Engineering Optimization
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/geno20

Modelling the Pareto-optimal set using B-spline basis functions for continuous multi-objective optimization problems

Piyush Bhardwaj [a], Bhaskar Dasgupta [a] & Kalyanmoy Deb [b]
[a] Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, PIN 208016, India
[b] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48824, USA
Published online: 05 Sep 2013.

To cite this article: Piyush Bhardwaj, Bhaskar Dasgupta & Kalyanmoy Deb, Engineering Optimization (2013): Modelling the Pareto-optimal set using B-spline basis functions for continuous multi-objective optimization problems, Engineering Optimization, DOI: 10.1080/0305215X.2013.812727

To link to this article: http://dx.doi.org/10.1080/0305215X.2013.812727

Engineering Optimization, 2013
http://dx.doi.org/10.1080/0305215X.2013.812727

Taylor & Francis
Taylor & Francis Group

Modelling the Pareto-optimal set using B-spline basis functions for continuous multi-objective optimization problems

Piyush Bhardwaj[a], Bhaskar Dasgupta[a] and Kalyanmoy Deb[b]*

[a]*Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, PIN 208016, India;* [b]*Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA*

(*Received 27 May 2012; final version received 22 May 2013*)

In the past few years, multi-objective optimization algorithms have been extensively applied in several fields including engineering design problems. A major reason is the advancement of evolutionary multi-objective optimization (EMO) algorithms that are able to find a set of non-dominated points spread on the respective Pareto-optimal front in a single simulation. Besides just finding a set of Pareto-optimal solutions, one is often interested in capturing knowledge about the variation of variable values over the Pareto-optimal front. Recent *innovization* approaches for knowledge discovery from Pareto-optimal solutions remain as a major activity in this direction. In this article, a different data-fitting approach for continuous parameterization of the Pareto-optimal front is presented. Cubic B-spline basis functions are used for fitting the data returned by an EMO procedure in a continuous 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 EMO as input and returns the control points of the B-spline manifold representing the Pareto-optimal set. Results for several standard and engineering, bi-objective and tri-objective optimization problems demonstrate the usefulness of the proposed procedure.

Keywords: multi-objective optimization; B-spline curves and surfaces; curve fitting; surface fitting

1. Introduction

Multi-objective optimization (MOO) often entails multiple conflicting objectives for which no unique single solution can be found. Therefore, the outcome of a MOO is often a Pareto-optimal set. The image of the Pareto-optimal set in the objective space constitutes the Pareto-optimal front. 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.

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 them (Pareto 1906; Steuer 1986). In a set, the points not dominated by any other set member are called the non-dominated points of that set. The non-dominated set of the entire search space constitutes the Pareto-optimal set.

Several classical and evolutionary methods exist to solve the multi-objective optimization problem (MOOP). Methods such as the weighted sum approach (Gass and Saarty 1955; Zadeh 1963),

*Corresponding author. Email: kdeb@egr.msu.edu

the normal boundary intersection (NBI) method (Das and Dennis 1998), the weighted min–max approach (Steuer and Choo 1983), goal programming (Charnes *et al.* 1967; Charnes and Cooper 1977) and physical programming (Messac 1996) give a single point on the Pareto-optimal front in each run. Evolutionary multi-objective optimization (EMO) algorithms attempt to find multiple Pareto-optimal solutions in a single run. A survey of methods to solve MOOP in engineering is done elsewhere (Marler and Arora 2004).

Most of these algorithms 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 give a user the flexibility of choosing any point that lies on the Pareto-optimal front, one needs a continuous parameterization of the front. In this work, a continuous parameterization is achieved by data fitting of non-dominated solutions returned by a multi-objective optimization algorithm (MOOA). As a data fitting procedure is used, the parameterization obtained is an approximation of the Pareto-optimal front; however, achieving a continuous parameterization has several benefits. A continuous parameterization of the front allows the user to produce points on the front at his own discretion. 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, generating a continuous parameterization in the variable space rather than the objective space allows us to map the parameterized entity in the variable space to the objective space by using the definition of the objective function and thereby get a continuous parameterization in both the spaces. This also facilitates 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. It must, however, be noted that EMO algorithms (used in this study) don't guarantee a truly global Pareto-optimal set. The non-dominated set of points returned by the EMO algorithms undergoes a local search procedure to take them close to the true optimum. These error-corrected points are then used for the purpose of parameterization and are loosely referred to as Pareto-optimal points here. 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.

A few algorithms attempt to give a continuous representation of the Pareto-optimal front as the result of the optimization process. Steuer *et al.* (2010) proposed continuous parametric curves and surfaces for bi-objective and tri-objective problems using parametric quadratic programming, but their algorithm was restricted to convex problems only. Grosan (2003) proposed a Pareto evolutionary continuous region algorithm (PECRA) to handle continuous regions on the Pareto-optimal front. However, it was restricted to single-variable MOOPs only. Dumitrescu *et al.* (2001) proposed an evolutionary approach called the continuous Pareto set (CPS) algorithm to find a continuous Pareto-optimal set (and the corresponding Pareto-optimal front). In their algorithm, individuals in the population are closed intervals or points. CPS is also restricted to single-variable MOOPs only. Approximation of the true Pareto-optimal front by using experimentally found solutions has been attempted by Veldhuizen (1999). A response surface approximation was proposed by Goel *et al.* (2007) to approximate the Pareto-optimal front. Their method gives

one of the objectives as a function of other objectives. In their method, the relationship exists in the objective space only. Moreover, their algorithm was tested on a problem with continuous Pareto-optimal front. Several evolutionary algorithms make use of the statistical properties of the solutions of the current generation to create solutions for the next generations (Okabe *et al.* 2004; Bosman and Thierens 2005; Zhou *et al.* 2005, 2006). However, these algorithms focus on producing points on the front rather than giving a continuous picture. Dellnitz *et al.* (2005) 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. Their method does not give parameterization of the Pareto-optimal set and variations of the design variables along the Pareto-optimal front.

Finding innovative design principles has been investigated in detail in another study (Bandaru and Deb 2011). The process of 'innovization' produces relationships between objectives, variables and constraints which are found by using them as basis functions. The algorithm presented in this article generates parameteric equations for all the variables which can be used to find quantitative relationships between the variables or verify known relationships. This remains as one of the major advantages of the techniques proposed in this article. Mathematical relationships between variables reveal critical insights about the decision variables as the solutions change along the Pareto-optimal front. Although the innovization process does not give a continuous parameterization of the Pareto-optimal set, its systematic and mathematically sound approach is capable of revealing optimal relationships among variables and objectives. Bonham and Parmee (2004) proposed cluster oriented genetic algorithms (COGA) to identify high performing regions in the search space. COGA also works in the design space so as rapidly to 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 high performance design space. This study is complementary to these past studies and, due to the use of a local search and mathematical optimality conditions, the proposed procedure may be more reliable.

In this article, a data fitting method is proposed to give a 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 for smooth objectives and constraint functions, under some mild regularity conditions, that the Pareto-optimal set of an M-objective MOOP is at most an $(M-1)$-dimensional entity (termed a *Hyposurface* in this study) (Hillermeier 2001). Also, Jin and Sendhoff (2003) conjecture that the order of the polynomial 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 (Jin and Sendhoff 2003). Dellnitz *et al.* (2005) also made use of this property to approximate a Pareto-optimal set using multilevel subdivision techniques. One can have confidence in the data fitting approach because a Pareto-optimal set often consists of connected or piecewise connected points in a continuous search space (Jin and Sendhoff 2003). Even though the whole Pareto-optimal front may be disconnected, it may consist of a finite number of disconnected sections. This means that, between two close points of the Pareto-optimal set returned by the MOOA, one can find more points 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 Ehrgott and Klamroth (1997), Jin and Sendhoff (2003) and Daniilidis *et al.* (1997).

To detect disconnected parts of the Pareto-optimal set for a piecewise connected Pareto-optimal set, a clustering procedure is used. For data fitting, a cubic B-spline basis function with uniform knot vector (Mortenson 1985) is used. Many methods exist for curve fitting of the data points. However, most of the methods need some initial ordering of the data (Fang and Gossard 1995). The present algorithm assumes no order in the data.

The article is structured in the following manner. Section 2 covers the mathematical formulation of the problem and provides a brief background of the subject. Sections 3 and 4 provide details of the error reduction and the clustering steps, respectively. Section 5 gives a detailed explanation of the data fitting approach. In Section 6, results for standard and engineering design optimization problems are presented. Limitations of the algorithm are discussed in Section 7 followed by conclusions in Section 8.

2. Mathematical formulation

This section gives the mathematical formulations and a brief background necessary for the understanding of the article. MOOP in its general form is stated as:

$$\text{Minimize/Maximize} \quad f_m(\mathbf{x}), \quad m = 1, 2, \ldots, M, \tag{1}$$

$$\text{subject to} \quad g_j(\mathbf{x}) \geq 0, \ j = 1, 2, \ldots, J, \tag{2}$$

$$h_e(\mathbf{x}) = 0, \ e = 1, 2, \ldots, E, \tag{3}$$

we here f_m, g_j and h_e represent the objective functions, the inequality constraints and the equality constraints, respectively. Any variable bound must be considered as an inequality constraint in the above formulation. Part of the search space where no constraints are violated is the feasible search space (Ω) of MOOP. For a MOOP with differentiable objectives and constraints, a point in the Pareto-optimal set will necessarily satisfy the Karush-Kuhn-Tucker (KKT) conditions. The KKT conditions can be stated as follows (Kuhn and Tucker 1950; Karush 1939):

$$\begin{aligned}
&\sum_{m=1}^{M} \lambda_m \nabla f_m(\mathbf{x}) - \sum_{j=1}^{J} \mu_j \nabla g_j(\mathbf{x}) - \sum_{e=1}^{E} v_e \nabla h_e(\mathbf{x}) = \mathbf{0}, \\
&\mu_j g_j(\mathbf{x}) = 0, \quad \text{for all } j \in \{1, \ldots, J\}, \\
&\lambda_m \geq 0, \quad \text{for all } m \in \{1, \ldots, M\}, \\
&\mu_j \geq 0, \quad \text{for all } j \in \{1, \ldots, J\}, \\
&g_j(\mathbf{x}) \geq 0, \quad \text{for all } j \in \{1, \ldots, J\}, \\
&h_e(\mathbf{x}) = 0, \quad \text{for all } e \in \{1, \ldots, E\},
\end{aligned} \tag{4}$$

where λ and μ represent the Lagrange multipliers for the objectives and the constraints, respectively.

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

(Zhang *et al.* 2008). Here, this property is used to ensure that, for an M-objective problem, $(M-1)$ parameters are needed to parameterize the Pareto-optimal set.

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

DEFINITION 2.1 *An N-dimensional hyposurface in a K-dimensional space $(1 \leq N < K)$ is an entity which requires N independent parameters for its complete description.*

Thus, a one-dimensional hyposurface is a curve and a two-dimensional 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, the general term *hyposurface fitting* will be used.

Cubic B-spline basis functions with uniform knot vector are used for modelling the hyposurface. They provide a convenient matrix notation for hyposurface fitting. The control points of a B-spline hyposurface provide good global and local control. The number of control points (C_n) required to model a single segment of the B-spline hyposurface depends on the dimension of the hyposurface and the order of the B-spline basis functions used for the modelling. The total number of control points (C_{tot}) that model the hyposurface is decided by the modeller. The total number of segments (T) that comprise the whole hyposurface is directly dependent on C_{tot} and the degree of the B-spline basis function for uniform B-splines. The detailed theory of B-spline curves and surfaces is covered by Mortenson (1985).

The data fitting problem is presented here. Given a set of data points $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_W}$, where each data point is a K-dimensional vector ($\mathbf{x_w} = [x_{w1}\ x_{w2}\ \ldots\ x_{wK}]^{\mathrm{T}}$), and a function $S_t(\mathbf{u_w}, \mathbf{P})\colon \mathbb{R}^{N+C_n \times K} \rightarrow \mathbb{R}^K$, where $\mathbf{u_w} = [u_{w1}\ u_{w2}\ \ldots\ u_{wN}]^{\mathrm{T}}$ is a vector of N parameters representing the wth point on the tth segment of an N-dimensional hyposurface in space and $\mathbf{P}$ is a $C_n \times K$ dimensional matrix of control points that model the tth segment of the hyposurface, the problem of hyposurface fitting is to find the appropriate values of $\mathbf{P}$ and parameters $\mathbf{u_w}$ corresponding to each data point w such that $S_t(\mathbf{u_w}, \mathbf{P}) = \mathbf{x_w}$ for all $w \in \{1, \ldots, W\}$. The problem of hyposurface fitting inherently leads to a least-squares problem which is mathematically stated as follows:

$$\text{Minimize} \quad \sum_{w=1}^{W} (S_t(\mathbf{u_w}, \mathbf{P}) - \mathbf{x_w})^2. \tag{5}$$

The problem, stated in other words, is the minimization of the distance between the input points and the hyposurface modelled by the function S, where the parameters assigned to each data point ($\mathbf{u_w}$) 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 follows.

PROBLEM 2.2 *Given the non-dominated set of points of an M-objective MOOP in W variables, returned by some MOOA, find the control points and the parameterization that model the Pareto-optimal set of the given MOOP.*

An algorithm is proposed 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 ($\mathbf{X}$) undergo an error reduction step and a clustering step. The second part of the algorithm deals with the hyposurface fitting. Figure 1 shows the major steps of the algorithm pictorially. In the following sections each step of the algorithm is described.

Figure 1. Main steps of the algorithm.

3. KKT-error reduction

As statistical methods for approximating the Pareto-optimal set are used in this study, it is important that the input data have a low error. The lower the error, the better will be the modelled Pareto-optimal set. Non-dominated points returned by certain MOOAs like the EMO algorithms, may not necessarily be Pareto-optimal. It has been observed for several continuous problems that using NSGA-II (Deb *et al.* 2002), which is a popular EMO algorithm, the population quickly gets close to the Pareto-optimal front (or some local optima in the case of misconvergence) and all the members in the population become non-dominated after a few generations. These non-dominated members may have some Pareto-optimal as well as some sub-optimal solutions. The chances of creating better solutions decrease 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, some 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 in Goel *et al.* (2007). One possible solution suggested by them is the use of an archive to store all the non-dominated points. However, the convergence to the true Pareto-optimal front is slow. Local search of the sub-optimal points to take them closer to the Pareto-optimal front can be used to reduce the error in the solutions found by MOOA.

KKT-error (ϵ) is defined here to estimate the error in the input data. Using the KKT conditions defined in Equation (4), the KKT-error for an input point $\mathbf{x}_\mathbf{w}$ is defined as follows:

$$\epsilon(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \left\| \sum_{m=1}^{M} \lambda_m \frac{\nabla f_m(\mathbf{x}_\mathbf{w})}{\|\nabla f_m(\mathbf{x}_\mathbf{w})\|} - \sum_{j=1}^{J} \mu_j \frac{\nabla g_j(\mathbf{x}_\mathbf{w})}{\|\nabla g_j(\mathbf{x}_\mathbf{w})\|} - \sum_{e=1}^{E} v_e \frac{\nabla h_e(\mathbf{x}_\mathbf{w})}{\|\nabla h_e(\mathbf{x}_\mathbf{w})\|} \right\|^2. \tag{6}$$

In the computation of the above expression, if any of the gradients becomes zero then that particular term is ignored and the value of the rest of the expression is taken as the value of ϵ. Other KKT-error reduction methods exist and can also be used instead (Tulshyan *et al.* 2010; Haeser and Schuverdt 2009). For brevity, the equality constraints are dropped here. At a point satisfying the KKT conditions, the vectors $\lambda\nabla f$ and $\mu\nabla g$ are in equilibrium and are linearly dependent. The KKT-error term (ϵ) 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 $\nabla f_(\mathbf{x}_\mathbf{w})/\|\nabla f_m(\mathbf{x}_\mathbf{w})\|$ and $\nabla g_j(\mathbf{x}_\mathbf{w})/\|\nabla g_j(\mathbf{x}_\mathbf{w})\|$ can be used. This has the advantage of making ϵ independent of the scale of the objectives, hence permitting the use of a common threshold error value (ϵ_{thresh}) for all problems.

At a point on the Pareto-optimal front, $\epsilon = 0$. To calculate the error at a particular point $\mathbf{x}_\mathbf{w}$, the problem is to find the vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ such that ϵ is minimized. The bounds on the variables are converted into inequality constraints. Active constraints are found and their gradients are calculated. For weights, $\boldsymbol{\lambda} \neq \mathbf{0}$ and each $\lambda_m \geq 0$, normalization is done by using $\sum_{m=1}^{M} \lambda_m = 1$. The problem is mathematically defined as

$$\text{Minimize} \quad \epsilon(\boldsymbol{\lambda}, \boldsymbol{\mu}),$$

$$\begin{aligned} \text{such that} \quad & \sum_{m=1}^{M} \lambda_m = 1, \\ & \lambda_m \geq 0, \quad \text{for all } m \in \{1, \ldots, M\}, \\ & \mu_j \geq 0, \quad \text{for all } j \in \{1, \ldots, J\}. \end{aligned} \tag{7}$$

The above problem is quadratic in λ_m and μ_j. The quadratic programming (QP) method is used here to solve the above optimization problem. The minimum value of ϵ for each input point $\mathbf{x_w}$, as returned by QP, is stored and compared with a minimum threshold value (ϵ_{thresh}). Points for which $\epsilon > \epsilon_{\text{thresh}}$ are tagged as erroneous points and a local search is conducted for them, to reduce the KKT-error.

A local search procedure based on minimizing the achievement scalarizing function (ASF) (Wierzbicki 1979) is conducted here. The ASF procedure has been successfully used by Sindhya *et al.* (2009) to conduct the local search. ASF uses a reference point $\mathbf{z}$, which is the objective function vector 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 follows:

$$\begin{aligned} \text{Minimize} \quad & \max_{m=1}^{M} [wt_m(f_m(\mathbf{x}) - z_m)], \\ \text{such that} \quad & x \in \Omega, \end{aligned} \tag{8}$$

where $\mathbf{wt}$ is a vector of weights used to scalarize the function and Ω is the feasible search space. The reference point $\mathbf{z}$ helps to focus on a particular part of the Pareto-optimal front whereas the weight vector provides a finer tradeoff between the objectives leading to convergence on a particular point on the Pareto-optimal front (Deb *et al.* 2006). The $\boldsymbol{\lambda}$ vector found during the process of KKT-error calculation is used to calculate the weights for the local search procedure. $wt_m = \lambda_m/(up_m^{bd} - lw_m^{bd})$ is used as the weight for each objective function, where up_m^{bd} and lw_m^{bd} are the maximum and the minimum values, respectively, of the mth objective function in the supplied set of non-dominated points. The 'max' function in the problem definition causes the problem to be non-smooth. The problem is converted into an equivalent smooth problem (Miettinen 1999) and is solved here using the sequential quadratic programming (SQP) routine of MATLAB™. For the smooth variation of the problem refer to Miettinen (1999) and Sindhya *et al.* (2009).

Figure 2 shows the KKT-error reduction for the 'POL' problem (Deb 2001; Coello, VanVeldhuizen, and Lamont 2002). Figure 2(a) shows the points before the KKT-error reduction. In

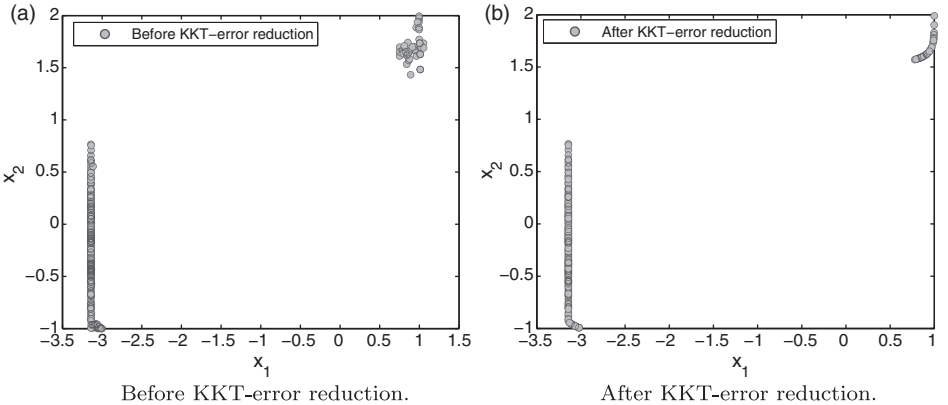


Figure 2. KKT-error reduction leads to a better organization of points in variable space.

Figure 2(a), the points on the left seem to have a lower error as a curve can be thought 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 the KKT-error reduction by the ASF procedure, points have arranged themselves in a better order for both the clusters. A regularity in the Pareto-optimal points is expected and the arrangement of points in an orderly fashion after the KKT-error reduction technique indicates the same, thereby proving the usefulness of the proposed local search approach.

4. Clustering

The Pareto-optimal set of an MOOP may be disconnected in certain problems in which the Pareto-optimal set consists of several disconnected parts. To take care of this, a clustering procedure is suggested to locate the gaps that may exist in a Pareto-optimal set. The proposed clustering process divides the corrected input data into N_d clusters, where N_d is the number of disconnected Pareto-optimal regions that may not be known a priori. The automated procedure should produce exactly the same number of clusters as the number of disconnected Pareto-optimal regions. For example, for connected data the procedure should not divide the data into more than one cluster; for a problem with two disconnected Pareto-optimal regions, exactly two clusters should be returned.

As the number of clusters that exist in the Pareto-optimal data is unknown, the clustering process is carried out in three steps: (i) an estimate of the number of clusters is approximated, (ii) K-means clustering is carried out based on the estimate, and (iii) clusters that are close to each other are updated. Figure 3 shows this three-step process pictorially.

First, the subtractive clustering method (Chiu 1994) is used to make an initial estimate of the number of clusters and cluster centres. Subtractive clustering gives only an approximate estimate of the number of clusters in the data set and doesn't associate a cluster with every data point. Therefore, the number of clusters and the cluster centres returned by subtractive clustering are fed as an estimate of the number of cluster and initial centre estimates, respectively. Thereafter, in the second step, each point is associated with a particular cluster. Use of such a hybridized clustering procedure to obtain improved results has been advocated by several researchers (Ratrout 2011; Verma *et al.* 2012). The above clustering procedure may still produce more than the expected number of clusters. Thereafter in the third step, a cluster correction procedure 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 on the data. One such situation is shown in Figure 4(a). Here, the initial clustering step gives eight clusters while it is clear from the plot of the data that there exists only one big cluster. The second step will associate each point with a different cluster, as shown in the figure. The cluster correction (third) step joins all the clusters together to indicate that there is a single large cluster associated with the supplied points (Figure 4(b)). On the 'POL' problem, Figure 5 shows the outcome of the proposed three-step clustering approach. Two clusters are correctly identified by the proposed clustering approach.

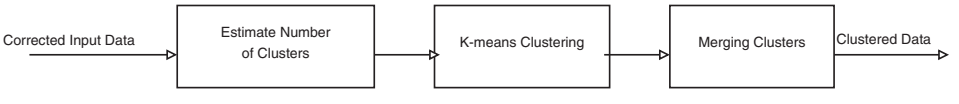
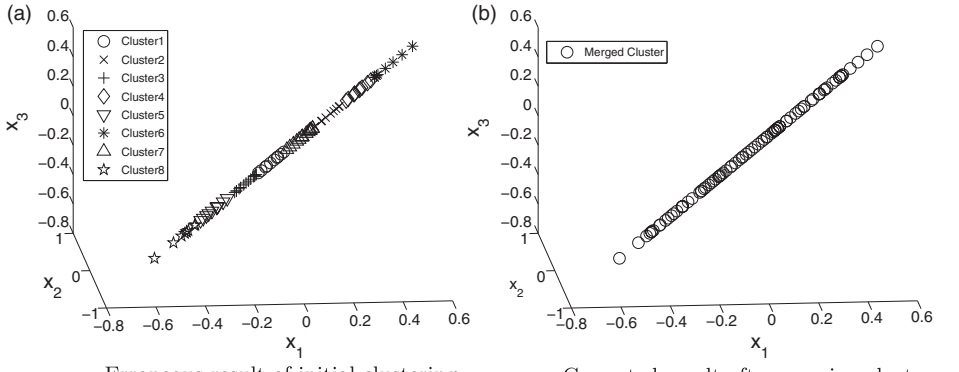


Figure 3. Pictorial representation of the three-step clustering process.



Erroneous result of initial clustering. Corrected result after merging clusters.

Figure 4. Erroneous and corrected clustering on a hypothetical problem.

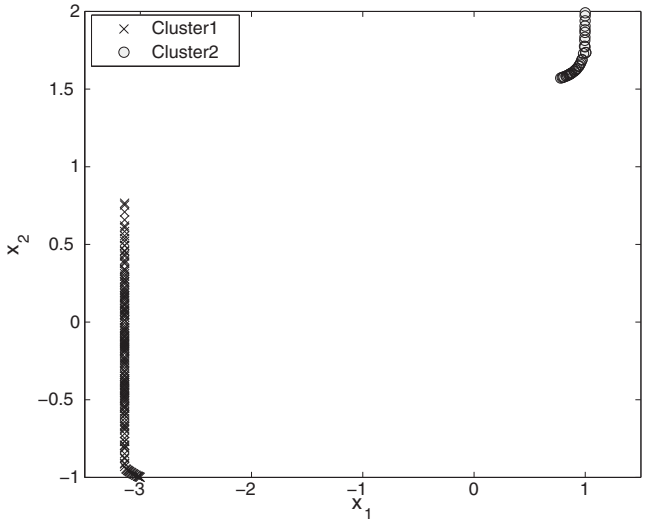


Figure 5. Two clusters in variable space correctly returned by the proposed three-step clustering process for the 'POL' problem.

5. Hyposurface fitting

In Section 2, the task of hyposurface fitting was discussed and a method was proposed based on solving the optimization problem given in Equation (5). For uniform open cubic B-spline data fitting, the objective $S_t(\mathbf{u_w}, \mathbf{P})$ can be defined by using a convenient matrix notation (Mortenson 1985). For example, the tth segment of a one-dimensional hyposurface can be written as follows:

$$S_t(\mathbf{u_w}, \mathbf{P}) = \begin{bmatrix} u_{w1}^3 & u_{w1}^2 & u_{w1} & 1 \end{bmatrix} \begin{bmatrix} -1/6 & 3/6 & -3/6 & 1/6 \\ 3/6 & 6/6 & 3/6 & 0/6 \\ -3/6 & 0/6 & 3/6 & 0/6 \\ 1/6 & 4/6 & 1/6 & 0/6 \end{bmatrix} \begin{bmatrix} \mathbf{p_{t-1}} \\ \mathbf{p_t} \\ \mathbf{p_{t+1}} \\ \mathbf{p_{t+2}} \end{bmatrix}, \tag{9}$$

where $w \in \{1, \ldots, W\}$ and $t \in \{1, \ldots, T\}$. Similar matrix notation can also be written for a two-dimensional hyposurface.

Given the data, **u** and **P** are the unknowns for the minimization problem (Equation 5). The problem is linear in control points **P** and nonlinear in parameter **u**. Starting with some initial value for **u**, an iterative procedure is used to find the unknowns that minimize the error value. The iterative procedure consists of following two steps.

(1) Find the control points: for fixed values of the parameters find the optimal control points that minimize the error.
(2) Update the parameters: keeping the control points found in the previous step fixed, find the parameter values that minimize the error.

Both steps are repeated until either the error is reduced below the fixed threshold or a pre-specified maximum number of iterations is reached. If the error is large, the data are subdivided into several nodes using a 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 process, with acceptable data fitting leading to leaf node creation. The procedure for tree building goes as follows. Initially all the data are 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 child nodes. For every child node, the above process is repeated recursively. At the end of the recursion, the hyposurface models stored in every leaf node are joined together to give a continuous hyposurface. This process of hyposurface fitting is performed independently for each cluster identified in the clustering step. See Figure 6 for an example of a two-way tree. Here, the initial fit for the data in the root node had an error greater than the threshold. Therefore, the data were 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 the child 2 node, fitting was still not good and subdivision of the data into a further two child nodes—child 3 and child 4—takes place. Fitting for these two nodes is satisfactory and they become the leaf node.

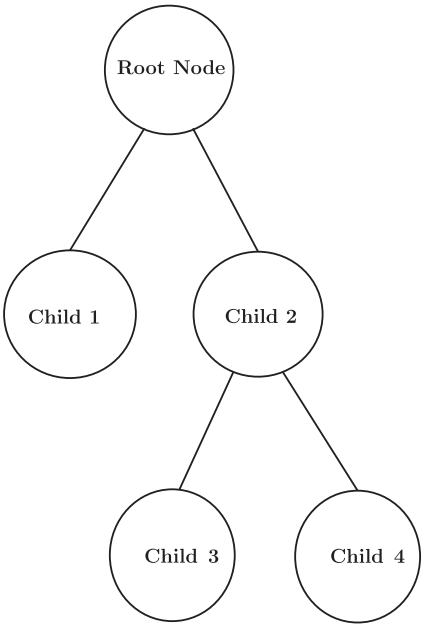


Figure 6. Two-way tree.

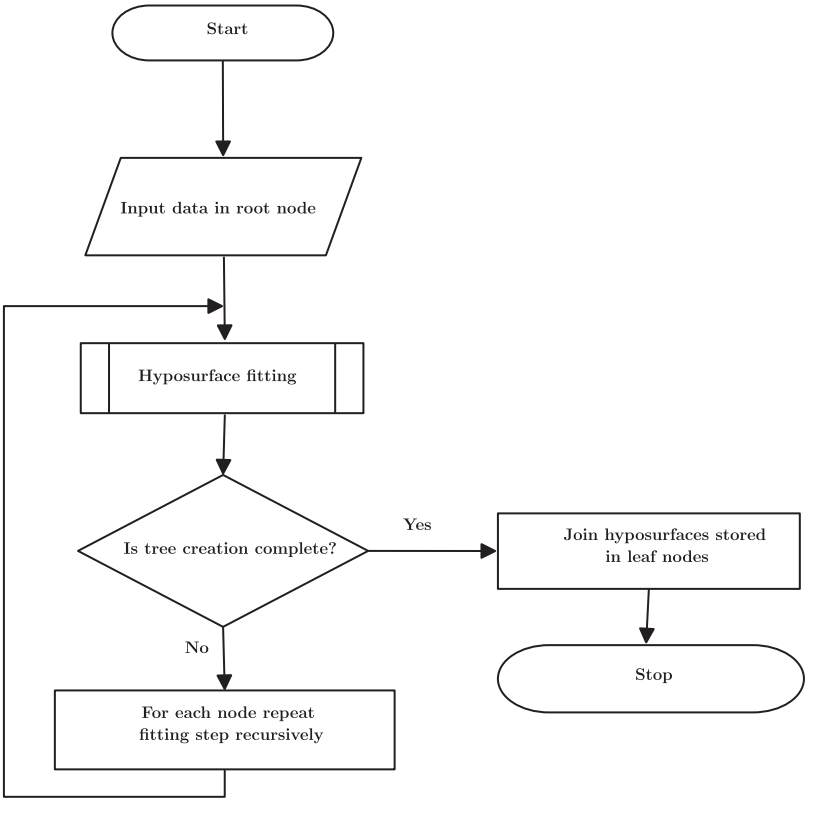


Figure 7. Flowchart of the proposed algorithm.

Figure 7 shows a flowchart of the proposed algorithm. For the data in each node, several steps are carried out for hyposurface fitting. Figure 8 shows a flowchart of these steps. A brief description of these steps follows.

5.1. *Finding outliers*

The purpose of this subroutine is to find those data members in a node that are dramatically different from the rest of the data members in the same node. An archive of outliers is maintained, the members of which are added to the members of the current node. After this, the centroid of the data is found. Data members that 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 that are outliers for one node may represent valid members for other nodes. This process ensures that no information is lost.

5.2. *Principal component analysis*

The next step is to perform a principal component analysis (PCA) of the data members in the node. The aim of the 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. It can be mathematically shown that the principal

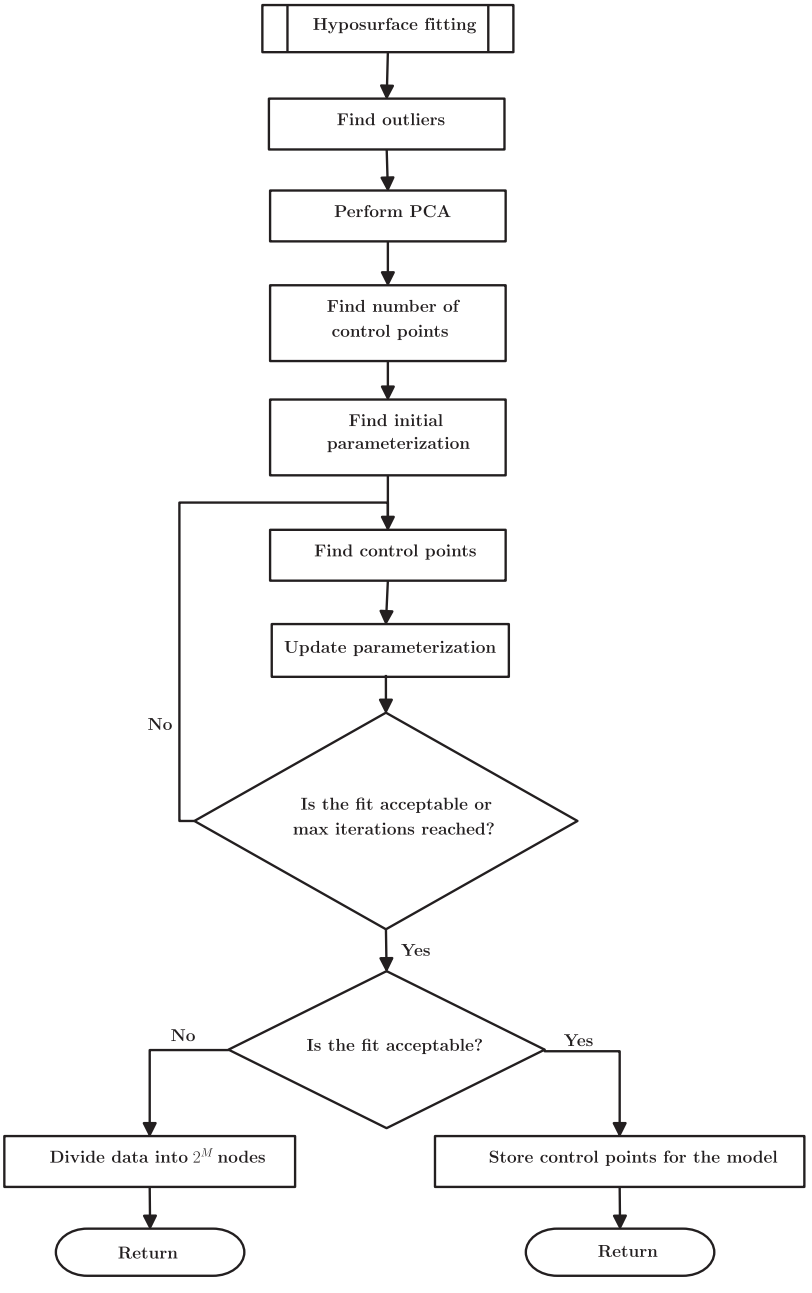


Figure 8. Flowchart showing the hyposurface fitting procedure.

components of the data matrix $\mathbf{X}$ are the eigenvectors of matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$. The principal directions are sorted according to the corresponding eigenvalues in descending order and the data are 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 the boundary points are well captured by the MOOA, as the boundary points are assigned an initial parameter value of zero. If the boundary points are not well captured, some parts of the Pareto-set may be lost.

It is important that all the neighbouring nodes have similar orientation of direction returned by the PCA, as this helps maintain proper continuity in the parameterization necessary for joining the modelled segments with C^2 continuity in the last step of joining hyposurfaces stored in leaf nodes. 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, the direction of the eigenvector for the child node is flipped. This process ensures that the principal directions of the parent and the children are similarly oriented.

PCA is also used to decide the number of control points (C^m_{node}, $m \in \{1, \ldots, M\}$) in each of the M principal direction for modelling the B-spline hyposurface. The total number of control points (C_{node}) is 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 the proportion of the eigenvalues associated with the corresponding eigenvectors.

5.3. *Iterative hyposurface fitting*

The steps for finding the control points and parameter update are iteratively repeated to reduce the error, as described earlier. The problem of finding the control points for fixed parameterization is a linear least-squares problem with linear constraints and is solved using the 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 nonlinear 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 is 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.

5.4. *Joining hyposurface segments*

The k-way tree creation process results in discontinuous segments modelled using B-spline basis functions stored in the leaf nodes of the tree (as shown in Figure 9(g)). Next step is to join together these segments to create a C^2 continuous hyposurface. The two step iterative hyposurface fitting approach is used again, this time for modelling the whole hyposurface in one go. To model a hyposurface using this iterative procedure one needs to find a good initial parameterization as well as figure out the total number of control points necessary for modelling the whole hyposurface. A parameterization as well as the number of control points required for modelling each of the disconnected segments was found in the previous step of hyposurface fitting for data in each leaf node and is stored in the leaf nodes of the k-way tree. The sum of the number of control points required to model each segment serves as the number of control points required to model the complete hyposurface. To join the segments, it is necessary to find the order in which the segments are arranged. The position of leaf nodes in the tree is used to find the order in which the segments are arranged. Scaling the parameterization of the data stored in each node in the required range, based on the position of the node in the tree, an initial parameterization for modelling the complete data is worked out. The iterative procedure for error reduction is now used to find the control points, which model the whole hyposurface. The iterations are carried until the threshold error is not satisfied. At the end of this process one can get a C^2 continuous hyposurface whose parametric equation is known.

Figure 9 shows the steps of the algorithm for the data points in the first cluster for the 'POL' problem. Figure 9(a) shows the data in the root node. No outliers were found in the data. Figure 9(b)

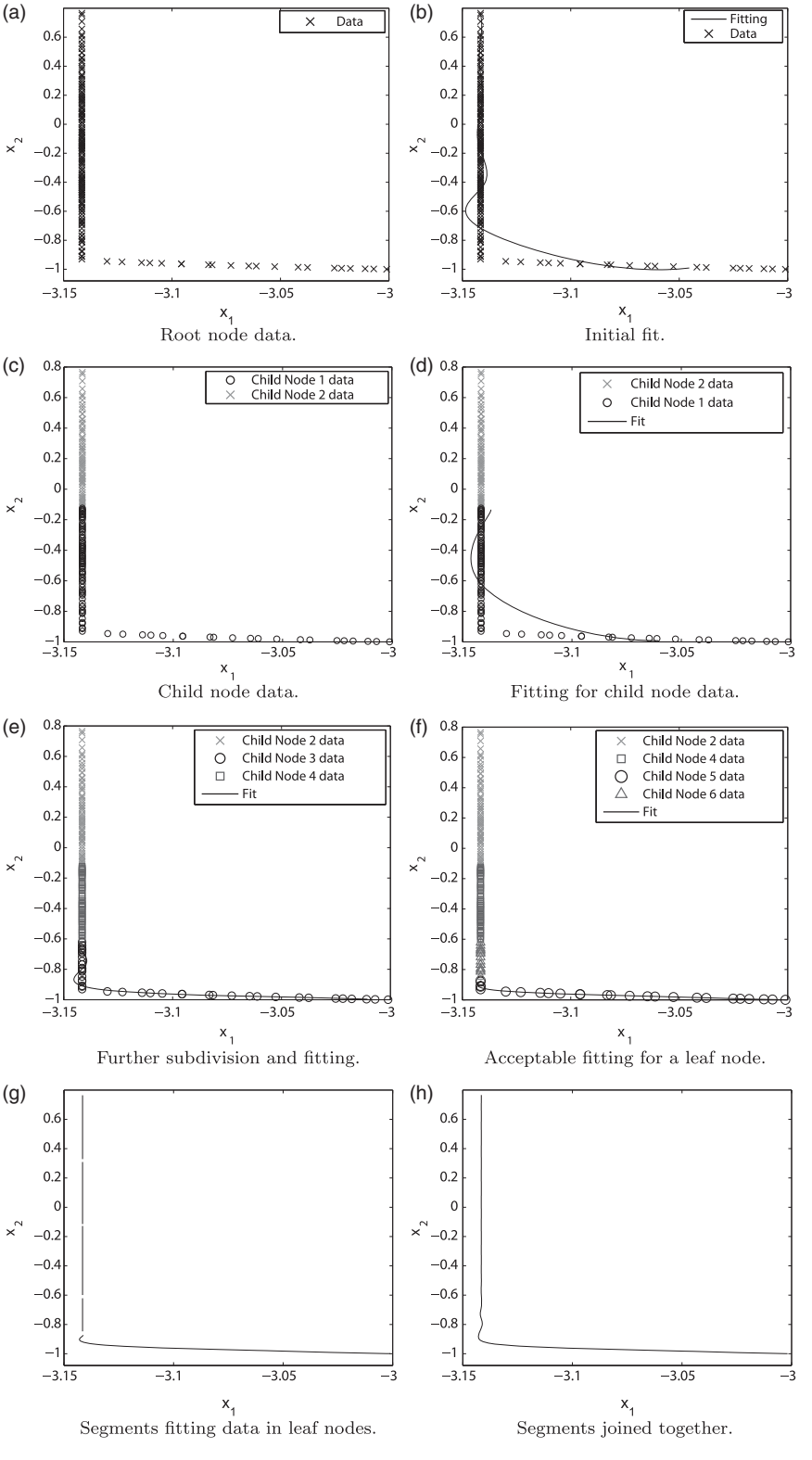


Figure 9. Fitting algorithm illustrated in variable space.

shows the initial fitting. As the error value for this fit is greater than the threshold error value, the data are divided into two nodes. Figure 9(c) shows the subdivided data in the children nodes. Figure 9(d) 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 are subdivided until an acceptable fit is found in Figure 9(f). Figure 9(g) 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 9(h).

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 are considered. The following standard MOOPs are used: FON, ZDT2, ZDT3, DTLZ2, DTLZ7 and OSY (Deb 2001; Coello, VanVeldhuizen, and Lamont 2002). A two-bar truss design problem is also considered to show the efficacy of the method on engineering design problems. For all of these problems, both the Pareto-optimal front and the Pareto-optimal set are $(M-1)$-dimensional hyposurfaces.

The input points are generated by NSGA-II, although another EMO (Deb 2001; Coello, VanVeldhuizen, and Lamont 2002) or a classical generating method (Miettinen 1999) can also be used for the purpose. NSGA-II is run with a population size of 400 for 400 generations and the results are passed to the proposed procedure as input data. The threshold value of the KKT-error is fixed at 10^{-4}. If the KKT-error is greater than this threshold, a local search is conducted using the SQP routine of MATLAB™. The maximum number of function evaluations for SQP is fixed at 1000. The gradients for the KKT-error calculation are calculated exactly for the FON and OSY problems; for the other problems, the gradients are estimated using the forward difference method. For the subtractive clustering process, the radii of influence are fixed at 0.10. For merging clusters, the threshold distance between the two clusters is taken as 3% of the length of the 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 threshold error value for dividing the data recursively into 2^M child nodes is taken to be 10^{-4}. The results for the problems mentioned above are presented in the following subsections.

6.1. *FON*

FON is a two-objective problem with n-variables. The results for $n = 3$ are shown for better visualization of solutions in the objective and variable space. Figure 10 shows the results for the FON problem. Figure 10(a) shows the input data set obtained by the NSGA-II procedure, which is updated to Figure 10(b) by the KKT-error reduction technique. The KKT-error reduction routine reduces the error in the data and now the data are better organized for curve fitting. This is demonstrated in Figure 11. Figures 10(c) and 10(d) show the Pareto-optimal set and the Pareto-optimal front in the variable and the objective space, respectively. The parametric equation of the Pareto-optimal set is obtained as follows:

$$x_1(u) = x_2(u) = x_3(u) = 1.14u - 0.577 \quad \text{for } u \in [0, 1]. \tag{10}$$

The Pareto-optimal front is obtained by mapping the fitted curve to the objective space. Parameter values of points in the Pareto-optimal set are shown above their respective image in the objective

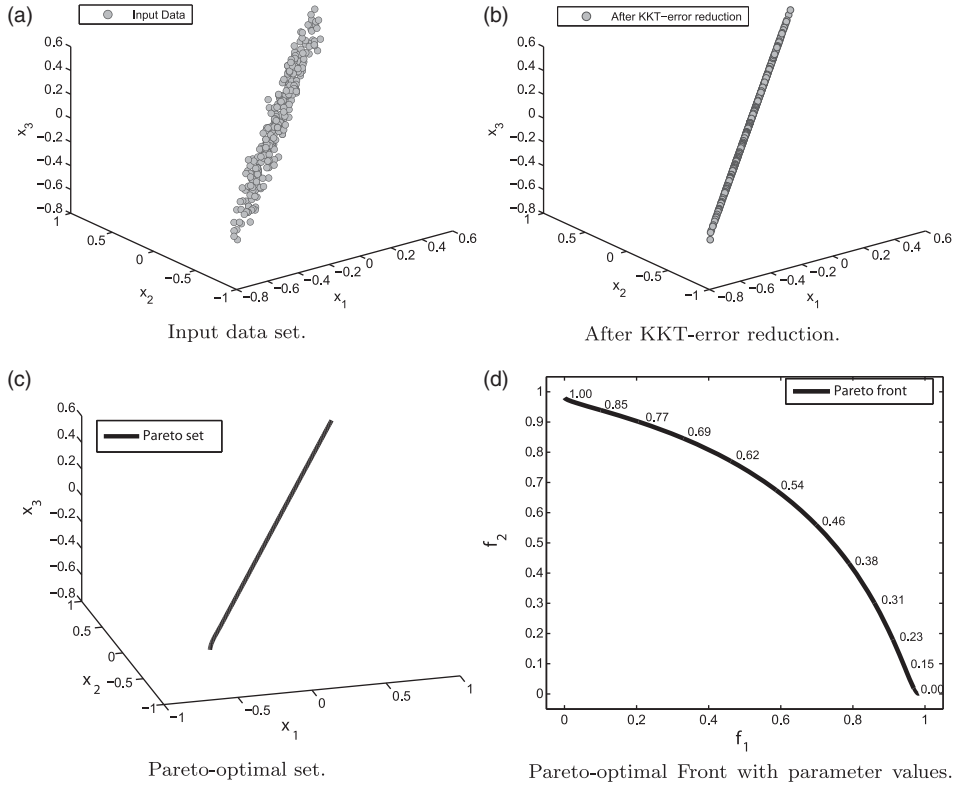


Figure 10. Results for three-variable FON problem.

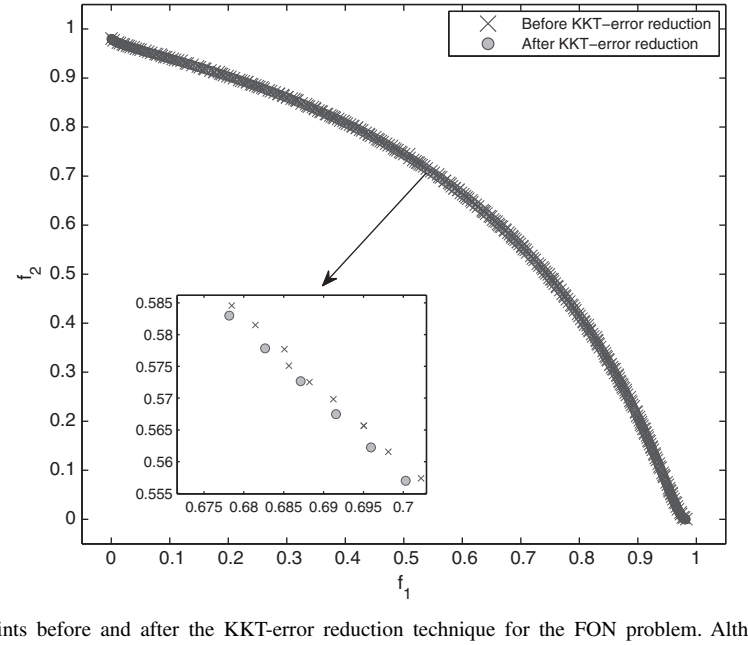


Figure 11. Points before and after the KKT-error reduction technique for the FON problem. Although marginal, improvements can be seen through the inset figure.

space in Figure 10(d). 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. Also, the parameter values allow the user to choose any point on the front.

6.2. *ZDT2*

ZDT2 has a non-convex Pareto-optimal front. First the results of a three-variable ZDT2 problem are shown followed by those for a 30-variable problem. Figure 12(a) shows the initial input and the corrected data for a three-variable ZDT2 problem. The '×' markers show the corrected data. It can be seen that the data after error correction are much more organized and lie on a straight line. Figure 12(b) 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 zero to one. These values are the same as those shown by Deb (2001). The parametric equation of the curve is given as follows:

$$x_1(u) = \begin{cases} -3.293u^3 + 1.098u^2 + 0.883u + 0.014, & u \in [0, 0.143), \\ 0.926u^3 - 0.686u^2 + 1.146u + 0.002, & u \in [0.143, 0.286), \\ -0.366u^3 + 0.421u^2 + 0.829u + 0.032, & u \in [0.286, 0.428), \\ 0.211u^3 - 0.321u^2 + 1.148u - 0.014, & u \in [0.428, 0.571), \\ -0.211u^3 + 0.404u^2 + 0.733u + 0.065, & u \in [0.571, 0.714), \\ 0.274u^3 - 0.637u^2 + 1.477u - .112, & u \in [0.714, 0.857), \\ -0.846u^3 + 2.244u^2 - 0.993u + 0.594, & u \in [0.857, 1], \end{cases} \tag{11}$$

$$x_2(u) = 0, \quad u \in [0, 1],$$

$$x_3(u) = 0, \quad u \in [0, 1].$$

Figure 12(c) shows the image of the Pareto-optimal set in the objective space. Figure 12(d) shows the Pareto-optimal front for a 30-variable ZDT2 problem. The Pareto-optimal set for a 30-variable ZDT2 problem cannot be shown on a plot due to the high number of dimensions of the variable space. However, it can be seen that the Pareto-optimal front of a 30-variable ZDT2 problem is similar to its three-variable counterpart.

6.3. *ZDT3*

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 13(a) shows the initial input and the corrected data for the ZDT3 problem. Several points with high error values have been corrected and the corrected data are better organized for all the five disconnected Pareto-optimal sets. The clustering routine identified five clusters in the corrected data. Figure 13(b) shows the curve fitting for the Pareto-optimal set. Curve fitting is carried out, independently, for each of the clusters, as a result of which five different parameterizations are obtained. Figure 13(c) shows the image of the Pareto-optimal set in the objective space. From Figure 13(b) 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 lead to a disconnected Pareto-optimal front. The Pareto-optimal front with parameter values for the 30-variable problem is shown in Figure 13(d).

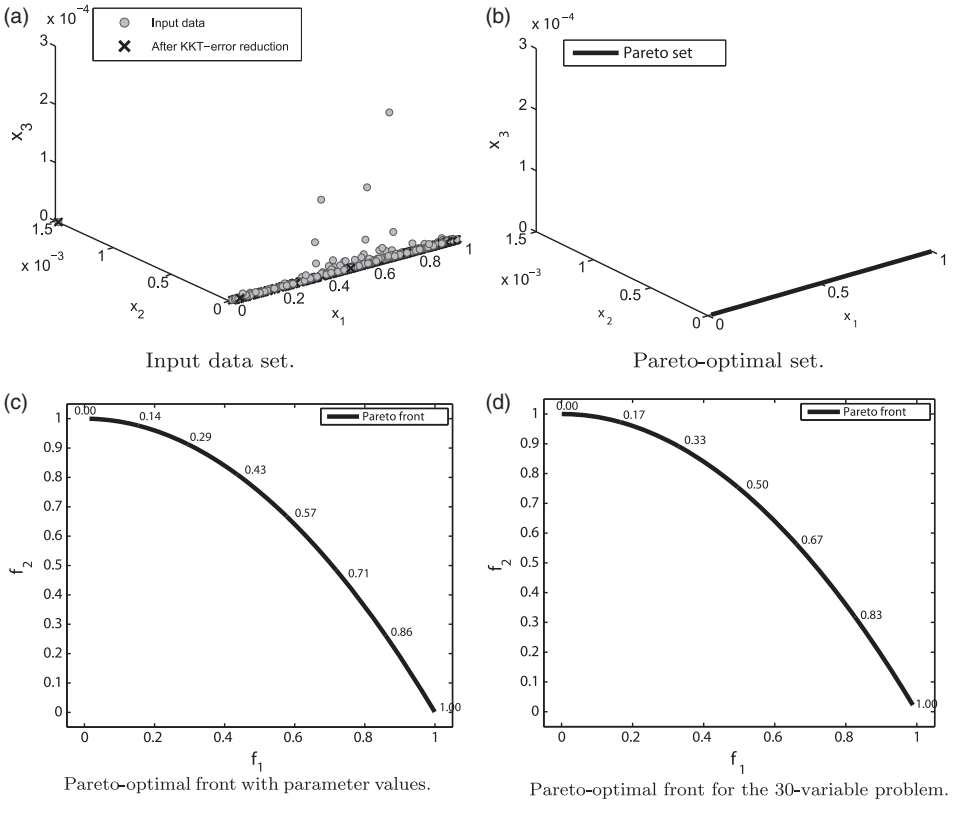


Figure 12. Results for the ZDT2 problem.

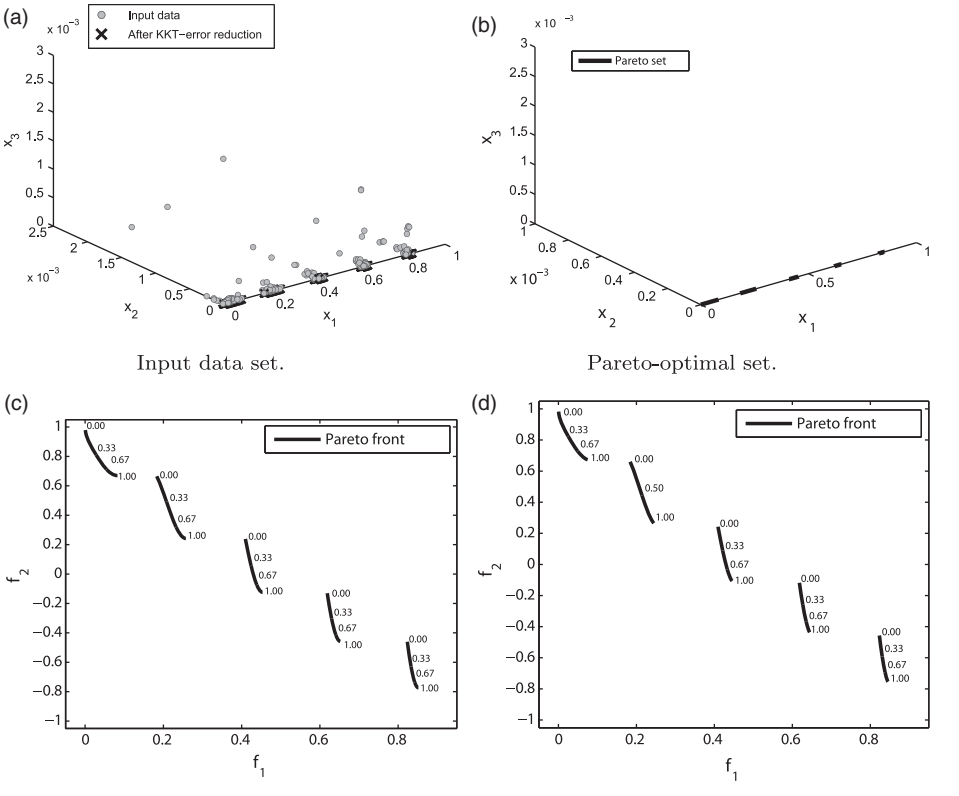
6.4. *DTLZ2*

A three-objective DTLZ2 problem has a spherical Pareto-optimal front. The Pareto-optimal set is a plane with $x_3 = 0.5$ and the Pareto-optimal front is part of a sphere with $f_1^2 + f_2^2 + f_3^2 = 1$. Results for the problem using NSGA-II are shown in Figure 14. NSGA-II's crowding distance operator is not adequate for three or more objective problems. A clustering operator (Deb *et al.* 2005) can be used instead, but here the performance enhancement issue of the NSGA-II procedure is not addressed. The KKT-error reduction routine reduces the error from 0.0011 to 8×10^{-10}. Figures 14(a) and 14(b) show the Pareto-optimal set and Pareto-optimal front for the problem, respectively.

6.5. *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.

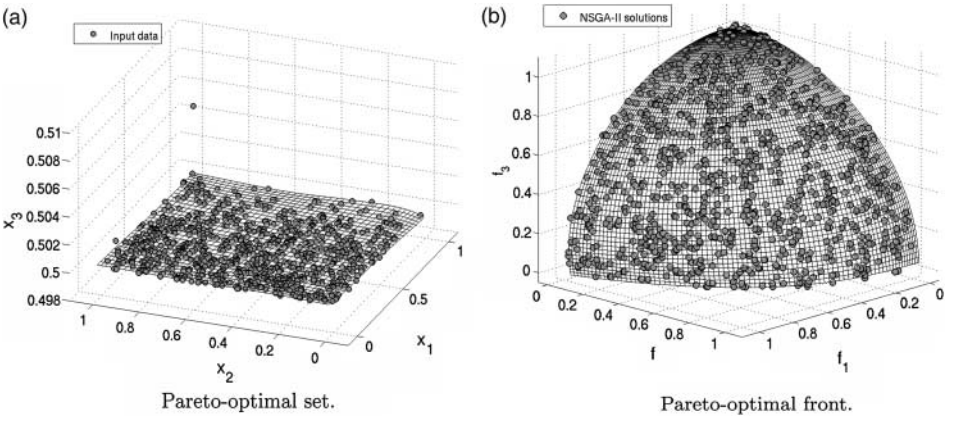
Figure 15 shows the results for a three-objective and three-variable DTLZ7 problem. There are four disconnected Pareto regions here which have been identified by the clustering procedure. The average KKT-error for the input points reduced from 8.1×10^{-3} to 1.7×10^{-6} after the error reduction routine. The fitted Pareto-optimal set and the Pareto-optimal front along with the input points are shown in Figures 15(a) and 15(b), respectively. It can be noticed that some of the points in the Pareto-optimal set have high error. Owing to this, the surface fitting is not exact and the generated surface is not planar.



Input data set. Pareto-optimal set.

Pareto-optimal front for the three-variable problem. Pareto-optimal front for the 30-variable problem.

Figure 13. Results for the ZDT3 problem.



Pareto-optimal set. Pareto-optimal front.

Figure 14. Results for the three-variable DTLZ2 problem.

6.6. *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.

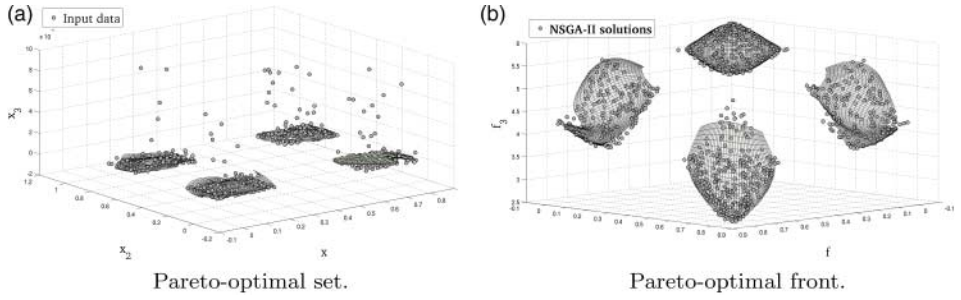
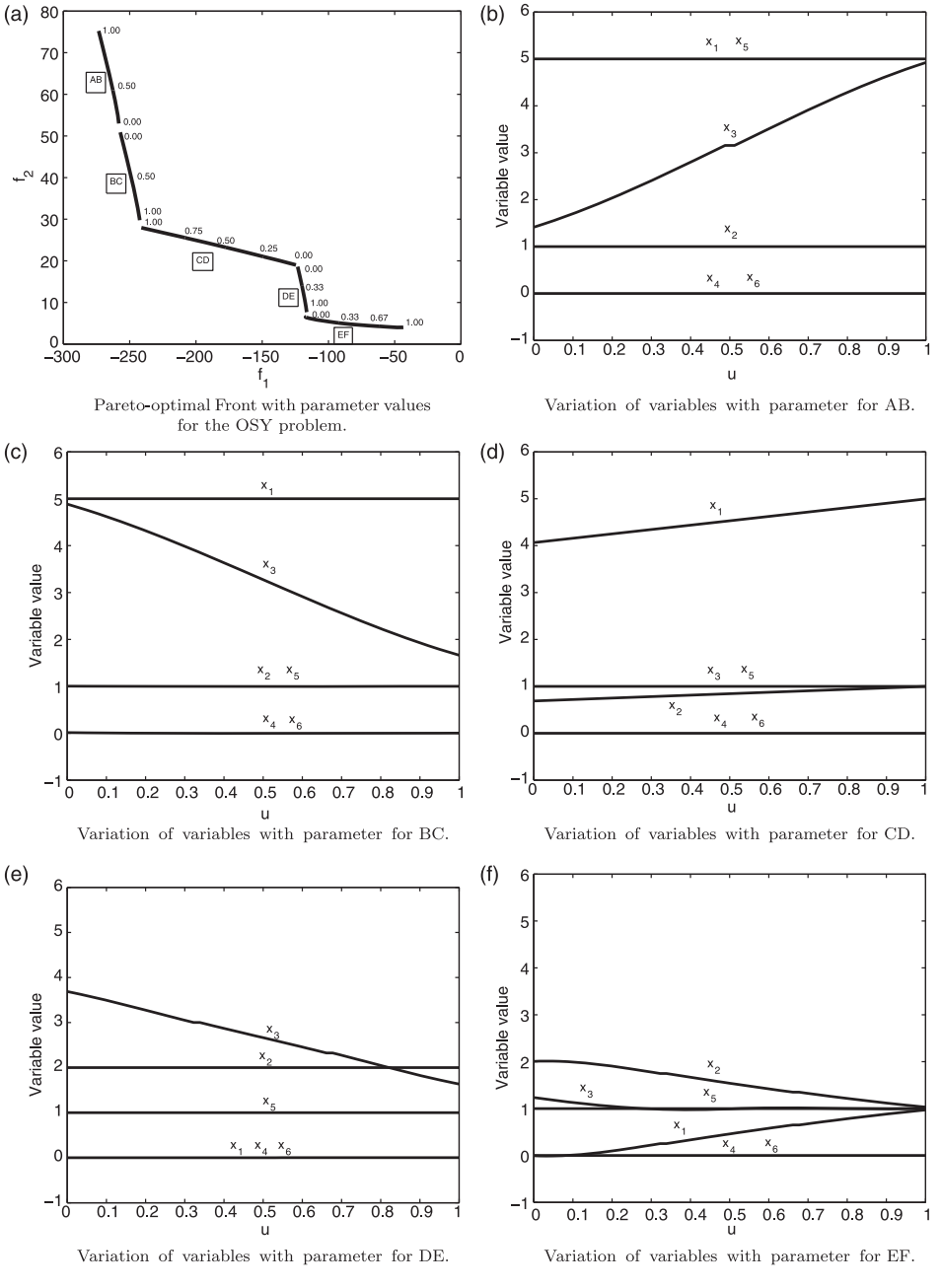


Figure 15. Results for the three-variable DTLZ7 problem.

Figure 16(a) show the Pareto-optimal front for the OSY problem. 'AB', 'BC', 'CD', 'DE' and 'EF' show the five parts of the front. The data are 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. As the B-spline used is cubic, one can get cubic equations as relationships; however, the coefficient for some of the terms may be negligible. These relationships correct to two decimal places are shown in Table 1. The values of variables obtained by the proposed method are very close to those tabulated elsewhere (Deb 2001). There are small differences which are described below.

To show the variation of variables with the parameter u, graphs between variable values and parameter values are plotted for each of the five parts of the Pareto-optimal front. Figure 16 shows these plots. For AB, as one moves from a parameter value of zero to one, x_1, x_2, x_4, x_5 and x_6 remain constant at five, one, zero, five and zero, 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 five, one, zero, one and zero, 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 for $x_1(u)$ 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 from a previous study by Deb (2001). For DE, it can be noticed that x_1, x_2, x_4, x_5 and x_6 are constant at zero, two, zero, four and zero, respectively. Value of x_3 varies between 1.63 and 3.69. For the part EF of the front, from parameter values zero 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 zero, one and zero respectively. For the rest 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 images of each other about the line $x = 1$. This makes $x_2 = 2 - x_1$, which was also found in Deb (2001). Variable x_3 is nearly constant and variable x_5 is constant at a variable value of one while x_4 and x_6 have a constant value of zero. Some interesting properties of optimal points can be inferred by this procedure which can be useful to the decision maker. Some inferences are enumerated as follows.

(1) $x_4 = x_6 = 0$ along the front.
(2) x_5 is constant at five for part AB and at one 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, $x_2 = 2 - x_1$.



Pareto-optimal Front with parameter values for the OSY problem.

Variation of variables with parameter for AB.

Variation of variables with parameter for BC.

Variation of variables with parameter for CD.

Variation of variables with parameter for DE.

Variation of variables with parameter for EF.

Figure 16. Variation of six variables with parameter u for the OSY problem.

6.7. *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 (Deb 2001). Figure 17 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

Table 1. Relationships between variables inferred from the plots for the OSY problem.

Front part	x_1	x_2	x_3	x_4	x_5	x_6
AB	5	1	$(1.41, \ldots, 5)$	0	5	0
BC	5	1	$(1.66, \ldots, 4.88)$	0	1	0
CD	$(4.06, \ldots, 5)$	$(0.69, \ldots, 1)$	1	0	1	0
DE	0	2	$(1.63, \ldots, 3.69)$	0	1	0
EF($u \in [0, 0.1]$)	$(0, \ldots, 0.01)$	$(2.00, \ldots, 1.99)$	$(1.24, \ldots, 1.14)$	0	1	0
EF($u \in (0.1, 1]$)	$(0.01, \ldots, 0.97)$	$(1.99, \ldots, 1.03)$	$(1.14, \ldots, 1)$	0	1	0

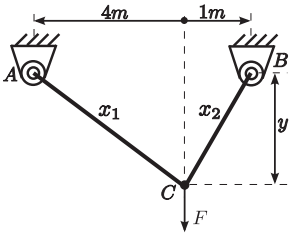


Figure 17. A two-bar truss.

the horizontal distance between the hinges is 5 m. 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.

$$\begin{aligned} \text{Minimize} \quad & f_1(\mathbf{x}, y) = x_1\sqrt{16 + y^2} + x_2\sqrt{1 + y^2}, \\ \text{minimize} \quad & f_2(\mathbf{x}, y) = \max(\sigma_{\text{AC}}, \sigma_{\text{BC}}), \\ \text{subject to} \quad & \max(\sigma_{\text{AC}}, \sigma_{\text{BC}}) \leq 10^8, \\ & 1 \leq y \leq 3, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{12}$$

where

$$\sigma_{\text{AC}} = \frac{F\sqrt{16 + y^2}}{5yx_1}, \quad \sigma_{\text{BC}} = \frac{4F\sqrt{16 + y^2}}{5yx_2}.$$

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 alternative formulation is considered. The constraint can be split into two different constraints, $\sigma_{\text{AC}} \leq 10^8$ and $\sigma_{\text{BC}} \leq 10^8$. 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 alternative smooth formulation of the 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.

$$\begin{aligned} \text{Minimize} \quad & \epsilon_1 + \epsilon_2, \\ \text{subject to} \quad & w_1(f_1(\mathbf{x}, y) - z_1) \leq \epsilon_1, \\ & w_2(\epsilon_2 - z_2) \leq \epsilon_1, \end{aligned}$$

$$\begin{aligned} &\sigma_{AC} \leq \epsilon_2, \\ &\sigma_{BC} \leq \epsilon_2, \\ &1 \leq y \leq 3, \\ &\mathbf{x} \geq \mathbf{0}, \\ &\epsilon_1 \leq 0. \end{aligned} \tag{13}$$

This formulation helps 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 18. The original formulation doesn't lead to any considerable correction in the data while the new formulation leads to a substantial reduction in the error.

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 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 are normalized to the interval $[0, 1]$. The data are rescaled back to show the results. Figure 19 shows the results for the above problem. The input data set and the data set after error correction are shown in Figure 19(a). Figure 19(b) shows the Pareto-optimal set for this problem. The mapped Pareto-optimal front is shown in Figure 19(d). It is very close to the non-dominated points shown in Figure 19(c) returned by the MOOA.

The variation of the variables with parameter values is shown in Figure 20. Notice that after parameter values of around 0.89 there is a sudden change in the behaviour 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 values of x_1 and x_2 rise steadily. It can also be noticed that variation of x_1 and x_2 with the 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 the cross-sectional area of BC (x_2) is always greater than the cross-sectional area of AC (x_1) for all the optimal solutions. There is little change in the value of x_2 for parameter values greater than 0.89; x_2 reaches its maximum value of 0.01 and remains constant there. Thus, for the variable x_2, the bound becomes active for $u > 0.89$. This may be the reason for the sudden change in behaviour of the variables at $u = 0.89$. The value of the variable x_1 decreases slightly for $u > 0.89$. However, a drastic change is seen for the variable y in the same range. This means that, beyond the parameter u value of 0.89, the change in the optimal value of both functions is mostly caused by the change in the value of y.

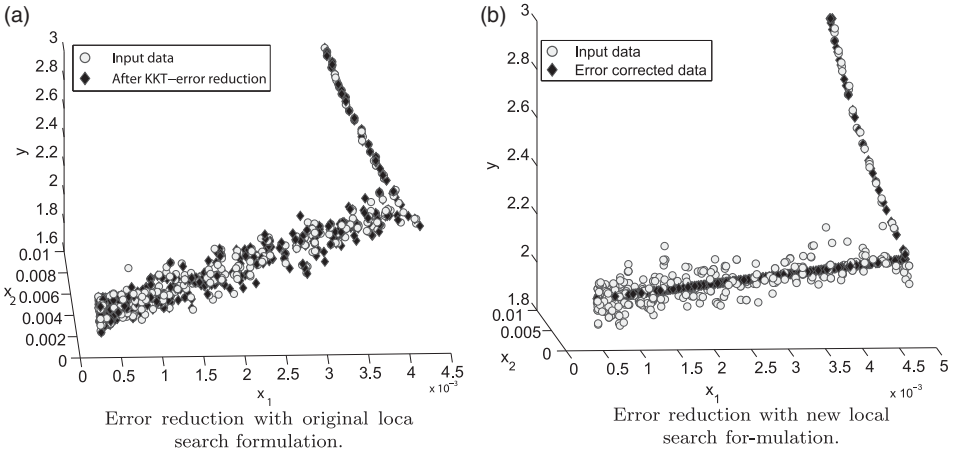


Figure 18. Error reduction results for the two-bar truss problem.

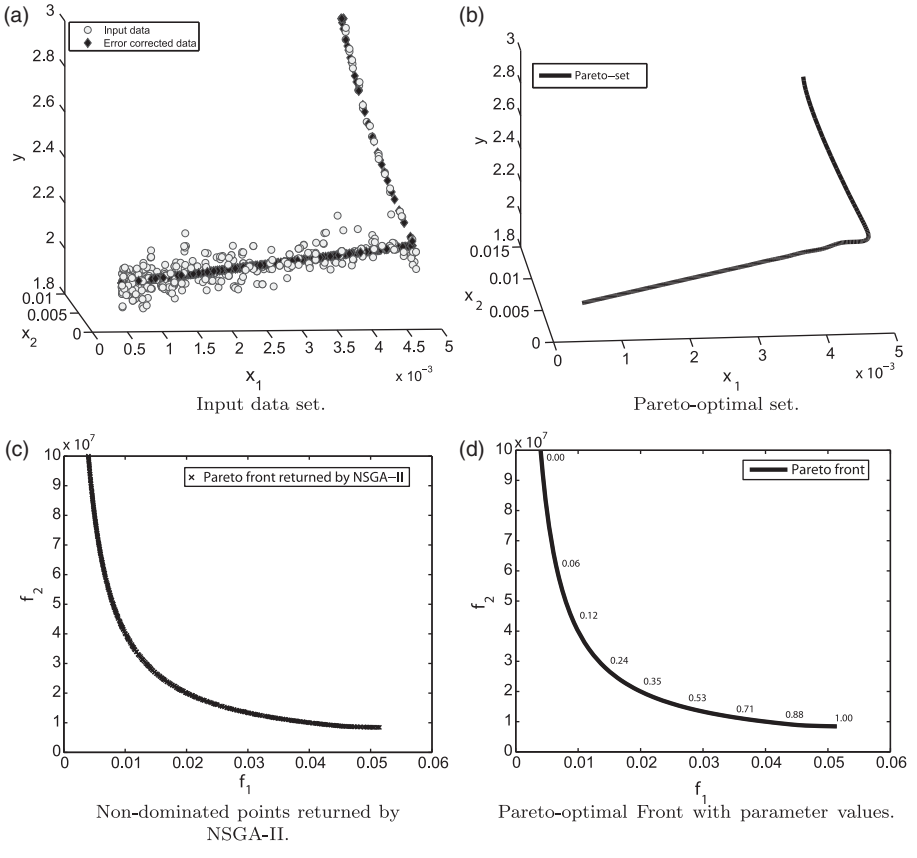


Figure 19. Results for the three-variable truss problem.

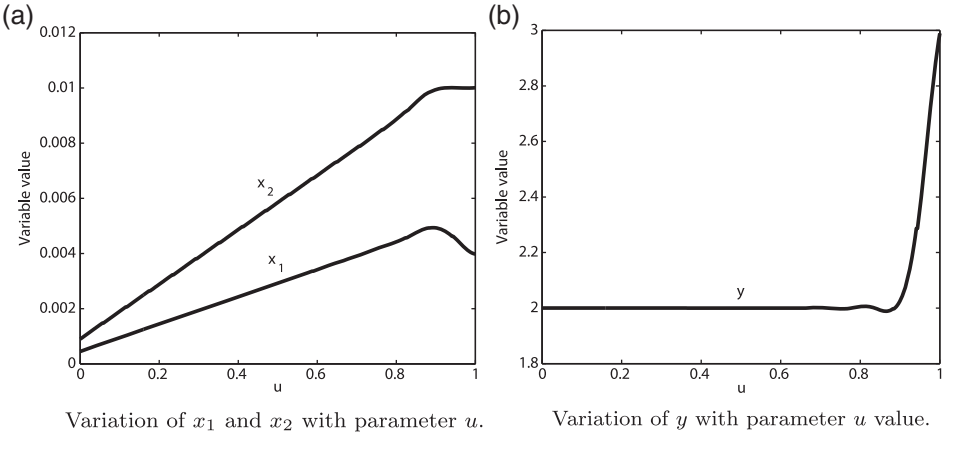


Figure 20. Variation of three variables with parameter *u* for the truss problem.

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

Several interesting inferences that can be drawn from the plots are enumerated below. These observations are similar to observations made in Bandaru and Deb (2011), but they can be found using a different approach. The relationships that exist between the variables as found in Bandaru and Deb (2011) are $y = 2$ and $x_2 = 2x_1$.

(1) x_2 is greater than x_1 throughout the Pareto-optimal front.
(2) The relationship $x_2 = 2x_1$ can be confirmed from Figure 20(a).
(3) Variable y, remains constant at $y = 2$ for parameter $u < 0.89$ and rises rapidly for higher u values.
(4) After a parameter u value of around 0.89, x_2 reaches its upper bound and remains constant there.
(5) $\sigma_{AC} = \sigma_{BC}$ throughout the front.

7. Limitations of the algorithm

As a statistical model is used, it cannot be said with certainty that all the points on the modelled hyposurface are the 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 are dense, more confidence in the modelled hyposurface can be obtained as sudden changes in the Pareto-optimal front are not expected. However, with sparse data one can get an erroneous Pareto-optimal front even though the least-squares distance error is sufficiently low. In the proposed 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 situations, the points modelled 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, the constraints of any point generated using this model must be checked.

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. In such cases, the sub-differential based KKT-error minimization procedure due to Deb et al. (2007) can be adopted.

Here, only problems on which regularity conditions are met are considered. 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 exists a class of problems for which this may not be so and the Pareto-optimal set can be a lower dimensional manifold for an $(M - 1)$-dimensional Pareto-optimal front—the DTLZ6 problem (Deb et al. 2005), for example. Several such problems have been considered in Zhou *et al.* (2009) and can be considered as an extension of this study.

8. Conclusions

This article has proposed an algorithm to model the Pareto-optimal set of an MOOP using the data returned by an MOOA in a continuous search space. A continuous model of the Pareto-optimal surface has been obtained from the discrete data set returned by an MOOA. The algorithm has been designed to take advantage of the regularity and connectedness property of the Pareto-optimal front

to do modelling for a certain class of optimization problem. Cubic B-spline basis functions are used for the process of data fitting. Results have been shown for some standard test problems indicating that the Pareto-optimal front obtained by the model is close to the true Pareto-optimal front.

Such a procedure helps to provide a continuous picture of the front which gives a sense of completion in solving an optimization problem. As a by-product, using the parametric hyposurface, points not returned by the MOOA can also 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. On account of this, the data-fitting procedure proposed here can be implemented within an EMO algorithm and can be applied after every few generations so as to help generate new and near-optimal points quickly using the fitted hyposurface. In this article, results for several standard and engineering-design optimization problems were discussed. In future studies more engineering design problems can be analysed. In principle, the algorithm can be extended to higher objective problems as well. In higher dimensions, issues with clustering and merging of clusters may be important. The data fitting procedure will also be used for designing MOOAs that use data fitting as an intermediate step to figure out relationships in non-dominated frontier data and use that for faster MOOA convergence.

Acknowledgements

The third author acknowledges his JC Bose National Fellowship from the Government of India and its support towards this study.

References

Bandaru, S., and K. Deb. 2011. "Towards Automating the Discovery of Certain Innovative Design Principles Through a Clustering-Based Optimization Technique." *Engineering Optimization* 43 (9): 911–941.

Bonham, C. R., and I. C. Parmee. 2004. "Developments of the Cluster Oriented Genetic Algorithm." *Engineering Optimization* 36 (2): 249–279.

Bosman, P. A. N., and D. Thierens. 2005. "The Naive MIDEA: A Baseline Multiobjective EA." In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005).* Vol. 3410 of *Lecture Notes in Computer Science*. Guanajuato, Mexico: Springer, 428–442.

Charnes, A., R. W. Clower and K. O. Kortanek. 1967. "Effective Control Through Coherent Decentralization with Preemptive Goals." *Econometrica* 35: 294–320.

Charnes, A., and W. W. Cooper. 1977. "Goal Programming and Multiple Objective Optimization—Part 1." *European Journal of Operations Research* 1: 39–54.

Chiu, S. 1994. "Fuzzy Model Identification Based on Cluster Estimation." *Journal of Intelligent and Fuzzy Systems* 2 (3), 267–278.

Coello, C. A. C., D. A. VanVeldhuizen, and G. Lamont. 2002. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer.

Daniilidis, A., N. Hadjisavvas, and S. Schaible. 1997. "Connectedness of the Efficient Set for Three-Objective Quasiconcave Maximization Problems." *Journal of Optimization Theory and Applications* 93 (3): 517–524.

Das, I., and J. E. Dennis. 1998. "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems." *SIAM Journal on Optimization* 8: 631–657.

Deb, K. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley.

Deb, Kalyanmoy, J. Sundar, N. Udaya Bhaskara Rao, and Shamik Chaudhuri. 2006. "Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms." *International Journal of Computational Intelligence Research* 2 (3): 273–286.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan, 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Deb, K., L. Thiele, M. Laumanns, and E. Zitzler. 2005. "Scalable Test Problems for Evolutionary Multi-Objective Optimization." In *Evolutionary Multiobjective Optimization,* edited by A. Abraham, L. Jain, and R. Goldberg, 105–145. London: Springer-Verlag.

Deb, K., R. Tiwari, M. Dixit, and J. Dutta. 2007. "Finding Trade-Off Solutions Close to KKT Points Using Evolutionary Multi-Objective Optimization." In *Proceedings of the Congress on Evolutionary Computation (CEC-2007)*, 2109–2116. Piscataway, NJ: IEEE Press.

Dellnitz, M., O. Schutze, and T. Hestermeyer. 2005. "Covering Pareto Sets by Multilevel Subdivision Techniques." *Journal of Optimization Theory and Applications* 124 (1): 113–136.

Dumitrescu, D., C. Groşan, and M. Oltean. 2001. "Genetic Chromodynamics for Obtaining Continuous Representation of Pareto Regions." *Studia Universitatis Babeş–Bolyai, Informatica* XLVI (1): 15–30.

Ehrgott, M., and K. Klamroth. 1997. "Connectedness of Efficient Solutions in Multiple Criteria Combinatorial Optimization." *European Journal of Operational Research* 150, 475–497.

Fang, L., and D. C. Gossard. 1995. "Multidimensional Curve Fitting to Unorganized Data Points by Nonlinear Minimization." *Computer-Aided Design* 27 (1): 48–58.

Gass, S. I., and T. L. Saaty. 1955. "The Computational Algorithm for the Parametric Objective Function." *Naval Research Logistics Quarterly* 2: 39–45.

Goel, T., R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker, 2007. "Response Surface Approximation of Pareto Optimal Front in Multi-Objective Optimization." *Computer Methods in Applied Mechanics and Engineering* 196: 879–893.

Grosan, C. 2003. "A New Evolutionary Technique for Detecting Pareto Continuous Regions." In *2003 Genetic and Evolutionary Computation Conference. Workshop Program*, July, edited by A. Barry, 304–307. Chicago, IL: AAAI.

Haeser, G., and M. L. Schuverdt. 2009. "Approximate KKT Conditions for Variational Inequality Problems." *Optimization Online*. http://www.optimization-online.org/DB_HTML/2009/10/2415.html.

Hillermeier, C. 2001. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Vol. of *International Series of Numerical Mathematics*. Basel, Switzerland: Birkhaüser Verlag.

Jin, Y., and B. Sendhoff. 2003. "Connectedness, Regularity and the Success of Local Search in Evolutionary Multi-Objective Optimization." In Vol. 3 of *Proceedings of the IEEE Congress on Evolutionary Computation*, 1910–1917. Piscataway, NJ: IEEE Press.

Karush, W. 1939. "Minima of Functions of Several Variables with Inequalities as Side Conditions." Master's thesis, Department of Mathematics, University of Chicago.

Kuhn, H. W., and A. W. Tucker. 1950. "Nonlinear Programming." In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, edited by J. Neyman, 481–492. Berkeley, CA: University of California Press.

Marler, R. T., and J. S. Arora. 2004. "Survey of Multi-Objective Optimization Methods for Engineering." *Structural and Multidisciplinary Optimization* 26 (6): 369–395.

Messac, A. 1996. "Physical Programming: Effective Optimization for Computational Design." *AIAA Journal* 34: 149–158.

Miettinen, K. 1999. "*Nonlinear Multiobjective Optimization*. Boston, MA: Kluwer.

Mortenson, M. E. 1985. "*Geometric Modeling*. Chichester, UK: Wiley.

Okabe, T., Y. Jin, B. Sendoff, and M. Olhofer, 2004. "Voronoi-Based Estimation of Distribution Algorithm for Multi-Objective Optimization." In *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, Portland, OR, 1594–1601. IEEE Press.

Pareto, V. 1906. "Manuale di Economica Politica." [Manual of Political Economy]. Milan: Societa Editrice Libraria. Translated into English by A. S. Schwier. New York: Macmillan, 1971.

Ratrout, N. T. 2011. "Subtractive Clustering-Based K-Means Technique for Determining Optimum Time-of-Day Breakpoints." *Journal of Computing in Civil Engineering* 25 (5): 380–387.

Sindhya, K., K. Deb, and K. Miettinen, 2009. "Local Search Based Evolutionary Multi-Objective Optimization Algorithm for Constrained and Unconstrained Problems." In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)* May, Trondheim, Norway, 2919–2926. Piscataway, NJ: IEEE Press.

Steuer, R. 1986. *Multiple Criteria Optimization: Theory, Computation, and Application*. New York: Wiley.

Steuer, R., and E. Choo. 1983. "An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming." *Mathematical Programming* 26: 326–344.

Steuer, R. E., M. Hirschberger, and Y. Qi. 2010. "Large-Scale MV Efficient Frontier Computation Via a Procedure of Parametric Quadratic Programming." *European Journal of Operational Research* 204: 581–588.

Tulshyan, R., R. Arora, K. Deb, and J. Dutta. 2010. "Investigating EA Solutions for Approximate KKT Conditions for Smooth Problems." In *Proceedings of Genetic and Evolutionary Algorithms Conference (GECCO-2010)*, 689–696. New York: ACM Press.

Veldhuizen, D. A. V. 1999. "Multiobjective Evolutionary Algorithms: Classification, Analyses and New Innovations." PhD diss., Air Force Institute of Technology, Wright-Patterson AFB, Montgomery, OH.

Verma, M., M. Srivastava, N. Chack, A. K. Diswar, and N. Gupta, 2012. "A Comparative Study of Various Clustering Algorithms in Data Mining." *International Journal of Engineering Research and Applications* 2: 1379–1384.

Wierzbicki, A. P. 1979. "The Use of Reference Objectives in Multiobjective Optimization—Theoretical Implications and Practical Experiences," 1–32. Laxenburg, Austria: International Institute for Applied System Analysis.

Zadeh, L. 1963. "Optimality and Non-Scalar-Valued Performance Criteria." *IEEE Transactions on Automatic Control* 8: 59–60.

Zhang, Q., A. Zhou, and Y. Jin. 2008. "RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm." *IEEE Transactions on Evolutionary Computation* 12 (1): 41–63.

Zhou, A., Q. Zhang, and Y. Jin. 2009. "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.

Zhou, A., Q. Zhang, Y. Jin, E. Tsang and T. Okabe, 2005. "A Model-Based Evolutionary Algorithm for Bi-Objective Optimization." In *Proceedings of the Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, 2568–2575. Piscataway, NJ: IEEE Press.

Zhou, A., *et al.* 2006. "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)*, Vancouver, 3234–3241. Piscataway, NJ: IEEE Press.