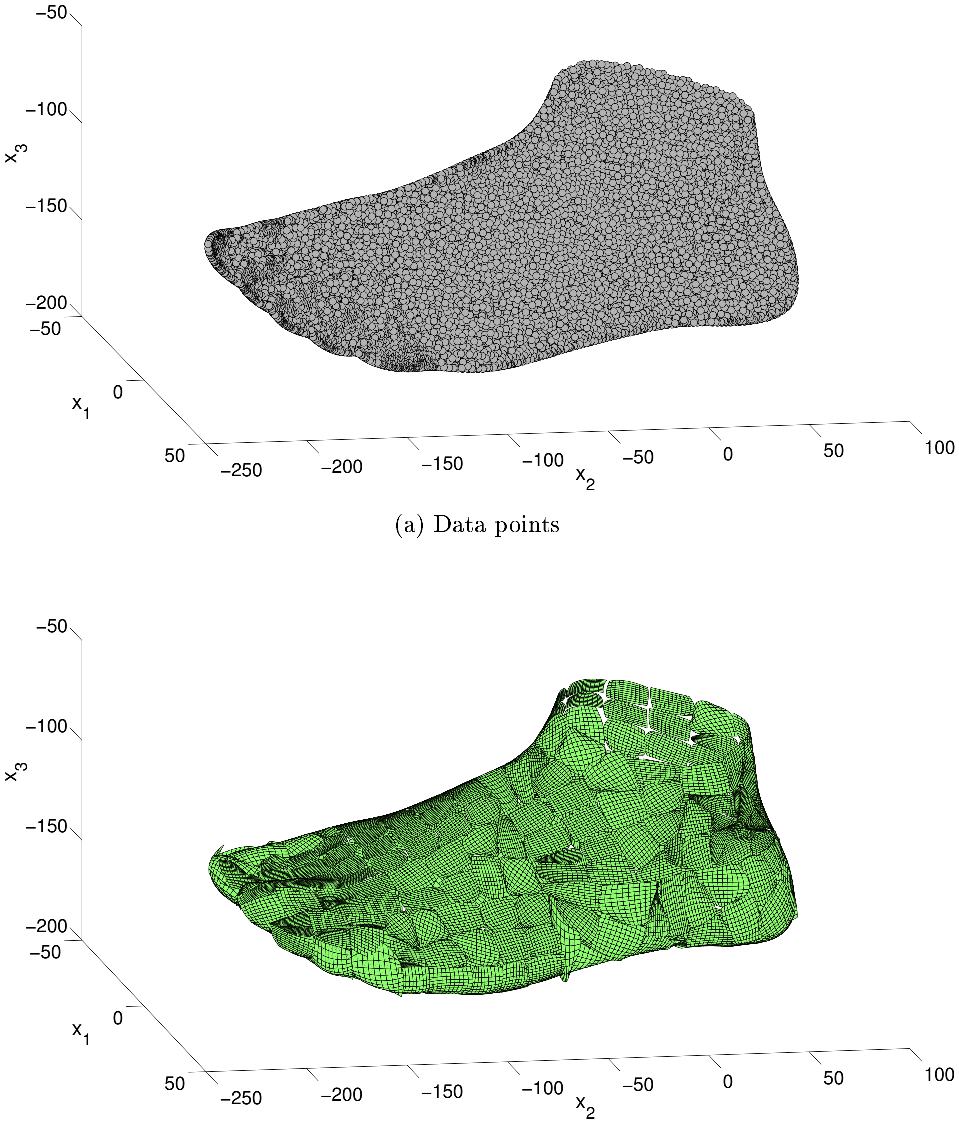
(a) Hemisphere data points

(b) Hemisphere fitted surface

Figure 7.1: Hemisphere data and fitted surface

- In principle the algorithm can be extended to higher-objective problems as well. An alternate formulation of B-spline basis functions will be required for higher dimensions as the matrix notation is good up to two-dimensional manifolds only. A tensor notation is required for higher-dimensional hyposurface. In higher-dimensions issues with clustering and merging of clusters may turn up, which can be addressed in future studies.
- The procedure of data fitting can also be used for designing MOOAs which use data fitting as an intermediate step to figure out relationships in non-dominated frontier data and use that for faster convergence of the algorithm. Moreover problems with complicated Pareto-optimal set can be tackled using such kind of integration.
- The procedure can be used as a stand-alone surface fitting algorithm. Initial results in this direction are motivating. Figure 7.1 shows data points of a hemisphere fitted using the algorithm presented here. Four B-spline patches are found and stitched together. For complicated surface as that shown in Figure 7.2[1], the process gives good surface patches, but joining them together with $C^2$ continuity turns up to be a cumbersome task. The example shown here has 20021 number of data points.

[1] Data points taken from `http://research.microsoft.com/en-us/um/people/hoppe/proj/bspline/`



(a) Data points

(b) Fitted surface

Figure 7.2: Foot data points and fitted surface

Appendix A

# Control Points of Modeled Curves and Surfaces for Test Problems

In this appendix control points and the equations of B-spline curve and surface that represent the Pareto-optimal set are given.

## A.1 FON problem

For a three variable FON problem the Pareto-optimal set is modeled by 15 control points shown below.

$$\mathbf{P} = \begin{bmatrix} -0.6727 & -0.6727 & -0.6727 \\ -0.5773 & -0.5773 & -0.5773 \\ -0.4819 & -0.4819 & -0.4819 \\ -0.3865 & -0.3865 & -0.3865 \\ -0.2911 & -0.2911 & -0.2911 \\ -0.1957 & -0.1957 & -0.1957 \\ -0.1003 & -0.1003 & -0.1003 \\ -0.0049 & -0.0049 & -0.0049 \\ 0.0905 & 0.0905 & 0.0905 \\ 0.1859 & 0.1859 & 0.1859 \\ 0.2813 & 0.2813 & 0.2813 \\ 0.3767 & 0.3767 & 0.3767 \\ 0.4721 & 0.4721 & 0.4721 \\ 0.5675 & 0.5675 & 0.5675 \\ 0.6629 & 0.6629 & 0.6629 \end{bmatrix} \tag{A.1}$$

Equations of 12 segments,correct to three decimal places, that model the Pareto-optimal set are shown in table A.1. As there are 12 segments parameter $u \in [0, 12]$ with each segment taking parameter values in gap of one.

| Parameters | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $[0, 12]$ | 0.095u-0.577 | 0.095u-0.577 | 0.095u-0.577 |

Table A.1: Parametric Equations for FON problem

| Parameters | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $[0, 1]$ | $-0.010u^3+0.023u^2+0.126u+0.014$ | 0 | 0 |
| $[1, 2]$ | $0.003u^3$-$0.014u^2+0.164u+0.002$ | 0 | 0 |
| $[2, 3]$ | $-0.001u^3+0.009u^2+0.118u+0.032$ | 0 | 0 |
| $[3, 4]$ | $0.001u^3$-$0.007u^2+0.165u$-$0.014$ | 0 | 0 |
| $[4, 5]$ | $-0.001u^3+0.008u^2+0.104u+0.066$ | 0 | 0 |
| $[5, 6]$ | $0.001u^3$-$0.013u^2+0.212u$-$0.114$ | 0 | 0 |
| $[6, 7]$ | $-0.002u^3+0.046u^2$-$0.141u+0.594$ | 0 | 0 |

Table A.2: Parametric Equations for ZDT2 problem

## A.2 ZDT2

For a three variable ZDT2 problem the Pareto-optimal set is modeled by 10 control points shown correct upto four places of decimal below.

$$\mathbf{P} = \begin{bmatrix} -0.0967 & -0.0000 & 0.0000 \\ 0.0063 & 0.0000 & 0.0000 \\ 0.1560 & 0.0000 & 0.0000 \\ 0.2939 & -0.0000 & 0.0000 \\ 0.4362 & 0.0000 & 0.0000 \\ 0.5765 & 0.0000 & 0.0000 \\ 0.7185 & 0.0000 & 0.0000 \\ 0.8585 & 0.0000 & 0.0000 \\ 1.0013 & 0.0000 & 0.0000 \\ 1.1321 & 0.0000 & -0.0000 \end{bmatrix} \tag{A.2}$$

Equations of seven segments,correct to three decimal places, that model the Pareto-optimal set are shown in table A.2. As there are seven segments parameter $u \in [0, 7]$ with each segment taking parameter values in a range of one.

## A.3 ZDT3

There are five clusters identified in this problem and hence five different B-splines. Each part of the Pareto-optimal set is modeled by 6 control points. Moving left to right along the Pareto-optimal front, the control points for the five clusters are represented by $P_1$, $P_2$, $P_3$, $P_4$ and $P_5$.

$$\mathbf{P_1} = \begin{bmatrix} -0.0159 & 0 & -0.0000 \\ -0.0029 & 0 & 0.0000 \\ 0.0289 & 0.0000 & -0.0000 \\ 0.0528 & 0.0000 & 0.0000 \\ 0.0852 & -0.0000 & 0.0000 \\ 0.0971 & 0 & -0.0000 \end{bmatrix} \tag{A.3}$$

$$\mathbf{P_2} = \begin{bmatrix} 0.1683 & 0.0000 & -0.0000 \\ 0.1801 & 0.0000 & 0.0000 \\ 0.2091 & -0.0000 & 0.0000 \\ 0.2315 & 0.0000 & 0.0000 \\ 0.2603 & -0.0000 & 0.0000 \\ 0.2723 & -0.0000 & -0.0000 \end{bmatrix} \tag{A.4}$$

$$\mathbf{P_3} = \begin{bmatrix} 0.4009 & 0.0000 & 0.0000 \\ 0.4079 & 0.0000 & 0.0000 \\ 0.4249 & 0.0000 & 0.0000 \\ 0.4380 & 0.0000 & 0.0000 \\ 0.4551 & 0.0000 & 0.0000 \\ 0.4624 & -0.0000 & -0.0000 \end{bmatrix} \tag{A.5}$$

$$\mathbf{P_4} = \begin{bmatrix} 0.6125 & 0.0000 & 0.0000 \\ 0.6174 & -0.0000 & 0.0000 \\ 0.6305 & 0.0000 & 0.0000 \\ 0.6403 & 0.0000 & 0.0000 \\ 0.6533 & -0.0000 & 0.0000 \\ 0.6587 & 0.0000 & 0.0000 \end{bmatrix} \tag{A.6}$$

$$\mathbf{P_5} = \begin{bmatrix} 0.8178 & 0.0000 & 0.0000 \\ 0.8224 & 0.0000 & -0.0000 \\ 0.8334 & -0.0000 & 0.0000 \\ 0.8418 & 0.0000 & -0.0000 \\ 0.8528 & 0.0000 & 0.0000 \\ 0.8575 & -0.0000 & 0.0000 \end{bmatrix} \tag{A.7}$$

| Front part | Parameters | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|---|
| 1 | $[0,1]$ | $-0.0045u^3 + 0.0095u^2 + 0.0224u + 0.0002$ | 0 | 0 |
| | $[1,2]$ | $0.0028u^3 - 0.0123u^2 + 0.0442u - 0.0070$ | 0 | 0 |
| | $[2,3]$ | $-0.0048u^3 + 0.0334u^2 - 0.0473u + 0.0540$ | 0 | 0 |
| 2 | $[0,1]$ | $-0.0039u^3 + 0.0085u^2 + 0.0204u + 0.1830$ | 0 | 0 |
| | $[1,2]$ | $0.0022u^3 - 0.0097u^2 + 0.0386u + 0.1769$ | 0 | 0 |
| | $[2,3]$ | $-0.0039u^3 + 0.0264u^2 - 0.0337u + 0.2251$ | 0 | 0 |
| 3 | $[0,1]$ | $-0.0023u^3 + 0.0050u^2 + 0.0120u + 0.4095$ | 0 | 0 |
| | $[1,2]$ | $0.0013u^3 - 0.0058u^2 + 0.0228u + 0.4059$ | 0 | 0 |
| | $[2,3]$ | $-0.0023u^3 + 0.0158u^2 - 0.0203u + 0.4347$ | 0 | 0 |
| 4 | $[0,1]$ | $-0.0019u^3 + 0.0040u^2 + 0.0090u + 0.6187$ | 0 | 0 |
| | $[1,2]$ | $0.0011u^3 - 0.0048u^2 + 0.0178u + 0.6158$ | 0 | 0 |
| | $[2,3]$ | $-0.0018u^3 + 0.0123u^2 - 0.0164u + 0.6386$ | 0 | 0 |
| 5 | $[0,1]$ | $-0.0015u^3 + 0.0032u^2 + 0.0078u + 0.8234$ | 0 | 0 |
| | $[1,2]$ | $0.0008u^3 - 0.0038u^2 + 0.0148u + 0.8211$ | 0 | 0 |
| | $[2,3]$ | $-0.0015u^3 + 0.0102u^2 - 0.0133u + 0.8398$ | 0 | 0 |

Table A.3: Parametric equation for ZDT3 problem

The parametric equation for each cluster is shown in Table A.3

## A.4 POL

POL problem has two clusters. Cluster A is modeled by 12 control points where as cluster B is modeled by 24 control points. Control points of both clusters are given below.

$$\mathbf{P_A} = \begin{bmatrix} 0.7341 & 1.5557 \\ 0.7769 & 1.5710 \\ 0.8368 & 1.5838 \\ 0.8959 & 1.6110 \\ 0.9335 & 1.6350 \\ 0.9841 & 1.6997 \\ 1.0072 & 1.7624 \\ 0.9892 & 1.8213 \\ 1.0023 & 1.8764 \\ 0.9951 & 1.9331 \\ 1.0037 & 1.9943 \\ 0.9876 & 2.0329 \end{bmatrix} \mathbf{P_B} = \begin{bmatrix} -2.9781 & -1.0153 \\ -2.9993 & -0.9985 \\ -3.0873 & -0.9713 \\ -3.1551 & -0.9330 \\ -3.1367 & -0.8099 \\ -3.1442 & -0.7018 \\ -3.1401 & -0.6310 \\ -3.1423 & -0.5346 \\ -3.1412 & -0.4412 \\ -3.1418 & -0.3525 \\ -3.1415 & -0.2618 \\ -3.1416 & -0.1678 \\ -3.1416 & -0.0756 \\ -3.1416 & 0.0049 \\ -3.1416 & 0.0876 \\ -3.1416 & 0.1687 \\ -3.1416 & 0.2471 \\ -3.1416 & 0.3464 \\ -3.1416 & 0.4306 \\ -3.1416 & 0.5176 \\ -3.1416 & 0.5947 \\ -3.1416 & 0.6859 \\ -3.1416 & 0.7657 \\ -3.1416 & 0.8442 \end{bmatrix} \tag{A.8}$$

## A.5 DTLZ1

DTLZ1 is modeled by 30 control points. Control points are given below.

$$\mathbf{P} = \begin{bmatrix} 1.1171 & 1.1307 & 0.5000 \\ 1.0253 & 1.1070 & 0.5000 \\ 0.4879 & 1.0951 & 0.5000 \\ 0.2846 & 1.1009 & 0.5000 \\ -0.1875 & 1.1990 & 0.5000 \\ 1.0922 & 0.8316 & 0.5000 \\ 0.9413 & 0.8794 & 0.5000 \\ 0.5484 & 0.9346 & 0.5000 \\ 0.1715 & 0.9617 & 0.5000 \\ -0.1722 & 1.0437 & 0.5000 \\ 1.0943 & 0.5366 & 0.5000 \\ 0.9200 & 0.5867 & 0.5000 \\ 0.4907 & 0.6222 & 0.5000 \\ 0.1388 & 0.6789 & 0.5000 \\ -0.1134 & 0.6958 & 0.5000 \\ 1.1181 & 0.2977 & 0.5000 \\ 0.8678 & 0.3229 & 0.5000 \\ 0.4725 & 0.3740 & 0.5000 \\ 0.1146 & 0.4080 & 0.5000 \\ -0.1026 & 0.4818 & 0.5000 \\ 1.1416 & -0.0282 & 0.5000 \\ 0.8522 & 0.0200 & 0.5000 \\ 0.4444 & 0.0653 & 0.5000 \\ 0.0522 & 0.1221 & 0.5000 \\ -0.0766 & 0.1532 & 0.5000 \\ 1.0934 & -0.1101 & 0.5000 \\ 0.6903 & -0.1021 & 0.5000 \\ 0.4547 & -0.1005 & 0.5000 \\ 0.0044 & -0.0940 & 0.5000 \\ -0.1265 & -0.1996 & 0.5000 \end{bmatrix} \tag{A.9}$$

## A.6 DTLZ2

DTLZ2 is modeled by 30 control points. Control points are given below.

$$\mathbf{P} = \begin{bmatrix} 1.0621 & 1.1065 & 0.5000 \\ 0.9547 & 1.1149 & 0.5000 \\ 0.5077 & 1.0953 & 0.5000 \\ 0.2255 & 1.1042 & 0.5000 \\ -0.1989 & 1.0978 & 0.5000 \\ 1.1092 & 0.9052 & 0.5000 \\ 0.8638 & 0.8810 & 0.5000 \\ 0.5264 & 0.9132 & 0.5000 \\ 0.1610 & 0.9390 & 0.5000 \\ -0.1332 & 0.9559 & 0.5000 \\ 1.0578 & 0.5629 & 0.5000 \\ 0.8567 & 0.6102 & 0.5000 \\ 0.5065 & 0.6293 & 0.5000 \\ 0.1404 & 0.6539 & 0.5000 \\ -0.1333 & 0.6808 & 0.5000 \\ 1.1336 & 0.3420 & 0.5000 \\ 0.8274 & 0.3560 & 0.5000 \\ 0.4937 & 0.3809 & 0.5000 \\ 0.1160 & 0.4016 & 0.5000 \\ -0.1013 & 0.4241 & 0.5000 \\ 1.1783 & 0.0386 & 0.5000 \\ 0.8089 & 0.0741 & 0.5000 \\ 0.4737 & 0.0985 & 0.5000 \\ 0.0974 & 0.1293 & 0.5000 \\ -0.0936 & 0.1523 & 0.5000 \\ 0.9292 & -0.0949 & 0.5000 \\ 0.7858 & -0.1066 & 0.5000 \\ 0.4730 & -0.0936 & 0.5000 \\ 0.0647 & -0.1208 & 0.5000 \\ -0.1088 & -0.0710 & 0.5000 \end{bmatrix} \tag{A.10}$$

## A.7 DTLZ7

DTLZ7 has four clusters. Each cluster is modeled by different number of control points. Cluster 1 is modeled by 12, cluster 2 is modeled by 20 and cluster 3 and 4 are modeled by 15 number of control points each. Control points for each cluster are shown below.

$$\mathbf{P_1} = \begin{bmatrix} 0.7232 & 0.9117 & -0.0000 \\ 0.5878 & 0.9117 & -0.0000 \\ 0.5878 & 0.6765 & 0.0001 \\ 0.9096 & 0.9117 & -0.0000 \\ 0.6807 & 0.8600 & 0.0000 \\ 0.5878 & 0.5760 & -0.0000 \\ 0.9096 & 0.9117 & -0.0000 \\ 0.8506 & 0.6605 & 0.0000 \\ 0.5878 & 0.5760 & -0.0000 \\ 0.9027 & 0.6970 & -0.0001 \\ 0.9096 & 0.5760 & -0.0000 \\ 0.6561 & 0.5760 & 0.0001 \end{bmatrix} \mathbf{P_2} = \begin{bmatrix} 0.0180 & -0.0505 & -0.0000 \\ 0.0067 & 0.0441 & -0.0000 \\ -0.0519 & -0.0511 & -0.0000 \\ 0.0052 & -0.0102 & 0.0000 \\ -0.0047 & 0.0537 & 0.0003 \\ -0.0168 & 0.2919 & -0.0001 \\ -0.0332 & 0.1339 & 0.0000 \\ 0.0489 & 0.0554 & 0.0000 \\ 0.1387 & -0.0301 & 0.0000 \\ 0.2432 & -0.0253 & 0.0000 \\ -0.0280 & 0.2746 & 0.0002 \\ 0.1152 & 0.2906 & -0.0000 \\ 0.1937 & 0.1971 & 0.0000 \\ 0.2833 & 0.1115 & -0.0000 \\ 0.2989 & -0.0126 & 0.0000 \\ 0.3115 & 0.2609 & -0.0000 \\ 0.2166 & 0.2540 & 0.0000 \\ 0.3114 & 0.3123 & -0.0001 \\ 0.2789 & 0.2276 & 0.0001 \\ 0.0958 & 0.3132 & 0.0004 \end{bmatrix} \tag{A.11}$$

$$\mathbf{P_3} = \begin{bmatrix} 0.6091 & 0.3153 & -0.0000 \\ 0.8516 & 0.3153 & 0.0001 \\ 0.5818 & 0.3153 & -0.0000 \\ 0.5818 & 0.3153 & -0.0000 \\ 0.5818 & 0.3153 & 0.0000 \\ 0.9057 & 0.3153 & 0.0000 \\ 0.8881 & 0.0582 & -0.0000 \\ 0.7265 & 0.2849 & 0.0000 \\ 0.6247 & -0.0298 & 0.0000 \\ 0.5818 & 0.0598 & 0.0003 \\ 0.9057 & -0.0526 & -0.0000 \\ 0.7994 & -0.0526 & 0.0001 \\ 0.9057 & -0.0526 & -0.0000 \\ 0.9057 & -0.0526 & 0.0000 \\ 0.5818 & -0.0526 & 0.0004 \end{bmatrix} \mathbf{P_4} = \begin{bmatrix} -0.0513 & 0.9090 & 0.0002 \\ -0.0513 & 0.8185 & -0.0000 \\ -0.0513 & 0.9095 & -0.0000 \\ -0.0513 & 0.9095 & -0.0000 \\ -0.0513 & 0.5829 & 0.0001 \\ 0.3138 & 0.9095 & -0.0000 \\ 0.1394 & 0.9003 & -0.0000 \\ 0.2380 & 0.7333 & 0.0000 \\ -0.0125 & 0.6653 & -0.0000 \\ 0.0018 & 0.5829 & -0.0000 \\ 0.3138 & 0.9095 & -0.0000 \\ 0.3138 & 0.5829 & 0.0000 \\ 0.3138 & 0.5829 & -0.0000 \\ 0.3138 & 0.5829 & 0.0001 \\ 0.3137 & 0.5829 & -0.0000 \end{bmatrix} \tag{A.12}$$

## A.8 OSY

OSY has 5 different clusters identified as AB, BC, CD, DE and EF. Cluster AB is modeled by 7 control points, BC by 5 control points, CD by 4 control points, DE and EF by 6 control points each. The control points for each of these 5 parts of the Pareto-optimal set are shown below.

$$\mathbf{P_{AB}} = \begin{bmatrix} 3.9165 & 0.6388 & 1.0006 & 0.0000 & 1.0000 & -0.0000 \\ 4.0467 & 0.6822 & 0.9998 & 0.0000 & 1.0000 & -0.0000 \\ 4.2990 & 0.7663 & 1.0002 & 0.0000 & 1.0000 & 0.0000 \\ 4.5312 & 0.8437 & 0.9997 & -0.0000 & 1.0000 & 0.0000 \\ 4.7603 & 0.9201 & 1.0008 & 0.0000 & 1.0000 & 0.0000 \\ 5.0156 & 1.0052 & 0.9961 & 0.0000 & 1.0000 & 0.0000 \\ 5.1470 & 1.0490 & 1.0565 & 0.0000 & 1.0000 & 0.0000 \end{bmatrix} \tag{A.13}$$

$$\mathbf{P_{BC}} = \begin{bmatrix} 5.0000 & 1.0000 & 0.5947 & 0 & 5.0000 & 0.0000 \\ 5.0000 & 1.0000 & 1.1839 & 0 & 5.0000 & 0.0000 \\ 5.0000 & 1.0000 & 3.1459 & 0 & 5.0000 & 0.0000 \\ 5.0000 & 1.0000 & 5.1633 & 0 & 5.0000 & 0.0000 \\ 5.0000 & 1.0000 & 5.7323 & 0 & 5.0000 & 0.0000 \end{bmatrix}$$

$$\mathbf{P_{CD}} = \begin{bmatrix} 5.0000 & 1.0000 & 5.6903 & 0 & 1.0475 & 0.1934 \\ 5.0000 & 1.0000 & 5.6903 & 0 & 0.9924 & -0.0316 \\ 5.0000 & 1.0000 & 0.8582 & 0 & 1.0048 & 0.0198 \\ 5.0000 & 1.0000 & 0.8582 & 0 & 0.9919 & -0.0322 \end{bmatrix}$$

$$\mathbf{P_{DE}} = \begin{bmatrix} 0.4877 & 1.5165 & 1.5989 & 0.0000 & 1.0000 & -0.0000 \\ -0.1975 & 2.1963 & 1.2288 & 0.0000 & 1.0000 & 0.0000 \\ 0.2622 & 1.7385 & 0.9002 & -0.0000 & 1.0000 & 0.0000 \\ 0.6616 & 1.3376 & 1.0747 & 0.0000 & 1.0000 & -0.0000 \\ 0.9934 & 1.0080 & 0.9002 & -0.0000 & 1.0000 & 0.0000 \\ 1.1851 & 0.8219 & 1.3931 & -0.0000 & 1.0000 & 0.0001 \end{bmatrix}$$

$$\mathbf{P_{EF}} = \begin{bmatrix} 0.0000 & 2.0000 & 4.1123 & 0.0000 & 1.0000 & -0.0014 \\ 0.0000 & 2.0000 & 3.7614 & 0.0000 & 1.0000 & 0.0005 \\ -0.0000 & 2.0000 & 2.9727 & -0.0000 & 1.0000 & -0.0005 \\ -0.0000 & 2.0000 & 2.3564 & -0.0000 & 1.0000 & 0.0007 \\ 0.0000 & 2.0000 & 1.5512 & 0.0000 & 1.0000 & 0.0015 \\ 0.0001 & 2.0000 & 1.2481 & 0.0000 & 1.0000 & -0.0016 \end{bmatrix}$$

## A.9 Two Bar Truss Problem

In this problem Pareto-optimal set is modeled by 20 control points. The control points are given below.

$$\mathbf{P} = \begin{bmatrix} 0.0002 & 0.0003 & 1.9987 \\ 0.0004 & 0.0009 & 2.0003 \\ 0.0007 & 0.0015 & 2.0001 \\ 0.0010 & 0.0021 & 2.0001 \\ 0.0013 & 0.0026 & 2.0000 \\ 0.0016 & 0.0032 & 2.0002 \\ 0.0019 & 0.0038 & 2.0000 \\ 0.0022 & 0.0044 & 2.0001 \\ 0.0025 & 0.0050 & 1.9998 \\ 0.0028 & 0.0056 & 2.0004 \\ 0.0031 & 0.0061 & 1.9991 \\ 0.0034 & 0.0067 & 2.0016 \\ 0.0037 & 0.0073 & 1.9969 \\ 0.0039 & 0.0079 & 2.0053 \\ 0.0043 & 0.0085 & 1.9899 \\ 0.0045 & 0.0091 & 2.0210 \\ 0.0051 & 0.0100 & 1.9542 \\ 0.0048 & 0.0100 & 2.1470 \\ 0.0036 & 0.0100 & 3.1787 \\ 0.0046 & 0.0102 & 3.0752 \end{bmatrix} \tag{A.14}$$

## A.10 Metal Cutting problem

Pareto-optimal set in this problem is modeled by 20 control points. These control points are given below.

$$\mathbf{P} = \begin{bmatrix} 250.0537 & 0.1395 & 6.6309 \\ 249.9867 & 0.1587 & 5.9960 \\ 250.0098 & 0.1802 & 5.3867 \\ 249.9870 & 0.2053 & 4.8333 \\ 250.0223 & 0.2342 & 4.3298 \\ 249.9615 & 0.2668 & 3.8864 \\ 250.0701 & 0.3029 & 3.4931 \\ 249.8747 & 0.3437 & 3.1511 \\ 250.2373 & 0.3841 & 2.8639 \\ 249.5173 & 0.4366 & 2.5883 \\ 251.0063 & 0.4693 & 2.4170 \\ 247.6387 & 0.5570 & 2.1182 \\ 259.0589 & 0.5482 & 2.0905 \\ 287.9872 & 0.5506 & 1.9405 \\ 306.7920 & 0.5496 & 1.8696 \\ 331.3558 & 0.5501 & 1.7748 \\ 353.2345 & 0.5498 & 1.7053 \\ 376.9374 & 0.5501 & 1.6329 \\ 399.6038 & 0.5499 & 1.5747 \\ 424.5796 & 0.5497 & 1.5079 \end{bmatrix} \tag{A.15}$$

# Bibliography

[1] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley, 2001. (Cited on pages 3, 33, 36 and 47.)

[2] T. W. Athan and P. Y. Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27:155–176, 1996. (Cited on page 4.)

[3] V. Chankong and Y.Y. Haimes. *Multiobjective Decision Making: Theory and Methodology.* Dover Books on Engineering Series. Dover Publications, 2008. (Cited on page 4.)

[4] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8:631–657, 1998. (Cited on page 4.)

[5] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, April 2004. (Cited on page 4.)

[6] A. Charnes, R. W. Clower, and K. O. Kortanek. Effective control through coherent decentralization with preemptive goals. *Econometrica*, 35:294–320, 1967. (Cited on page 4.)

[7] A. Charnes and W. W. Cooper. Goal programming and multiple objective optimization; part 1. *European Journal of Operations Research*, 1:39–54, 1977. (Cited on page 4.)

[8] A. Messac. Physical programming: effective optimization for computational design. *AIAA Journal*, 34:149–158, 1996. (Cited on page 4.)

[9] J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, NJ, USA, 1985. L. Erlbaum Associates Inc. (Cited on page 5.)

[10] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, 1993. (Cited on page 5.)

[11] Jeffrey Horn, Jeffrey Horn, Nicholas Nafpliotis, Nicholas Nafpliotis, David E. Goldberg, and David E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 82–87, 1994. (Cited on page 5.)

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Comp.*, 6(2):182–197, April 2002. (Cited on pages 5 and 19.)

[13] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999. (Cited on page 5.)

[14] Joshua Knowles and David Corne. Approximating the non-dominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8:149–172, 1999. (Cited on page 5.)

[15] C. Korsemov Guliashki V., H. Toshev. Survey of evolutionary algorithms used in multiobjective optimization. *Problems of Engineering Cybernetics and Robotics*, 60:42–54, 2009. (Cited on page 5.)

[16] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011. (Cited on page 5.)

[17] M. Hirschberger, Y. Qi, and R. E. Steuer. Large-scale mv efficient frontier computation via a procedure of parametric quadratic programming. *European Journal of Operational Research*, 204:581–588, 2010. (Cited on page 7.)

[18] H. Markowitz. Portfolio Selection. *The Journal of Finance*, 7(1):77–91, 1952. (Cited on page 7.)

[19] Crina Grosan. A new evolutionary technique for detecting Pareto continuous regions. In Alwyn Barry, editor, *2003 Genetic and Evolutionary Computation Conference. Workshop Program*, pages 304–307, Chicago, Illinois, USA, July 2003. AAAI. (Cited on page 7.)

[20] D. Dumitrescu, Crina Groşan, and Mihai Oltean. Genetic Chromodynamics for Obtaining Continuous Representation of Pareto Regions. *Studia Universitatis Babeş–Bolyai, Informatica*, XLVI(1):15–30, 2001. (Cited on page 7.)

[21] D. A. V. Veldhuizen. *Multiobjective Evolutionary Algorithms: Classification, Analyses and New Innovations.* PhD thesis, Air Force Institute of Technology, Air University, 1999. (Cited on page 8.)

[22] Thomas Bäck. *Evolutionary algorithms in theory and practice - evolution strategies, evolutionary programming, genetic algorithms.* Oxford University Press, 1996. (Cited on page 8.)

[23] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker. Response surface approximation of pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering*, 196:879–893, 2007. (Cited on pages 8 and 19.)

[24] M. Dellnitz, O. Schutze, and T. Hestermeyer. Covering pareto sets by multilevel subdivision techniques. *Journal of optimization theory and applications*, 124(1):113–136, January 2005. (Cited on pages 8 and 10.)

[25] T. Okabe, Y. Jin, B. Sendhoff, and M. Olhofer. Voronoi-based estimation of distribution algorithm for multi-objective optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, pages 1594–1601, Portland, Oregon, USA, 2004. IEEE press. (Cited on page 8.)

[26] P. A. N. Bosman and D. Thierens. The naive midea: A baseline multiobjective ea. In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005).*, volume 3410 of *LNCS*, pages 428–442, Guanajuato, Mexico, 2005. Springer. (Cited on page 8.)

[27] A. Zhou, Q. Zhang, Y. Jin, E. Tsang, and T. Okabe. A model-based evolutionary algorithm for bi-objective optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC 2005).*, pages 2568–2575, Edinburgh,U.K, 2005. IEEE Press. (Cited on page 8.)

[28] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In *Proceedings of the Congress on Evolutionary Computation (CEC 2006)*, pages 3234–3241, Vancouver, BC,Canada, 2006. IEEE Press. (Cited on pages 8 and 9.)

[29] Q. Zhang, A. Zhou, and Y. Jin. Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. Evolutionary Computation*, 12(1):41–63, 2008. (Cited on pages 9 and 14.)

[30] S. Bandaru and K. Deb. Towards automating the discovery of certain innovative design principles through a clustering-based optimization technique. *Engineering optimization*, pages 1–31, 2011. (Cited on pages 9 and 52.)

[31] S. Bandaru and K. Deb. Automated discovery of vital knowledge from pareto-optimal solutions: First results from engineering design. In *IEEE Congress on Evolutionary Computation (CEC-2010)*, pages 1224–1231. IEEE Press, 2010. (Cited on page 9.)

[32] Sunith Bandaru and Kalyanmoy Deb. Automated innovization for simultaneous discovery of multiple rules in bi-objective problems. In Ricardo Takahashi, Kalyanmoy

Deb, Elizabeth Wanner, and Salvatore Greco, editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin / Heidelberg, 2011. (Cited on page 9.)

[33] P.Y. Papalambros and D.J. Wilde. *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, 2000. (Cited on page 9.)

[34] C. R. Bonham and I. C. Parmee. Developments of the cluster oriented genetic algorithm. *Engineering Optimization*, 36(2):249–279, April 2004. (Cited on page 10.)

[35] B. Williams and J. Cagan. Activity analysis: Simplifying optimal design problems through qualitative partitioning. *Engineering Optimization*, 27(2):109–138, 1995. (Cited on page 10.)

[36] Y. Jin and B. Sendhoff. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1910–1917, 2003. (Cited on pages 10, 11 and 15.)

[37] Juairgen Guddat. *Multiobjective and stochastic optimization based on parametric optimization*. Akademie-Verlag GmbH, Berlin, Germany, Germany, 1987. (Cited on page 10.)

[38] J. Rakowska, R.T. Haftka, L.T. Watson, Virginia Polytechnic Institute, and State University. Dept. of Computer Science. *Tracing the Efficient Curve for Multi-objective Control-structure Optimization*. TR (Virginia Polytechnic Institute and State University. Dept. of Computer Science). Department of Computer Science, Virginia Polytechnic Institute and State University, 1991. (Cited on page 10.)

[39] C. Hillermeier. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. International Series of Numerical Mathematics. Birkhaüser Verlag, 2001. (Cited on page 10.)

[40] Oliver Schutze, Carlos A. Coello Coello, Sanaz Mostaghim, El-Ghazali Talbi, and Michael Dellnitz. Hybridizing evolutionary strategies with continuation methods for solving multi-objective problems. *Engineering Optimization*, 40:383–402, 2008. (Cited on page 10.)

[41] Oliver Schütze, Alessandro Dell'Aere, and Michael Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In *Practical Approaches to Multi-Objective Optimization*, 2005. (Cited on page 10.)

[42] Maik Ringkamp, Sina Ober-Blobaum, Michael Dellnitz, and Oliver Schutze. Handling high-dimensional problems with multi-objective continuation methods via successive

approximation of the tangent space. *Engineering Optimization*, pages 1–30, 2012. (Cited on page 10.)

[43] M. Ehrgott and K. Klamroth. Connectedness of efficient solutions in multiple criteria combinatorial optimization. *European Joumal of Opemtional Research*, 1997. (Cited on pages 11 and 15.)

[44] A. Daniilidis, N. Hadjisavvas, and S. Schaible. Connectedness of the efficient set for three-objective quasiconcave maximization problems. *J. Optim. Theory Appl.*, 93(3):517–524, 1997. (Cited on pages 11 and 15.)

[45] M. E. Mortenson. *Geometric Modeling.* John Wiley and Sons, 1985. (Cited on pages 11, 16 and 25.)

[46] L. Fang and D. C. Gossard. Multidimensional curve fitting to unorganized data points by nonlinear minimization. *Computer-Aided Design*, 27(1):48–58, 1995. (Cited on page 11.)

[47] Nina Amenta, Marshall Bern, and David Eppstein. The crust and the beta-skeleton: Combinatorial curve reconstruction. In *Graphical Models and Image Processing*, pages 125–135, 1998. (Cited on page 11.)

[48] A. Atieg and G. A. Watson. A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances. *J. Comput. Appl. Math.*, 158(2):277–296, September 2003. (Cited on page 11.)

[49] Å. Björck. *Numerical methods for least squares problems.* Miscellaneous Bks. SIAM, 1996. (Cited on page 11.)

[50] Tamal K. Dey, Kurt Mehlhorn, and Edgar A. Ramos. Curve reconstruction: Connecting dots with good reason. *In Proc. 15th Annu. ACM Sympos. Comput. Geom*, 15:229–244, 1999. (Cited on page 11.)

[51] E.T.Y. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6):363 – 370, 1989. (Cited on page 11.)

[52] Josef Hoschek and Dieter Lasser. *Fundamentals of computer aided geometric design.* A. K. Peters, Ltd., Natick, MA, USA, 1993. (Cited on page 11.)

[53] G.E. Farin. *Curves and surfaces for computer-aided geometric design: a practical guide.* Number v. 1 in Computer science and scientific computing. Academic Press, 1997. (Cited on page 11.)

[54] Michael Plass and Maureen Stone. Curve-fitting with piecewise parametric cubics. *SIGGRAPH Comput. Graph.*, 17(3):229–239, July 1983. (Cited on page 11.)

[55] J. Hoschek. Intrinsic parametrization for approximation. *Comput. Aided Geom. Des.*, 5(1):27–31, June 1988. (Cited on page 11.)

[56] A. Ardeshir Goshtasby. Grouping and parameterizing irregularly spaced points for curve fitting. *ACM Trans. Graph*, pages 185–203, 2000. (Cited on page 11.)

[57] Eric Saux and Marc Daniel. An improved hoschek intrinsic parametrization. *Comput. Aided Geom. Des.*, 20(8-9):513–521, December 2003. (Cited on page 11.)

[58] Hugues Hoppe, Tony Derose, Tom Duchamp, Mark Halstead, Hubert Jin, John McDonald, Jean Schweitzer, and Werner Stuetzle. Piecewise smooth surface reconstruction. pages 295–302, 1994. (Cited on page 11.)

[59] David R. Forsey and Richard H. Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14:134–161, 1995. (Cited on page 11.)

[60] Hugues Hoppe. Progressive meshes. In *SIGGRAPH*, pages 99–108, 1996. (Cited on page 11.)

[61] Weiyin Ma and JP Kruth. Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces. *Computer-Aided Design*, 27(9):663 – 675, 1995. (Cited on page 11.)

[62] Jörg Haber, Frank Zeilfelder, Oleg Davydov, and Hans Peter Seidel. Smooth approximation and rendering of large scattered data sets. In *Proceedings of the conference on Visualization '01*, VIS '01, pages 341–348, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 11.)

[63] Xiaohuan Corina Wang and Cary Phillips. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 129–138, New York, NY, USA, 2002. ACM. (Cited on page 11.)

[64] M. Djebali, M. Melkemi, and N. Sapidis. Range-image segmentation and model reconstruction based on a fit-and-merge strategy. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, SMA '02, pages 127–138, New York, NY, USA, 2002. ACM. (Cited on page 11.)

[65] Lian Fang and David C Gossard. Multidimensional curve fitting to unorganized data points by nonlinear minimization. *Computer-Aided Design*, 27(1):48 – 58, 1995. (Cited on page 11.)

[66] Hong Yan. Convergence condition and efficient implementation of the fuzzy curve-tracing (fct) algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(1):210–221, 2004. (Cited on page 12.)

[67] Yang Liu, Huaiping Yang, and Wenping Wang. Reconstructing b-spline curves from point clouds–a tangential flow approach using least squares minimization. In *Proceedings of the International Conference on Shape Modeling and Applications 2005*, SMI '05, pages 4–12, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on page 12.)

[68] A. Ardeshir Goshtasby. Grouping and parameterizing irregularly spaced points for curve fitting. *ACM Trans. Graph*, pages 185–203, 2000. (Cited on page 12.)

[69] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):pp. 502–516, 1989. (Cited on page 12.)

[70] Rocco Furferi, Lapo Governi, Matteo Palai, and Yary Volpe. From unordered point cloud to weighted b-spline: a novel pca-based method. In *Proceedings of the 2011 American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications*, AMERICAN-MATH'11/CEA'11, pages 146–151, Stevens Point, Wisconsin, USA, 2011. World Scientific and Engineering Academy and Society (WSEAS). (Cited on page 12.)

[71] Benjamin Gregorski and Bernd Hamann at Kenneth I. Joy. Reconstruction of b-spline surfaces from scattered data points. In *Proceedings of the International Conference on Computer Graphics*, CGI '00, pages 163–, Washington, DC, USA, 2000. IEEE Computer Society. (Cited on page 12.)

[72] William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939. (Cited on page 13.)

[73] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In Jerzy Neyman, editor, *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, Berkeley, CA, USA, 1950. (Cited on page 13.)

[74] M. Slater. *Lagrange multipliers revisited: a contribution to non-linear programming.* 1950. (Cited on page 14.)

[75] O. Mangasarian and S. Fromovitz. The Fritz-John necessary optimality conditions in the presence of equality and inequality constraints. *Journal of Mathematical Analysis and Applications*, 7:37–47, 1967. (Cited on page 14.)

[76] C. Hillermeier. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach.* International Series of Numerical Mathematics. Birkhaüser Verlag, 2001. (Cited on pages 14 and 15.)

[77] A P Wierzbicki. The use of reference objectives in multiobjective optimization-theoretical implications and practical experiences. *Int Inst Applied System Analysis*, pages 1–32, 1979. (Cited on page 20.)

[78] Karthik Sindhya, Ankur Sinha, Kalyanmoy Deb, and Kaisa Miettinen. Local Search Based Evolutionary Multi-Objective Optimization Algorithm for Constrained and Unconstrained Problems. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 2919–2926, Trondheim, Norway, May 2009. IEEE Press. (Cited on page 20.)

[79] K. Deb, U. B. Rao N. J. Sundar, and S.Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(3):273–286, 2006. (Cited on page 21.)

[80] S. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3), September 1994. (Cited on page 22.)

[81] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010. (Cited on page 22.)

[82] K. Hammouda and F. Karray. A comparative study of data clustering techniques. 2006. (Cited on page 23.)

[83] Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, pages 600–607, New York, NY, USA, 2002. ACM. (Cited on page 23.)

[84] N. T. Ratrout. Subtractive clustering-based k-means technique for determining optimum time-of-day breakpoints. *Journal of comupting in civil engineering*, 25(5):380–387, Sep 2011. (Cited on page 23.)

[85] J. Shlens. A tutorial on principal component analysis. Technical report, Institute for Nonlinear Science, University of California, San Diego, Dec 2005. (Cited on page 27.)

[86] A. Zhou, Q. Zhang, and Y. Jin. Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE transactions on evolutionary computation*, 13(5):1167–1189, 2009. (Cited on pages 33 and 60.)

[87] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multi-objective optimization. Technical report, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH, 2001. (Cited on page 40.)

[88] Ramón Q. Sardiñas, Marcelino Rivas Santana, and Eleno A. Brindis. Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Eng. Appl. Artif. Intell.*, 19:127–133, March 2006. (Cited on page 53.)

[89] K. Deb and S. Bandaru. Higher and lower-level innovizations for knowledge discovery in practical problems. Technical report, Kanpur Genetic Algorithm Laboratory, 2012. (Cited on page 53.)