3-9-2009

Adaptive Pareto Set Estimation for Stochastic Mixed Variable Design Problems

Christopher D. Arendt

ADAPTIVE PARETO SET ESTIMATION FOR STOCHASTIC MIXED VARIABLE DESIGN PROBLEMS

THESIS

Christopher D. Arendt, Captain, USAF

AFIT/GOR/ENS/09-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/09-01

ADAPTIVE PARETO SET ESTIMATION FOR STOCHASTIC MIXED VARIABLE DESIGN PROBLEMS

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Christopher D. Arendt, BA

Captain, USAF

March 2009

AFIT/GOR/ENS/09-01

ADAPTIVE PARETO SET ESTIMATION FOR STOCHASTIC MIXED VARIABLE DESIGN PROBLEMS

Christopher D. Arendt, BA

Captain, USAF

Approved:

Dr. James W. Chrissis (Chairman)	date
Dr. Raymond R. Hill (Member)	date

AFIT/GOR/ENS/09-01

Abstract

Many design problems require the optimization of competing objective functions that may be too complicated to solve analytically. These problems are often modeled in a simulation environment where static input may result in dynamic (stochastic) responses to the various objective functions. System reliability, alloy composition, algorithm parameter selection, and structural design optimization are classes of problems that often exhibit such complex and stochastic properties. Since the physical testing and experimentation of new designs can be prohibitively expensive, engineers need adequate predictions concerning the viability of various designs in order to minimize wasteful testing. Presumably, an appropriate stochastic multi-objective optimizer can be used to eliminate inefficient designs through the analysis of simulated responses. This research develops an adaptation of Walston's [56] Stochastic Multi-Objective Mesh Adaptive Direct Search (SMOMADS) and Paciencia's NMADS [45] based on Kim and de Weck's [34] Adaptive Weighted Sum (AWS) procedure and standard distance to a reference point methods. This new technique is compared to standard heuristic based methods used to evaluate several real-world design problems. The main contribution of this paper is a new implementation of MADS for Mixed Variable and Stochastic design problems that drastically reduces dependence on subjective decision maker interaction.

Acknowledgements

I would like to thank Dr. James Chrissis for his insight and direction, and Capt Todd Paciencia for his foundational contributions. This research could not have been accomplished without their generous support.

Table of Contents

List of Figures

List of Tables

ADAPTIVE PARETO SET ESTIMATION FOR STOCHASTIC MIXED VARIABLE DESIGN PROBLEMS

I. Introduction

1.1. Problem Setting

A scalar-valued function maps an n-dimensional vector to a scalar value (a one-dimensional vector). Vector-valued functions, on the other hand, map an n-dimensional design vector to an m-dimensional response vector (where m is greater than 1). Thus, optimizing a single objective function involves finding the extrema of a scalar-valued function, whereas optimizing multiple objectives simultaneously involves finding the extrema of a vector-valued function. Unfortunately, for vector-valued functions, there is often no single n-dimensional design vector in the feasible domain whose response vector contains the extrema in each of the m dimensions of the range. That is, often a mapping will be better in one subset of objectives and worse in another. Therefore, in multi-objective optimization, the goal is to find the set of solutions that are non-dominated, known as the *Efficient set, Pareto set,* or *Pareto front*. According to Paciencia's [45] definition, a vector is in the set of non-dominated solutions if its response vector is not worse for every objective than the response vector of another solution in the set. In most cases, it is practically impossible to generate the entire Pareto set. Instead, various methods are used to approximate the set. However, each of these approximation methods has its own limitations.

If at least one element of the m-dimensional response vector exhibits randomness or uncertainty, the optimization problem is considered a *stochastic optimization problem,* and can be presented in the standard form given by Paciencia [45]:

$$\min\ Z(w) = F(x,w) \tag{1.1a}$$

subject to

$$g_i(x,w) \le 0, i = 1,\ldots,M \tag{1.1b}$$

$$x \in \mathbb{R}^{n_1} \tag{1.1c}$$

$$w \in \mathbb{R}^{n_1} \tag{1.1d}$$

where *x* represents the vector of deterministic design variables, *w* represents a vector of uncontrollable random variables, and all constraints are assumed to be deterministic. Commonly, the stochastic function $F(x,w)$ is replaced with an observation or mathematical expectation. If the observation is assumed to be an unbiased estimator of the actual function, the response can be written as $F(x,w) = f(x) + \varepsilon_w(x)$, where $f(x)$ is some deterministic function and $\varepsilon_w(x)$ is some random error function such that $E[\varepsilon_w(x)] = 0$.

If at least one of the variables of the *n*-dimensional design vector is discrete or categorical, the optimization problem is considered a *mixed variable problem*. In the case of mixed-variable problems, the set of design variables, Ω, is decomposed into two subsets: the set of continuous variables, Ω^c, and the set of categorical and discrete variables, Ω^d. If the set of categorical and discrete variables is mapped to a subset of integers, $\mathbb{Z}^{n_d}$, where n_d represents the number of categorical and discrete variables, any solution vector, $x \in \Omega$, can be denoted as $x = (x^c, x^d) \in \left(\mathbb{R}^{n_c} \times \mathbb{Z}^{n_d}\right)$ where $x^c \in \mathbb{R}^{n_c}$, $x^d \in \mathbb{Z}^{n_d}$, and $n = n_c + n_d$ is the total number of design variables. Incorporating the mixed variables with the stochastic optimization, the problem can be presented in the standard form given by Paciencia [45]:

$$\min\ E[F(x)] = E[f(x) + \varepsilon_w(x)] \tag{1.2a}$$

subject to

$$g_i(x, w) \le 0, i = 1, \ldots, M \tag{1.2b}$$

where $F(x): \left(\mathbb{R}^{n_c} \times \mathbb{Z}^{n_d}\right) \to \mathbb{R}^m$ such that $F = \left(F_1, F_2, ..., F_m\right)$. A solution x^* is *Pareto optimal* for the vector-valued function F if there are no other feasible solutions that give a better response in all objectives. For any Pareto optimal solution x^*, there may exist feasible solutions that gives a better response in some objectives.

1.2. Purpose of the Research

Walston [56] and Paciencia [45] developed implementations of Mesh Adaptive Direct Search with Ranking and Selection (MADS-RS) to solve mixed variable, stochastic multi-objective problems of the type described in Section 1.1. However, Paciencia [45] concluded that the quality of an initial Pareto front approximation in Walston's SMOMADS implementation was dependent upon the aspiration/reservation level ranges. That is, the decision maker interaction with SMOMADS can affect the quality of the Pareto set determined by SMOMADS. Additionally, Paciencia stated that "an algorithm that can identify gaps in n-dimensional space, likely based on indifference values, in an efficient and complete manner could be of great value". However, he also noted that since indifference values are not easy to determine, an algorithm "not based on indifference values would be useful as well". Finally, while the MADS-RS procedure has been evaluated on a set of standard test problems, these test problems may provide little insight into how well the MADS-RS procedure may perform in solving real-world design problems.

1.3. Problem Statement

This research was designed to improve or expand upon the methods developed in SMOMADS and NMADS. Specifically, the Pareto front generation algorithm developed in this research makes use of the SMOMADS method of finding objective-wise optima, but diverges from SMOMADS in that it does not depend on decision maker input in selecting desirable or undesirable objective function values. The Pareto front gap-finding algorithm developed in this research leverages the NMADS method of sorting Pareto optimal solutions one objective at a time, but searches for multiple neighbors, instead of performing the single-neighbor search used in NMADS. Additionally, this research incorporates Kim and de Weck's [34] recently developed Adaptive Weighted Sum (AWS) method for multi-objective optimization into the MADS-RS algorithm developed by Abramson, *et al.* [1]. The AWS-inspired method replaces the SMOMADS aspiration and reservation level method of Pareto front determination (which requires a decision maker to select values of the multiple objective functions that are "ideal" or "unacceptable") with an adaptive algorithm that automatically searches the objective function space to approximate the Pareto front. Furthermore, a novel algorithm is used to identify gaps in the *m*-dimensional Pareto front. Finally, this updated MADS algorithm is compared to existing methods for solving four common classes of real-world multi-objective mixed variable stochastic design problems.

1.4. Overview

This thesis is organized as follows. Chapter II reviews past implementations of MADS-RS as well as other methods for solving multi-objective optimization problems.

Furthermore, the AWS method is reviewed in comparison to other methods of determining the Pareto front. Finally, four classes of common design problems are discussed. Chapter III presents an adaptive constraint Pareto set estimation algorithm based on AWS, and incorporates it into the MADS-RS framework. Chapter IV presents the results of the new adaptive constraint implementations of MADS-RS and the analysis of its performance. Chapter V presents the final conclusions and recommendations for future research.

II. Literature Review

With few exceptions, most multi-objective optimization techniques transform the multiple objective functions into a single objective and use some standard non-linear optimization technique to optimize the single objective problem. In the case of stochastic optimization, some method of choosing a best solution in the presence of uncertainty must be employed. Therefore, this chapter presents various methods of scalarizing the multiple objectives functions into a single objective function. Then, methods for determining optimality in the presence of uncertainty are discussed. Additionally, some recently developed optimization algorithms are reviewed, including their current implementations as SMOMADS and NMADS, developed by Walston [56] and Paciencia [45], respectively. Finally, four classes of multi-objective design problems are described, along with techniques that have been used to solve them.

2.1. Pareto Front Estimation

This section describes various common methods for generating or approximating the set of Pareto optimal solutions. This is not meant to be a comprehensive description.

2.1.1. Pareto Set Definition

A Pareto optimal solution is defined by Erghott [28] as follows:

Definition 2.1.1. *A solution to a multi-objective optimization problem of the form* $\min_{x \in \Theta} F(x,w),\ F:\Theta \to \mathbb{R}^m$ *is said to be Pareto optimal at the point* $\hat{x}$ *if there is no* $x \in \Theta$ *such that* $F_k(x) \le F_k(\hat{x})$ *for* $k = 1,\ldots,m$ *and* $F_i(x) < F(\hat{x})$ *for some* $i \in \{1,\ldots,m\}$.

A solution is said to be *dominated* if it is not Pareto optimal.

Since it is generally not possible to generate the complete set of Pareto optimal solutions, several techniques have been developed to estimate the Pareto set. Paciencia [45] outlines several Pareto set quality metrics that have been developed.

2.1.2. Weighted Sum

The simplest (often called naïve) approach to generating a Pareto optimal solution is to optimize a weighted sum of the multiple objective functions. Often in this method, a vector of random weights $\gamma = \{\gamma_1, \ldots, \gamma_m\}$ is generated such that $\sum_{i=1}^{m} \gamma_i = 1$ and $\gamma_i \geq 0$ for all $i = 1, \ldots, m$. These weights form the single-objective optimization problem of the form

$$\min Z(x) = \sum_{i=1}^{m} \gamma_i \hat{F}_i(x), \tag{2.1}$$

where $\hat{F}_i(x)$ may be some normalized version of the *i*th objective function (the objective functions may be normalized so only the random weights determine which objective will dominate the weighted sum). Das and Dennis [21] demonstrate that for every unique random vector γ, solving (2.1) generates a Pareto optimal solution.

While this is a very efficient generator, its principal limitation is that it cannot find Pareto optimal points in any non-convex portion of the Pareto front. For a thorough explanation of this drawback, see Das and Dennis [21].

2.1.3. Normal Boundary Intersection

Das and Dennis [22] describe a new method for generating the Pareto front, known as the Normal Boundary Intersection (NBI). This method solves a series of single-objective formulations of the multi-objective optimization problem with an additional equality constraint based on previously determined objective function values. The intersection of this new equality constraint and the boundary of the surface determined by feasible objective function values is added to the set of Pareto optimal solutions. This is based on the idea that the intersection of the boundary of the set of feasible objective function values with a normal vector emanating from a point that can be represented as a convex combination of feasible objective function values towards the origin lies on the portion of the boundary that contains non-dominated points. While NBI has been shown to generate an evenly distributed set of points on the Pareto front, it is not guaranteed to avoid selecting dominated solutions. That is, if the boundary of feasible objective function values is not sufficiently convex, the point of intersection may be a dominated solution.

2.1.4. Distance to a Reference Point

Instead of optimizing a weighted sum of the multiple objective functions, Collette and Siarry [20], explain that the distance to a reference point method minimizes the distance from a feasible point in the objective space to some reference point, usually the vector whose elements equal the optimal values for the individual objectives, known as the *utopia point*. While this method is capable of finding Pareto optimal solutions on non-convex regions of the Pareto front, Walston [56] explains its performance is highly dependent on the choice of reference point.

As shown in Figure 2.1, this method may fail to include points, p_i, on the Pareto front that a weighted sum method would find, since point q is closer to reference point r than all other points on the convex Pareto front.

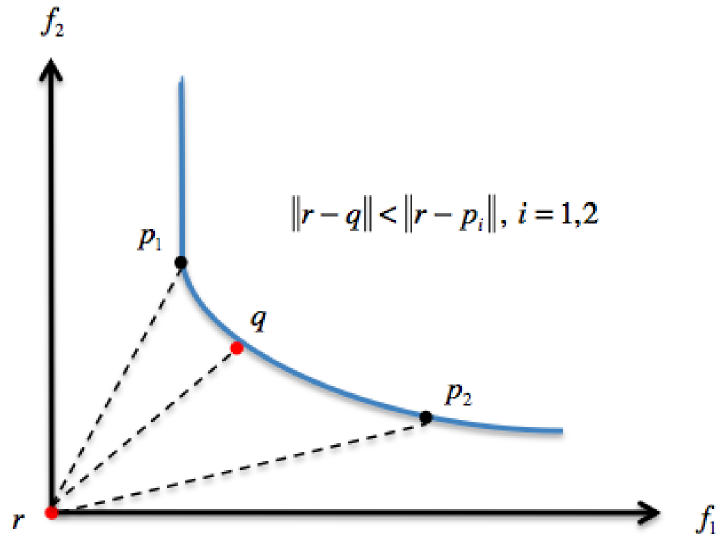

Figure 2.1: Distance to r minimized by single point q on convex Pareto front

2.1.5. Aspiration and Reservation Levels

The aspiration and reservation levels method is an interactive approach that takes decision maker input regarding acceptable (*aspiration*) and unacceptable (*reservation*) objective function values and uses that information to generate Pareto optimal solutions. As shown in Figure 2.2, points on the Pareto front can be found by varying the relative importance of the distance to a given point. Using the utopia point **U**, any point between points **D** and **E** can be found, where **D** and **E** represent the objective function values obtained by optimizing each objective independently. Walston [56] demonstrates how using aspiration point **A** as a starting point and connecting it with a ray to the reservation point **R**, points along the Pareto front between **B** and **C** can be found. According to Walston, this method "is based on the assumption that the decision maker has an idea of what is desired for each objective, as well as what minimum, or maximum, values are acceptable."

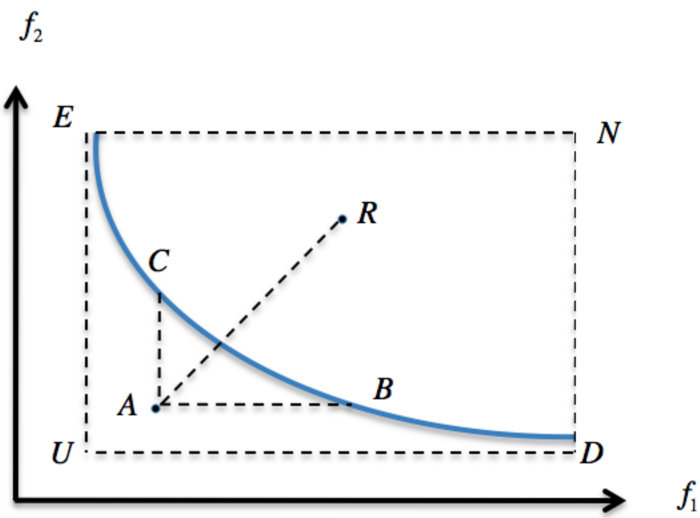

Figure 2.2: Pareto front generation from aspiration *A* and reservation *R*

In addition to the aspiration and reservation values assigned by the decision maker, this method also makes use of the *nadir point*, **N**. Ehrgott, *et al.* [28], characterize the nadir point $y^N \in \mathbb{R}^m$ by the component-wise supremum of all the Pareto optimal solutions:

$$y_m^N := \sup_{x \in Pareto} f^m(x), \; m = 1, \ldots, M. \tag{2.2}$$

Since the utopia point is the objective-wise minimum over the feasible set, as shown in Figure 2.2, the nadir point can be estimated as the reflection of the utopia point opposite the Pareto front.

In addition to requiring input from a knowledgeable decision maker, using aspiration and reservation levels to determine the Pareto front may become computationally expensive, since a new aspiration or reservation point must be selected in order to generate a new Pareto-optimal solution. Since it is impossible to sample every possible combination of aspiration and reservation levels, Paciencia [45] evaluates the use of several different experimental designs to select aspiration and reservation points that would generate a sufficient representation of the Pareto front.

2.1.6. Adaptive Weighted Sum

Generalizing their previous work in the bi-objective case, Kim and de Weck [34] developed the Adaptive Weighted Sum (AWS) method for multi-objective optimization. AWS begins by estimating the Pareto front using a typical weighted sum technique. It then searches the resulting set of points for gaps that may indicate the existence of a non-convex region in the Pareto front.

In the bi-objective case, a gap is defined as the distance from one point, p_1, to its nearest neighbor, p_2. If this distance is greater than some threshold value ξ, two new inequality constraints, c_1 and c_2, that define a region slightly smaller than the region contained within p_1 and p_2 are added to the optimization problem. A weighted sum is again used to optimize this adapted optimization problem. If the gap represents a non-convex region of the surface, as depicted in Figure 2.3, the weighted sum optimization produces two new points, q_1 and q_2, at the end points of the feasible region.

If the distance between these new points is greater than the threshold value, the new points are used to define their own new constraints, and the process is repeated until the distance between the points produced by the AWS optimization problem is less than the threshold value. This procedure is continued for each gap identified in the original

Pareto set. Thus, the AWS method can estimate the actual Pareto front with arbitrarily good accuracy without permitting any dominated points into the estimated Pareto set.

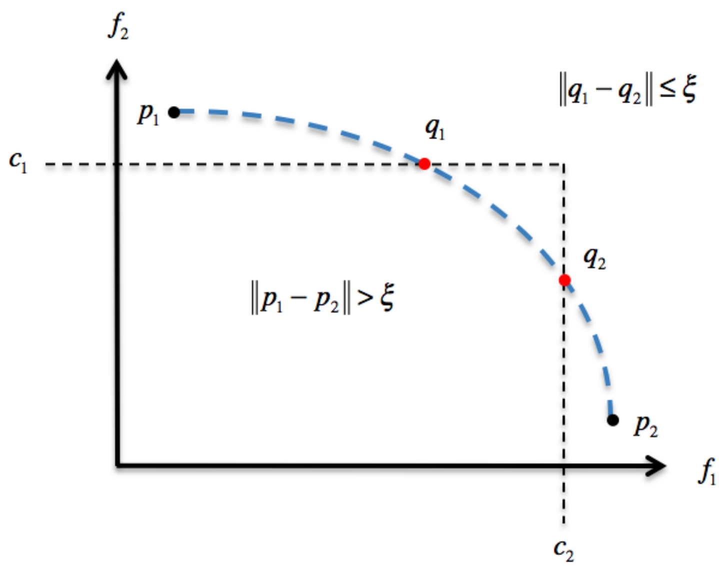

Figure 2.3: Points q_1 and q_2 generated via Adaptive Weighted Sum

Whereas inequality constraints are used to adapt the bi-objective weighted sum optimization problem, in the case of more than two objectives, the AWS method introduces an equality constraint to generate a new candidate for the Pareto set. First, a standard weighted sum method is used to generate an initial set of Pareto optimal solutions. For simplicity, or if the entire feasible surface in the objective space is non-convex, this initial set may be limited to the solutions corresponding to the components of the utopia point, that is, the points defined by the objective-wise minima. A subset of these initial points can then be selected to define the hyper-plane containing the entire subset of points. To generate a new solution, a line is constructed to emanate from the nadir point and intersect the hyper-plane at a point interior to the defining subset. The equation of this line is added as an equality constraint to adapt the optimization problem.

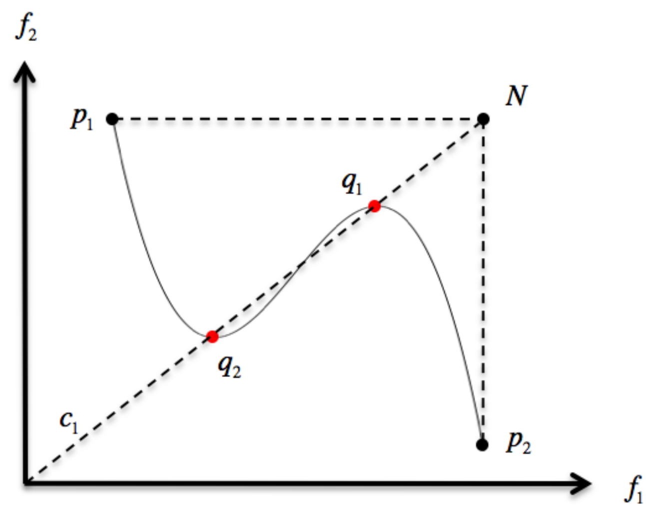

Figure 2.4: Point q_1 satisfying c_1 and dominated by q_2

However, as shown in Figure 2.4, since the solution to this equality-adapted optimization problem is not guaranteed to be non-dominated (that is, it is possible for a dominated solution to satisfy the new problem), each new point must be checked for dominance before being added to the Pareto set.

2.2. Optimality in the Presence of Uncertainty

For multi-objective optimization problems, if at least one objective (or element of the *m*-dimensional response vector) exhibits randomness or uncertainty, the optimization problem is considered a stochastic optimization problem, and can be presented in the standard form given by Paciencia [45]:

$$\min\ Z(w) = F(x, w) \tag{2.3a}$$

subject to

$$g_i(x, w) \le 0, i = 1, \ldots, M \tag{2.3b}$$

$$x \in \mathbb{R}^{n_1} \tag{2.3c}$$

$$w \in \mathbb{R}^{n_1} \tag{2.3d}$$

where *x* represents the vector of deterministic design variables, *w* represents a vector of uncontrollable random variables, and all constraints are assumed to be deterministic.

2.2.1. Deterministic Approach

The simplest method to handle the stochastic response vector is to substitute the uncertain vector with a deterministic value. Commonly, the stochastic function $F(x,w)$ is replaced with its observation or mathematical expectation. If the observation is assumed to be an unbiased estimator of the actual function, the response can be written as $F(x,w) = f(x) + \varepsilon_w(x)$, where $f(x)$ is some deterministic function and $\varepsilon_w(x)$ is some random error function such that $E[\varepsilon_w(x)] = 0$. Thus, according to Marti [42], the "true" value of the response, $f(x)$, is assumed to be given by $E[F(x,w)]$, and this value alone is used for optimality or improvement comparisons. While straightforward, this method may be misleading in that it fails to reflect the variability present in each optimal solution produced.

2.2.2. $\kappa\sigma$ Approach

The $\kappa\sigma$ Approach is similar to the Deterministic Approach in that it selects a single representative value for the response vector. However, in the $\kappa\sigma$ Approach, in order to avoid underestimating (in the case of a minimization problem) the response, the value used to represent the response at a given feasible solution x is increased by a function of the variance of the observed response. Mattson and Messac [44] formulate this representative value as:

$$f(x)+\kappa\sigma_{\varepsilon_w(x)} \ , \tag{2.4}$$

where $\sigma^2_{\varepsilon_w(x)}$ is the variance of ε_w at x and κ is some positive scalar.

2.2.3. Ranking and Selection

Instead of generating precise estimates of the stochastic response, Ranking and Selection (R&S) considers multiple candidates simultaneously.

Using the R&S formulation presented by Paciencia [45], let X_k denote the *k*th element of a sequence of random vectors and x_k denote a realization of X_k. For a finite set of candidate points $C=\{Y_1,\ldots,Y_{n_c}\}$ with $n_c \geq 2$, let $f_q = f(Y_q) = E\left[F(Y_q)\right]$ denote the true mean of the response function F at Y_q for each $q=1,\ldots,n_c$. These means can be ordered as $f_{[1]},\ldots,f_{[n_c]}$, where $f_{[1]}$ represents the minimum mean and $f_{[n_c]}$ represents the maximum. Denote by $Y_{[q]} \in C$ the candidate from C with the *q*th lowest true objection function value.

Given some $\delta > 0$, called the *indifference zone parameter*, no distinction is made between two candidate points whose true means satisfy $f_{[2]} - f_{[1]} < \delta$. In such a case, the method is *indifferent* in choosing either candidate as best. The probability of correct selection (*CS*) is defined as

$$P\left[CS\right] = P\left[\text{select } Y_{[1]} \,\middle|\, f_{[q]} - f_{[1]} \geq \delta,\ q=1,\ldots,n_c\right] \geq 1-\alpha, \tag{2.5}$$

where $\alpha \in (0,1)$ is the statistical significance level. Because random sampling guarantees $P[CS] \geq \frac{1}{n_c}$, the significance level must satisfy $0 < \alpha < 1 - \frac{1}{n_c}$.

Since the true objection function means are unavailable, it is necessary to use the sample means of F to select the best candidate. For each $q \in \{1, \ldots, n_c\}$, let s_q be the total number of replications at Y_q, and let $\left\{F_{qs}\right\}_{s=1}^{s_q} = \left\{F\left(Y_{qs}, W_{qs}\right)\right\}_{s=1}^{s_q}$ be the set of simulated responses, where $\left\{Y_{qs}\right\}_{s=1}^{s_q}$ are the replications at candidate point Y_q, and W_{qs} are realizations of the random noise. For each $q \in \{1, \ldots, n_c\}$, the sample mean $\bar{F}$ is given by

$$\bar{F}_q = \frac{1}{s_q} \sum_{s=1}^{s_q} F_{qs} . \tag{2.6}$$

The sample means can be ordered and indexed, letting $\hat{Y}_{[1]} \in C$ denote the candidate with the qth lowest estimated objection function value as determined by the R&S procedure. The candidate corresponding to the minimum mean response $\hat{Y}_{[1]} = \arg(\bar{F}_{[1]})$ is chosen as the best point. A generic R&S procedure is shown in Figure 2.5. Bechhofer, *et al.* [14], provide an in depth discussion of several methods used to determine s_q for the R&S procedure.

Ranking and Selection Procedure $RS(C, \alpha_r, \delta_r)$

Inputs: $C = \{Y_1, \ldots, Y_{n_c}\}$, $\alpha_r \in (0,1)$, $\delta_r > 0$.

Step 1. For each $Y_q \in C$, use an appropriate statistical technique to determine the number of samples s_q required to meet the probability of correct selection guarantee in (2.5), as a function of α_r, δ_r and response variation of Y_q.

Step 2. For each $q \in \{1, \ldots, n_c\}$, obtain replicated responses F_{qs}, $s \in \{1, \ldots, s_q\}$, and compute the sample mean $\overline{F}_q$, according to (2.6).

Return: $\hat{Y}_{[1]} = \arg\left(\overline{F}_{[1]}\right)$

Figure 2.5: Generic R&S Procedure [45]

2.3. Existing Optimization Techniques

This section describes the development of the Generalized Pattern Search (GPS) and Mesh Adaptive Direct Search (MADS) optimization techniques that are used in this research. These optimization methods were selected for study because they have been shown to be able to converge to optimal solutions for stochastic problems. Additionally, two recent implementations of these methods, SMOMADS and NMADS, are presented. SMOMADS and NMADS represent the foundational work upon which this research expands.

2.3.1. Generalized Pattern Search

Generalized pattern search (GPS) algorithms are defined through a finite set of directions used at each iteration. The direction set and a step length parameter are used to generate a discrete set of points, or mesh, around the current iterate. Paciencia [45] defines the mesh at iteration *k* to be

$$M_k = \bigcup_{x \in O_k} \{x + \Delta_k^m Dz : z \in \mathbb{N}^{n_D}\}, \tag{2.7}$$

where O_k is the set of points for which the objective function f has been evaluated as of iteration *k*, Δ_k^m is called the *mesh size parameter*, and *D* is a set of positive spanning directions for $\mathbb{R}^n$. Torczon and Trosset [55] state an additional restriction on *D* is that each direction $d_j \in D,\ j \in \{1,\ldots,n_D\}$, must be the product of some fixed nonsingular generating matrix $G \in \mathbb{R}^{n \times n}$ and an integer vector $z_j \in \mathbb{Z}^n$. For bound and linearly constrained problems, the directions in *D* must be sufficiently rich to ensure that *polling* directions (directions used to obtain a *poll set*) can be chosen that conform to the geometry of the constraint boundaries, and that these directions be used infinitely many times. A finite set of trial points, called the *poll set*, is then chosen from the mesh, evaluated, and compared to the incumbent solution. If improvement is found, the incumbent is replaced and the mesh is retained or coarsened by increasing the mesh size parameter Δ_k^m. If not, the mesh is refined and a new set of trial points is selected.

Audet and Dennis [6] as well as Lewis and Torczon [36] have extended this approach to handle nonlinear constraints. Furthermore, Audet and Dennis [5] extended GPS to handle mixed variable problems with bound constrains by including user-specified discrete neighborhoods in the definition of the mesh. Abramson *et al.* extended the mixed variable results of Audet and Dennis [5] to linear [1] and non-linear constraints [3].

This method was applied with ranking and selection by Sriver [52] to address stochastic mixed-variable problems. In this method, the poll set at each iteration is given by $P_k(x_k) \bigcup N(x_k)$ where $N(x_k)$ is a user-defined set of discrete neighbors around x_k and

$$P_k = \left\{ x_k + \Delta_k (d,0) : d \in D_k^i \right\}, \tag{2.8}$$

where $(d,0)$ denotes that continuous variables have been partitioned and that the discrete variables remain unchanged. The set of discrete neighbors is defined by a set-valued function $N : \Omega \to 2^{\Omega}$, where 2^{Ω} denotes the power set of Ω. By convention, $x \in N(x)$ for each $x \in \Omega$, and it is assumed that $N(x)$ is finite. A generic indifference-zone ranking and selection procedure $RS(P_k, \alpha, \delta)$ with indifference-zone parameter δ and significance level α is used to select among points in the poll set for improved solutions (see Figure 2.5). Given a fixed rational number $\tau > 1$ and two integers $m^- \leq -1$ and $m^+ \geq 0$, the mesh size parameter Δ_k^m is updated according to

$$\Delta_{k+1}^m = \tau^{w_k} \Delta_k^m , \tag{2.9}$$

where

$$w_k \in \begin{cases} \{0,1,\ldots,m^+\}, & \text{if an improved mesh point is found} \\ \{m^-, m^- + 1, \ldots, -1\}, & \text{otherwise.} \end{cases} \tag{2.10}$$

If no improvement is found, an extended poll step is conducted to search about any discrete neighbor $y \in N(x_k)$ that satisfies $f(x_k) \leq f(y) + \xi_k$, where ξ_k is called the *extended poll trigger*. Paciencia [45] explains how each neighbor satisfying this criteria, in turn, becomes the poll center, and the extended poll continues until either a better point than the current iterate is found, or else they are all worse than the extended poll center. Sriver [52] showed that this algorithm has an iteration subsequence with almost sure convergence to a stationary point “appropriately defined” in the mixed-variable domain. That is, as the number of iterations goes to infinity, the probability that the algorithm will converge to a stationary point in the mixed-variable domain grows to 1. A general mixed variable pattern search with ranking and selection (General MVPS-RS) algorithm is shown in Figure 2.6

2.3.2. Mesh-Adaptive Direct Search

Mesh Adaptive Direct Search (MADS) was developed by Audet and Dennis for minimization of non-smooth function of the type $f : \mathbb{R}^n \rightarrow \mathbb{R} \bigcup \{+\infty\}$ under general constraints $x \in \Omega \subseteq \mathbb{R}^n$ where $\Omega \neq \varnothing$.

A General MVPS-RS Algorithm

- INITIALIZATION: Let $X_0 \in \Omega$, $\Delta_0 > 0$, $\xi > 0$, $\alpha_0 \in (0,1)$, and $\delta_0 > 0$. Set the iteration and R&S counters $k = 0$ and $r = 0$, respectively.

- SERCH STEP (OPTIONAL): Employ a finite strategy to select a subset of candidate solutions, $S_k \subset M_k(X_k)$ defined in (2.8) for evaluation. Use R&S procedure $RS(S_k \cup \{X_k\}, \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]} \in S_k \cup \{X_k\}$, update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq X_k$, the step is *successful*, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$, see (2.9)-(2.10), and $k = k+1$, and repeat SEARCH STEP. Otherwise, proceed to POLL STEP.

- POLL STEP: Set extended poll trigger $\xi_k \geq \xi$. Use R&S procedure $RS(P_k(X_k) \cup N(X_k), \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq X_k$, the step is *successful*, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$, see (2.9)-(2.10), and $k = k+1$, and return to SEARCH STEP. Otherwise, proceed to EXTENDED POLL STEP.

- EXTENDED POLL STEP: For each discrete neighbor $Y \in N(X_k)$ that satisfies the extended poll trigger condition $\overline{F}(Y) < \overline{F}(X_k) + \xi_k$, set $j = 1$ and $Y_k^j = Y$, and do the following.
  - Use R&S procedure $RS(P_k(Y_k^j), \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq Y_k^j$, set $Y_k^j = \hat{Y}_{[1]}$ and $j = j+1$, and repeat this step. Otherwise, set $Z_k = Y_k^j$ and go to the next step
  - Use R&S procedure $RS(X_k \cup Z_k, \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} = Z_k$, the step is *successful*, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1} \geq \Delta_k$, see (2.9)-(2.10), and $k = k+1$, and return to the SEARCH STEP. Otherwise, repeat the extended poll trigger condition. If no such discrete neighbors remain in $N(X_k)$, set $X_{k+1} = X_k$, $\Delta_{k+1} < \Delta_k$, and $k = k+1$, and return to the SEARCH STEP.

Figure 2.6: General MVPS-RS Algorithm [52]

While MADS is similar to GPS in the generation of the mesh as well as the rules for updating the mesh, one key difference is that in MADS a separate poll size parameter Δ_k^p is introduced which controls the magnitude of the distance between the incumbent solution and trial points generated for the poll step, and satisfies $\Delta_k^m \le \Delta_k^p$ for all *k* such that $\lim_{k \in K} \Delta_k^m = 0 \Leftrightarrow \lim_{k \in K} \Delta_k^p = 0$ for any infinite subset of indices *K*.

In MADS, the frame is defined to be

$$P_k = \{x_k + \Delta_k^m d : d \in D_k\} \subset M_k, \tag{2.11}$$

where D_k is a positive spanning set such that $0 \notin D_k$ and for each $d \in D_k$ the following conditions must be met, according to Audet and Dennis [7]:

- d can be written as a non-negative integer combination of the direction in *D*:

  $d = Du$ for some vector $u \in \mathbb{N}^{n_{D_k}}$ that may depend on the iteration number *k*,

- the distance from the frame center x_k to a frame point $x_k + \Delta_k^m d : d \in P_k$ is bounded above by a constant times the poll size parameter:

$$\Delta_k^m \|d\| \le \Delta_k^p \max\{\|d'\| : d' \in D\},$$

- limits of the normalized sets $D_k = \left\{ \dfrac{d}{\|d\|} : d \in D_k \right\}$ are positive spanning sets.

Since the mesh size parameter tends to decrease to zero at a faster rate than the poll size parameter, the set of directions D_k used to define the MADS frame can be chosen from increasingly larger sets as a limit point is approached.

The general MADS algorithm is shown in Figure 2.7. The extended algorithm for stochastic mixed variable problems (MVMADS-RS) is shown in Figure 2.8.

A General MADS Algorithm

- INITIALIZATION: Let $x_0 \in \Omega$, $\Delta_0^m \le \Delta_0^p$, D, G, τ, w^-, and w^+ satisfy the requirements of a MADS frame set given in (2.11). Set the iteration counter $k = 0$.

- SERCH AND POLL STEP: Perform SEARCH and possibly the POLL step (or part of them) until an improved mesh point x_{k+1} is found on the mesh M_k, where M_k is defined as for GPS in (2.7).
  - OPTIONAL SEARCH: Evaluate f_Ω on a finite subset of trial points on the mesh M_k.
  - LOCAL POLL: Evaluate f_Ω on the frame P_k, where P_k is as given in (2.11).

- PARAMETER UPDATE: Update Δ_{k+1}^m and Δ_{k+1}^p. Set $k = k+1$ and go back to the SEARCH AND POLL STEP.

Figure 2.7: General MADS Algorithm [7]

2.3.3. SMOMADS

Stochastic Multi-Objective Mesh Adaptive Direct Search (SMOMADS) was developed by Walston [56] to incorporate the aspiration/reservation method of multi-objective scalarization with MVPS-RS or MVMADS-RS in order to solve stochastic, multi-objective mixed variable optimization problems. The single objective scalarization is obtained from the aspiration and reservation levels for each objective, a_i and r_i respectively, where $i \in \{1, \ldots, m\}$ and m is the number of objectives in the original problem, via the following function:

$$f = -\left(\min(u) + \varepsilon \cdot \sum_{i=1}^{m} u_i \right). \tag{2.12}$$

The function u_i, defined by

$$u_i = \begin{cases} \alpha_i \cdot w_i \cdot (a_i - f_i) + 1, & f_i < a_i \\ w_i \cdot (a_i - f_i) + 1, & a_i \le f_i \le r_i \\ \beta_i \cdot w_i \cdot (r_i - f_i) + 1, & r_i < f_i \end{cases} \tag{2.13}$$

is a strictly monotone function of the response vector components f_i, known as a component achievement function. Walston [56] showed that minimizing (2.12) will produce Pareto optimal solutions nearest the aspiration level (as described in Section 2.1.5.). However, due to the wide range of possible aspiration and reservation levels, Paciencia [45] describes how in it may be difficult to adequately represent the Pareto front without exploring a very large number of combinations of aspiration and reservation levels. Figure 2.10 shows a notional example of the SMOMADS algorithm.

2.3.4. NMADS

Paciencia [45] expanded the bi-MADS method developed by Audet, *et al.* [9], to explore the Pareto front for *m* multiple objectives. This new method uses a distance-to-a-reference point method for multi-objective scalarization, and optimizes the resulting single-objective problem with the MADS technique. Then, a gap finding algorithm is used to search for gaps in the current Pareto set. The gap finding algorithm attempts to determine if any given point in the current Pareto set is surrounded by other points in the set. Given a vector of indifference values (similar to the level of detail desired in each objective in the Pareto front) $\bar{\omega}$, each point in the set (except the points corresponding to extreme values in each objective) should have another point within ω_i and $-\omega_i$ (above and below) in each objective *i*.

A General MVMADS-RS Algorithm

- INITIALIZATION: Let $X_0 \in \Omega$, $\Delta_0^m \leq \Delta_0^p$, $\xi > 0$, $\alpha_0 \in (0,1)$, and $\delta_0 > 0$. Set the iteration and R&S counters $k = 0$ and $r = 0$, respectively.

- SERCH STEP (OPTIONAL): Employ a finite strategy to select a subset of candidate solutions, $S_k \subset M_k(X_k)$ defined in (2.8) for evaluation. Use R&S procedure $RS(S_k \cup \{X_k\}, \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]} \in S_k \cup \{X_k\}$, update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq X_k$, the step is *successful*, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1}^p \geq \Delta_k^p$, $\Delta_{k+1}^m \geq \Delta_k^m$ and $k = k+1$, and repeat SEARCH STEP. Otherwise, proceed to POLL STEP.

- POLL STEP: Set extended poll trigger $\xi_k \geq \xi$. Use R&S procedure $RS(P_k(X_k) \cup N(X_k), \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq X_k$, the step is *successful*, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1}^p \geq \Delta_k^p$, $\Delta_{k+1}^m \geq \Delta_k^m$ and $k = k+1$, and return to SEARCH STEP. Otherwise, proceed to EXTENDED POLL STEP.

- EXTENDED POLL STEP: For each discrete neighbor $Y \in N(X_k)$ that satisfies the extended poll trigger condition $\overline{F}(Y) < \overline{F}(X_k) + \xi_k$, set $j = 1$ and $Y_k^j = Y$, and do the following.
  - Use R&S procedure $RS(P_k(Y_k^j), \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} \neq Y_k^j$, set $Y_k^j = \hat{Y}_{[1]}$ and $j = j+1$, and repeat this step. Otherwise, set $Z_k = Y_k^j$ and go to the next step
  - Use R&S procedure $RS(X_k \cup Z_k, \alpha_r, \delta_r)$ to return the estimated best solution $\hat{Y}_{[1]}$. Update $\alpha_{r+1} < \alpha_r$, $\delta_{r+1} < \delta_r$, and $r = r+1$. If $\hat{Y}_{[1]} = Z_k$, the step is *successful*, update $X_{k+1} = \hat{Y}_{[1]}$, $\Delta_{k+1}^p \geq \Delta_k^p$, $\Delta_{k+1}^m \geq \Delta_k^m$ and $k = k+1$, and return to the SEARCH STEP. Otherwise, repeat the extended poll trigger condition. If no such discrete neighbors remain in $N(X_k)$, set $X_{k+1} = X_k$, $\Delta_{k+1}^p < \Delta_k^p$, $\Delta_{k+1}^m < \Delta_k^m$ and $k = k+1$, and return to the SEARCH STEP.

Figure 2.8: MVMADS-RS [45]

SMOMADS Algorithm

- Generate a set of Aspiration/Reservation levels.
- For each choice of Aspiration/Reservation levels, combine the objective functions into an achievement scalarization function, and solve using MVMADS-RS or MVPS-RS.
- In the case of stochastic problems, because the solution converges to an efficient point with probability one in infinite iterations, check to ensure that a point is non-dominated before adding to the efficient set by comparing to solutions found thus far.

Figure 2.9: SMOMADS Algorithm [56]

To detect if each point is surrounded, the current Pareto set is sorted one objective at a time. The points corresponding to the maximum and minimum in each objective are identified as extreme points. Then, a one-dimensional search of the objective space is conducted in which differences in the objective function values are compared to the respective indifference value for that objective. If differences in objective function value between points are larger than ω_i, the closest point above or below the current point is found. If the difference in objective function value for that point and the current is also larger than ω_i, a gap is considered to be found. This point is then used as a starting point in the MADS method in order to fill the gap.

2.6. Four Classes of Design Problems

This section describes four classes of design problems that represent real-world applications of multi-objective stochastic mixed-variable optimization.

2.6.1. System Reliability Optimization

This class of problems is usually characterized by the competing objectives of increasing reliability, availability, or a similar metric, while minimizing overall cost, weight, etc. Since component reliability cannot be known with absolute certainty, and system performance is usually modeled in a simulation environment, system reliability optimization is an instance of stochastic optimization. Additionally, some design variables are likely to be discrete or categorical (such as number and type of components), while others are continuous (such as cost and weight). To solve this class of problems, usually some form of robust parameter optimization is applied in which an additional objective is included to minimize performance variance. A specific example of this class is investigated in Chapter IV.

2.6.2. Alloy Composition

Alloy composition optimization problems are used to predict viable or promising chemical experiments. Usually, the competing objectives include maximizing alloy strength at high temperature while minimizing weight, cost, etc. Given a set of experimental results from previously tested chemical compositions, the goal of the optimization problem is to use stochastic multi-objective optimization techniques to identify a boundary to the Pareto front in order to predict dominated compositions that would thus be not worthwhile to construct and test in the laboratory.

Since the designs under test include categorical variables such as specific metal or chemical component as well as continuous variables such as amount by weight, alloy composition optimization is studied further in Chapter IV.

2.6.3. Algorithm Parameter Selection

Algorithm parameter selection refers to a class of problems in which the goal is to improve the performance of a parameter-based meta-heuristic (such as Tabu Search or

Simulated Annealing) or some stochastic algorithm. While many parameters may be discrete (such as maximum iterations), some may be categorical (such as type of cross-over for a Genetic Algorithm) or continuous (such as temperature or cooling for Simulated Annealing). Since there are many measures of algorithm performance (including time to completion, accuracy of result, etc.), there is usually no single set of parameter values that will simultaneously optimize all performance measures. Furthermore, since these algorithms themselves are stochastic in nature, their performance should not be optimized using deterministic techniques. However, since it is common to apply response surface methodology or robust parameter design techniques to this class of problems, a specific instance of this class is presented in Chapter IV to compare current results to those obtained using a method such as MVMADS-RS.

2.6.4. Structural Design

Structural design problems involve optimizing competing objectives such as minimum weight or volume and maximum load supported in order to design a support structure or gearbox. Design variables include component lengths, component offset or separation, and the number of components or sub-structures. Since these variables can be grouped into discrete or continuous sets (depending on the problem), structural design problems may be considered mixed-variable in nature. Additionally, if any of the objective functions cannot be measured with complete certainty, a structural design problem may require stochastic optimization techniques. Therefore, an instance of the gearbox design problem is investigated in Chapter IV.

III. Methodology

This chapter outlines the basic approach used to approximate the Pareto optimal set for stochastic mixed-variable design problems as well as the development of a suitable test implementation of this new method. The algorithms described in this chapter were designed to improve or expand upon the methods developed in SMOMADS and NMADS, while using the same MADS-based optimization technique. Specifically, the Pareto front generation algorithm developed in this chapter makes use of the SMOMADS method of finding objective-wise optima, but diverges from SMOMADS in that there is no use of decision maker input in selecting desirable or undesirable objective function values. The Pareto front gap-finding algorithm developed in this chapter leverages the NMADS method of sorting Pareto optimal solutions one objective at a time, but searches for multiple neighbors, instead of performing the single neighbor search used in NMADS.

3.1. Basic Approach

The basic approach used to approximate the Pareto optimal set for a stochastic mixed-variable problem is to use MVMADS-RS or MVPS-RS to find the objective-wise extrema and to use adaptive constraint generation and multi-objective scalarization until enough Pareto optimal responses have been generated so that the approximated Pareto front satisfies some minimum density and resolution. Although this approach is similar to the approach developed by Paciencia [45], the algorithms developed in this chapter are unique, and they will be contrasted to previously developed methods.

3.2. Objective-wise Extreme Points

The first step in this process is to determine the utopia point for the multi-objective minimization problem. Using MVMADS-RS or MVPS-RS, each one of the m multiple objectives is independently optimized. For a single objective function, $F_i(x)$, where $i \in \{1,\ldots,m\}$, if x_i^* is the optimal solution to $F_i(x)$, and the minimal objective function value $F_i^*(x)$ is given by $F_i^*(x) = F_i(x_i^*)$, then the column vector F^{i*} corresponding to this solution in the m-dimensional objective space is given by

$$F^{i*} = \left[F_1(x_i^*), F_2(x_i^*), \ldots, F_m(x_i^*) \right]^T. \tag{3.1}$$

Since $i \in \{1,\ldots,m\}$, $F_i(x_i^*) = F_i^*(x)$ is an element of F^{i*}. Furthermore, since $F_i(x_i^*)$ is optimal, then

$$F_i(x_i^*) \le F_i(x), \ \forall x \in \Theta, \tag{3.2}$$

where Θ is the feasible region of the original multi-objective optimization problem. Therefore, there is no $x \in \Theta$ such that $F_i(x) < F_i(x_i^*)$, and F^{i*} is Pareto optimal by Definition 2.1.1. Thus, the objective-wise extreme points of the Pareto front are given by (3.1) for $i \in \{1,\ldots,m\}$.

These extreme points are used to set bounds on the region to search for Pareto optimal solutions within the objective space. Let U be the matrix composed of the objective-wise extreme column vectors such that

$$U = \left[F^{1*}, \ldots, F^{m*} \right] = \begin{bmatrix} F_1(x_1^*) & \cdots & F_1(x_m^*) \\ \vdots & \ddots & \vdots \\ F_m(x_1^*) & \ldots & F_m(x_m^*) \end{bmatrix}. \tag{3.3}$$

Thus, the upper bound b^u for each objective function F_j in the Pareto set is given by

$$b_j^u = \max\{F_j(x_1^*),\ldots,F_j(x_m^*)\},\ j\in\{1,\ldots,m\} \tag{3.4}$$

and the lower bound b^l for each objective function F_j in the Pareto set is given by

$$b_j^l = \min\{F_j(x_1^*),\ldots,F_j(x_m^*)\},\ j\in\{1,\ldots,m\}. \tag{3.5}$$

For example, Figure 3.1 shows b^u and b^l for a bi-objective case.

3.2.1. Re-Orienting the Objective Space

Once the objective-wise upper and lower bounds of the Pareto front are known, they can be used to begin searching the objective space for more Pareto optimal solutions. However, in some instances, it may be useful to re-orient the objective space with an *affine transformation* so that the lower bound of each objective function is at the origin, $O = \mathbf{0}$, of the projected space. An *affine transformation* is a transformation with all the characteristics of a linear transformation, except that the origin of the original vector space is not mapped to the origin of the transformed space. This is especially helpful if the feasible region of some objectives is in the positive axis direction, while the feasible region of other objectives is in the negative axis direction.

If the objective-wise extreme points are linearly independent, they can be used as a basis set for the objective vector space, and an affine transformation can be constructed so that all points in the objective space can be written as coordinates in relation to the objective-wise extreme vectors. This affine transformation is constructed by finding the appropriate offset point, X_O, that will become the origin of the transformed vector space. The appropriate offset point must satisfy

$$(F^{i*} - X_O)^T(F^{j*} - X_O) = 0,\ \forall\ i, j\in\{1,\ldots,m\}\ \ni\ i\neq j\ . \tag{3.6}$$

That is, the vector defined as the difference vector between an objective-wise extreme point and the offset point must be orthogonal to the difference vector from all other objective-wise extreme points and the offset point. This point solves the optimization problem

$$\min \sum_{i=1}^{m} \left[\left(F^{i*} - X_0 \right)^T \left(F^{j*} - X_0 \right) \right]^2 \tag{3.7a}$$

subject to

$$m x_i \leq \sum_{j=1}^{m} F_i(x_j^*), \ \forall \ i = 1, \ldots, m. \tag{3.7b}$$

The inequality (3.7b) guarantees that the objective-wise extreme points will be mapped to the positive coordinate axes in the transformed space.

If the objective-wise extreme points are linearly independent, then normalizing the vector components of the matrix U given in (3.3) gives the change of coordinates matrix,

$$I_O = \left[\frac{1}{\left| F^{1*} \right|} F^{1*}, \ldots, \frac{1}{\left| F^{m*} \right|} F^{m*} \right], \tag{3.8}$$

and the vector of new coordinates, p_O, of any point p from the original objective space is given by

$$p_O = I_O^{-1} \left(p - X_O \right). \tag{3.9}$$

Figure 3.2 illustrates the affine transformation and change of coordinates process.

It is important to note that if this affine transformation process is used to re-orient the objective space, all distances and angles between points in the original space will be maintained, so the shape of the Pareto front will remain unchanged through the transformation.

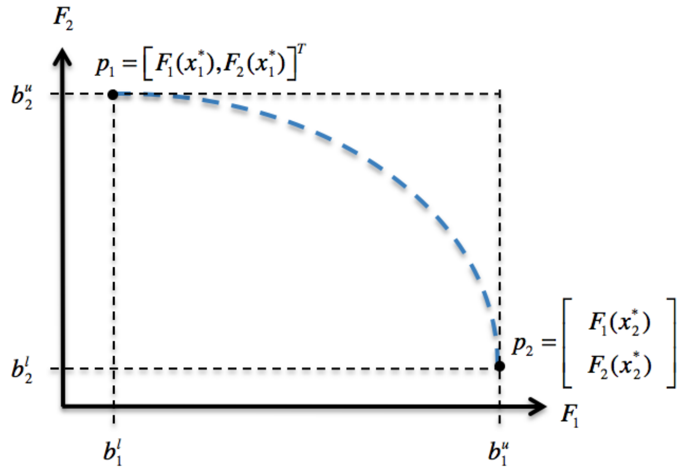

Figure 3.1: Upper and Lower Bounds for Bi-Objective Pareto front

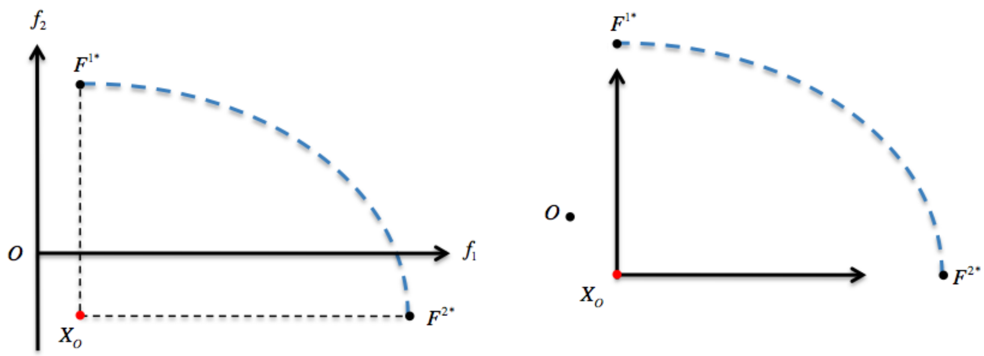

Figure 3.2: Affine Transformation of the Objective Space

3.3. Gap Finding

After the extreme points of the Pareto front have been discovered, the current Pareto set is checked for some threshold density of points within some resolution tolerance. This resolution tolerance is a function of the upper and lower bounds as formulated in (3.4) and (3.5), respectively.

3.3.1. Resolution Tolerance

In this formulation, the Pareto front resolution parameter, ρ, describes the number of sub-intervals of length $\rho \cdot \left\| b_i^l - b_i^u \right\|$ into which the objective space will be partitioned between the upper and lower bounds in each direction. The number of partitions is equal to $1/\rho$. For example, Figure 3.3 shows if $\rho = 1$, then each direction will have only one partition between b^u and b^l, but for $\rho = 0.25$, each direction will have four partitions of equal size between b^u and b^l. The effect this resolution tolerance has on Pareto front approximation is evaluated in Section 3.6.

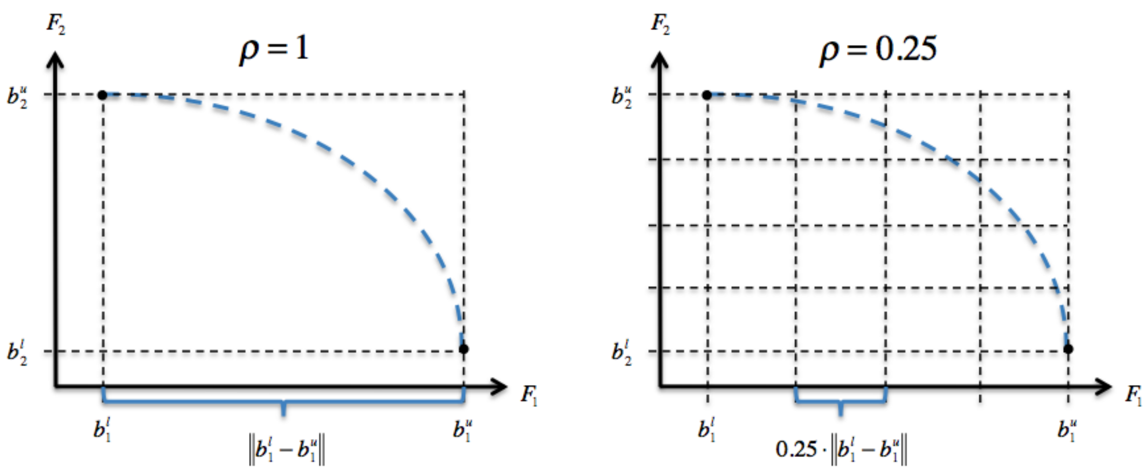

Figure 3.3: Partitioning for Resolution Parameter 1 and 0.25

3.3.2. Density Threshold

The density threshold P refers to the minimum number of Pareto optimal points desired within some resolution-related neighborhood in the objective space projected onto each $m-1$ dimensional sub-space, where m is the number of objectives in the multi-objective optimization problem. That is, when the Pareto set is projected onto each sub-space of dimension $m-1$, every member p of the projected Pareto set should have at least P other members of the projected Pareto set within some distance r_ρ^i. Formally, the density threshold is stated as

$$\left|\{q : \|(q-p)Q_i\| \le r_\rho^i\}\right| \ge \mathrm{P},\ \forall\ q, p \in S_E, \tag{3.10}$$

where S_E is the Pareto set, Q_i is the matrix that projects the Pareto set onto the $m-1$ dimensional objective space that does not include the F_i objective direction, and r_ρ^i is given by

$$r_\rho^i = \min\{\rho\left\|b_j^l - b_j^u\right\|,\ \forall\ j \in \{1,\ldots,m\}\ \ni\ j \ne i\} \tag{3.11}$$

That is, r_ρ^i is the smallest partition length of the $m-1$ objective directions onto which the Pareto set S_E has been projected. The effect these density threshold parameters have on approximating the Pareto front is discussed further in Section 3.6.

3.3.3. Gap Existence

If the density threshold is not met, a gap is considered to exist, and the gap filling algorithm must be used. Note the gap-finding technique presented here differs from that presented by Paciencia [45] in that he uses a one-dimensional gap finding algorithm that looks “above” and “below” each member of the Pareto set in one objective direction at a time to find a *single* nearest neighbor in that direction, while the method presented here partitions the objective space and uses the partition sizes to generate an $m-1$ dimensional neighborhood that must contain some sufficient number of elements from the Pareto set S_E. An example of the density threshold gap-finding algorithm is shown in Figure 3.4.

3.4. Gap Filling

Once a gap has been determined around some member p of the Pareto set S_E using the density threshold algorithm, a search neighborhood is constructed around p, and a series of scalarized optimization problems are solved iteratively using MVMADS-RS or MVPS-RS to increase the density of Pareto optimal points $q \in S_E$ around p. This gap filling algorithm uses an adaptive constraint formation that is based on the AWS for bi-objective problems developed by Kim and de Weck [34]. However, while AWS and the method presented in this section generate similar adaptive constraints for each sub-problem, the AWS sub-problem is to minimize a conventional weighted sum of the objective functions, while the sub-problem for the method presented here is to minimize the distance to a reference point.

Density Threshold Gap Finding

Step 0. Initialize algorithm parameters, including Pareto set S_E, test point $p \in S_E$, bounds b^u and b^l, resolution parameter ρ and density threshold P.

Step 1. For each objective direction i project S_E onto the sub-space that does not contain direction i and set

$$r_\rho^i = \arg\min\left\{\rho\left\|b_j^l - b_j^u\right\|,\ \forall\, j \in \{1,\ldots,m\} \ni j \neq i\right\}$$

Count the number of points $q \in S_E$ satisfying

$$\left\|(q-p)Q_i\right\| \leq r_\rho^i.$$

Step 2. If $\left|\left\{q : \left\|(q-p)Q_i\right\| \leq r_\rho^i\right\}\right| < \mathrm{P}$, a gap exists at p.

Step 3. Update p and return to Step 1. Stop when all points have been evaluated

Figure 3.4: Gap Finding Algorithm

3.4.1. Neighborhood Construction

For each point $p \in S_E$, if there exists some $i \in \{1,\dots,m\}$ such that (3.10) is not satisfied, determine the objective direction $\hat{i}_p$ in which p is "longest". That is, given the set $p_{\{\}}$ defined by the elements of the vector $p = \left[F_1(x_p),\dots,F_m(x_p)\right]$ such that

$$p_{\{\}} = \{F_1(x_p),\dots,F_m(x_p)\}, \tag{3.12}$$

$\hat{i}_p$ is given by

$$\hat{i}_p = \arg\max\{p_{\{\}}\}. \tag{3.13}$$

After $\hat{i}_p$ has been determined, define the set $S_E^p \subset S_E$ such that

$$S_E^p = S_E \setminus p \in S_E. \tag{3.14}$$

For all $j \in \{1,\dots,m\}$ such that $j \neq \hat{i}_p$, define the point $q_j \in S_E^p$ such that

$$F_j(x_{q_j}) = \arg\max\{F_j(x_q),\ \forall\ q \in S_E^p\}. \tag{3.15}$$

Define the set N_p of neighborhood vectors $N_p^j = q_j$ such that

$$N_p = \left\{N_p^j,\ \forall\, j \in \{1,\dots,m\}\ \ni\ j \neq \hat{i}_p\right\}. \tag{3.16}$$

Figure 3.5 gives an example of the neighborhood construction.

3.4.2. Neighborhood Search

For each member N_p^j of the neighborhood set N_p, an initial reference point, R_0^j, is defined such that

$$R_0^j = p - \left|F_{\hat{i}_p}(x_p) - F_{\hat{i}_p}(x_{N_p^j})\right| \cdot e_{\hat{i}_p}, \tag{3.17}$$

where $e_{\hat{i}_p} = \left[e_{\hat{i}_p}^1,\dots,e_{\hat{i}_p}^m\right]$ is the identity element vector such that

$$e_{\hat{i}_p}^k = \begin{cases} 1 & \text{if } k = \hat{i}_p \\ 0 & \text{otherwise} \end{cases}. \tag{3.18}$$

The idea is to replace the maximal component value of p with the corresponding component value from N_p^j in order to generate the reference point R_0^j. This reference point is updated iteratively based on the set of Pareto optimal solutions discovered using the initial reference point.

Given the initial reference point R_0^j defined from the neighbor vector N_p^j, a series of adaptive scalarized optimization problems is constructed:

$$\min \left\| R_t^j - Y(x) \right\| \tag{3.19a}$$

subject to

$$Y_{\hat{i}_p}(x) \le Y^k - \beta_{\hat{i}_p} \tag{3.19b}$$

$$Y_{\hat{i}_p}(x) \ge V_{\hat{i}_p} \tag{3.19c}$$

$$Y_i(x) \le b_i^u - \gamma\beta_i,\ i = 1, \ldots, m \tag{3.19d}$$

$$Y_i(x) \ge b_i^l,\ i = 1, \ldots, m \tag{3.19e}$$

$$x \in \Theta, \tag{3.19f}$$

where

$$Y^k = \begin{cases} F_{\hat{i}_p}(x_p) & \text{if } t = 0 \\ Y_{\hat{i}_p}(x_{k-1}^*) & \text{if } t > 0 \end{cases}, \tag{3.20}$$

$$\beta = \varepsilon \cdot \rho \cdot \left[\left\| b_1^l - b_1^u \right\|, \ldots, \left\| b_m^l - b_m^u \right\| \right], \tag{3.21}$$

$$\varepsilon \in (0,1), \tag{3.22}$$

$$\gamma \in (0,1), \tag{3.23}$$

$$V = R_t^j. \tag{3.24}$$

Since ε determines the maximum feasible value in the $\hat{i}_p$ direction, it is referred to as the *ceiling parameter*, while the value used to determine the maximum feasible value in all other directions, γ, is referred to as the *boundary parameter*.

Once an optimal solution x_k^* to (3.19a) has been found, it is added to the set G_E^p, the Pareto optimal points surrounding p. All iterative constraints are updated, and the resulting optimization problem is solved. This process continues until

$$Y^k - \beta_{\hat{i}_p} \leq V_{\hat{i}_p}. \tag{3.25}$$

Figure 3.6 depicts a notional example of this neighborhood search.

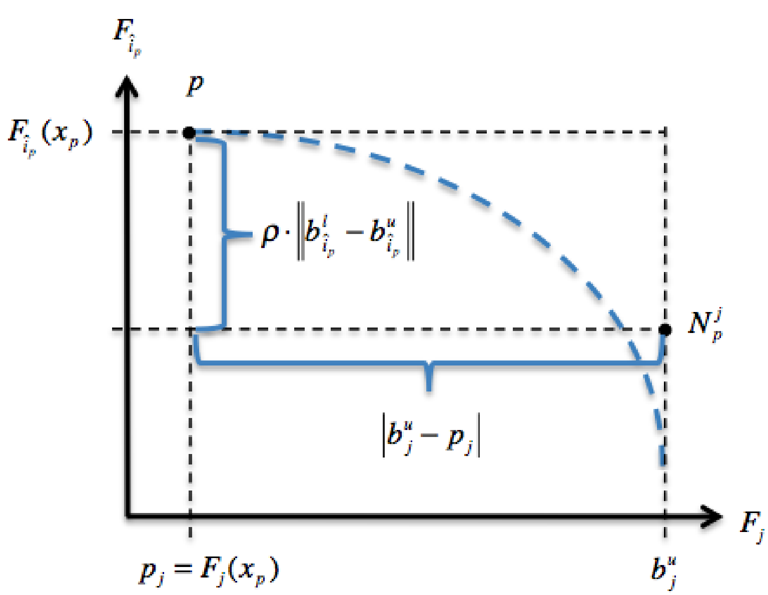

Figure 3.5: Neighborhood Construction

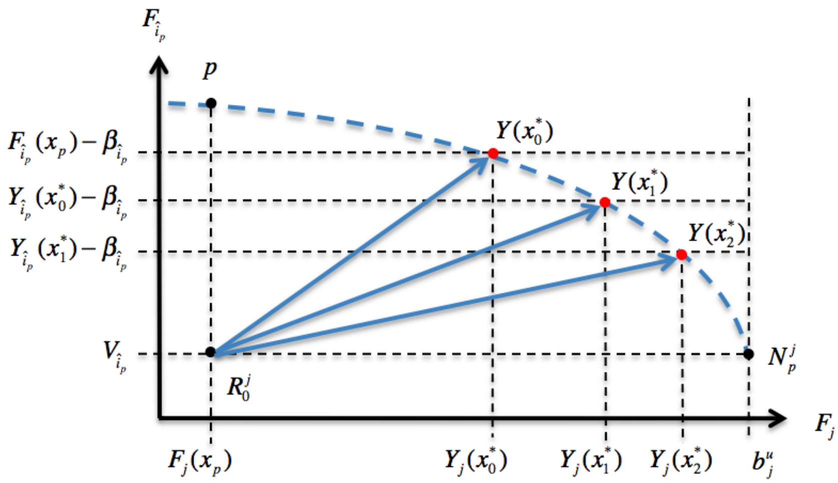

Figure 3.6: Notional Neighborhood Search

This series of searches may fail to find Pareto optimal points in the neighborhood, depending on the shape of the actual Pareto front. Figure 3.7 shows how a neighboring Pareto optimal solution could be missed by the search.

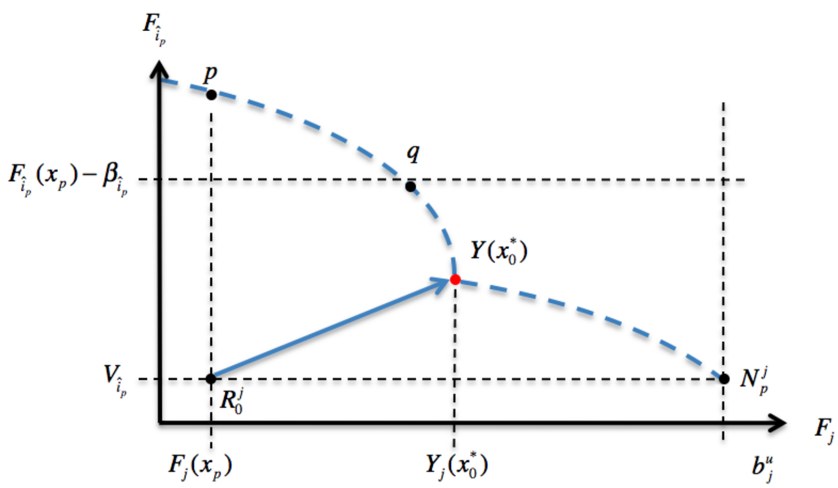

Figure 3.7: Point q missed by Neighborhood Search

In an effort to find points that may have been missed in the preceding series of searches, once the termination criteria (3.25) is met, a new reference point R_{t+1}^{j} is determined based on the set G_E^p such that

$$R_{t+1}^{j} = p - \left| F_{\hat{i}_p}(x_p) - g_{\hat{i}_p}^{R_t^j} \right| \cdot e_{\hat{i}_p}, \tag{3.26}$$

where

$$g^{R_t^j} = \hat{g} \in G_E^p \,\middle|\, \hat{g} = \arg\max\{F_{\hat{i}_p}(x_g),\ \forall\, g \in G_E^p\} \tag{3.27}$$

That is, $g^{R_t^j}$ is the member of G_E^p with the largest $\hat{i}_p$ component value in the set. This process of updating R_t^j is continued (and all new points are added to G_E^p) until

$$\left| F_{\hat{i}_p}(x_p) - g_{\hat{i}_p}^{R_t^j} \right| \le \rho \cdot \left\| b_{\hat{i}_p}^{l} - b_{\hat{i}_p}^{u} \right\|. \tag{3.28}$$

Figure 3.8 displays a notional update of R_t^j.

Once the termination criterion (3.28) is met, all members of G_E^p are added to the set G_E of all gap filling Pareto optimal points. Additionally, if there are more than three points in G_E^p, the three points representing the minimum, maximum and median $\hat{i}_p$ component values are selected to replace p in the set S_E, and p is added to the set G_E. Finally, all constraints and variables of the minimization sub-problem are reset to their initial values, and the process is continued for all remaining neighborhood vectors of N_p. The Pareto front estimation is considered complete when only one point fails to satisfy the neighborhood density threshold (3.10).

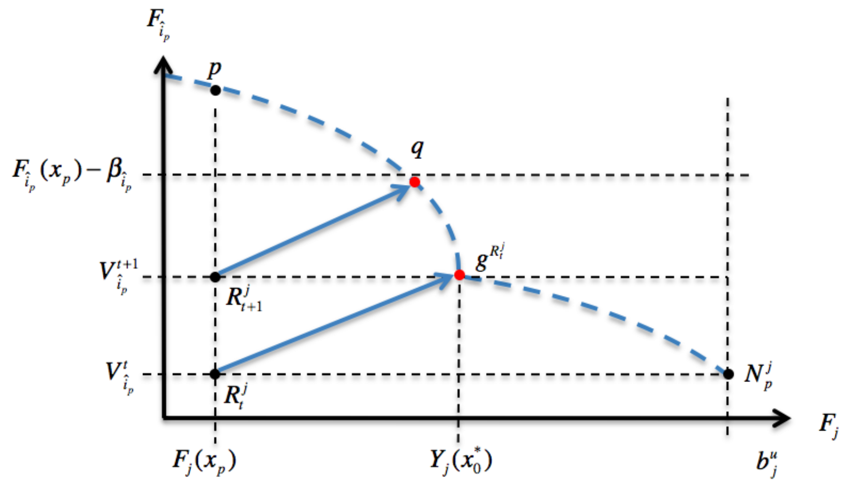

Figure 3.8: Update of Reference Point admits previously omitted q

3.5. Dominance Filtering

If the true optimal solution is found for (3.19a), no dominated point will be admitted to the set of Pareto optimal solutions using the adaptive constraint method developed in Section 3.4. However, since MVMADS-RS and MVPS-RS *estimate* best solutions for stochastic problems, it is possible to terminate the MVMADS-RS or MVPS-RS search with a sub-optimal solution, thus generating a dominated point in the adaptive constraint gap filling algorithm. Therefore, all generated points in the set G_E must be checked to see if they are dominated by any other point in the set, and all dominated points must be removed from G_E.

3.6. Parameter Investigation

This section describes a brief investigation to determine appropriate parameter settings for the gap checking and gap filling algorithms described in Sections 3.3 and 3.4, respectively. The parameters studied include the Density Threshold, P, the resolution parameter, ρ, and the adaptive constraint parameters, ε and γ.

In order to determine the effect of these parameters, the gap checking and gap filling algorithms were applied with varying parameter values to two known surfaces and the optimization sub-problems were evaluated using the MATLAB® Optimization toolbox to simulate the performance of MVMADS-RS and MVPS-RS.

The first test surface is the non-negative portion of a three-dimensional ellipsoid given by

$$\frac{x^2}{25}+\frac{y^2}{25}+z^2=1. \tag{3.29}$$

Figure 3.9 shows the ellipsoid region. This surface was investigated to evaluate algorithm performance in estimating a Pareto front that is entirely non-convex.

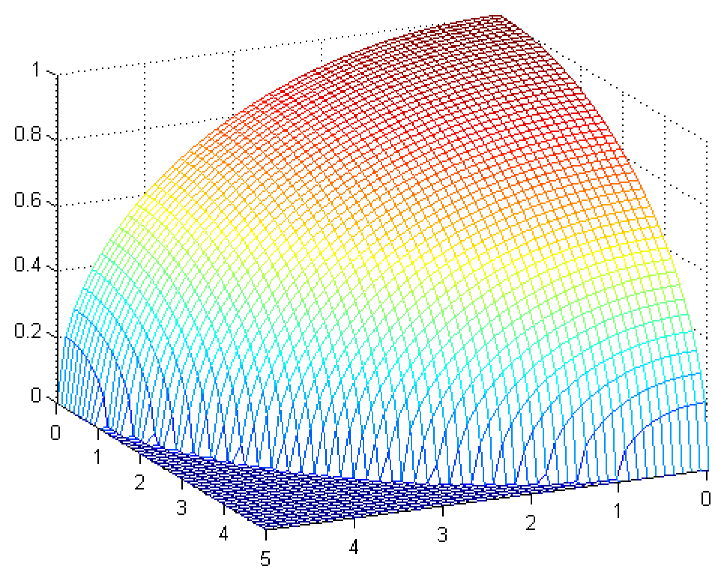

Figure 3.9: Simulated Ellipsoid Pareto Front

The second test surface is a non-negative three-dimensional curled sheath given by

$$z = \frac{1}{100}\left[(x-5)^2(y-5)^2 + x^2y^2\right] \tag{3.30}$$

Figure 3.10 shows the region of interest. This surface was investigated to evaluate algorithm performance in estimating a Pareto front that contains both convex and non-convex regions.

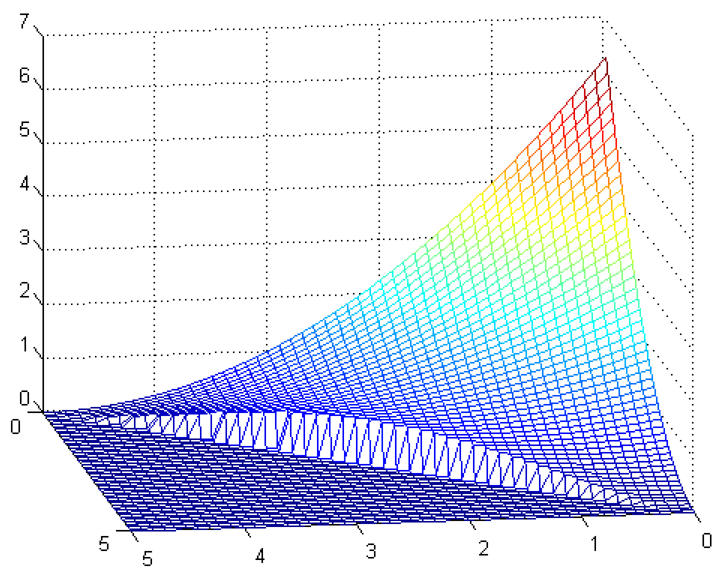

Figure 3.10: Simulated Curled Sheath Pareto Front

Table 3.1 displays the values assigned to each parameter of interest that were used to estimate both test surfaces. There are four parameters with four possible values each, giving a total of 256 unique algorithm implementations (or *runs*). The range of values for each parameter was empirically derived from initial experiments used to test the algorithm as it was being developed.

Table 3.1: Tested Algorithm Parameter Values

Name	Parameter Symbol	Tested Values			
Density Threshold	P	2,	4,	6,	8
Resolution Parameter	ρ	0.10,	0.15,	0.20,	0.25
Ceiling Parameter	ε	0.60,	0.65,	0.70,	0.75
Boundary Parameter	γ	0.20,	0.30,	0.40,	0.50

For the purposes of this research, the basic measure of algorithm performance is the total number of points generated. However, since the Density Threshold value directly determines how many points should be generated, for each Density Threshold value, the four runs that generated the most points for the ellipsoid test surface and the four runs that generated the most points for the curled sheath test surface were selected. Table 3.2 displays the four maximum point generating runs for each surface and Density Threshold value. The highlighted rows represent parameter combinations that resulted in a maximal number of points produced for both surfaces at some Density Threshold value (referred to as *best-in-common*). This indicates that the parameter combination corresponding to that row may be appropriate for estimating both convex and non-convex Pareto fronts. Figures 3.11 through 3.14 show the surface estimations corresponding to the best-in-common runs.

Table 3.2 indicates that for Density Threshold values $\mathrm{P} = \{2, 6, 8\}$, the best-in-common parameter combinations were identical, with

$$\rho = 0.10,\ \varepsilon = 0.75,\ \gamma = 0.20.$$

For $\mathrm{P} = 4$, the best-in-common parameter combination was given by

$$\rho = 0.10,\ \varepsilon = 0.75,\ \gamma = 0.30,$$

differing from the other best-in-common values only in the Boundary Parameter value.

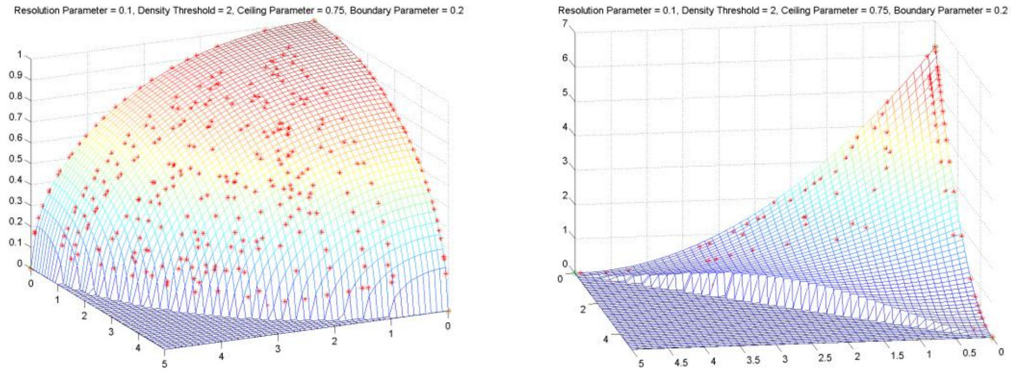

Figure 3.11: Surface Estimations for Run 13

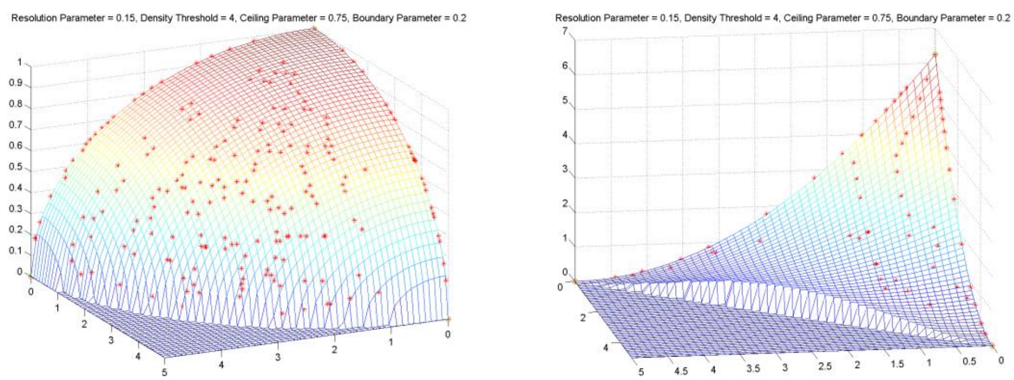

Figure 3.12: Surface Estimations for Run 78

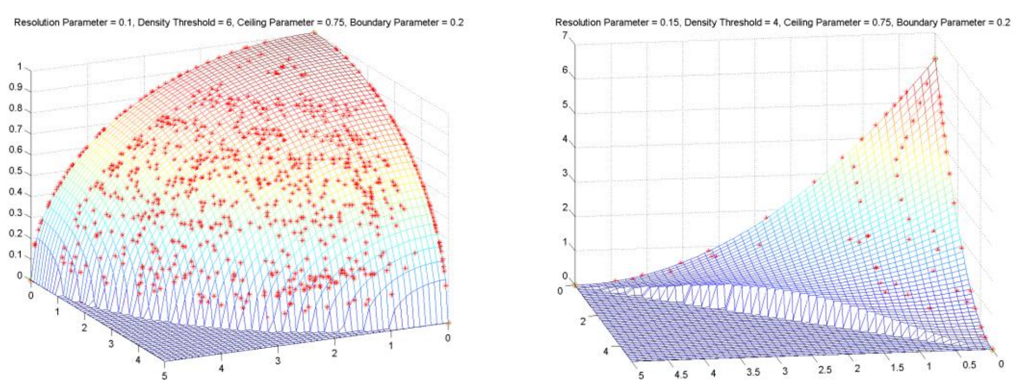

Figure 3.13: Surface Estimations for Run 141

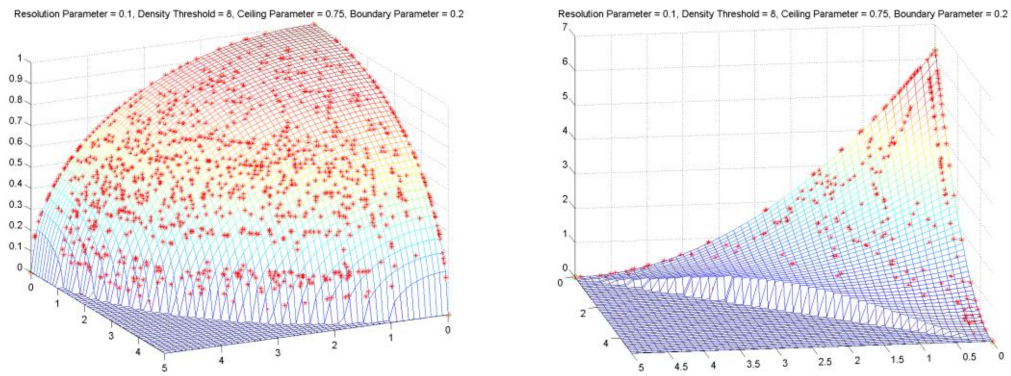

Figure 3.14: Surface Estimations for Run 205

Figures 3.11 through 3.14 indicate that overall coverage of the Pareto front improves at the best-in-common settings as Density Threshold increases. Therefore, two more experiments were conducted with $\mathrm{P} = 10,\ \rho = 0.10,\ \varepsilon = 0.75$, and $\gamma = \{0.20, 0.30\}$ to look for increased point coverage. However, there was no noticeable improvement in coverage compared to the $\mathrm{P} = 8$ best-in-common runs. The results for these runs are displayed in Table 3.3, and the surface estimations are shown in Figures 3.15 and 3.16.

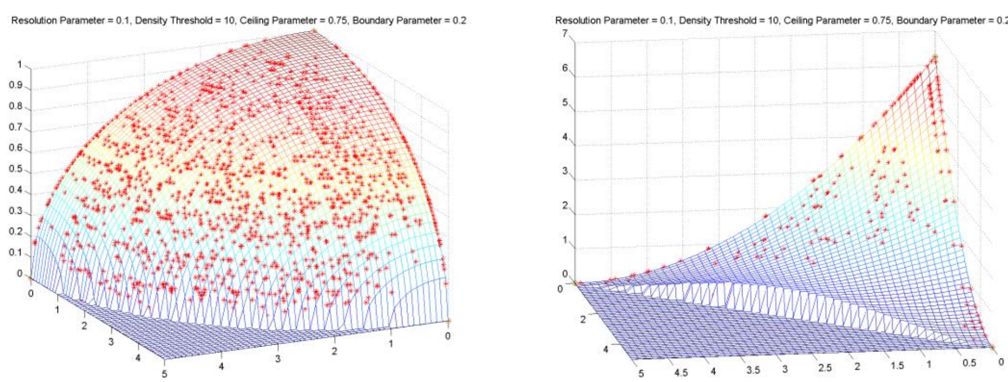

Figure 3.15: Surface Estimation for $\mathrm{P} = 10,\ \gamma = 0.20$

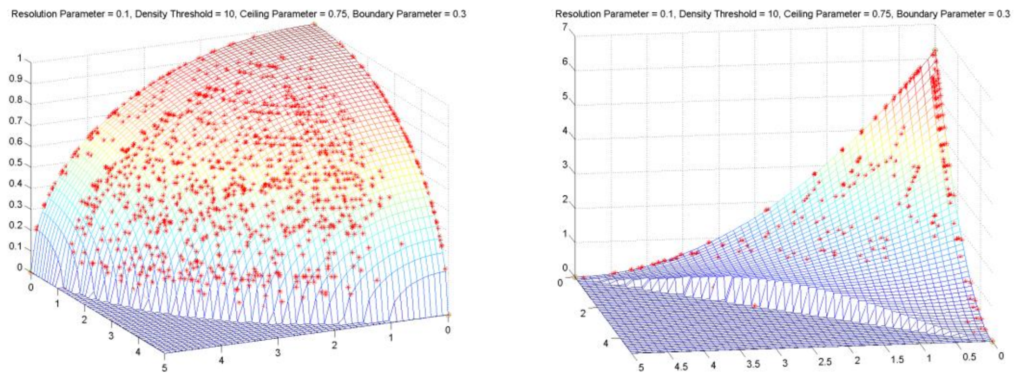

Figure 3.16: Surface Estimation for $\mathrm{P} = 10,\ \gamma = 0.30$

Table 3.2: Top Four Runs per Density Threshold

P	Surface	Run	ρ	ε	γ	Points
2	Ellipsoid	5	0.10	0.65	0.20	336
		15	0.10	0.75	0.40	399
		13	0.10	0.75	0.20	408
		6	0.10	0.65	0.30	440
	Curled Sheath	25	0.15	0.70	0.20	68
		11	0.10	0.70	0.40	68
		14	0.10	0.75	0.30	71
		13	0.10	0.75	0.20	96
4	Ellipsoid	74	0.10	0.70	0.30	782
		78	0.10	0.75	0.30	830
		73	0.10	0.70	0.20	845
		69	0.10	0.65	0.20	910
	Curled Sheath	80	0.10	0.75	0.50	118
		78	0.10	0.75	0.30	119
		79	0.10	0.75	0.40	130
		76	0.10	0.70	0.50	138
6	Ellipsoid	130	0.10	0.60	0.30	1301
		141	0.10	0.75	0.20	1418
		137	0.10	0.70	0.20	1438
		129	0.10	0.60	0.20	1539
	Curled Sheath	134	0.10	0.65	0.30	167
		140	0.10	0.70	0.50	190
		141	0.10	0.75	0.20	193
		138	0.10	0.70	0.30	209

P	Surface	Run	ρ	ε	γ	Points
8	Ellipsoid	202	0.10	0.70	0.30	1691
		205	0.10	0.75	0.20	1894
		197	0.10	0.65	0.20	1938
		193	0.10	0.60	0.20	1999
	Curled Sheath	199	0.10	0.65	0.40	230
		208	0.10	0.75	0.50	261
		206	0.10	0.75	0.30	266
		205	0.10	0.75	0.20	276

Table 3.3: Points Generate for $P = 10$

P	ρ	ε	Surface	γ	Points
10	0.10	0.75	Ellipsoid	0.20	2566
				0.30	2273
			Curled Sheath	0.20	250
				0.30	312

The parameter values for the algorithms that are used in Chapter IV are based on the experimental results from Section 3.6. The final Adaptive Constraint Pareto Set Estimation for Stochastic Mixed Variable Optimization (ACPSE-SMVO) algorithm is shown in Figure 3.17.

3.7. Summary

This chapter developed the algorithms and techniques that were used to estimate the Pareto front for the problems that were researched. The basic idea is to optimize each objective function one at a time and use the resulting points as bounds on the area to search adaptively for more Pareto optimal solutions. Chapter IV presents the results of applying the methodology and the specific algorithm parameters developed in this chapter to four mixed variable, stochastic multi-objective optimization problems.

Final ACPSE-SMVO Algorithm

- **INITIALIZATION:** Determine m objective-wise extreme points $p \in S_E$ and $\{b^u, b^l\}$ for S_E. Set $\mathrm{P} = 6$, $\rho = 0.10$, $\varepsilon = 0.75$, $\gamma = 0.20$, $G_E = \varnothing$.
- **GAP CHECK:** If $|S_E| \le 1$, proceed to **TERMINATION**. Otherwise, for each objective function $F_i, i \in \{1, \ldots, m\}$, determine Q_i, the matrix that projects S_E onto the $m-1$ dimension objective space that does not include the F_i objective direction, and set

$$r_\rho^i = \min\{\rho \|b_j^l - b_j^u\|, \ \forall\, j \in \{1, \ldots, m\} \ \ni \ j \ne i\}$$

    - For each $p \in S_E$, set $\mathrm{P}_p = \left|\left\{q : q \in S_E \cup G_E,\ q \ne p, \text{ and } \|(q-p)Q_i\| \le r_\rho^i\right\}\right|$.
    - If $\mathrm{P}_p < \mathrm{P}$ proceed to **GAP FILL**. Otherwise, remove p from S_E, set $G_E = G_E \cup \{p\}$, and continue **GAP CHECK**.
- **GAP FILL**: Determine $\hat{i}_p$ and construct search neighborhood N_p (see Section 3.4.1)
    - For each $N_p^j \in N_p$, determine R_0^j according to (3.13).
    - Given R_0^j, ρ, ε, and γ, perform optimization series (3.15) and construct G_E^p until criteria (3.24) is met. Remove p from S_E and set $G_E = G_E \cup \{p\}$.
    - If $|G_E^p| < 3$, set $S_E = S_E \cup G_E^p$. Otherwise, set $S_E = S_E \cup \{\widehat{q}, \breve{q}, \underset{\smile}{q} \in G_E^p\}$, where $\widehat{q}, \breve{q}, \underset{\smile}{q} \in G_E^p$ are the three points representing the maximum, median and minimum $\hat{i}_p$ component values in G_E^p.
    - Set $G_E = G_E \cup G_E^p$ and proceed to **GAP CHECK**.
- **TERMINATION**: Remove all dominated points from G_E and return final Pareto set estimation, G_E.

Figure 3.17: ACPSE-SMVO Algorithm

IV. Results and Analysis

This chapter presents the results of applying the ACPSE-SMVO algorithm developed in Chapter III to four optimization problems: an alloy composition problem, a gearbox design problem, an algorithm parameter optimization problem, and a system reliability problem. For each case, the background information and analysis and the ultimate optimization problem formulation are presented. Finally, the optimization results are analyzed.

4.1. Case Study Formulations

4.1.1. Alloy Composition Problem

The alloy composition problem presented here is based on the data and studies performed by Dulikravich, *et al.* [24,25,26,27]. Their study involved analyzing the melting temperature T_1, glass transition temperature T_g, and density, ρ,of 53 different alloy compositions in order to predict a combination of seven different alloy compounds that would maximize T_1 and T_g, and minimize ρ.

To formulate the optimization problem, the 53 data points were used to generate a regression function for the three responses based on the seven different compound variables. The 53 data points are shown in Table 4.1. The value of each variable represents the percentage by weight of the compound in the resulting alloy; therefore, the decision variables are inherently continuous. However, one of the variables occurs at only three levels, so it was treated as a discrete variable with allowable values at 0, 16.5, and 16.8.

A stepwise regression was performed in JMP version 8.0 for each response. Table 4.2 summarizes the fit of the resulting regression models for each response (the model regression equations are provided in Appendix A as they appeared in the MATLAB® code used for this research). The error in the regression model was used to define a normally distributed random variable with a mean of zero and a standard deviation equal to the root-mean-square error (RMSE) of the regression model. This random variable, $\varepsilon_i \sim N(0, RMSE_i)$, was used to represent the stochastic element in the mixed variable optimization problem.

The formulation of the alloy composition optimization problem is given by

$$\min \left(-T_1(x)+\varepsilon_1, -T_g(x)+\varepsilon_2, \rho(x)\right) \tag{4.1a}$$

subject to

$$\sum_{i=1}^{7} x_i = 100 \tag{4.1b}$$

$$x_i \geq 0 \tag{4.1c}$$

Table 4.1: Alloy Composition Data Set [25]

x_1	x_2	x_3	x_4	x_5	x_6	x_7	$-F_1$	$-F_2$	F_3
% Zr	%Cu	%Al	%La	%Cu,Ni	%Pd	%Si	T_1	T_g	ρ
50	36	14	0	0	0	0	724	1188	6.8636
50	38	12	0	0	0	0	722	1170	6.9888
50	40	10	0	0	0	0	714	1176	7.114
50	43	7	0	0	0	0	703	1181	7.3018
49	44	7	0	0	0	0	704	1184	7.3262
48	45	7	0	0	0	0	708	1186	7.3506
49	45	6	0	0	0	0	704	1187	7.3888
48	46	6	0	0	0	0	706	1192	7.4132
49	46	5	0	0	0	0	701	1195	7.4514
49	47	4	0	0	0	0	697	1208	7.514
45	49	6	0	0	0	0	717	1178	7.4864
45	50	5	0	0	0	0	714	1185	7.549
44	51	5	0	0	0	0	719	1189	7.5734
45	48	7	0	0	0	0	720	1188	7.4238
45	47	8	0	0	0	0	722	1195	7.3612

x_1	x_2	x_3	x_4	x_5	x_6	x_7	$-F_1$	$-F_2$	F_3
% Zr	%Cu	%Al	%La	%Cu,Ni	%Pd	%Si	T_1	T_g	ρ
46	49	5	0	0	0	0	711	1193	7.5246
47	49	4	0	0	0	0	704	1204	7.5628
54	38	8	0	0	0	0	692	1190	7.1416
56	36	8	0	0	0	0	685	1212	7.0928
52	38	10	0	0	0	0	705	1163	7.0652
54	36	10	0	0	0	0	698	1176	7.0164
54	40	6	0	0	0	0	684	1216	7.2668
0	0	12.4	70	17.6	0	0	403	759	6.216
0	0	13.2	68	18.8	0	0	407	742	6.2212
0	0	14	66	20	0	0	405	674	6.2265
0	0	14.6	64.6	20.8	0	0	414	696	6.2277
0	0	15.2	63.1	21.7	0	0	420	699	6.2316
0	0	15.7	62	22.3	0	0	422	722	6.2308
0	0	15.9	61.4	22.7	0	0	426	729	6.2348
0	0	16.3	60.5	23.2	0	0	423	727	6.2347
0	0	16.6	59.6	23.8	0	0	426	743	6.2408
0	0	17	58.6	24.4	0	0	431	764	6.2434
0	0	17.5	57.6	24.9	0	0	435	783	6.2399
0	0	17.9	56.5	25.6	0	0	440	813	6.2452
0	0	18.4	55.4	26.2	0	0	436	844	6.2444
0	0	20.5	50.2	29.3	0	0	435	930	6.2568
0	0	14	70	16	0	0	404	763	6.1166
0	0	14	68	18	0	0	405	724	6.1716
0	0	14	66	20	0	0	405	674	6.2265
0	0	14	64	22	0	0	411	715	6.2814
0	0	14	62	24	0	0	417	738	6.3363
0	0	14	59	27	0	0	422	773	6.4187
0	0	14	57	29	0	0	427	815	6.4736
0	2	0	0	0	81.5	16.5	633	1097.3	10.3624
0	4	0	0	0	79.5	16.5	635	1086	10.3011
0	6	0	0	0	77.5	16.5	637	1058.1	10.2398
0	8.2	0	0	0	75	16.8	645	1135.9	10.1434
0	10.2	0	0	0	73	16.8	652	1153.6	10.0821
0	36	14	50	0	0	0	428	862.7	6.6846
0	26	14	60	0	0	0	404	785.6	6.4048
0	20	14	66	0	0	0	395	731	6.2369
0	14	14	72	0	0	0	391	792.7	6.069
0	10	14	76	0	0	0	361	825.5	5.9571

Table 4.2: Summary of Regression Fit

T_1 Regression Model	
RSquare	0.99548
RSquare Adj	0.993973
Root Mean Square Error	16.56018
Mean of Response	971.9132
Observations	53
T_g Regression Model	
RSquare	0.999394
RSquare Adj	0.999231
Root Mean Square Error	3.965904
Mean of Response	557.5283
Observations (or Sum Wgts)	53

Density Regression Model	
RSquare	1
RSquare Adj	1
Root Mean Square Error	0.000018
Mean of Response	7.067474
Observations (or Sum Wgts)	53

4.1.2. Gearbox Design Problem

The gearbox design problem presented here is based on the dual-shaft problem formulated by Azarm, *et al.* [10]. Their summary states that the first objective is to minimize the gearbox volume, while the second and third objectives are to minimize the stresses on the first and second shafts, respectively. The formulation of the gearbox design optimization problem is given by

$$\min \ \left(F_1(x), F_2(x), F_3(x)\right) \tag{4.2a}$$

subject to

$$2.6 \le x_1 \le 3.6, \tag{4.2b}$$

$$0.7 \le x_2 \le 0.8, \tag{4.2c}$$

$$17 \le x_3 \le 28, \tag{4.2d}$$

$$7.3 \le x_4 \le 8.3, \tag{4.2e}$$

$$7.3 \le x_5 \le 8.3, \tag{4.2f}$$

$$2.9 \le x_6 \le 3.9, \tag{4.2g}$$

$$5.0 \le x_7 \le 5.5, \tag{4.2h}$$

$$27 x_1^{-1} x_2^{-1} x_3^{-1} - 1 \le 0, \tag{4.2i}$$

$$397.5 x_1^{-1} x_2^{-1} x_3^{-2} - 1 \le 0, \tag{4.2j}$$

$$5 x_2^{-1} x_3^{-1} x_4^{3} x_6^{-4} - 1 \le 0, \tag{4.2k}$$

$$50 x_2^{-1} x_3^{-1} x_5^{3} x_7^{-4} - 1 \le 0, \tag{4.2l}$$

$$40 x_2 x_3 - 1 \le 0, \tag{4.2m}$$

$$12x_1 / x_2 - 1 \le 0, \tag{4.2n}$$

$$1 - 5x_1 / x_2 \le 0, \tag{4.2o}$$

$$1.9\left(x_4 - 1.5x_6\right) - 1 \le 0, \tag{4.2p}$$

$$1.9\left(x_5 - 1.1x_7\right) - 1 \le 0, \tag{4.2q}$$

where x_1 is the gear face width (cm), x_2 is the teeth module (cm), x_3 is the number of teeth of pinion (discrete value), x_4 is the distance between bearings 1 (cm), x_5 is the distance between bearings 2 (cm), x_6 is the diameter of shaft 1 (cm), and x_7 is the diameter of shaft 2 (cm). The constraint (4.2i) is the upper bound of the bending stress of the gear tooth, constraint (4.2j) is the upper bound of the contact stress of the gear tooth, constraints (4.2k) and (4.2l) are the upper bounds of the transverse deflection of the two shafts, constraints (4.2m) - (4.2o) are the dimensional restrictions based on space, and constraints (4.2p) and (4.2q) are empirically-based design requirements on the shafts. Finally, the objective functions are defined as

$$F_1(x) = \begin{array}{l} 0.7854\left(x_4 x^2{}_6 + x_5 x^2{}_7 + x_1 x_2{}^2\left(3.333x^2{}_3 + 14.93x_3 - 43.09\right)\right) \\ -1.508x_1\left(x^2{}_6 + x^2{}_7\right) + 7.477\left(x^3{}_6 + x^3{}_7\right) \end{array}, \tag{4.3a}$$

$$F_2(x) = \left(1 + \varepsilon_2\right)\frac{\sqrt{\left(\dfrac{745x_4}{x_2 x_3}\right)^2 + 1.69 \times 10^7}}{0.1x_6^3}, \tag{4.3b}$$

$$F_3(x) = \left(1 + \varepsilon_3\right)\frac{\sqrt{\left(\dfrac{745x_5}{x_2 x_3}\right)^2 + 1.575 \times 10^7}}{0.1x_7^3}, \tag{4.3c}$$

where ε_2 and ε_3 are both normally distributed random variables with a mean of zero and a standard deviation of 0.01.

4.1.3. Algorithm Parameter Optimization Problem

The algorithm parameter optimization problem studied in this research is based on the AutoGAD algorithm developed by Johnson [33]. Table 4.3 gives the algorithm's parameters and their ranges. Table 4.4 gives the algorithm performance measures that are the optimization problem objective functions. A D-optimal design was developed by Davis [23] to investigate the relationship between the parameter settings and the algorithm performance measures. His results (displayed in Appendix B) were used to perform a stepwise regression in JMP version 8.0 to model each response as a function of the parameters. Table 4.5 summarizes the fit of the resulting regression models for each response (the model regression equations are provided in Appendix A as they appeared in the MATLAB® code used for this research). The error in the regression model was used to define a normally distributed random variable with a mean of zero and a standard deviation equal to the root mean square error of the regression model. This random variable was used to represent the stochastic element in the mixed variable optimization problem.

Table 4.3: Algorithm Parameters as Optimization Variables

x	**Parameter**	**Type**	**Range**
x_1	Dimension Adjust	Discrete	[-2, 2]
x_2	Max Score Threshold	Continuous	[6, 14]
x_3	Bin Width SNR	Continuous	[0.01, 0.1]
x_4	PT SNR Threshold	Continuous	[1, 6]
x_5	Bin Width Identify	Continuous	[0.01, 0.1]
x_6	Smooth Iterations High	Discrete	[50, 150]

x	Parameter	Type	Range
x_7	Smooth Iterations Low	Discrete	[5, 45]
x_8	Low SNR	Continuous	[4, 14]
x_9	Window Size	Discrete	[1, 9]
x_{10}	Threshold Both Sides	Categorical	[0, 1]
x_{11}	Clean Signal	Categorical	[0, 1]

Table 4.4: Algorithm Performance as Objective Functions

F	Performance Measure	Units	Range
F_1	Time to completion	Seconds	$[0, \infty)$
F_2	False Positive Fraction (FPF)	N/A	[0, 1]
F_3	True Positive Fraction (TPF)	N/A	[0, 1]
F_4	Total Fraction of Positives (TFP)	N/A	[0, 1]

Table 4.5: Summary of Regression Fit

Time Regression Model	
RSquare	0.998666
RSquare Adj	0.994182
Root Mean Square Error	2.839837
Mean of Response	19.92367
Observations	158
FPF Regression Model	
RSquare	0.954398
RSquare Adj	0.914768
Root Mean Square Error	0.006655
Mean of Response	0.010716
Observations (or Sum Wgts)	158
TPF Regression Model	
RSquare	0.965074
RSquare Adj	0.921666
Root Mean Square Error	0.115054
Mean of Response	0.640272
Observations (or Sum Wgts)	158
TFP Regression Model	
RSquare	0.919272
RSquare Adj	0.852624
Root Mean Square Error	0.16359
Mean of Response	0.502776
Observations (or Sum Wgts)	158

4.1.4. System Reliability Optimization

The system reliability optimization problem studied in this research is a notional example based on the problems presented by Coit, *et al.* [18]. For this problem, the goal is to maximize the reliability of a system with two sub-systems in series, each consisting of a variable number of components in parallel, to minimize the cost of the system, and to minimize the total weight of the system. Since the basic reliability system can be formulated as a 0-1 integer programming problem, two continuous variables were included to force a mixed-variable formulation. Additionally, an artificial noise component was added to the reliability objective function to simulate the stochastic nature of an actual component reliability optimization problem. The system reliability optimization problem is given as

$$\min \ \left(F_1(x), F_2(x), F_3(x)\right) \tag{4.4a}$$

subject to

$$0 \le x_1 \le 3, \tag{4.4b}$$

$$0 \le x_2 \le 3, \tag{4.4c}$$

$$0 \le x_3 \le 2, \tag{4.4d}$$

$$0 \le x_4 \le 3, \tag{4.4e}$$

$$0 \le x_5 \le 3, \tag{4.4f}$$

$$0 \le x_6 \le 2, \tag{4.4g}$$

$$0 \le x_7 \le 50, \tag{4.4h}$$

$$0 \le x_8 \le 50, \tag{4.4i}$$

where x_1 is the number of components of type 1 in sub-system 1, x_2 is the number of components of type 2 in sub-system 1, x_3 is the number of components of type 3 in sub-system 1, x_4 is the number of components of type 1 in sub-system 2, x_5 is the number of components of type 2 in sub-system 2, x_6 is the number of components of type 3 in sub-system 2, x_7 is the operating temperature of sub-system 1, and x_8 is the operating temperature of sub-system 2. Finally, the objective functions are defined as

$$F_1(x) = ((1-R_{1,1})^{x_1}(1-R_{2,1})^{x_2}(1-R_{3,1})^{x_3})((1-R_{1,2})^{x_4}(1-R_{2,2})^{x_5}(1-R_{3,2})^{x_6}), \tag{4.5a}$$

$$F_2(x) = (1+\varepsilon_2)\left(\begin{array}{c} (50-x_7)(x_1+x_2+x_3)+(50-x_7)^2(x_4+x_5+x_6) \\ +5(x_1+x_4)+10(x_2+x_5)+20(x_3+x_6) \end{array}\right), \tag{4.5b}$$

$$F_3(x) = (1+\varepsilon_3)(20(x_1+x_4)+10(x_2+x_5)+5(x_3+x_6)), \tag{4.5c}$$

where F_1 is the unreliability of the system, F_2 is the total system cost, F_3 is the total system weight, ε_2 and ε_3 are both normally distributed random variables with a mean of zero and a standard deviation of 0.005, and $R_{i,j}$ is the mean reliability of component *i* in sub-system *j* given as a function of the sub-system operating temperature such that

$$R_{1,1} = 0.8 - 0.001x_7, \tag{4.6a}$$

$$R_{2,1} = 0.85 - 0.001x_7, \tag{4.6b}$$

$$R_{3,1} = 0.89 - 0.005x_7, \tag{4.6c}$$

$$R_{1,2} = 0.8 - 0.001x_8, \tag{4.6d}$$

$$R_{2,2} = 0.85 - 0.001x_8, \tag{4.6e}$$

$$R_{3,2} = 0.89 - 0.005x_8. \tag{4.6f}$$

4.2. Case Study Evaluation

The ACPSE-SMVO method (described in Chapter III) was applied to each of the four test problems described in Section 4.1 to estimate their respective Pareto optimal fronts. For each problem, the objective-wise optima were estimated for each objective function using Abramson's [2] NOMADm MATLAB® implementation of MVMADS-RS. For the Alloy Composition and Algorithm Parameter optimization problems, the objective-wise optima were used to re-orient the objective space using the affine transformation described in Section 3.2.1. After the objective-wise optima were estimated, the ACPSE-SMVO process was continued until either 200 estimated Pareto optimal solutions had been found, or the objective space density threshold had been met. Finally, the set of estimated Pareto optimal solutions was filtered to remove all dominated points.

4.2.1. Alloy Composition Problem

The ACPSE-SMVO process was completed five times for the Alloy Composition problem. The objective-wise extreme points for each replication are given in Table 4.6. Figures 4.1 - 4.10 display the estimated and filtered Pareto sets for each replication.

Although the plots for the replications appear to vary greatly in estimating the Pareto front for the Alloy Composition problem, the points generated for each replication seem to be all estimating a planar triangle in the objective space, with vertices at the estimated objective-wise extreme points.

Table 4.6: Alloy Composition Extreme Points

Replication	$-T_1$	$-T_g$	ρ
1	-156505.1103	-6109.48505	7.667573253
	-135466.4663	-6456.405323	7.835653776
	0	0	2.63636809
2	-156282.5365	-3978.948826	6.944260729
	-156233.1607	-3994.836093	6.941744748
	0	0	2.63636809
3	-156326.9172	-5221.777148	5.825269187
	-156205.2871	-5281.644979	5.816981886
	0	0	2.637323064
4	-156626.9106	-5081.340907	7.60019441
	-156083.0101	-5224.865222	5.814228964
	0	0	2.63636809
5	-156285.7669	-4000.42395	6.941473361

Replication	$-T_1$	$-T_g$	ρ
	-1185.513279	-4177.312583	7.843737884
	0	0	2.657931848

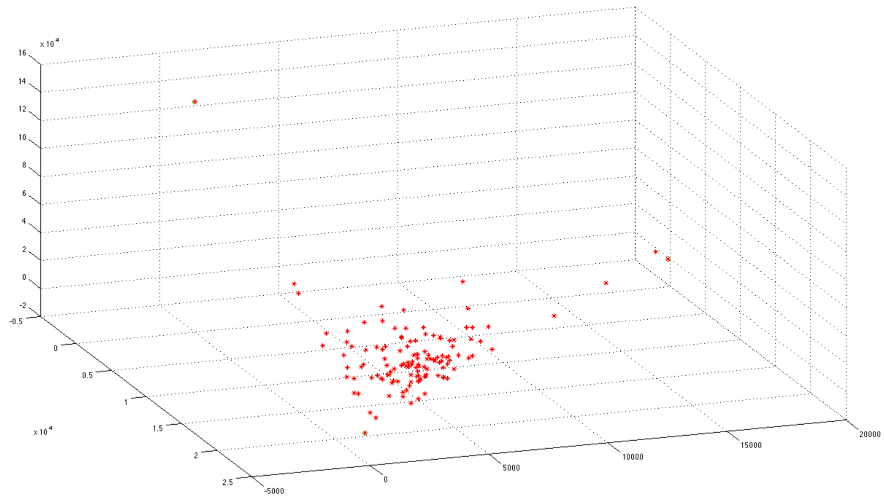

Figure 4.1: Alloy Composition Replication 1 Estimated Pareto Set

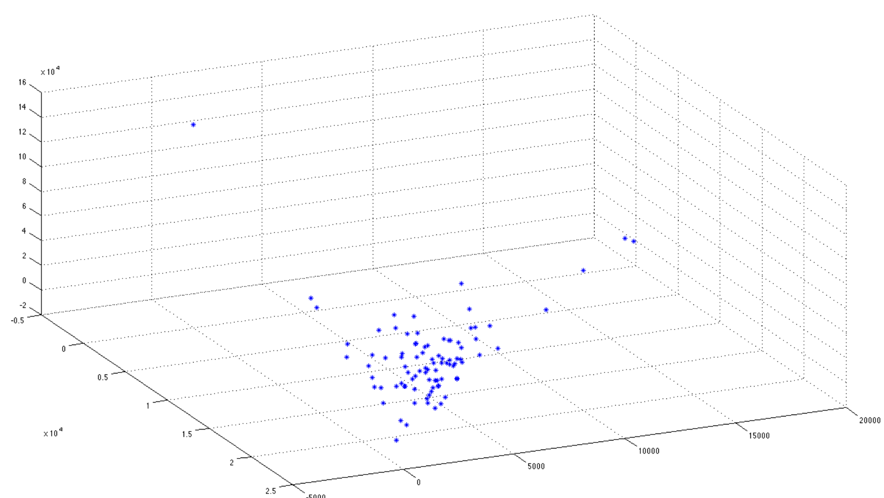

Figure 4.2: Alloy Composition Replication 1 Filtered Pareto Set

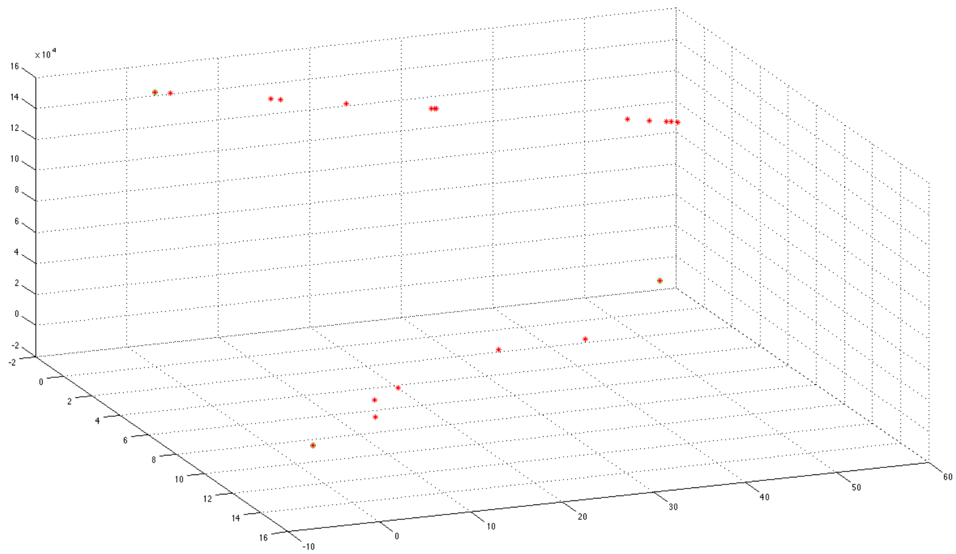

Figure 4.3: Alloy Composition Replication 2 Estimated Pareto Set

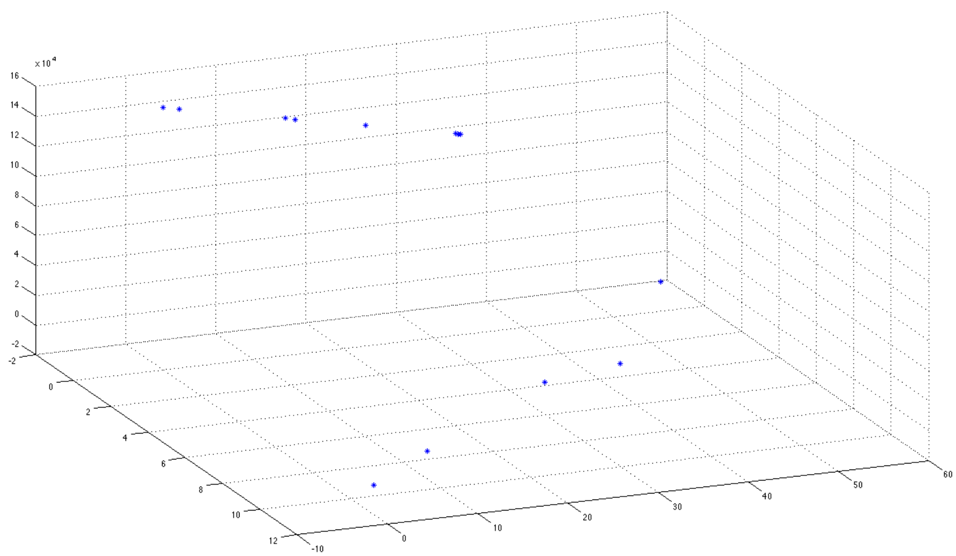

Figure 4.4: Alloy Composition Replication 2 Filtered Pareto Set

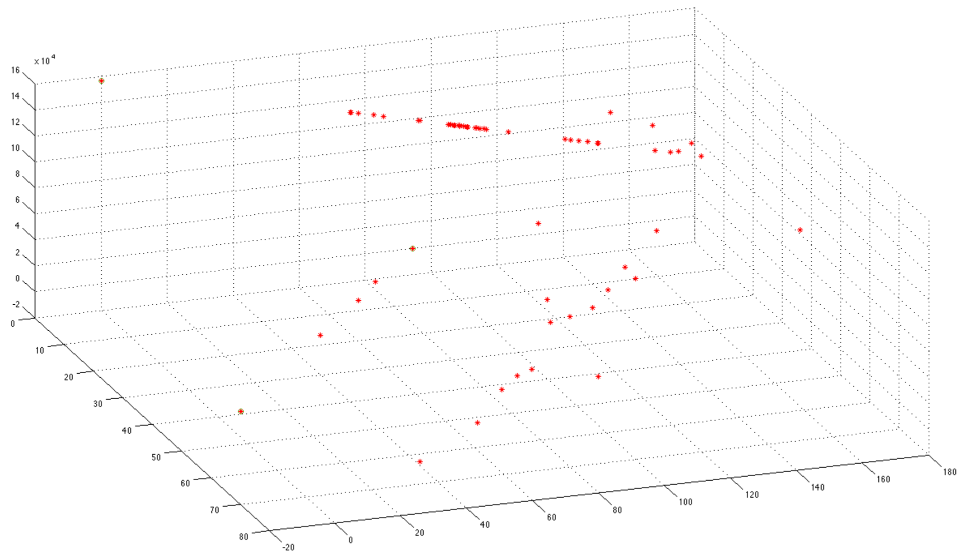

Figure 4.5: Alloy Composition Replication 3 Estimated Pareto Set

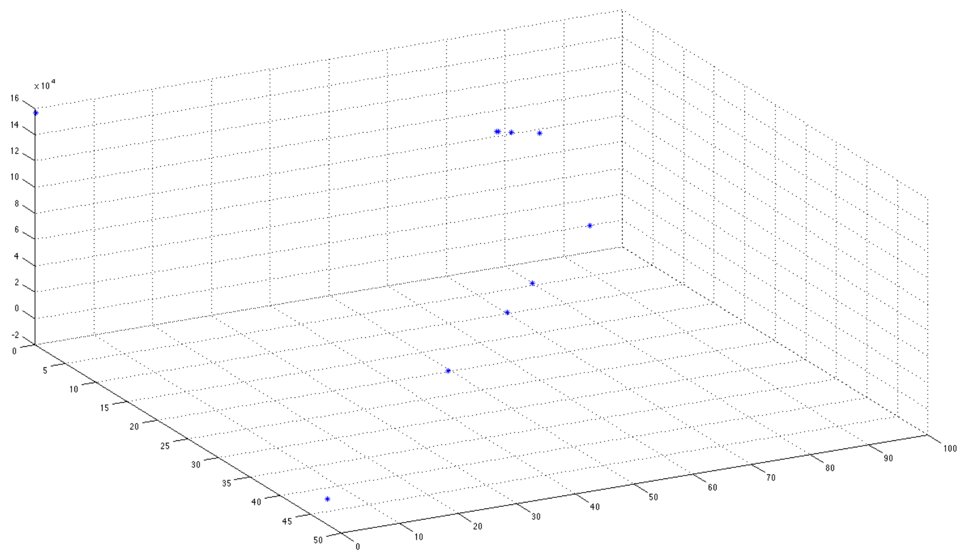

Figure 4.6: Alloy Composition Replication 3 Filtered Pareto Set

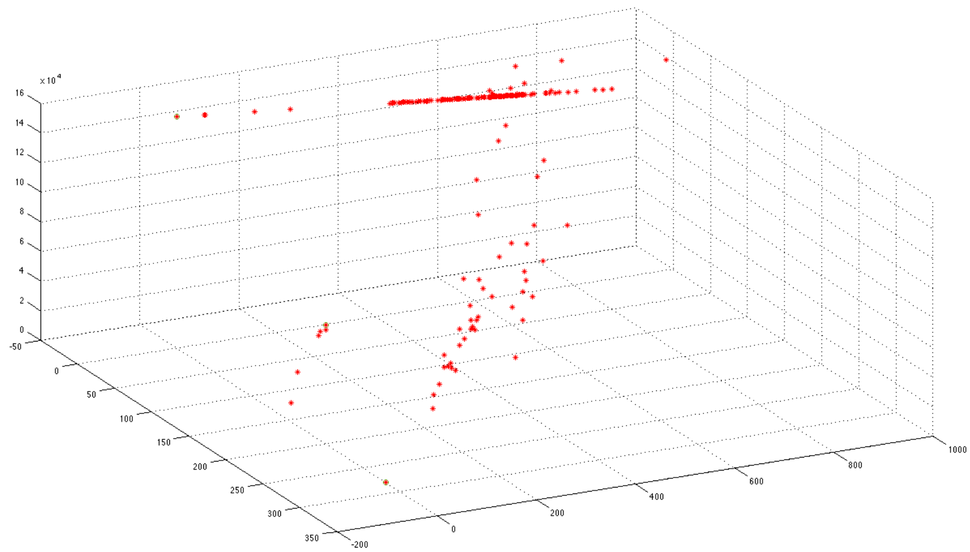

Figure 4.7: Alloy Composition Replication 4 Estimated Pareto Set

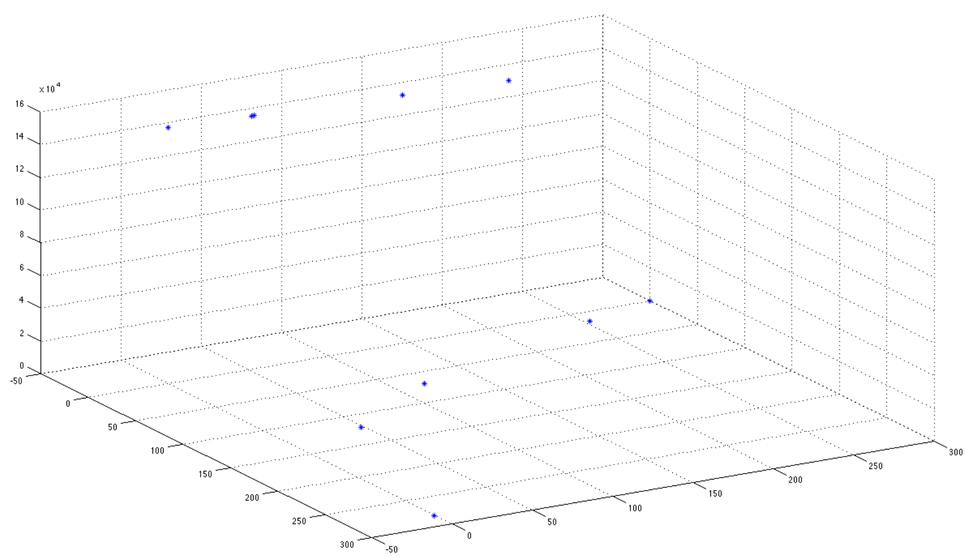

Figure 4.8: Alloy Composition Replication 4 Filtered Pareto Set

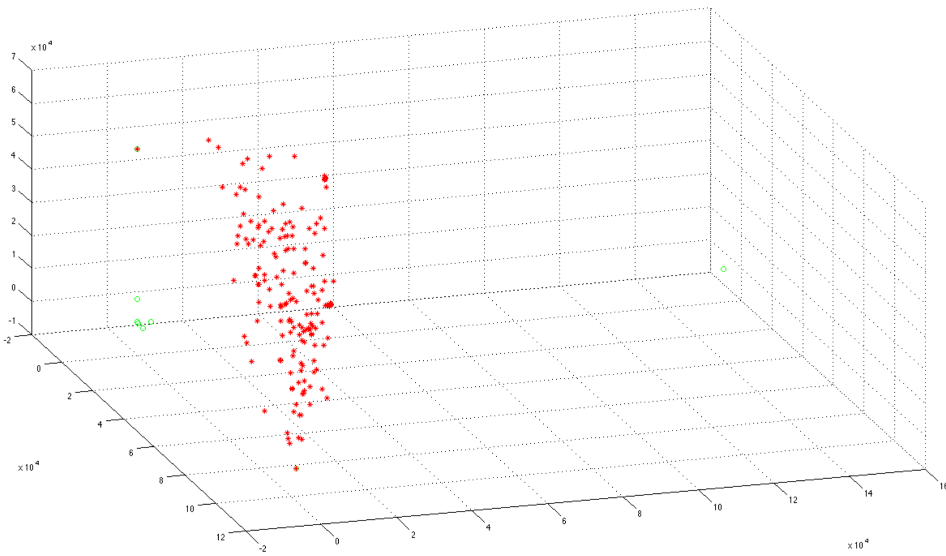

Figure 4.9: Alloy Composition Replication 5 Estimated Pareto Set

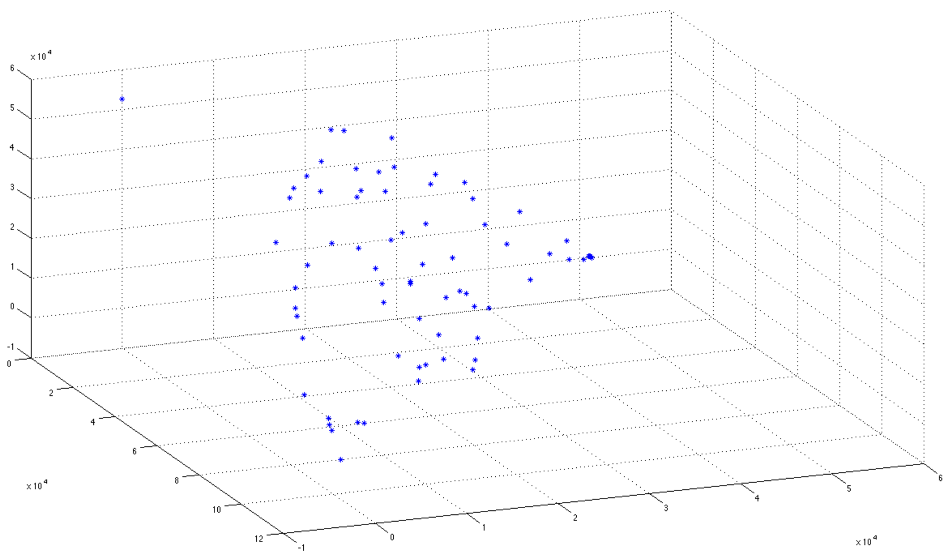

Figure 4.10: Alloy Composition Replication 5 Filtered Pareto Set

4.2.2. Gearbox Design Problem

The ACPSE-SMVO process was completed five times for the Gearbox Design problem. The objective-wise extreme points for each replication are given in Table 4.7. Figures 4.11 – 4.20 display the estimated and filtered Pareto sets for each replication.

Table 4.7: Gearbox Design Extreme Points

Replication	Volume	Stress 1	Stress 2
1	2379.091562	1681.458775	853.3359867
	2948.241488	1668.682313	956.8364233
	2907.900311	1678.372485	763.8310525
2	2525.590227	1691.26222	928.8753356
	3338.93746	1667.845363	886.5397223
	2871.051741	1694.905633	766.0451259
3	2420.578331	1683.297334	971.6242271
	2982.888272	1668.720697	892.506765
	2986.925049	1691.64357	763.1128618
4	2460.505591	1698.416374	943.8695953
	3222.279822	1675.338632	843.9620105
	2759.516806	1691.429411	759.758313
5	2447.505962	1692.214839	917.4229836
	2990.712725	1670.539656	829.4049937
	2916.60226	1690.416531	760.3073499

The plots for the replications appear to vary greatly across the replications: some estimating a three-dimensional convex curled sheath, others approximating a non-convex curled sheath, and another estimating a planar triangle. However, the plot that appears to be estimating a planar triangle may actually be estimating the bottom curve and top "crest" of the same curled sheath being estimated by other plots.

To investigate the variability across the replications, the filtered Pareto sets from all five replications were combined into a "master" set that was subsequently filtered to remove any resulting dominated points. The plot of this filtered master set is shown in

Figure 4.21. The master set appears to approximate a front with two distinct regions, one circled in green on the lower left of Figure 4.21 and the other circled in black. If the actual Pareto front is as discontinuous as the master set appears to be, this may explain why the Gearbox Design Pareto sets varied so drastically across the replications.

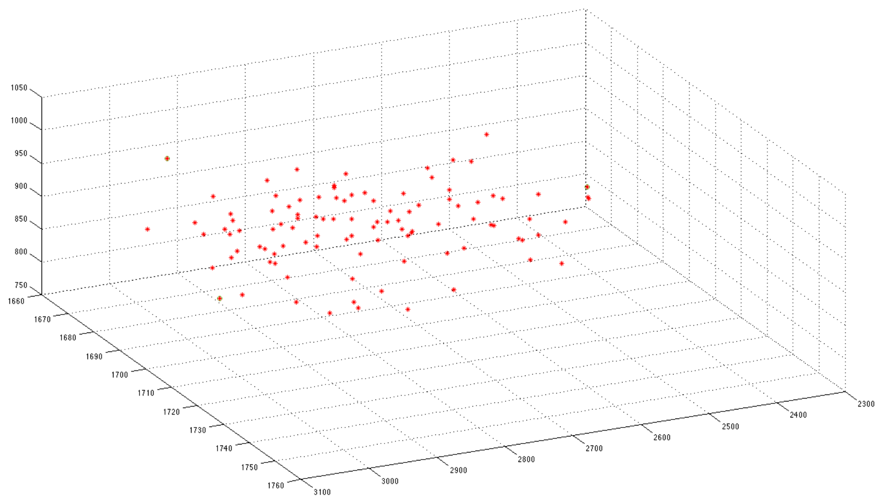

Figure 4.11: Gearbox Design Replication 1 Estimated Pareto Set

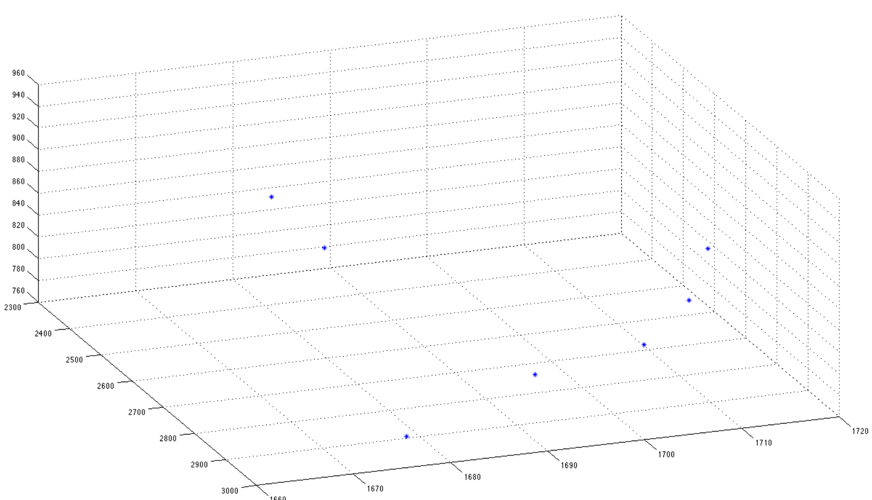

Figure 4.12: Gearbox Design Replication 1 Filtered Pareto Set

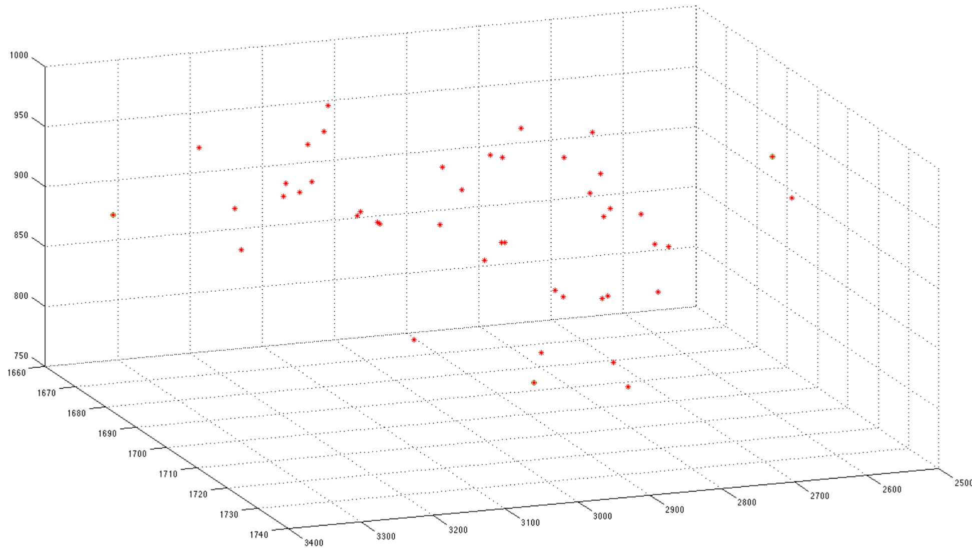

Figure 4.13: Gearbox Design Replication 2 Estimated Pareto Set

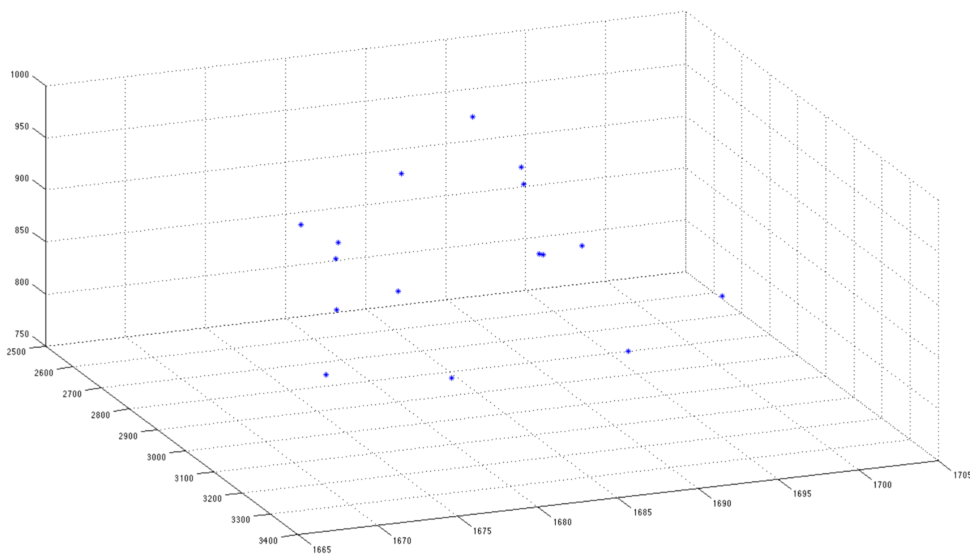

Figure 4.14: Gearbox Design Replication 2 Filtered Pareto Set

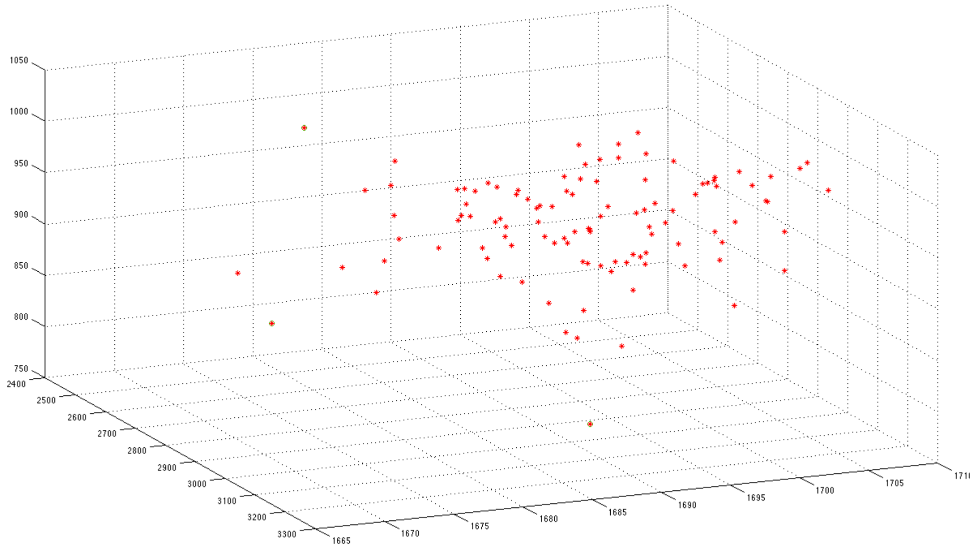

Figure 4.15: Gearbox Design Replication 3 Estimated Pareto Set

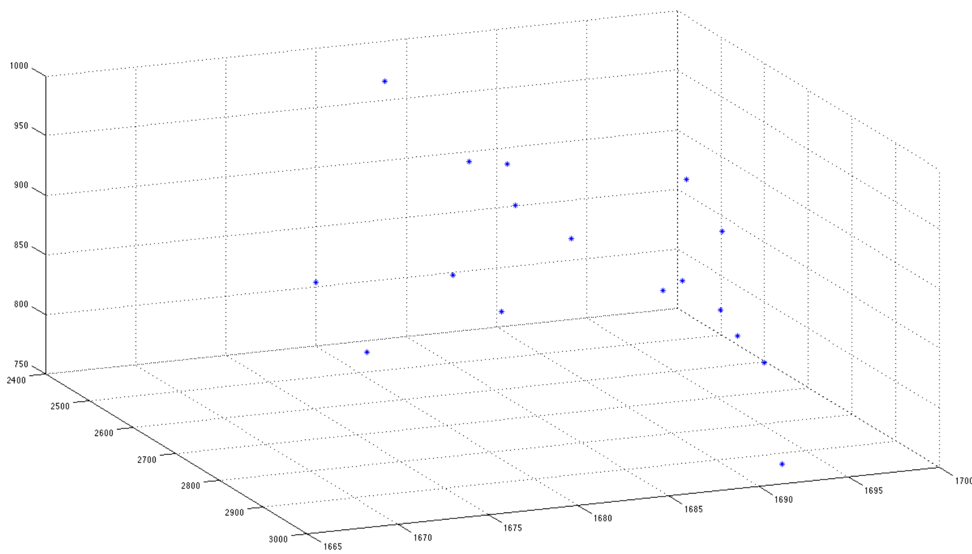

Figure 4.16: Gearbox Design Replication 3 Filtered Pareto Set

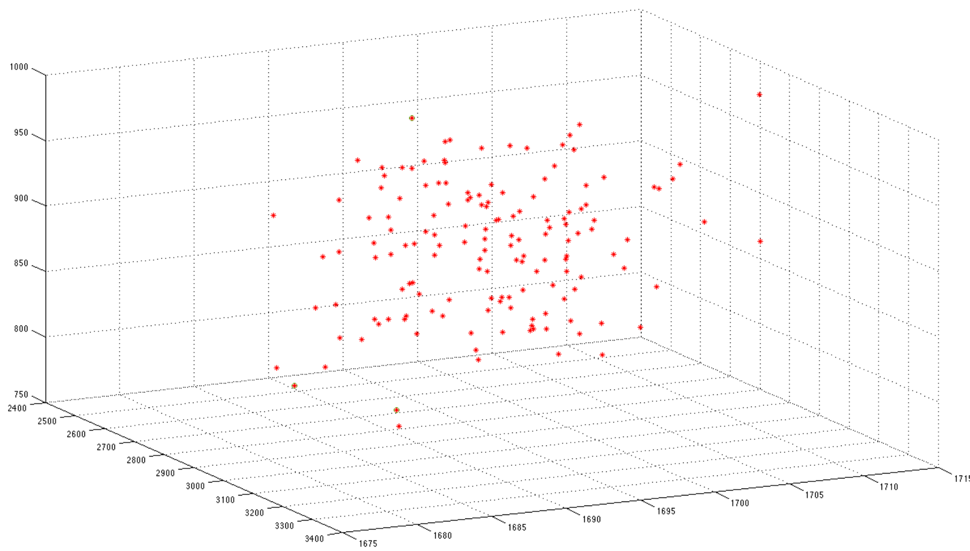

Figure 4.17: Gearbox Design Replication 4 Estimated Pareto Set

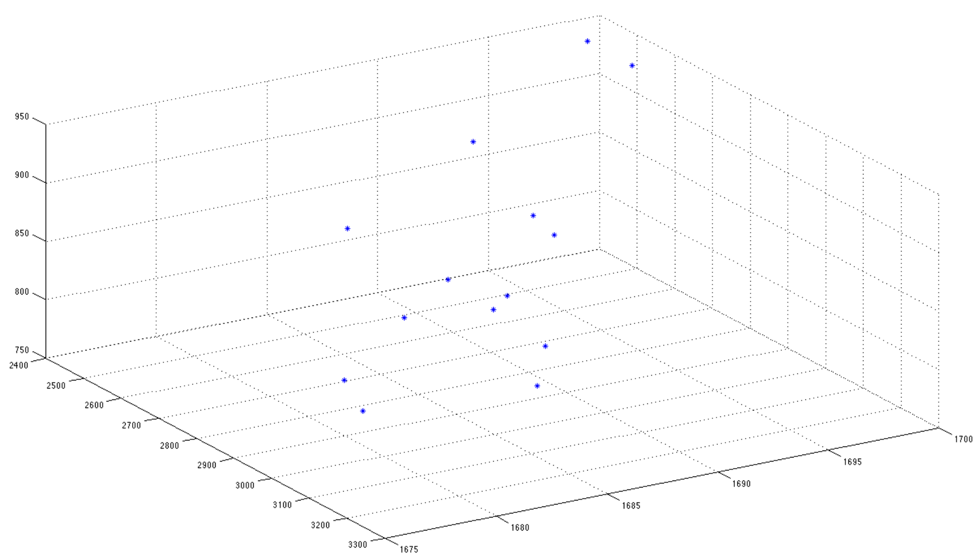

Figure 4.18: Gearbox Design Replication 4 Filtered Pareto Set

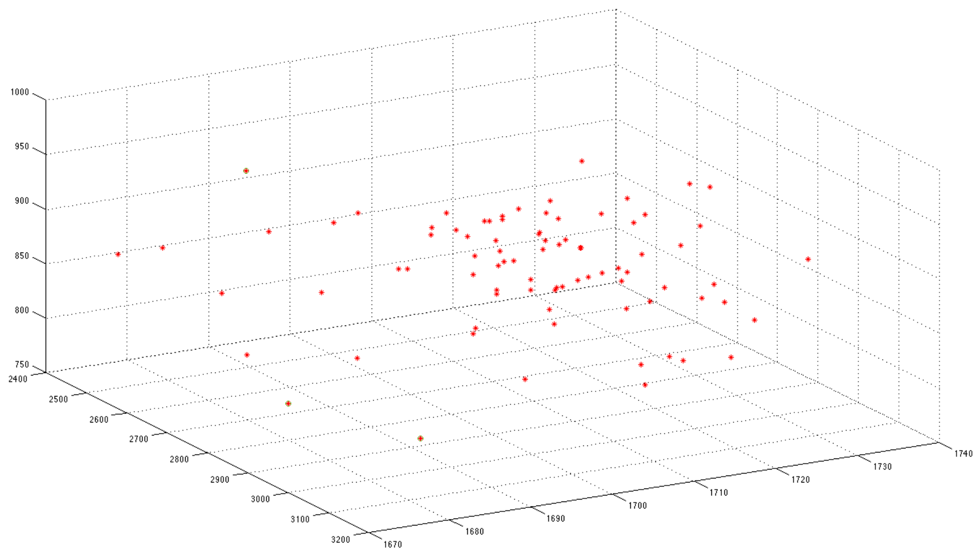

Figure 4.19: Gearbox Design Replication 5 Estimated Pareto Set

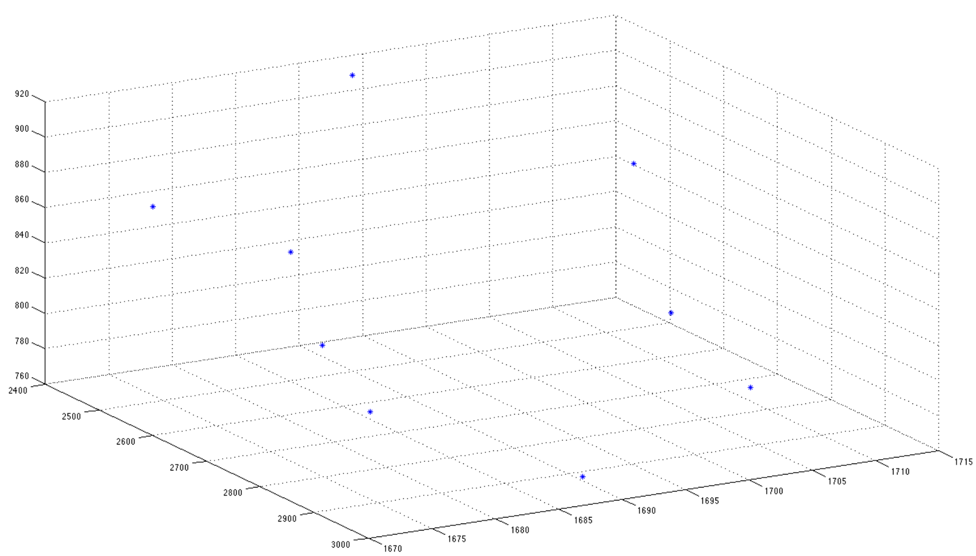

Figure 4.20: Gearbox Design Replication 5 Filtered Pareto Set

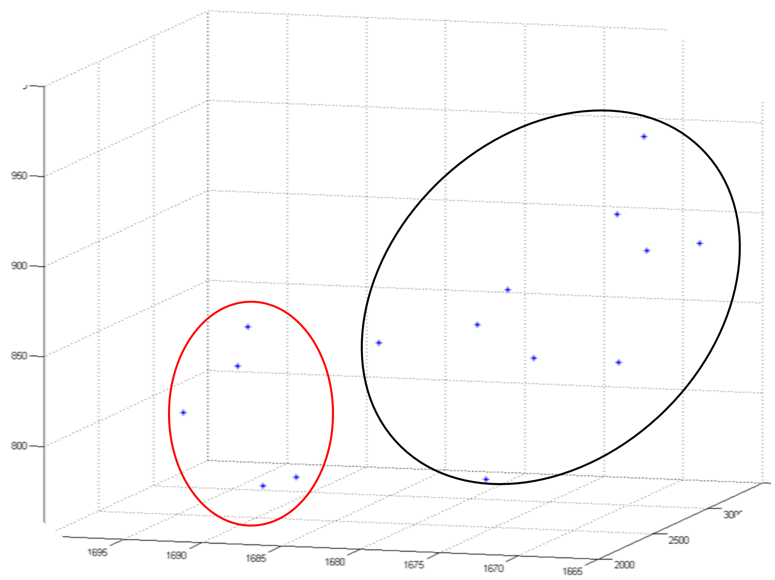

Figure 4.21: Gearbox Design Master Pareto Set with Two Regions

4.2.3. Algorithm Parameter Optimization Problem

The ACPSE-SMVO process was completed five times for the Algorithm Parameter Optimization problem. The objective-wise extreme points for each replication are given in Table 4.8. Figures 4.22 – 4.31 display the estimated and filtered Pareto sets for each replication, projected onto the sub-space of the first three objectives. Figures 4.32 – 4.35 display the filtered Pareto set of all replications combined, projected onto all four sub-spaces.

Table 4.8: Algorithm Parameter Optimization Extreme Points

Replication	FPF	-TPF	Time	-TFP
1	0	-1	22.28691413	-0.952489225
	0.013849069	-1	0	-1
	0	-1	0	-1
	0.068240196	0	58.99805124	-1
2	0	-0.365103331	0	-0.624316924
	0	-1	14.70248504	-1

Replication	FPF	-TPF	Time	-TFP
	0.012178492	-1	0	-1
	0.072040748	0	65.56160718	-1
3	0	-0.526293336	27.82307318	-0.529289615
	0	-1	0	-1
	0	-0.728498782	0	-0.73631103
	0.064546375	0	69.17977705	-1
4	0	-1	15.8967135	-0.590042969
	0	-1	0	-1
	0.017199261	-1	0	-1
	0.069804593	0	69.41282045	-1
5	0	-0.895466518	6.251030822	-0.539900539
	0	-1	72.31698138	-1
	0	-0.137773106	0.139318505	-1
	0.075744837	0	64.10370994	-1

The plots of the three-dimensional projection of the estimated Pareto set appear consistent across all replications. The projected Pareto front appears to be a concave "heel", similar to the bottom eighth of a sphere.

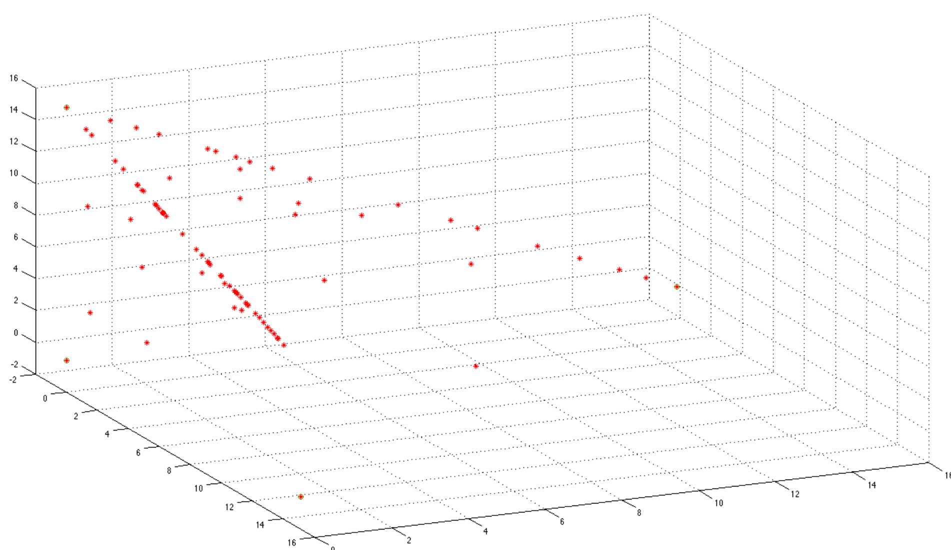

Figure 4.22: Algorithm Parameter Optimization Replication 1 Estimated Pareto Set

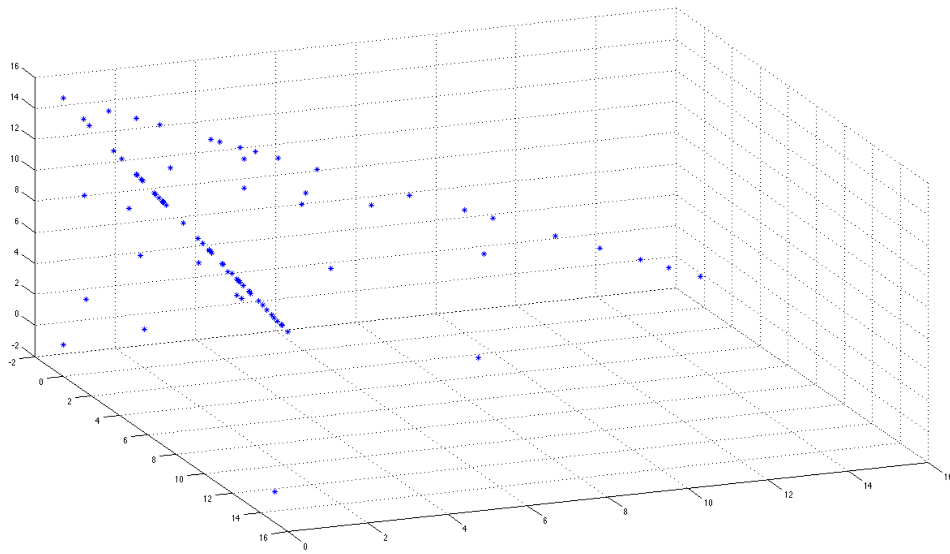

Figure 4.23: Algorithm Parameter Optimization Replication 1 Filtered Pareto Set

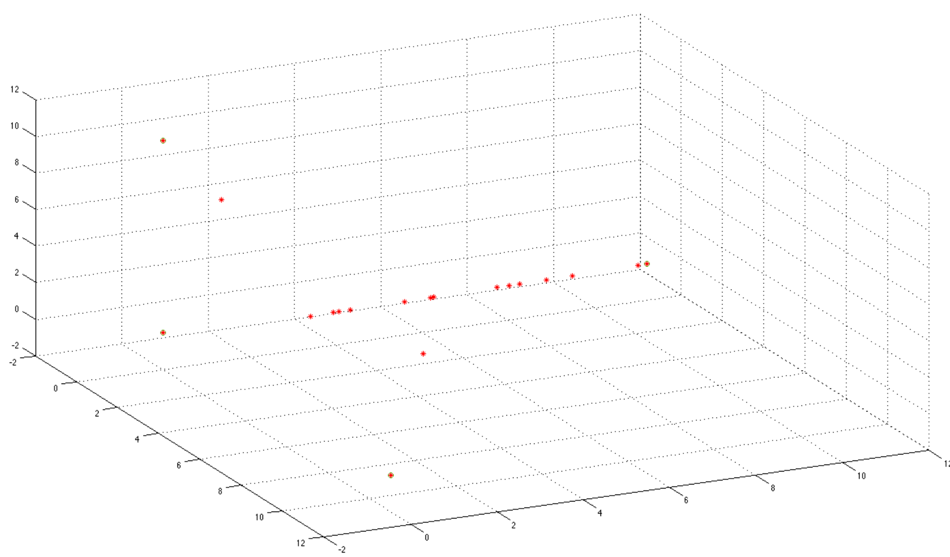

Figure 4.24: Algorithm Parameter Optimization Replication 2 Estimated Pareto Set

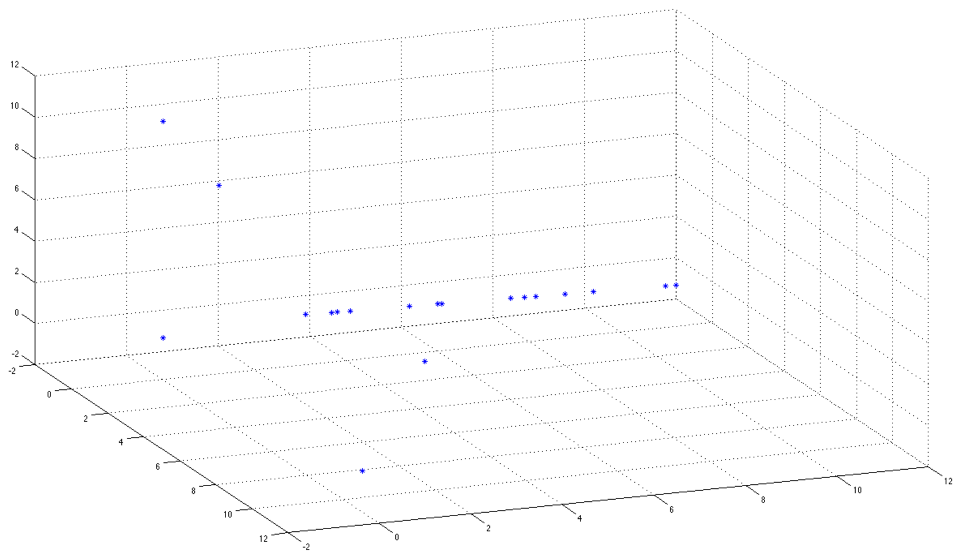

Figure 4.25: Algorithm Parameter Optimization Replication 2 Filtered Pareto Set

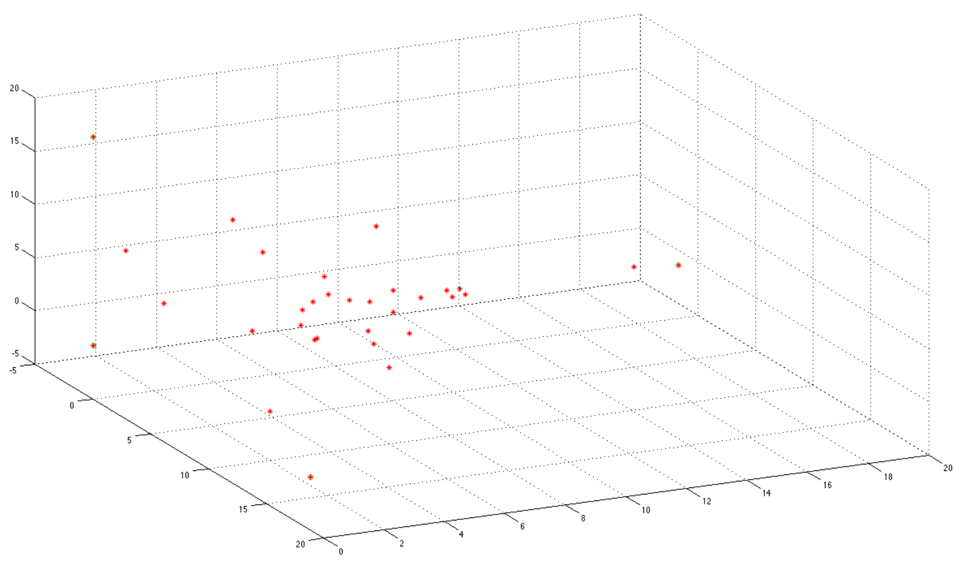

Figure 4.26: Algorithm Parameter Optimization Replication 3 Estimated Pareto Set

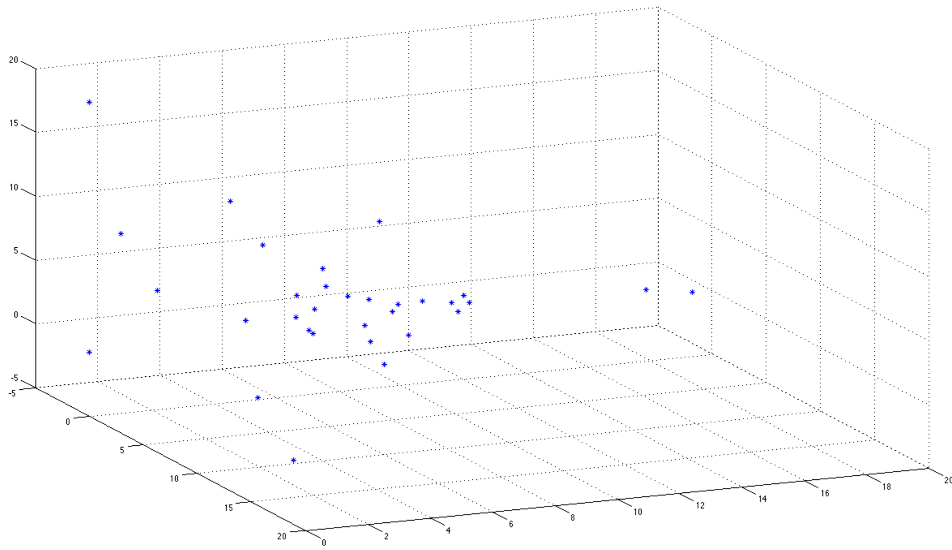

Figure 4.27: Algorithm Parameter Optimization Replication 3 Filtered Pareto Set

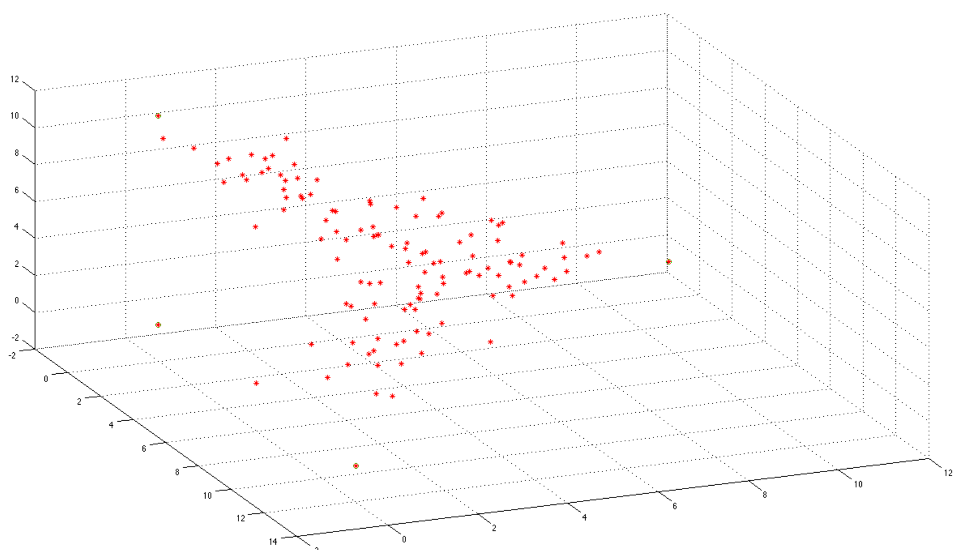

Figure 4.28: Algorithm Parameter Optimization Replication 4 Estimated Pareto Set

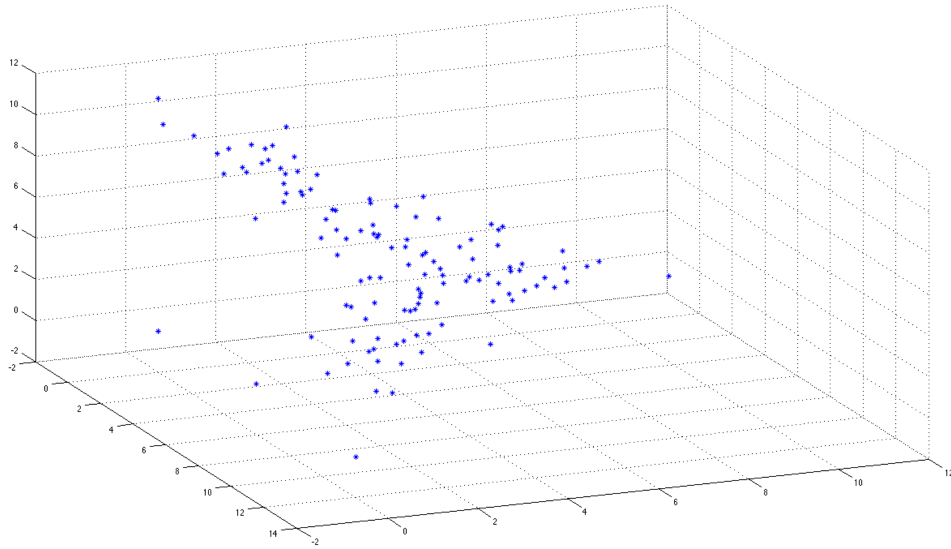

Figure 4.29: Algorithm Parameter Optimization Replication 4 Filtered Pareto Set

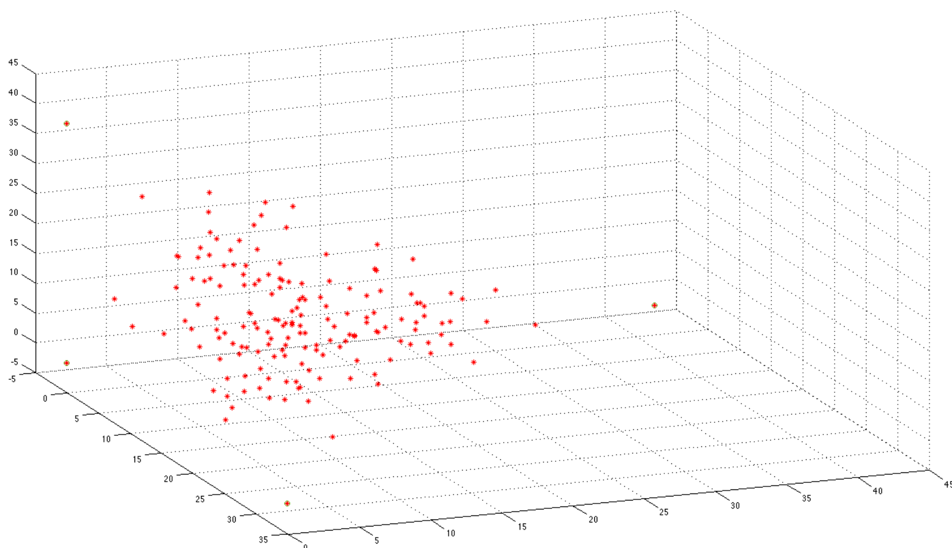

Figure 4.30: Algorithm Parameter Optimization Replication 5 Estimated Pareto Set

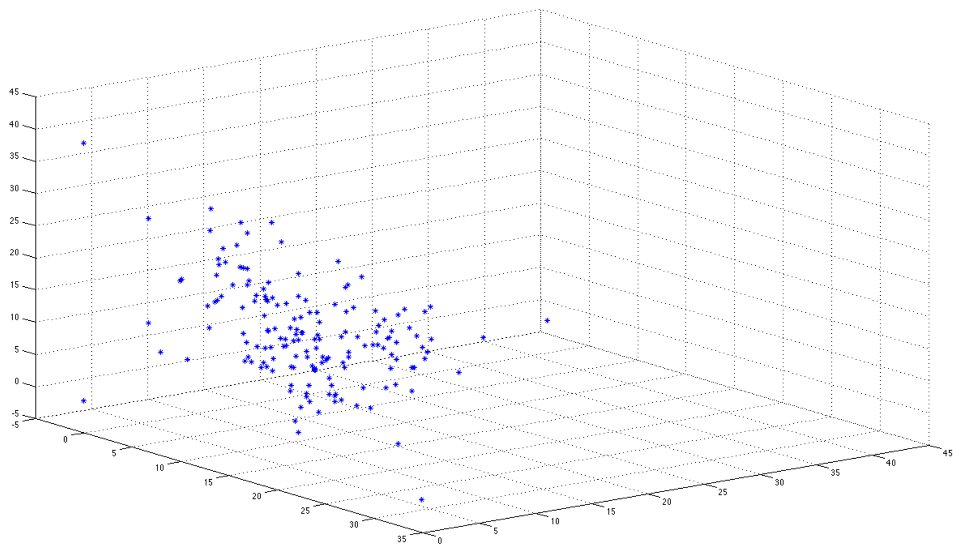

Figure 4.31: Algorithm Parameter Optimization Replication 5 Filtered Pareto Set

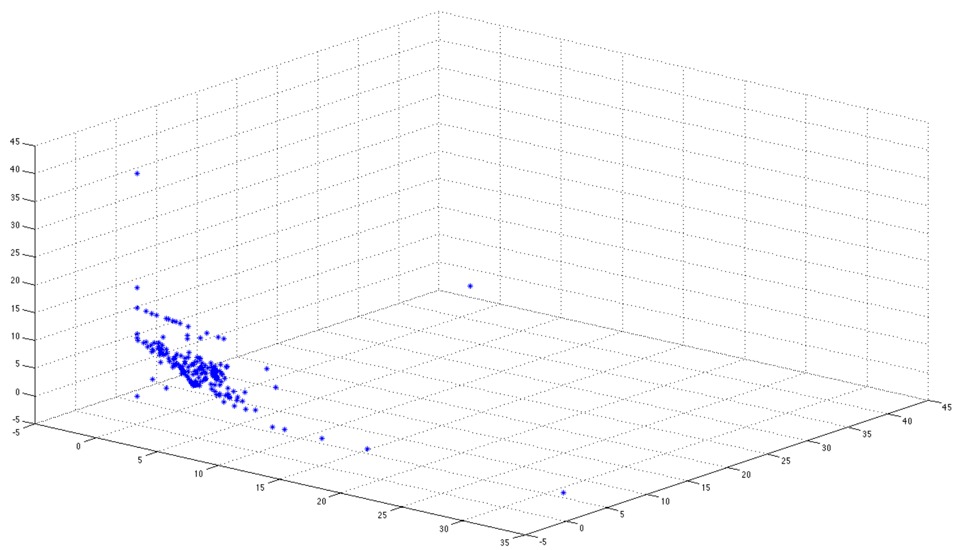

Figure 4.32: Algorithm Parameter Optimization Pareto Set Projection 1

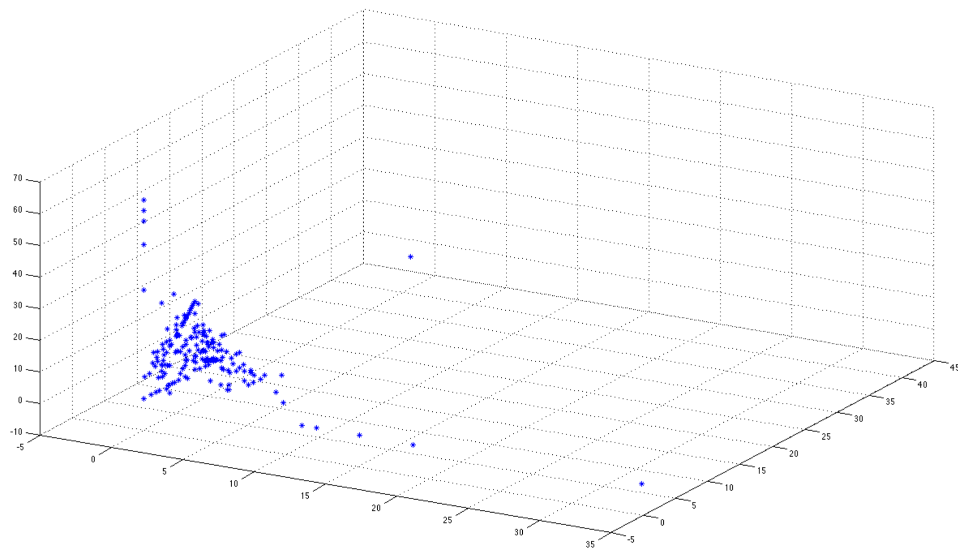

Figure 4.33: Algorithm Parameter Optimization Pareto Set Projection 2

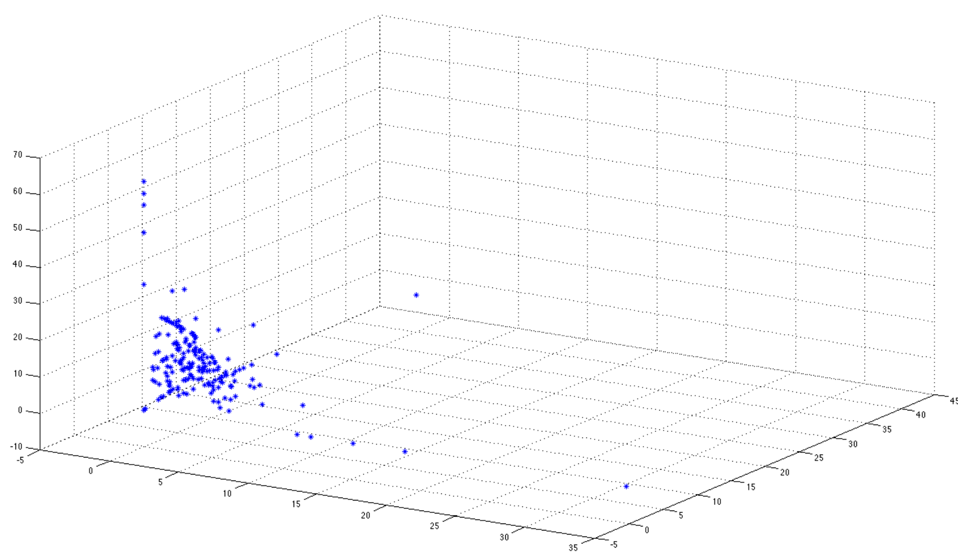

Figure 4.34: Algorithm Parameter Optimization Pareto Set Projection 3

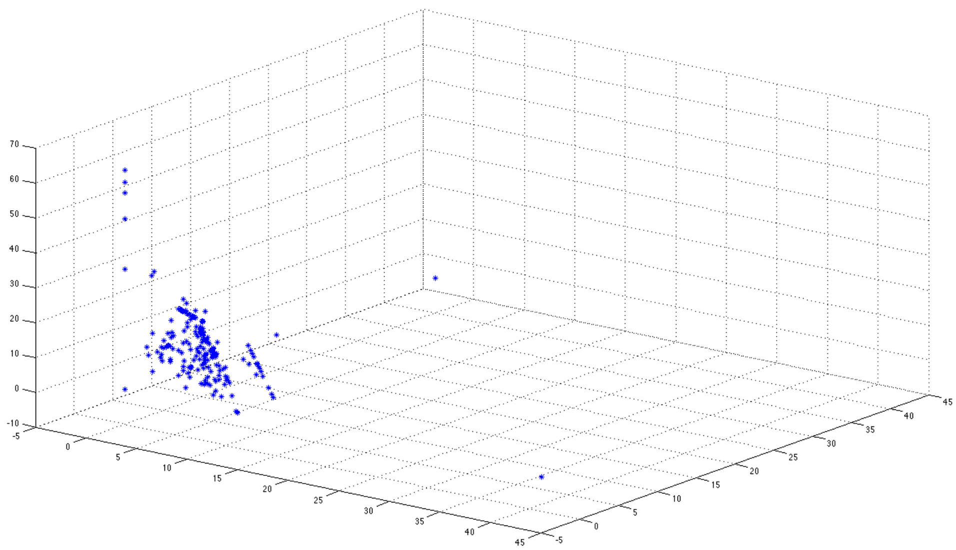

Figure 4.35: Algorithm Parameter Optimization Pareto Set Projection 4

4.2.4. System Reliability Problem

The ACPSE-SMVO process was completed four times for the System Reliability problem. The objective-wise extreme points for each replication are given in Table 4.9. Figures 4.36 – 4.43 display the estimated and filtered Pareto sets for each replication.

Table 4.9: System Reliability Extreme Points

Replication	Unreliability	Cost	Weight
1	3.95886E-19	925397.6218	290.230489
	5.832E-06	105.3103891	105.7286258
	5.82533E-06	105.7411821	104.446037
2	1.45466E-18	716990.0438	289.6193566
	0.0625	9.990128026	40.3932388
	0.0547536	71.04343095	39.96006224
3	1.559E-11	15103.84759	170.025366
	0.0625	10.07472107	39.73246985
	0.11523	53.69702519	9.943238729
4	1.65313E-13	19713.60355	200.4267172
	0.0625	9.971925081	40.4318991
	0.020708625	114.6073867	14.92594576

The plots of the estimated and filtered Pareto sets for the System Reliability problem appear unique from the other problems in that the estimated Pareto front appears to be a parameterized two-dimensional function, rather than an *m*-dimensional surface. This is probably due to the close relationship between the second (Cost) and third (Weight) objectives, indicating that the second and third objectives do not compete with each other in the multi-objective optimization problem.

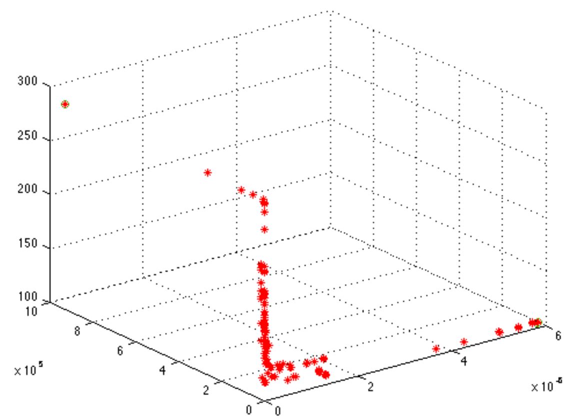

Figure 4.36: System Reliability Replication 1 Estimated Pareto Set

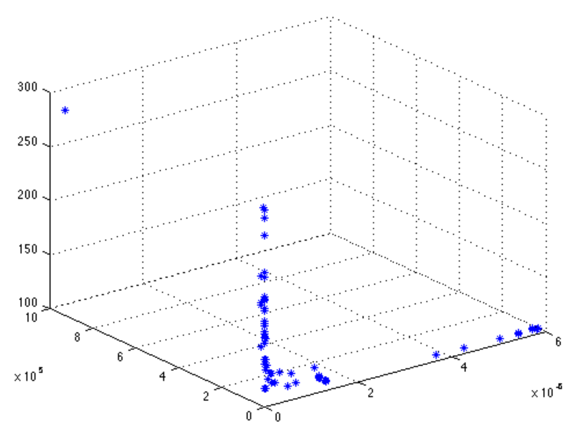

Figure 4.37: System Reliability Replication 1 Filtered Pareto Set

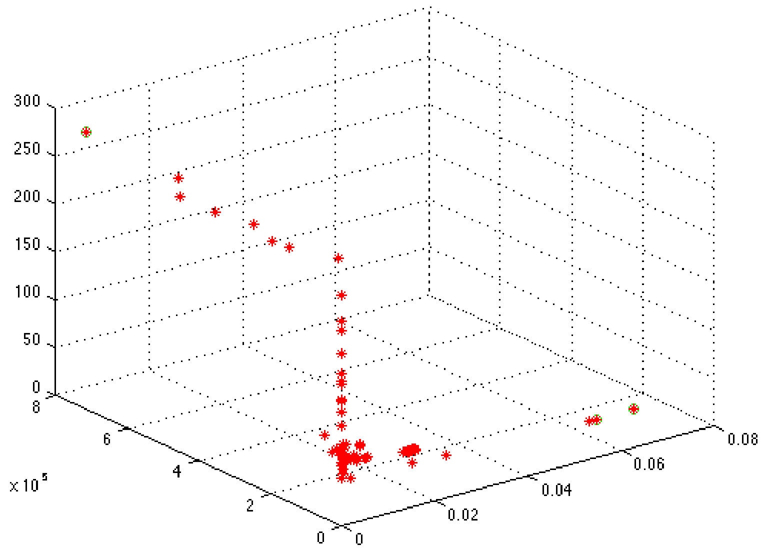

Figure 4.38: System Reliability Replication 2 Estimated Pareto Set

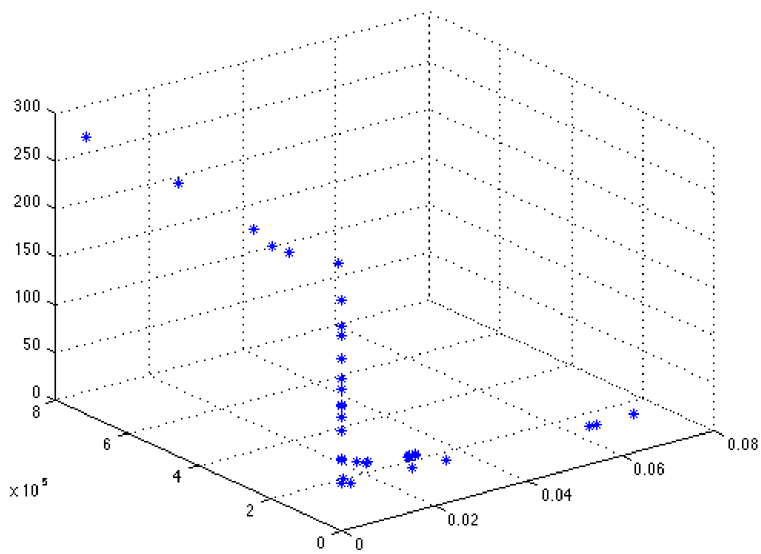

Figure 4.39: System Reliability Replication 2 Filtered Pareto Set

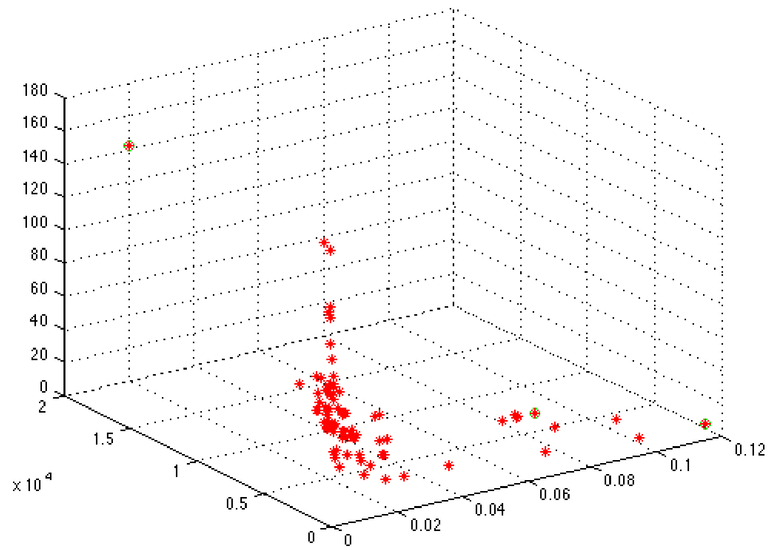

Figure 4.40: System Reliability Replication 3 Estimated Pareto Set

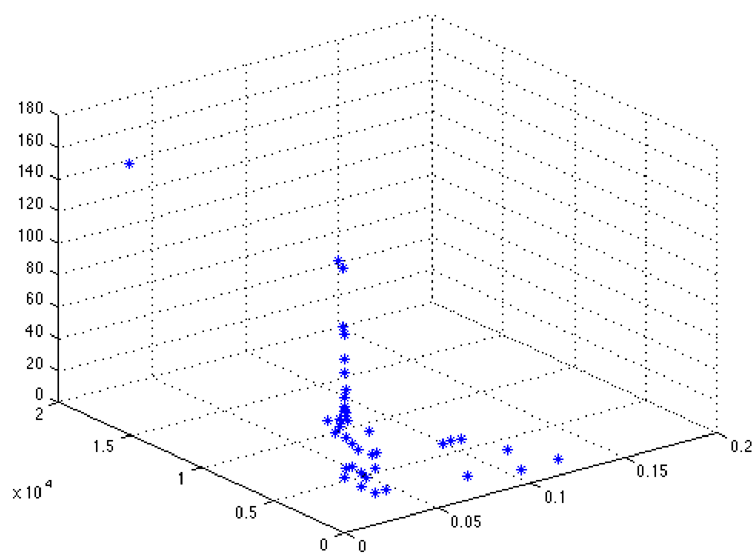

Figure 4.41: System Reliability Replication 3 Filtered Pareto Set

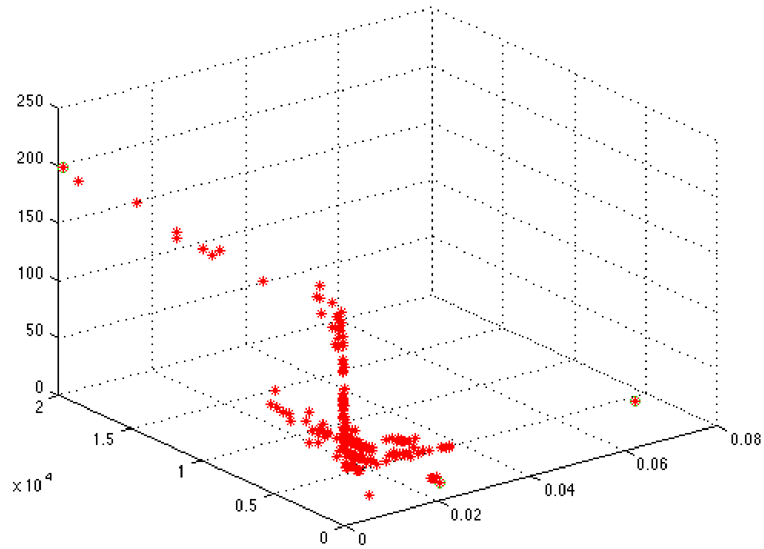

Figure 4.42: System Reliability Replication 4 Estimated Pareto Set

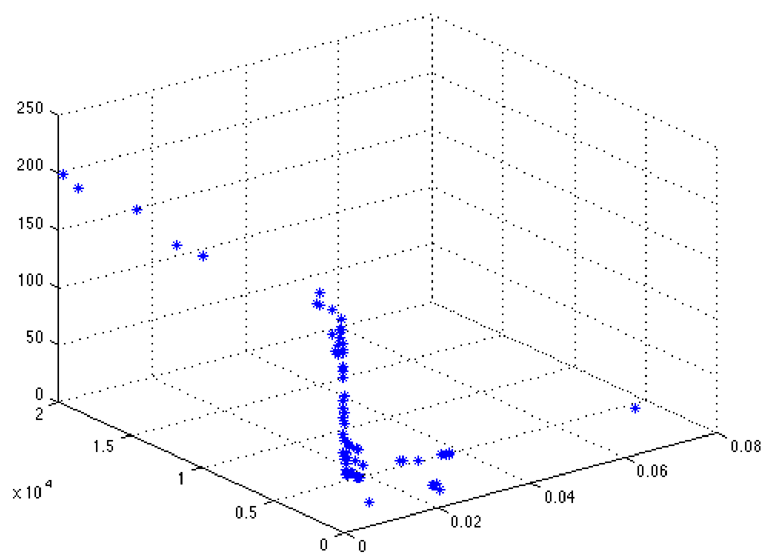

Figure 4.43: System Reliability Replication 4 Filtered Pareto Set

4.3. *Case Study Comparison*

The Alloy Composition and Gearbox Design problems have been previously studied, so some comparison can be made between the results from the ACPSE-SMVO process and previously published results.

4.3.1. Alloy Composition Problem

The Alloy Composition Problem presented in Chapter IV is based on the work by Dulikravich, *et al.* [24,25,26,27]. Their estimated Pareto set is presented in Table 4.10. The objective function values are based on an Artificial Neural Network (ANN) model constructed to predict T_1, T_g, and ρ based on the same 53 data points presented in Table 4.1. These multiple objective functions were then optimized using the Indirect Optimization based upon Self-Organization (IOSO) algorithm. The IOSO algorithm is described by Dulikravich, *et al.*[26], as a "semi-stochastic multi-objective optimization algorithm incorporating aspects of a selective search on a continuously updated multi-dimensional response surface."

Table 4.10: IOSO Generated Pareto Optimal Solutions [26]

x_1	x_2	x_3	x_4	x_5	x_6	x_7	$-F_1$	$-F_2$	F_3
% Zr	%Cu	%Al	%La	%Cu,Ni	%Pd	%Si	T_1	T_g	ρ
57.999	30.77	0.002	0.001	0	0.001	11.227	673.5	1232.7	6.799
53.33	29.935	0.77	0	0	0	15.964	675.1	1230.7	6.552
56.866	38.133	4.974	0	0	0	0.022	679.1	1222.5	7.26
50.227	47.223	1.06	0	0	0	1.489	694.3	1213.3	7.569
39.138	46.942	2.272	0	0.001	0.001	11.645	705.5	1204.7	7.09
32.645	50.993	11.075	0.001	0	0.001	5.282	730.1	1197.5	7.119
38.96	50.384	9.723	0	0	0	0.931	727.7	1196.3	7.335
48.256	34.613	16.486	0	0	0	0.643	727	1193.4	6.705
40.999	43.022	15.859	0.001	0	0.001	0.117	726.4	1190.3	6.955
37.97	41.55	15.124	0.001	0	0	5.344	726.1	1189.2	6.73

44.287	50.864	4.847	0	0	0	0	718.7	1189	7.577
0.393	17.233	0.216	0.064	0	73.758	8.334	653.5	1157.1	10.64
0.001	0	1.053	2.568	0.008	81.017	15.353	632.1	1095.4	10.285
0.5	0.134	0.223	9.268	0.024	81.111	8.731	631.6	1093	10.582
3.368	0.546	1.437	12.258	0.036	81.5	0.853	631.3	1091.5	10.888
0.005	8.134	6.062	4.713	0.321	70.319	10.449	638.5	1060.1	9.905
0.006	5.965	0.008	0	0	77.33	16.689	637.1	1058.8	10.215
0	0.032	19.88	45	29.282	0	5.805	434.9	934.5	6.055
0	1.449	20.497	41.613	26.367	0	10.074	435.1	920.3	5.83
0	36.014	13.996	49.973	0.002	0	0.008	428	862.5	6.684
0	0.001	16.255	50.1	28.67	0	4.971	432.9	830.5	6.197
0	25.735	15.3	42.81	2.378	0	13.77	417.1	827.1	5.888
0	0.001	13.918	56.909	29.09	0	0.08	427	815.8	6.478
0	9.318	16.13	63.464	0	0	11.087	389.5	796.5	5.438
0	26.62	13.917	59.456	0	0	0	404.8	789.1	6.426
0	0.0137	12.416	70.057	17.513	0	0.001	402.9	759.3	6.216
0	0.001	19.297	59.59	20.46	0	0.651	426.4	739.3	6.032
0	5.422	18.15	50.853	17.858	0	7.717	423	724.7	5.881

In order to compare the ACPSE-SMVO results to those from Dulikravich, *et al*., the 23 estimated Pareto optimal solutions from Table 4.10 were used to predict T_1, T_g, and ρ using the Alloy Composition regression model described in Section 4.1.1. However, these 23 estimated Pareto optimal solutions indicate that Dulikravich, *et al.*, considered all variables to be continuous. Therefore, the Alloy Composition regression model was adjusted to assign all values of %Si from the Dulikravich data to a "High", "Medium", or "Low" setting, corresponding to the discrete values of 16.8, 16.5, and 0 used in the ACPSE-SMVO method. The %Si assignment function is given as

$$\%Si_{regression} = \begin{cases} High & \text{if } \%Si_{Dulikravich} > 10 \\ Medium & \text{if } 5.0 < \%Si_{Dulikravich} < 10 \\ Low & \text{if } \%Si_{Dulikravich} < 5.0 \end{cases} \tag{4.1a}$$

The predicted objective function values were then re-oriented with the affine transformation method described in Section 3.2.1, using the Extreme Points from Replication 1 (displayed in Table 4.6) as the transformation basis. Finally, these predicted and re-oriented objective function values based on the 23 Pareto optimal solutions from Dulikravich, *et al.*, were plotted along with the combined and re-filtered Pareto sets from Replications 1 through 5 of Section 4.2.1. This plot is shown in Figure 4.44.

When the combined plot was projected onto the first two dimensions, as shown in Figure 4.45, the transformed Dulikravich-based Pareto set and the ACPSE-SMVO Pareto set appeared to estimate the same front. However, in another two-dimensional projection, shown in Figure 4.46, most of the Dulikravich-based points appear dominated with respect to the transformed origin. This suggests that the set of Pareto optimal solutions found using ACPSE-SMVO dominate many of the solutions found using the IOSO method.

However, when no affine transformation is performed, the plot of the two sets (shown in Figure 4.47) suggests that there may exist some reference point that allows both sets to be non-dominated. This compatible reference point would exist in the extreme lower-right region of Figure 4.48. Although they are not conclusive, these plots suggest that the ACPSE-SMVO process and the Dulikravich solutions may have estimated the same Pareto front, and that both approaches discovered Pareto optimal solutions that the other was not able to find.

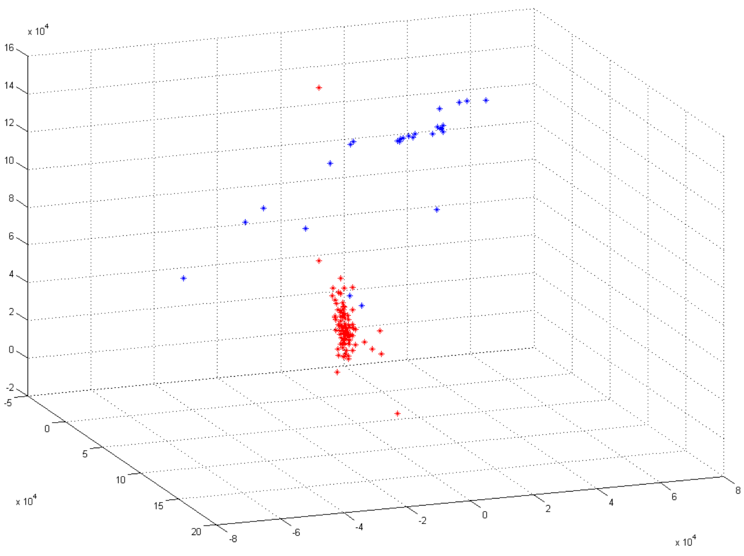

Figure 4.44: Transformed Dulikravich (blue) and ACPSE-SMVO (red) Pareto Sets

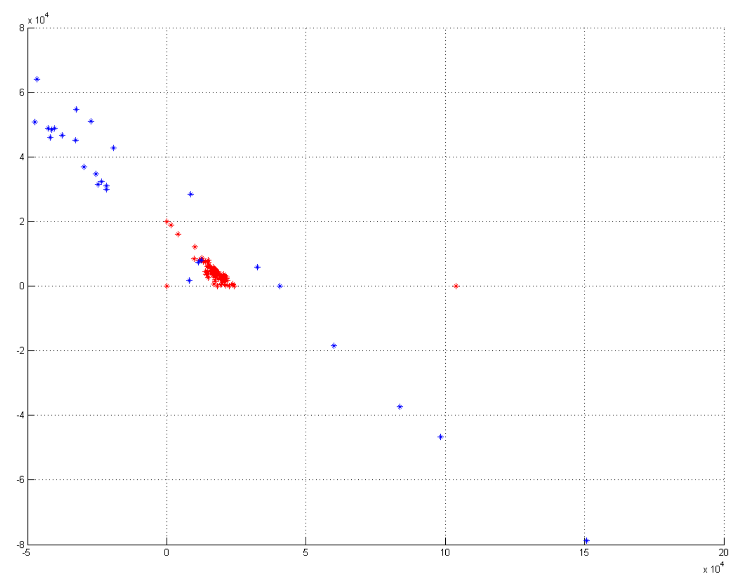

Figure 4.45: Projected Dulikravich (blue) and ACPSE-SMVO (red) Pareto Sets

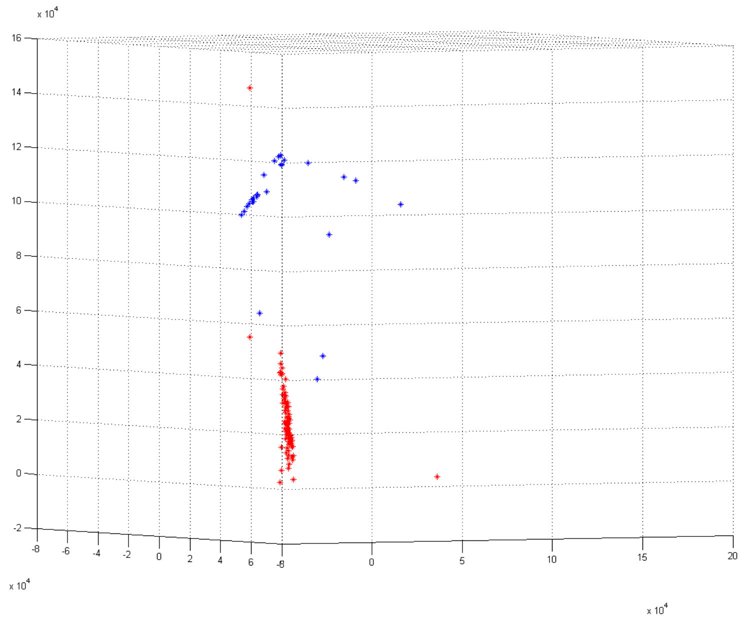

Figure 4.46: Projection Suggesting Distinct Surfaces

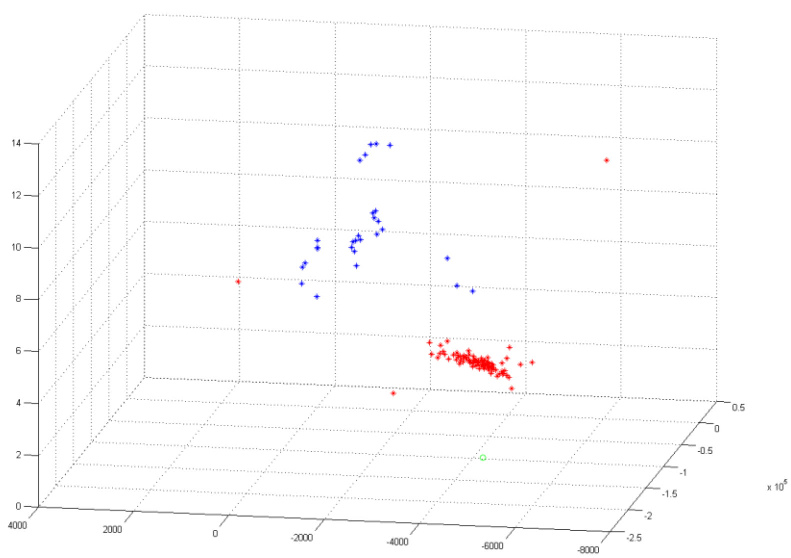

Figure 4.47: Dulikravich (blue) and ACPSE-SMVO (red) Pareto Sets on Original Axes

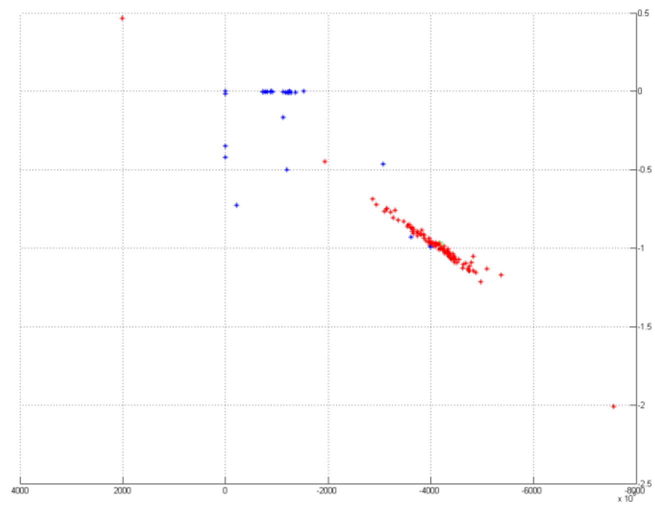

Figure 4.48: Compatible Reference Point in Bottom Right Corner

4.3.2. Gearbox Design Problem

The Gearbox Design solutions presented by Azarm, *et al.* [10], are not intended to estimate the Pareto optimal set (nor was the problem solved as a stochastic optimization problem) but they can be used as a reference to compare individual Pareto optimal solutions obtained from the ACPSE-SMVO process. The final objective function values obtained by Azarm, *et al.*, are shown in Table 4.11. From Table 4.7, the best Volume result obtained from the ACPSE-SMVO process is 2379.1, approximately 1.25% greater than the minimum Volume obtained by Azarm, *et al.* However, all Stress 1 and Stress 2 values from the ACPSE-SMVO process outperform the Stress 1 and Stress 2 values from Table 4.11.

For Stress 1, the best value from Table 4.7 is 1.9% smaller than the Stress 1 value in Table 4.11, the worst value is only 0.09% smaller than the Stress 1 value in Table 4.11, and the average value from Table 4.7 is 1.1% smaller than the Stress 1 value in Table

4.11. For Stress 2, the best value from Table 4.7 is 21.8% smaller than the Stress 2 value in Table 4.11, the worst value is only 0.04% smaller than the Stress 2 value in Table 4.11, and the average value from Table 4.7 is 11.89% smaller than the Stress 2 value in Table 4.10.

Table 4.11: Optimal Objective Function Values [10]

Volume	Stress 1	Stress 2
2350	1700	972

4.3. Summary

For three out of the four problems studied, the estimated and filtered Pareto sets appear fairly densely populated, indicating the ACPSE-SMVO was successful in adaptively and automatically searching for and finding Pareto optimal solutions within the bounds set by the objective-wise optima.

The estimated Pareto set for the Alloy Composition problem was compared to the results from Dulikravich, *et al.* [26], and shown to either dominate those previous solutions, or perhaps estimate a region of the actual Pareto front that was not discovered by previous research.

While the Gearbox Design problem did not produce a densely populated Pareto front, when compared to previously obtained optimal solutions, the objective-wise optima are just as good or better. This indicates that the optimization process within ACPSE-SMVO is performing adequately, and that the low density of the estimated front may not be due to the ACPSE-SMVO algorithm, but the complex shape of the actual feasible region defined by the various constraints.

V. Summary, Conclusions and Recommendations

This chapter summarizes this research, presents some final conclusions, addresses known issues, and provides recommendations for future improvements or extensions of this research.

5.1. Summary

This research reviewed the current state of multi-objective optimization for mixed variable stochastic design problems, with particular emphasis on developments in the area of Mixed Variable Mesh Adaptive Direct Search with Ranking and Selection (MVMADS-RS). Using the MVMADS-RS optimization framework, a novel approach to Pareto set estimation was developed based on the SMOMADS method of objective-wise optima determination, the NMADS method of gap detection and the Adaptive Weighted Sum (AWS) method for constructing neighborhood searches. This new method, called Adaptive Constraint Pareto Set Estimation for Stochastic Mixed Variable Optimization (ACPSE-SMVO), was parameterized for robust performance across convex and non-convex surfaces, and was then applied to four test cases, representing distinct classes of real-world design problems.

5.2. Conclusions

The adaptive constraint generation method developed in this research was demonstrated to eliminate, or at least drastically reduce, the need for decision maker input in estimating the Pareto front of multiple classes of stochastic, mixed-variable multi-objective optimization problems. Additionally, the sub-space density method of

Pareto front gap finding proved to be a promising method of automatically detecting regions to search for Pareto optimal solutions.

However, the density threshold method still requires arbitrary, or at least subjective, parameter selection. For example, the full factorial design used in this research to select appropriate parameter values for the ACPSE-SMVO process only resulted in a "best guess" at good values, and applying ACPSE-SMVO at those values did not always produce densely populated Pareto fronts.

Additionally, one of the most promising features of the adaptive constraint method developed in Chapter III was that its formulation seemed to preclude dominated points from entering the estimated Pareto set. When implemented, however, dominated points were admitted into the set, and a final dominance filter had to be applied. It is unclear if this is due to the adaptive constraint formulation or if it is caused by the stochastic optimization step of the ACPSE-SMVO process. In order to be considered a robust method for Pareto set estimation, the ACPSE-SMVO process must be shown to admit no dominated points when applied to deterministic optimization problems.

Finally, the affine transformation developed in Chapter III proved useful in generating a more understandable Pareto front for problems with both minimization and maximization objectives. However, the affine transformation failed to properly re-orient the objective space of problems that had no maximization functions, so these problems had to be evaluated in their original objective space.

5.3. Recommendations for Future Research

5.3.1. Adaptive Constraint Method and Dominated Points

Since the implementation of ACPSE-SMVO resulted in some dominated points being selected for the estimated Pareto front, further study is required to determine if this is due to the adaptive formulation of the constraints or simply caused by the stochastic nature of the optimization step. An attempt should be made to mathematically demonstrate that the adaptive constraint formulation developed in Chapter III will admit no dominated points given a deterministic optimization step. If the adaptive constraint method is shown to reject dominated solutions for all deterministic problems, it should follow that dominated points are admitted simply due to the estimation error of the stochastic optimization step.

5.3.2. ACPSE-SMVO Process

Although an attempt was made to determine the best parameter settings for the ACPSE-SMVO process, the results presented in Chapter IV suggest that there is room for improvement. A more rigorous experiment of parameter settings across a broader range of test surfaces might yield more robust parameters. Additionally, the use of the minimum, median, and maximum gap filling points to replace the Pareto optimal gap generator is arbitrary. A different method of gap generator replacement should be studied and compared to the current method.

Additionally, it may be feasible to decrease the amount of time necessary to complete the optimization step of the ACPSE-SMVO process. Since the ACPSE-SMVO process constructs a series of search neighborhoods based on two-objectives, it may be possible to apply bi-objective optimization techniques to the ACPSE-SMVO process, instead of relying on complex and time consuming multiple-objective techniques. Furthermore, the use of surrogate functions should also be studied as a method of decreasing the number of iterations needed to complete the gap-filling step of the ACPSE-SMVO process.

5.3.3. MVMADS-RS

The standard Ranking & Selection procedure used in this research and described in Section 2.2.3 could be updated to use more recently developed selection methods. The sequential selection procedures developed by Pichitlamken, *et al.* [46], and by Chick and Inoue [18] may be able to reduce the number of function evaluations required for the R&S portion of the MVMADS technique. The quantile method developed by Bekki, *et al.* [15], could also be incorporated to the MVMADS method as an alternative to the standard R&S procedure.

5.3.4. Algorithm Parameter Optimization Problem

There are two main areas of improvement concerning the ACPSE-SMVO investigation of the AutoGAD performance measure Pareto front. First, generating a regression model to fit the results of a test design (as performed for this research) is unnecessary. Instead of using a regression model as an objective function, AutoGAD performance could be directly measured by running the algorithm at the parameter settings selected by MVMADS-RS and the ACPSE-SMVO. While this would increase the time necessary to complete the optimization step of the ACPSE-SMVO process, it would provide optimal values based on actual performance, rather than predicted values.

The second area for improvement is to validate the Pareto optimal solutions based on the regression model by running AutoGAD at the parameter settings that are predicted to belong in the Pareto set. However, both of these areas for improvement require knowledge concerning the automatic updating of AutoGAD parameters and algorithm execution that is beyond the scope of this research.

APPENDIX A. MATLAB® Code

Regression Model Objective Functions for Algorithm Parameter Optimization

```
function [XX,YY]=ObjFunc(x,p)

%XX(1)=FPF: Minimize;
%XX(2)=TFP: Maximize;
%XX(3)=TIME: Minimize;
%XX(4)=TPF: Maximize;
%p{1}=Dim Adj
%x(2)=Max Score
%x(3)=Bin width SNR
%x(4)=PT SNR thresh
%x(5)=Bin Width Ident
%x(6)=Smooth iter hi
%x(7)=Smooth iter lo
%x(8)=Low SNR
%x(9)=Window Size
%p{2}=Threshold Both Sides
%p{3}=Clean Signal

x(6)=round(x(6));
x(7)=round(x(7));
x(9)=round(x(9));

noise(1)= 0.006655*randn;
XX(1)=(...
  0.0256894114702926...
  + -0.0805017213692844*x(3)...
  + -0.155362540913017*x(5)...
  + -0.000238838141475098*x(8)...
  + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*0.0261666947382813)...
  + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*0.0211002302462076)...
  + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0000145388227577559)...
  + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*0.0000599450479063906)...
  + (x(2)-10.1757587359102)*((x(9)-4.72784810126582)*-0.000336878794516759)...
  + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*0.0372393225712019)...
  + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*3.90561673756179)...
  + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*0.000907974806650315)...
  + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*0.00354180766250731)...
  + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.00116189700958767)...
  + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.00430770537399931)...
  + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*-0.00925528299002467)...
  + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*-0.0199601034951385)...
  + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.0000038570725140484)...
  + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.0000077890598086508)...
  + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*0.249303506493041))...
  + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(7)-24.746835443038)*-0.00127033198999695))...
  + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-0.00674816865359704))...
  + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(6)-103.03164556962)*-0.0000054211855470922))...
  + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(7)-24.746835443038)*-0.0000109035024164093))...
  + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(9)-4.72784810126582)*0.0000328074068201622))...
  + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.000381660196391127))...
```

```
    + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.0000049073687867604))...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.0360774888548907))...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*-0.209991434825651))...
    + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000245690791925806))...
    + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.000731478427519515))...
    + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.000907736414737403))...
    + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*((x(7)-24.746835443038)*0.0000005648808451628))...
    + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.0000085179179207463))...
    + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.0000134656719077507))...
    + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.0000738058078761663))...
    + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.00043401736862272))...
    + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.00133604609305646))...
    + (x(6)-103.03164556962)*((x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.0000005431002695669))...
    + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.0000030716744510132))...
    );

if isequal(p{1},2)
    XX(1)=XX(1)+(...
        (x(3)-0.0557025001720556)*-0.0955747164287934...
        + (x(4)-3.55601517830807)*0.000623781761495857...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*0.0293501576114639)...
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0000239255643907588)...
        + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*0.0433021094372379)...
        + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*2.94240349906786)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-0.0126273178124193)...
        + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*-0.000521804332560101)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.00385445399656002)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.000002194994844654)...
        + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.0000745681075927964)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.0000642559545488319)...
        );
else
    XX(1)=XX(1)-(...
        (x(3)-0.0557025001720556)*-0.0955747164287934...
        + (x(4)-3.55601517830807)*0.000623781761495857...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*0.0293501576114639)...
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0000239255643907588)...
        + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*0.0433021094372379)...
        + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*2.94240349906786)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-0.0126273178124193)...
        + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*-0.000521804332560101)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.00385445399656002)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.000002194994844654)...
        + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.0000745681075927964)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.0000642559545488319)...
        );
end

if isequal(p{2},0)
    XX(1)=XX(1)+(...
        -0.00349643817808737...
        + (x(9)-4.72784810126582)*-0.000802253501553507...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*0.000404671755300744)...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*0.0159943145956149)...
        + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*0.0274185955008617)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*-0.00185816635893043)...
```

```
        + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*0.00489270610250974)...
        + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*-0.000206649491504432)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.00461893164812979)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.0000036038261148686)...
        + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.0000150332373369121)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.0000665068574050734)...
        );
else
    XX(1)=XX(1)-(...
        -0.00349643817808737...
        + (x(9)-4.72784810126582)*-0.000802253501553507...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*0.000404671755300744)...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*0.0159943145956149)...
        + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*0.0274185955008617)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*-0.00185816635893043)...
        + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*0.00489270610250974)...
        + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*-0.000206649491504432)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.00461893164812979)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.0000036038261148686)...
        + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.0000150332373369121)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.0000665068574050734)...
        );
end

if isequal(p{3},0)
    XX(1)=XX(1)+(...
        (x(2)-10.1757587359102)*-0.000364013706398382...
        + (x(3)-0.0557025001720556)*-0.068782342691512...
        + (x(5)-0.0557406029241196)*-0.141028110138679...
        + (x(7)-24.746835443038)*0.000169992054271948...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*0.000416982976333291)...
        + (x(2)-10.1757587359102)*((x(9)-4.72784810126582)*0.000263200632500861)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-0.00669637224106221)...
        + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.0222149173543354)...
        + (x(4)-3.55601517830807)*((x(9)-4.72784810126582)*-0.00046086769699504)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.0000159132598163982)...
        );
else
    XX(1)=XX(1)-(...
        (x(2)-10.1757587359102)*-0.000364013706398382...
        + (x(3)-0.0557025001720556)*-0.068782342691512...
        + (x(5)-0.0557406029241196)*-0.141028110138679...
        + (x(7)-24.746835443038)*0.000169992054271948...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*0.000416982976333291)...
        + (x(2)-10.1757587359102)*((x(9)-4.72784810126582)*0.000263200632500861)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-0.00669637224106221)...
        + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.0222149173543354)...
        + (x(4)-3.55601517830807)*((x(9)-4.72784810126582)*-0.00046086769699504)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.0000159132598163982)...
        );
end

XX(1)=XX(1)+noise(1);

noise(2)=0.16359*randn;
XX(2)=(...
```

```
    0.165578044367919...
    + -0.0129808651503392*x(4)...
    + 4.08235109484774*x(5)...
    + 0.000886940955334082*x(6)...
    + -0.00281165281765342*x(7)...
    + 0.0280914163084165*x(9)...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*-0.845255954588247)...
    + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*-0.260120520808717)...
    + (x(2)-10.1757587359102)*((x(9)- 4.72784810126582)*0.00293808193358571)...
    + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*0.00120690202051978)...
    + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*-0.00631506499336382)...
    + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*0.236240751806905)...
    + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.000717415247481852)...
    + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.000686940106098636)...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*0.152615053296013))...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*-13.0217827230349))...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*0.0484705571014071))...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.134211248704195))...
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*0.0986298966339441))...
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(6)-103.03164556962)*0.00015777415278073))...
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(7)-24.746835443038)*0.000297503681216606))...
    + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.00801843991472878))...
    + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*0.0498651870058177))...
    + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.0000156963278183928))...
    + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.000138472705449725))...
    + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.00118580498670985))...
    + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*((x(8)-8.95158620211076)*0.0746216481290795))...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(6)-103.03164556962)*0.585702796341571))...
    + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.00719726549853942))...
    + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.0258753920728689))...
    + (x(4)-3.55601517830807)*((x(5)- 0.0557406029241196)*((x(6)-103.03164556962)*0.0105448927965717))...
    + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.000246945490772528))...
    + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.000665078533093979))...
    + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.00126305545030453))...
    + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.00336310826013376))...
    );

if isequal(p{1},2)
    XX(2)=XX(2)+(...
        -0.100444017204813...
        +(x(2)-10.1757587359102)*0.0318821568353435...
        +(x(3)-0.0557025001720556)*-0.856863381576973...
        +(x(2)-10.1757587359102)*((x(9)-4.72784810126582)*0.00261205921903866)...
        +(x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*-0.414308785796218)...
        +(x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*0.44189361816745)...
        +(x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.373373201973405)...
        +(x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*-0.641177294911055)...
        +(x(6)-103.03164556962)*((x(7)-24.746835443038)*0.000116620377679345)...
        +(x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.000539889681160408)...
        +(x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.00149930830087245)...
        );
else
    XX(2)=XX(2)-(...
        -0.100444017204813...
        +(x(2)-10.1757587359102)*0.0318821568353435...
        +(x(3)-0.0557025001720556)*-0.856863381576973...
```

```
        +(x(2)-10.1757587359102)*((x(9)-4.72784810126582)*0.00261205921903866)...
        +(x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*-0.414308785796218)...
        +(x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*0.44189361816745)...
        +(x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.373373201973405)...
        +(x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*-0.641177294911055)...
        +(x(6)-103.03164556962)*((x(7)-24.746835443038)*0.000116620377679345)...
        +(x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.000539889681160408)...
        +(x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.00149930830087245)...
        );
end

if isequal(p{2},0)
    XX(2)=XX(2)+(...
        0.0799932960602052...
        +(x(3)-0.0557025001720556)*(0.612668774978395)...
        +(x(4)-3.55601517830807)*(-0.0274762666517147)...
        +(x(7)-24.746835443038)*(-0.00313608589066527)...
        +(x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*-0.517349086412935)...
        +(x(3)-0.0557025001720556)*((x(6)-103.03164556962)*-0.0164044791116466)...
        +(x(4)-3.55601517830807)*((x(9)-4.72784810126582)*-0.00322210425384521)...
        +(x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.0417626628623927)...
        +(x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-0.0956595935930405)...
        +(x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000325127534930812)...
        +(x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.000774019087293837)...
        +(x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.00193244787157434)...
        );
else
    XX(2)=XX(2)-(...
        0.0799932960602052...
        +(x(3)-0.0557025001720556)*(0.612668774978395)...
        +(x(4)-3.55601517830807)*(-0.0274762666517147)...
        +(x(7)-24.746835443038)*(-0.00313608589066527)...
        +(x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*-0.517349086412935)...
        +(x(3)-0.0557025001720556)*((x(6)-103.03164556962)*-0.0164044791116466)...
        +(x(4)-3.55601517830807)*((x(9)-4.72784810126582)*-0.00322210425384521)...
        +(x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.0417626628623927)...
        +(x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-0.0956595935930405)...
        +(x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000325127534930812)...
        +(x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.000774019087293837)...
        +(x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.00193244787157434)...
        );
end

if isequal(p{3},0)
    XX(2)=XX(2)+(...
        0.167498386946838...
        +(x(3)-0.0557025001720556)*(1.78334347248485)...
        +(x(5)-0.0557406029241196)*(1.48635412636791)...
        +(x(9)-4.72784810126582)*(-0.0304321776978525)...
        +(x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*32.0081709514872)...
        +(x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.736576657140496)...
        +(x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*-0.54616688237893)...
        +(x(4)-3.55601517830807)*((x(6)-103.03164556962)*0.000976447597367798)...
        +(x(4)-3.55601517830807)*((x(8)-8.95158620211076)*0.00587143486896801)...
        +(x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*0.525907138371367)...
        +(x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000410709791663283)...
```

```
        +(x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.000209578061445395)...
        +(x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.00155420212016607)...
        +(x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.00127536741797038)...
        );
else
    XX(2)=XX(2)-(...
        0.167498386946838...
        +(x(3)-0.0557025001720556)*(1.78334347248485)...
        +(x(5)-0.0557406029241196)*(1.48635412636791)...
        +(x(9)-4.72784810126582)*(-0.0304321776978525)...
        +(x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*32.0081709514872)...
        +(x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.736576657140496)...
        +(x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*-0.54616688237893)...
        +(x(4)-3.55601517830807)*((x(6)-103.03164556962)*0.000976447597367798)...
        +(x(4)-3.55601517830807)*((x(8)-8.95158620211076)*0.00587143486896801)...
        +(x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*0.525907138371367)...
        +(x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000410709791663283)...
        +(x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.000209578061445395)...
        +(x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.00155420212016607)...
        +(x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.00127536741797038)...
        );
end

XX(2)=-1*(XX(2)+noise(2));

noise(3)=2.839837*randn;
XX(3)=(...
    45.7149627856836...
    + -2.31144399533137*x(2)...
    + -3.00415850485113*x(4)...
    + 154.704094122475*x(5)...
    + -0.0431968769158141*x(6)...
    + 0.240174245048667*x(7)...
    + -0.393162278304518*x(8)...
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*-1.52879418328491)...
    + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0380428252396837)...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*-1429.88552945108)...
    + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*7.11443835425053)...
    + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-46.5633873826699)...
    + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*-103.487955222288)...
    + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*0.105251687946118)...
    + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*-0.478098099612669)...
    + (x(4)-3.55601517830807)*((x(9)-4.72784810126582)*-0.77467604974633)...
    + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-7.3840772841507)...
    + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-8.40901667311317)...
    + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*55.658273194977)...
    + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*-91.8026355045603)...
    + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.00464627455117163)...
    + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.00318496435826446)...
    + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.034239391626387)...
    + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.0638609917820322)...
    + (x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.089511430949877)...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*-962.494474014905))...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(7)-24.746835443038)*1.16459536146175))...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*3.418210017844))...
    + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-23.7212757458549))...
```

```
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(6)-103.03164556962)*-0.00292921669633108))...
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(7)-24.746835443038)*0.0752396147254257))...
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(8)-8.95158620211076)*-0.0437276636644996))...
    + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(9)-4.72784810126582)*-0.184582567367813))...
    + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.504068583898552))...
    + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*16.2225700776081))...
    + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*((x(7)-24.746835443038)*0.00102049763874207))...
    + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.00233911734990856))...
    + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.00516090300176989))...
    + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*-1236.64449534935))...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(6)-103.03164556962)*53.8305182295834))...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-165.289306701291))...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*528.943878927993))...
    + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*635.404493219827))...
    + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.215503190481792))...
    + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.557035587356886))...
    + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*-1.93555788446261))...
    + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-2.18225375658647))...
    + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*5.74407990803628))...
    + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.0301612327126137))...
    + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.108307552308221))...
    + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.126379074190482))...
    + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.295821026038062))...
    + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.695484440527229))...
    + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-3.82411019490071))...
    + (x(6)-103.03164556962)*((x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.000665991141949981))...
    + (x(6)-103.03164556962)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.00314784482738907))...
    + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.00593970451475243))...
    + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.0419979234305606))...
    );

if isequal(p{1},2)
    XX(3)=XX(3)+(...
        (x(2)-10.1757587359102)*-2.42096142740308...
        + (x(4)-3.55601517830807)*-0.401314661370035...
        + (x(5)-0.0557406029241196)*-72.3153991784875...
        + (x(7)-24.746835443038)*0.430113567257772...
        + (x(8)-8.95158620211076)*-0.457088752409798...
        + (x(9)-4.72784810126582)*0.342108469251351...
        + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*16.3363348983188)...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*15.3765019176104)...
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*0.073049934106511)...
        + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*0.0664514296427968)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*-2.51487829474736)...
        + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*-14.3229806662654)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-30.6383776830326)...
        + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*13.8393774937646)...
        + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*118.79863927424)...
        + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*-0.0174400902430602)...
        + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*0.412614603680372)...
        + (x(4)-3.55601517830807)*((x(9)-4.72784810126582)*0.275661305045246)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-5.59632018259808)...
        + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*4.19636212466563)...
        + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*35.7786673689535)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.0208374768478553)...
        );
```

```
else
    XX(3)=XX(3)-(...
        (x(2)-10.1757587359102)*-2.42096142740308...
        + (x(4)-3.55601517830807)*-0.401314661370035...
        + (x(5)-0.0557406029241196)*-72.3153991784875...
        + (x(7)-24.746835443038)*0.430113567257772...
        + (x(8)-8.95158620211076)*-0.457088752409798...
        + (x(9)-4.72784810126582)*0.342108469251351...
        + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*16.3363348983188)...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*15.3765019176104)...
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*0.073049934106511)...
        + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*0.0664514296427968)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*-2.51487829474736)...
        + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*-14.3229806662654)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*-30.6383776830326)...
        + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*13.8393774937646)...
        + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*118.79863927424)...
        + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*-0.0174400902430602)...
        + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*0.412614603680372)...
        + (x(4)-3.55601517830807)*((x(9)-4.72784810126582)*0.275661305045246)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-5.59632018259808)...
        + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*4.19636212466563)...
        + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*35.7786673689535)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.0208374768478553)...
        );
end

if isequal(p{2},0)
    XX(3)=XX(3)+(...
        + (x(3)-0.0557025001720556)*-79.6007832726288...
        + (x(4)-3.55601517830807)*-3.37934355176912...
        + (x(6)-103.03164556962)*0.0561870659818837...
        + (x(7)-24.746835443038)*0.0883614952268015...
        + (x(8)-8.95158620211076)*-0.431410435908149...
        + (x(9)-4.72784810126582)*-1.23832799208976...
        + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*106.362636381367)...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*2.5362235870835)...
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0229798753435252)...
        + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*0.347860472727556)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*0.657335735553628)...
        + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*-1.97216332257642)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*50.9085875433265)...
        + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*88.0183452801527)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*3.57541792186564)...
        + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*39.2229753181889)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.0031896630936398)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.00722320469816253)...
        + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.0601431356896265)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.0209593969894853)...
        + (x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.177493073191604)...
        );
else
    XX(3)=XX(3)-(...
        + (x(3)-0.0557025001720556)*-79.6007832726288...
        + (x(4)-3.55601517830807)*-3.37934355176912...
        + (x(6)-103.03164556962)*0.0561870659818837...
```

```
        + (x(7)-24.746835443038)*0.0883614952268015...
        + (x(8)-8.95158620211076)*-0.431410435908149...
        + (x(9)-4.72784810126582)*-1.23832799208976...
        + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*106.362636381367)...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*2.5362235870835)...
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0229798753435252)...
        + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*0.347860472727556)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*0.657335735553628)...
        + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*-1.97216332257642)...
        + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*50.9085875433265)...
        + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*88.0183452801527)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*3.57541792186564)...
        + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*39.2229753181889)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.0031896630936398)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.00722320469816253)...
        + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.0601431356896265)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.0209593969894853)...
        + (x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.177493073191604)...
        );
end


if isequal(p{3},0)
    XX(3)=XX(3)+(...
        1.12079151663465...
        + (x(3)-0.0557025001720556)*-133.17527743101...
        + (x(6)-103.03164556962)*-0.384488851757642...
        + (x(7)-24.746835443038)*0.301762343037636...
        + (x(8)-8.95158620211076)*0.535950903868837...
        + (x(9)-4.72784810126582)*-1.22269252186118...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*-1.16015146692986)...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*73.7966751036054)...
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0213032007120818)...
        + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*-0.476845188733536)...
        + (x(2)-10.1757587359102)*((x(9)-4.72784810126582)*-0.329431264149946)...
        + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*-45.2385755952501)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*4.66368504029263)...
        + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*139.346645238898)...
        + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*-0.0124653320242662)...
        + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*-0.127077131762122)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*0.00740386697861435)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.106606235064834)...
        + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.0305827320558507)...
        + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.0364888306192935)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.101790107791787)...
        );
else
    XX(3)=XX(3)-(...
        1.12079151663465...
        + (x(3)-0.0557025001720556)*-133.17527743101...
        + (x(6)-103.03164556962)*-0.384488851757642...
        + (x(7)-24.746835443038)*0.301762343037636...
        + (x(8)-8.95158620211076)*0.535950903868837...
        + (x(9)-4.72784810126582)*-1.22269252186118...
        + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*-1.16015146692986)...
        + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*73.7966751036054)...
```

```
        + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*-0.0213032007120818)...
        + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*-0.476845188733536)...
        + (x(2)-10.1757587359102)*((x(9)-4.72784810126582)*-0.329431264149946)...
        + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*-45.2385755952501)...
        + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*4.66368504029263)...
        + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*139.346645238898)...
        + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*-0.0124653320242662)...
        + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*-0.127077131762122)...
        + (x(6)-103.03164556962)*((x(7)-24.746835443038)*0.00740386697861435)...
        + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.106606235064834)...
        + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.0305827320558507)...
        + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*-0.0364888306192935)...
        + (x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.101790107791787)...
        );
end

XX(3)=XX(3)+noise(3);

noise(4)=0.115054*randn;
XX(4)=(...
   0.662789842196492...
   + -0.00866473634318355*x(2)...
   + -2.02695967600563*x(3)...
   + -1.81170568580437*x(5)...
   + 0.000585949630377693*x(6)...
   + 0.00400179610128431*x(8)...
   + 0.0327145563420182*x(9)...
   + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*-0.771226529564022)...
   + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*-0.00227883906367036)...
   + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*0.000393632643786534)...
   + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*0.191226881676559)...
   + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*16.9428211281166)...
   + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*0.240560397997758)...
   + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*-1.10594609099351)...
   + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*-0.000494173749224278)...
   + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*0.00185403380612153)...
   + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*0.00988551246296919)...
   + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.0750176644079506)...
   + (x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.000194126565383995)...
   + (x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.00421323177313059)...
   + (x(2)-10.1757587359102)*((x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*-2.6134104933345))...
   + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*0.126178438788599))...
   + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(6)-103.03164556962)*0.000129529899101966))...
   + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*((x(8)-8.95158620211076)*0.00203618920256686))...
   + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-0.0422685000157967))...
   + (x(2)-10.1757587359102)*((x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*-0.0255192924888868))...
   + (x(2)-10.1757587359102)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*-0.000164916508402042))...
   + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.000174762975207503))...
   + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.000716102285301043))...
   + (x(3)-0.0557025001720556)*((x(4)-3.55601517830807)*((x(8)-8.95158620211076)*0.280933194479036))...
   + (x(3)-0.0557025001720556)*((x(5)-0.0557406029241196)*((x(6)-103.03164556962)*1.0337524461506))...
   + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*((x(7)-24.746835443038)*-0.00169976812511343))...
   + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.00429744017521218))...
   + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.00470302489292601))...
   + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.00923517180681528))...
   + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*0.0332216585406472))...
```

```
    + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*0.0684378063280906))...
    + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*((x(7)-24.746835443038)*-0.0248700491577205))...
    + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*((x(8)-8.95158620211076)*0.000162823994515373))...
    + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*((x(9)-4.72784810126582)*0.000768657652452225))...
    + (x(4)-3.55601517830807)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.000853628886501693))...
    + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.0793741689770156))...
    + (x(6)-103.03164556962)*((x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.0000158997056323332))...
    + (x(6)-103.03164556962)*((x(7)-24.746835443038)*((x(9)-4.72784810126582)*-0.0000101649680209043))...
    );
if isequal(p{1},2)
    XX(4)=XX(4)+(...
        + (x(2)-10.1757587359102)*0.00806274348580709...
        + (x(4)-3.55601517830807)*-0.0180914133284487...
        + (x(6)-103.03164556962)*-0.000953346512619605...
        + (x(7)-24.746835443038)*-0.000912241469116581...
        + (x(9)-4.72784810126582)*0.0135014087576002...
        + (x(2)-10.1757587359102)*((x(4) -3.55601517830807)*0.00451972642003055)...
        + (x(2)-10.1757587359102)*((x(5) -0.0557406029241196)*-0.380196340906971)...
        + (x(3)-0.0557025001720556)*((x(8) -8.95158620211076)*0.651338440118295)...
        + (x(3)-0.0557025001720556)*((x(9) -4.72784810126582)*-0.759817023661298)...
        + (x(4)-3.55601517830807)*((x(5) -0.0557406029241196)*-0.470078182808422)...
        + (x(4)-3.55601517830807)*((x(7) -24.746835443038)*-0.00139001828288159)...
        + (x(4)-3.55601517830807)*((x(9) -4.72784810126582)*-0.00718031559859441)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.0703174374349332)...
        + (x(5)-0.0557406029241196)*((x(9) -4.72784810126582)*-0.291654642539285)...
        + (x(6)-103.03164556962)*((x(7) -24.746835443038)*0.0000923502745840698)...
        + (x(7)-24.746835443038)*((x(8) -8.95158620211076)*0.000411998835257628)...
        + (x(7)-24.746835443038)*((x(9) -4.72784810126582)*-0.000839022274019635)...
        );
else
    XX(4)=XX(4)-(...
        + (x(2)-10.1757587359102)*0.00806274348580709...
        + (x(4)-3.55601517830807)*-0.0180914133284487...
        + (x(6)-103.03164556962)*-0.000953346512619605...
        + (x(7)-24.746835443038)*-0.000912241469116581...
        + (x(9)-4.72784810126582)*0.0135014087576002...
        + (x(2)-10.1757587359102)*((x(4) -3.55601517830807)*0.00451972642003055)...
        + (x(2)-10.1757587359102)*((x(5) -0.0557406029241196)*-0.380196340906971)...
        + (x(3)-0.0557025001720556)*((x(8) -8.95158620211076)*0.651338440118295)...
        + (x(3)-0.0557025001720556)*((x(9) -4.72784810126582)*-0.759817023661298)...
        + (x(4)-3.55601517830807)*((x(5) -0.0557406029241196)*-0.470078182808422)...
        + (x(4)-3.55601517830807)*((x(7) -24.746835443038)*-0.00139001828288159)...
        + (x(4)-3.55601517830807)*((x(9) -4.72784810126582)*-0.00718031559859441)...
        + (x(5)-0.0557406029241196)*((x(7)-24.746835443038)*0.0703174374349332)...
        + (x(5)-0.0557406029241196)*((x(9) -4.72784810126582)*-0.291654642539285)...
        + (x(6)-103.03164556962)*((x(7) -24.746835443038)*0.0000923502745840698)...
        + (x(7)-24.746835443038)*((x(8) -8.95158620211076)*0.000411998835257628)...
        + (x(7)-24.746835443038)*((x(9) -4.72784810126582)*-0.000839022274019635)...
        );
end

if isequal(p{2},0)
    XX(4)=XX(4)+(...
       + (x(4)-3.55601517830807)*-0.0100341113261308...
       + (x(7)-24.746835443038)*-0.00392353331349028...
       + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*0.098518552959542)...
```

```
      + (x(4)-3.55601517830807)*((x(9)-4.72784810126582)*0.00922435480034716)...
      + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.0137131165360334)...
      + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*0.417674895700106)...
      + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000514116029582448)...
      + (x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.00282999736705026)...
      );
else
   XX(4)=XX(4)-(...
      + (x(4)-3.55601517830807)*-0.0100341113261308...
      + (x(7)-24.746835443038)*-0.00392353331349028...
      + (x(3)-0.0557025001720556)*((x(7)-24.746835443038)*0.098518552959542)...
      + (x(4)-3.55601517830807)*((x(9)-4.72784810126582)*0.00922435480034716)...
      + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*-0.0137131165360334)...
      + (x(5)-0.0557406029241196)*((x(9)-4.72784810126582)*0.417674895700106)...
      + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000514116029582448)...
      + (x(8)-8.95158620211076)*((x(9)-4.72784810126582)*-0.00282999736705026)...
      );
end

if isequal(p{3},0)
   XX(4)=XX(4)+(...
      + 0.151677058248746...
      + (x(3)-0.0557025001720556)*1.77723281825986...
      + (x(4)-3.55601517830807)*-0.0106386599832117...
      + (x(5)-0.0557406029241196)*0.683506911461439...
      + (x(9)-4.72784810126582)*-0.0179980048351728...
      + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*0.00853000698695022)...
      + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*-0.000906326458817658)...
      + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*-0.00234762030570042)...
      + (x(2)-10.1757587359102)*((x(9)-4.72784810126582)*0.0104392305842542)...
      + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*0.0402758463681714)...
      + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*0.210690450257682)...
      + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.755949908435926)...
      + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*-0.471328207210384)...
      + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*0.000681870964039866)...
      + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*0.00405050894732)...
      + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*0.0471105814702395)...
      + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*0.546098962371161)...
      + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000309445664711597)...
      + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.00144953972114957)...
      );
else
   XX(4)=XX(4)-(...
      + 0.151677058248746...
      + (x(3)-0.0557025001720556)*1.77723281825986...
      + (x(4)-3.55601517830807)*-0.0106386599832117...
      + (x(5)-0.0557406029241196)*0.683506911461439...
      + (x(9)-4.72784810126582)*-0.0179980048351728...
      + (x(2)-10.1757587359102)*((x(4)-3.55601517830807)*0.00853000698695022)...
      + (x(2)-10.1757587359102)*((x(7)-24.746835443038)*-0.000906326458817658)...
      + (x(2)-10.1757587359102)*((x(8)-8.95158620211076)*-0.00234762030570042)...
      + (x(2)-10.1757587359102)*((x(9)-4.72784810126582)*0.0104392305842542)...
      + (x(3)-0.0557025001720556)*((x(6)-103.03164556962)*0.0402758463681714)...
      + (x(3)-0.0557025001720556)*((x(8)-8.95158620211076)*0.210690450257682)...
      + (x(3)-0.0557025001720556)*((x(9)-4.72784810126582)*-0.755949908435926)...
      + (x(4)-3.55601517830807)*((x(5)-0.0557406029241196)*-0.471328207210384)...
```

```
      + (x(4)-3.55601517830807)*((x(6)-103.03164556962)*0.000681870964039866)...
      + (x(4)-3.55601517830807)*((x(8)-8.95158620211076)*0.00405050894732)...
      + (x(5)-0.0557406029241196)*((x(6)-103.03164556962)*0.0471105814702395)...
      + (x(5)-0.0557406029241196)*((x(8)-8.95158620211076)*0.546098962371161)...
      + (x(6)-103.03164556962)*((x(8)-8.95158620211076)*-0.000309445664711597)...
      + (x(7)-24.746835443038)*((x(8)-8.95158620211076)*0.00144953972114957)...
      );
end

XX(4)=-1*(XX(4)+noise(4));

if XX(1)<0;
   XX(1)=0;
end
if XX(3)<0;
   XX(3)=0;
end
if XX(2)>0;
   XX(2)=0;
end
if XX(4)>0;
   XX(4)=0;
end
if XX(1)>1;
   XX(1)=1;
end
if XX(2)<-1;
   XX(2)=-1;
end
if XX(4)<-1;
   XX(4)=-1;
end
YY=XX;
YY(2)=-1*XX(2);
YY(4)=-1*XX(4);
return
```

Regression Model Objective Functions for Alloy Composition

```
function [XX,YY]=alloyObjFunc(x,p)

%x(1)=Zr
%x(2)=Cu
%x(3)=Al
%x(4)=La
%x(5)=Cu,Ni
%x(6)=Pd
%p(1)=Si
noise(1)=3.965904*randn;
   XX(1)=662.510024591388 -3.43114455035605*x(4) + (x(1)-20.3584905660377)*((x(2)-
20.7056603773585)*0.125590764945869) + (x(2)-20.7056603773585)*((x(4)-30.6320754716981)*-
0.0245239528558381) + (x(3)-10.4377358490566)*((x(5)-9.00566037735849)*-0.353747895035932)+ (x(4)-
30.6320754716981)*((x(5)-9.00566037735849)*0.0192651005300172);
   if isequal(p,{'H'})
      XX(1)=XX(1) + (x(2)-20.7056603773585)*(-1868.78692225644) + (x(3)-10.4377358490566)*(-9916.67570077209)
+ (x(6)-7.29245283018868)*(-1874.09254189245);
   end
   if isequal(p,{'L'})
      XX(1)=XX(1) + (x(2)-20.7056603773585)*(1.90597750742905) + (x(3)-10.4377358490566)*(6.23475623467384) +
(x(6)-7.29245283018868)*(10.0172161940532);
   end
   if isequal(p,{'M'})
      XX(1)=XX(1) + (x(2)-20.7056603773585)*(1866.88094474901) + (x(3)-10.4377358490566)*(9910.44094453742) +
(x(6)-7.29245283018868)*(1864.0753256984);
   end
   XX(1)=-1*(XX(1)+noise(1));
   YY(1)=XX(1);

noise(2)=16.56018*randn;
   XX(2)=2562.00152741729 -152.580856107031*x(3) -3.71158027025736*x(4)+ (x(1)-20.3584905660377)*((x(2)-
20.7056603773585)*-0.670385666856341) + (x(1)-20.3584905660377)*((x(3)-10.4377358490566)*-
3.21324447564255) + (x(1)-20.3584905660377)*((x(5)-9.00566037735849)*-1.46846246552511) + (x(2)-
20.7056603773585)*((x(3)-10.4377358490566)*-1.07635463033817) + (x(2)-20.7056603773585)*((x(4)-
30.6320754716981)*-0.603448527190867) + (x(2)-20.7056603773585)*((x(6)-7.29245283018868)*-
0.575672006041501) + (x(3)-10.4377358490566)*((x(4)-30.6320754716981)*-2.744659435206) + (x(3)-
10.4377358490566)*((x(5)-9.00566037735849)*-2.20178804726434) + (x(4)-30.6320754716981)*((x(5)-
9.00566037735849)*-0.881030796432491);
   if isequal(p,{'H'})
      XX(2)=XX(2)+(x(3)-10.4377358490566)*(-88.0363292870266);
   end

   if isequal(p,{'L'})
      XX(2)=XX(2)+(x(3)-10.4377358490566)*(167.228480366726);
   end

   if isequal(p,{'M'})
      XX(2)=XX(2)+(x(3)-10.4377358490566)*(-79.1921510796991);
   end
   XX(2)=-1*(XX(2)+noise(2));
   YY(2)=XX(2);

  XX(3)=7.74549394593709 + -0.0630035413934196*x(3) + (x(1)-20.3584905660377)*((x(4)-
```

```
30.6320754716981)*0.000843976498489926) + (x(1)-20.3584905660377)*((x(6)-7.29245283018868)*-
0.000144126156888867) + (x(2)-20.7056603773585)*((x(5)-9.00566037735849)*0.0000445452387552537) + (x(3)-
10.4377358490566)*((x(4)-30.6320754716981)*-0.0000000927103338128);
  if isequal(p,{'H'})
    XX(3)=XX(3)+(x(4)-30.6320754716981)*(0.00592447215213463)+(x(6)-
7.29245283018868)*(0.0273146497007321);
  end

  if isequal(p,{'L'})
    XX(3)=XX(3)+(x(4)-30.6320754716981)*(-0.011199450354896)+(x(6)-7.29245283018868)*(-
0.0546292994031862);
  end

  if isequal(p,{'M'})
    XX(3)=XX(3)+(x(4)-30.6320754716981)*(0.00527497820276136)+(x(6)-
7.29245283018868)*(0.0273146497024542);
  end
  if XX(3)<0;
    XX(3)=0;
  end
  YY(3)=XX(3);
  for i=1:2
    if XX(i)>0
      XX(i)=0;
      YY(i)=0;
    end
  end
return
```

Objective Function Equations for Gearbox Design

```
function [XX,YY]=ObjFunc(x,p)

%XX(1)=Minimize volume;
%XX(2)=Minimize stress in shaft 1;
%XX(3)=Minimize stress in shaft 2;
%p{1}=Number of teeth of pinion
%x(1)=Gear face width
%x(2)=Teeth module
%x(3)=Distance between bearings on shaft 1
%x(4)=Distance between bearings on shaft 2
%x(5)=Diameter of shaft 1
%x(6)=Diameter of shaft 2
%noise=Random noise as a percentage of objective function value

noise(1)=0;
noise(2)=.005;
noise(3)=.005;

XX(1)=(...
   .7854*x(1)*(x(2)^2)...
   *(((round(x(7))^2)*(4/3))+14.9334*(round(x(7)))-43.0934)...
   -1.508*((x(5)^2)+(x(6)^2))...
   +7.477*((x(5)^3)+(x(6)^3))...
   +0.7854*((x(3))*(x(5)^2)+(x(4))*(x(6)^2))...
   );

XX(1)=XX(1)+noise(1)*XX(1)*randn;

XX(2)=(...
   (1/(.1*x(5)^3))...
   *sqrt(...
   (((745*x(3))/(x(2)*p{1}))^2)...
   +(1.69*(10^7))...
   )...
   );

XX(2)=XX(2)+noise(2)*XX(2)*randn;

XX(3)=(...
   (1/(.1*x(6)^3))...
   *sqrt(...
   (((745*x(4))/(x(2)*p{1}))^2)...
   +(1.575*(10^8))...
   )...
   );

XX(3)=XX(3)+noise(3)*XX(3)*randn;

YY=XX;

return
```

Objective Function Equations for System Reliability

```
function [XX,YY]=ObjFunc(x,p)

%XX(1)=Minimize unreliability (1 - Reliability);
%XX(2)=Minimize Design Cost;
%XX(3)=Minimize Design Weight;
%x(1)=Number of type 1 components in sub-system 1
%x(2)=Number of type 2 components in sub-system 1
%x(3)=Number of type 3 components in sub-system 1
%x(4)=Number of type 1 components in sub-system 2
%x(5)=Number of type 2 components in sub-system 2
%x(6)=Number of type 3 components in sub-system 2
%x(7)=Operating temperature of sub-system 1
%x(8)=Operating temperature of sub-system 2
%R(1)=Reliability of type 1 component as a function of operating
%temperature
%R(2)=Reliability of type 2 component as a function of operating
%temperature
%Q(i)=Unreliability of sub-system i
%noise=Random noise as a percentage of objective function value

for i=1:9
   x(i)=round(x(i));
end

noise(1)=0;
noise(2)=.005;
noise(3)=.005;
if (sum(x(1:3))>0)&&(sum(x(4:6))>0)&&(sum(x(7:10))>0)
for i=1:3
   R(1)=.8-.001*x(9+i);
   R(2)=.85-.001*x(9+i);
   R(3)=.89-.005*x(9+i);
   Q(i)=((1-R(1))^x(3*i-2))*((1-R(2))^x(3*i-1))*((1-R(3))^x(3*i));
end
XX(1)=Q(1)*Q(2)*Q(3);
XX(1)=XX(1)+norm(noise(1)*XX(1)*randn);

XX(2)=(...
   (x(1)+x(2)+x(3))*(50-x(10))...
   + (x(4)+x(5)+x(6))*((50-x(11))^2)...
   + (x(7)+x(8)+x(9))*((50-x(12))^3)...
   + 5*(x(1)+x(4)+x(7))...
   + 10*(x(2)+x(5)+x(8))...
   + 20*(x(3)+x(6)+x(9))...
   );

XX(2)=XX(2)+noise(2)*XX(2)*randn;

XX(3)=(...
   20*(x(1)+x(4)+x(7))...
   + 10*(x(2)+x(5)+x(8))...
   + 5*(x(3)+x(6)+x(9))...
   );
```

```
XX(3)=XX(3)+noise(3)*XX(3)*randn;
else
   XX(1)=1;
   XX(2)=Inf;
   XX(3)=Inf;
end

YY=XX;

return
```

APPENDIX B. AutoGAD Output

AutoGAD Output data from D-optimal design performed by Matthew Davis [23].

Dimension Adjust	Max Score Threshold	Bin Width SNR	PT SNR Threshold	Bin Width Identify	Smooth Iterations High	Smooth Iterations Low	Low SNR	Window Size	Threshold Both Sides	Clean Signal	Time to Completion	True Positive Fraction	False Positive Fraction	Total Fraction of Positives
2	14.00	0.01	1.00	0.10	128	45	14.00	9	1	1	122.5947	0.8355	0.0010	0.8696
2	14.00	0.10	6.00	0.10	50	5	14.00	9	0	0	2.9468	0.8585	0.0006	0.9726
2	7.78	0.10	1.00	0.01	150	45	10.67	9	1	1	2.9306	1.0000	0.0000	1.0000
2	14.00	0.10	1.00	0.06	50	5	4.00	1	1	0	2.2418	0.9572	0.0016	0.9405
2	6.89	0.10	1.00	0.10	94	45	11.78	1	0	1	29.4813	0.0000	0.0000	0.0000
2	6.00	0.01	1.00	0.10	150	45	4.00	1	0	0	115.0929	0.8845	0.0051	0.5501
2	6.00	0.01	1.00	0.07	73	5	4.00	1	0	1	75.7169	0.0000	0.0000	0.0000
2	6.00	0.01	1.00	0.10	50	5	14.00	9	1	0	2.7026	1.0000	0.0060	0.0000
2	14.00	0.01	6.00	0.07	150	45	14.00	1	0	0	2.37	1.0000	0.0000	1.0000
2	14.00	0.10	6.00	0.01	150	5	14.00	1	0	1	20.3574	0.0000	0.0000	0.0000
2	7.78	0.01	6.00	0.01	94	45	14.00	1	1	1	13.8299	0.0000	0.0000	0.0000
2	6.00	0.10	2.67	0.10	50	45	4.00	9	0	0	4.2561	0.0085	0.0021	0.0165
2	6.00	0.01	6.00	0.10	50	5	4.00	1	1	1	3.8197	1.0000	0.0000	1.0000
2	6.00	0.10	1.00	0.01	50	5	14.00	1	0	1	2.174	1.0000	0.0000	1.0000
2	14.00	0.10	6.00	0.10	50	45	14.00	5	1	0	3.7498	0.4208	0.0000	0.9977
2	12.22	0.01	6.00	0.01	150	5	4.00	1	0	0	1.8924	0.9545	0.0661	0.1355
2	6.00	0.08	1.00	0.01	150	5	14.00	1	1	1	15.7267	0.0000	0.0000	0.0000
2	6.00	0.06	6.00	0.02	117	9	14.00	9	1	0	60.5903	0.9048	0.0311	0.1378
2	14.00	0.03	1.00	0.01	150	5	4.00	4	0	1	3.0282	0.9790	0.0113	0.7949
2	6.00	0.10	1.00	0.01	50	45	14.00	1	0	0	3.9133	0.9679	0.0230	0.6284
2	6.00	0.01	6.00	0.10	150	45	14.00	9	1	0	4.6393	0.5149	0.0016	0.5628
2	14.00	0.10	6.00	0.01	150	45	4.00	1	0	1	161.5426	0.0000	0.0000	0.0000
2	14.00	0.10	1.00	0.10	150	5	7.33	9	1	0	73.0666	0.8165	0.0038	0.5287
2	14.00	0.10	1.00	0.01	150	27	14.00	1	1	1	20.398	0.0000	0.0000	0.0000
2	6.00	0.10	1.00	0.01	150	5	4.00	9	0	1	115.3771	0.9723	0.0158	0.4145
2	14.00	0.01	6.00	0.01	50	45	4.00	1	0	1	2.0915	1.0000	0.0000	1.0000
2	14.00	0.10	1.00	0.01	50	45	14.00	1	0	1	17.643	0.0000	0.0000	0.0000
2	6.00	0.07	1.00	0.10	150	5	4.00	1	1	1	20.3076	0.0000	0.0000	0.0000
2	14.00	0.01	6.00	0.10	50	5	14.00	9	1	0	1.862	1.0000	0.0000	1.0000
2	14.00	0.10	1.00	0.10	50	5	14.00	9	1	1	4.9518	0.0000	0.0016	0.0000
2	6.00	0.01	3.22	0.01	50	5	8.51	1	1	0	3.262	1.0000	0.0720	0.3555
2	14.00	0.01	1.00	0.01	50	5	14.00	1	0	0	114.0148	0.9695	0.0357	0.1854

Dimension Adjust	Max Score Threshold	Bin Width SNR	PT SNR Threshold	Bin Width Identify	Smooth Iterations High	Smooth Iterations Low	Low SNR	Window Size	Threshold Both Sides	Clean Signal	Time to Completion	True Positive Fraction	False Positive Fraction	Total Fraction of Positives
2	14.00	0.01	1.00	0.10	106	45	4.00	9	0	1	2.2467	1.0000	0.0000	1.0000
2	14.00	0.01	3.22	0.08	50	5	4.00	1	1	0	3.1197	0.5200	0.0003	0.9820
2	14.00	0.01	1.00	0.01	150	5	4.00	6	1	0	1.8657	1.0000	0.0000	1.0000
2	6.00	0.10	1.00	0.10	150	5	4.00	9	1	0	3.1028	0.9126	0.0011	0.9534
2	14.00	0.01	6.00	0.10	150	5	14.00	1	1	1	15.3021	0.0000	0.0000	0.0000
2	8.67	0.01	6.00	0.06	150	45	4.00	9	0	1	6.3434	0.6726	0.0008	0.9610
2	10.43	0.06	1.00	0.10	150	5	14.00	9	0	0	3.4368	0.4479	0.0009	0.9450
2	14.00	0.10	6.00	0.01	50	45	7.35	9	0	1	3.0129	0.6505	0.0177	0.2966
2	6.00	0.01	3.22	0.01	150	5	14.00	9	0	0	2.4645	1.0000	0.1135	0.0000
2	14.00	0.04	1.00	0.01	50	45	7.33	1	1	0	4.9915	0.0000	0.0186	0.0000
2	14.00	0.10	1.00	0.03	150	45	14.00	9	1	1	2.4199	1.0000	0.0000	1.0000
2	14.00	0.10	1.00	0.10	150	45	4.00	1	0	0	2.4057	0.7971	0.0035	0.6587
2	6.00	0.10	6.00	0.10	50	32	4.00	1	1	0	2.7378	0.7820	0.0125	0.3377
2	6.00	0.10	6.00	0.10	150	5	4.00	1	0	0	4.2006	0.4293	0.0010	0.9389
2	14.00	0.10	1.00	0.01	150	5	4.00	1	1	0	4.0828	1.0000	0.0000	1.0000
2	14.00	0.10	6.00	0.10	150	40	10.25	1	1	1	9.542	0.0000	0.0000	0.0000
2	6.00	0.01	1.00	0.01	50	5	4.00	9	0	0	2.1665	0.9568	0.1120	0.0888
2	6.00	0.01	6.00	0.01	50	45	4.00	9	1	0	114.0467	0.9757	0.0966	0.0811
2	6.00	0.01	1.00	0.10	50	45	4.00	1	1	0	2.083	1.0000	0.0014	0.0000
2	6.00	0.01	6.00	0.01	50	5	14.00	5	0	0	4.4018	0.8275	0.0317	0.1034
2	14.00	0.10	6.00	0.01	150	45	4.00	1	0	1	161.8296	0.0000	0.0000	0.0000
2	6.00	0.10	5.44	0.10	150	5	14.00	9	0	1	3.9271	0.7688	0.0035	0.6833
2	12.22	0.01	6.00	0.10	50	32	8.44	9	0	0	78.1869	0.7949	0.0024	0.6294
2	6.00	0.10	6.00	0.10	50	5	14.00	1	1	0	114.2106	0.8906	0.0063	0.4949
2	14.00	0.10	6.00	0.01	50	5	4.00	1	1	1	3.6441	0.0000	0.0000	0.0000
2	14.00	0.01	6.00	0.01	150	45	4.00	9	1	1	68.5278	0.7101	0.0121	0.3135
2	14.00	0.01	6.00	0.01	150	45	14.00	9	0	0	3.4453	0.9663	0.0053	0.8741
2	6.00	0.10	6.00	0.01	83	45	4.00	1	0	0	2.4833	1.0000	0.0103	0.0000
2	8.64	0.10	4.89	0.10	150	5	14.00	1	1	0	2.0776	1.0000	0.0000	1.0000
2	6.00	0.01	6.00	0.10	150	5	6.36	9	0	1	2.0125	1.0000	0.0011	0.0000
2	14.00	0.01	1.00	0.01	50	45	4.00	3	1	0	2.3949	0.9581	0.0879	0.1139
2	6.00	0.10	6.00	0.01	150	45	14.00	1	1	0	2.2803	1.0000	0.0413	0.4575
2	14.00	0.01	1.00	0.10	150	45	14.00	1	1	0	3.1524	0.9093	0.0005	0.9761
2	14.00	0.01	1.00	0.10	50	45	14.00	9	1	1	4.9425	0.6561	0.0003	0.9850
2	14.00	0.03	6.00	0.01	50	5	4.00	9	1	1	2.6556	1.0000	0.0000	1.0000
2	6.00	0.10	1.00	0.01	150	5	4.00	9	0	1	114.5794	0.9675	0.0153	0.4408
2	6.00	0.10	1.00	0.07	150	45	4.00	5	1	1	7.6229	0.4730	0.0025	0.8761

Dimension Adjust	Max Score Threshold	Bin Width SNR	PT SNR Threshold	Bin Width Identify	Smooth Iterations High	Smooth Iterations Low	Low SNR	Window Size	Threshold Both Sides	Clean Signal	Time to Completion	True Positive Fraction	False Positive Fraction	Total Fraction of Positives
2	6.00	0.01	6.00	0.10	150	45	4.00	1	1	0	64.2793	0.7922	0.0048	0.4519
2	6.00	0.10	6.00	0.10	150	45	14.00	9	0	1	3.3736	1.0000	0.0012	0.0000
2	14.00	0.10	1.00	0.10	83	45	14.00	7	0	0	25.6183	1.0000	0.0000	1.0000
2	6.00	0.01	6.00	0.10	150	5	6.36	9	0	1	2.2722	1.0000	0.0000	1.0000
2	14.00	0.01	6.00	0.10	50	45	14.00	1	0	1	10.3465	0.0000	0.0000	0.0000
2	10.44	0.01	1.00	0.10	150	18	4.00	9	1	1	3.0974	0.6364	0.0037	0.6203
2	6.00	0.01	1.00	0.01	150	45	14.00	1	0	1	48.7212	0.0000	0.0000	0.0000
-2	12.22	0.10	2.11	0.01	139	5	14.00	9	1	1	3.8639	0.9377	0.0099	0.7861
-2	14.00	0.10	1.00	0.10	50	45	6.22	9	0	0	99.8503	0.8638	0.0007	0.9000
-2	14.00	0.10	1.00	0.10	150	5	14.00	1	0	1	112.049	0.0000	0.0000	0.0000
-2	14.00	0.01	1.00	0.01	150	45	14.00	9	1	0	2.1229	0.9545	0.0815	0.1128
-2	6.00	0.09	1.00	0.10	50	5	4.00	1	1	1	8.6633	0.0000	0.0000	0.0000
-2	6.00	0.10	3.78	0.01	50	5	4.00	6	1	0	1.162	1.0000	0.0000	1.0000
-2	14.00	0.01	6.00	0.10	94	5	14.00	9	0	1	2.5694	1.0000	0.0000	1.0000
-2	6.00	0.01	6.00	0.01	107	5	14.00	1	1	1	5.3885	1.0000	0.0000	1.0000
-2	14.00	0.10	6.00	0.10	50	45	14.00	9	1	0	2.6648	0.6959	0.0013	0.7203
-2	14.00	0.10	1.00	0.01	50	45	14.00	9	0	0	26.0713	1.0000	0.0000	1.0000
-2	6.00	0.10	6.00	0.01	50	45	14.00	1	1	1	3.3168	0.0000	0.0000	0.0000
-2	14.00	0.07	6.00	0.01	150	45	4.00	9	1	1	7.257	0.0000	0.0000	0.0000
-2	11.66	0.10	1.00	0.01	150	45	6.56	2	0	1	8.0544	0.0000	0.0074	0.0000
-2	14.00	0.03	1.00	0.10	150	5	4.00	9	1	0	24.5681	1.0000	0.0000	1.0000
-2	14.00	0.01	6.00	0.01	150	14	12.89	9	0	1	102.466	0.9288	0.0057	0.6784
-2	11.23	0.01	6.00	0.02	50	5	4.00	1	0	1	8.9165	0.0000	0.0000	0.0000
-2	14.00	0.01	4.89	0.10	150	45	14.00	5	1	1	1.2539	1.0000	0.0000	1.0000
-2	6.00	0.06	6.00	0.10	50	45	8.44	5	1	0	1.6472	0.9011	0.0011	0.9527
-2	6.00	0.07	1.56	0.01	106	45	14.00	9	0	1	5.241	0.6205	0.0067	0.8036
-2	6.00	0.01	6.00	0.01	50	5	14.00	9	1	1	2.1245	0.8049	0.0552	0.1558
-2	10.47	0.06	6.00	0.10	150	5	4.00	9	0	1	2.513	0.7661	0.0008	0.8397
-2	6.00	0.10	6.00	0.01	50	5	4.00	9	0	1	2.9034	0.9163	0.0117	0.7462
-2	14.00	0.01	6.00	0.04	50	45	4.00	9	1	1	3.4	0.8580	0.0000	1.0000
-2	14.00	0.01	1.00	0.10	50	5	7.33	1	0	1	1.1967	1.0000	0.0000	1.0000
-2	13.11	0.01	1.00	0.01	150	5	14.00	1	0	0	2.5269	0.9561	0.0056	0.8669
-2	6.00	0.01	1.00	0.10	150	45	4.00	1	1	1	22.7063	0.0000	0.0000	0.0000
-2	6.00	0.10	6.00	0.06	50	5	14.00	9	0	0	2.4379	0.4512	0.0016	0.9077
-2	6.00	0.10	1.00	0.01	116	5	14.00	9	0	0	5.652	0.0170	0.0144	0.0048
-2	6.00	0.01	6.00	0.01	150	45	4.00	9	1	1	2.1729	1.0000	0.0284	0.0000
-2	6.00	0.01	2.67	0.01	150	45	4.00	9	1	0	5.4065	0.0170	0.0150	0.0046

Dimension Adjust	Max Score Threshold	Bin Width SNR	PT SNR Threshold	Bin Width Identify	Smooth Iterations High	Smooth Iterations Low	Low SNR	Window Size	Threshold Both Sides	Clean Signal	Time to Completion	True Positive Fraction	False Positive Fraction	Total Fraction of Positives
-2	14.00	0.10	6.00	0.01	50	45	10.67	1	1	0	1.3248	1.0000	0.0000	1.0000
-2	6.00	0.01	6.00	0.01	150	5	14.00	9	0	0	2.1182	0.9838	0.0204	0.6009
-2	14.00	0.01	6.00	0.10	139	45	4.00	1	0	0	7.4978	0.0000	0.0004	0.0000
-2	6.00	0.10	1.00	0.10	50	45	14.00	9	1	1	3.0995	0.5203	0.0002	0.9506
-2	14.00	0.01	1.00	0.01	50	5	14.00	1	1	1	1.7164	1.0000	0.0000	1.0000
-2	14.00	0.10	1.00	0.10	150	45	14.00	3	1	0	6.8088	0.0000	0.0000	0.0000
-2	14.00	0.10	6.00	0.10	50	5	4.00	9	1	0	3.0377	0.0000	0.0000	0.0000
-2	14.00	0.10	1.00	0.01	50	5	4.00	1	0	1	12.753	0.0000	0.0000	0.0000
-2	14.00	0.01	6.00	0.01	106	23	4.00	9	1	0	2.1696	0.9928	0.0242	0.5661
-2	6.00	0.10	1.00	0.10	150	45	14.00	1	1	0	2.2943	1.0000	0.0000	1.0000
-2	14.00	0.07	6.00	0.06	150	45	4.00	1	1	0	2.6914	0.4579	0.0006	0.9657
-2	6.00	0.10	1.00	0.01	150	45	14.00	9	1	0	88.194	0.9659	0.0527	0.1293
-2	6.00	0.01	1.00	0.01	50	5	14.00	9	0	0	2.7577	0.8844	0.0859	0.0551
-2	9.56	0.10	6.00	0.01	150	45	14.00	5	0	0	1.6474	0.9609	0.0555	0.1604
-2	14.00	0.10	1.00	0.10	150	45	7.33	9	0	1	3.349	0.5932	0.0000	1.0000
-2	6.00	0.01	1.00	0.10	150	5	8.44	1	1	0	1.8088	0.8288	0.0052	0.5874
-2	6.00	0.01	2.63	0.10	50	45	14.00	1	1	1	15.5156	0.0000	0.0000	0.0000
-2	6.00	0.10	6.00	0.10	50	45	4.00	1	0	1	1.2303	1.0000	0.0000	1.0000
-2	6.00	0.01	1.00	0.10	50	45	4.00	9	1	1	8.5107	0.1568	0.0010	0.3814
-2	6.00	0.10	1.00	0.01	50	45	4.00	1	0	0	2.1368	0.9838	0.0267	0.5369
-2	6.00	0.10	6.00	0.01	150	5	10.67	1	0	1	3.4011	0.0000	0.0000	0.0000
-2	14.00	0.01	5.44	0.01	150	5	14.00	1	1	0	5.7714	0.0170	0.0153	0.0045
-2	14.00	0.01	6.00	0.10	50	45	4.00	9	0	0	2.0116	0.8194	0.0041	0.6413
-2	14.00	0.10	1.00	0.01	150	5	4.00	9	1	0	1.5736	0.9489	0.0775	0.1172
-2	14.00	0.01	6.00	0.10	150	5	14.00	1	0	0	2.799	0.6351	0.0011	0.7402
-2	6.00	0.01	2.67	0.10	150	45	4.00	9	0	0	2.6997	0.4518	0.0011	0.9343
-2	14.00	0.10	6.00	0.01	50	45	4.00	6	0	0	3.1509	0.8061	0.0085	0.3393
-2	14.00	0.01	6.00	0.10	94	5	14.00	9	0	1	1.6816	1.0000	0.0000	1.0000
-2	6.00	0.10	1.00	0.01	50	32	4.00	9	1	1	4.3013	0.6986	0.0039	0.5509
-2	6.00	0.01	1.00	0.01	50	18	4.00	1	0	0	2.6287	1.0000	0.0533	0.0000
-2	6.00	0.02	6.00	0.09	150	40	12.75	1	0	0	2.9558	0.4794	0.0013	0.9295
-2	14.00	0.10	1.00	0.10	50	5	4.00	5	0	0	2.7715	0.4518	0.0000	0.9979
-2	6.00	0.10	6.00	0.10	150	32	4.00	1	1	1	133.8812	0.0000	0.0000	0.0000
-2	14.00	0.10	5.44	0.10	150	5	4.00	1	0	0	2.9141	0.8642	0.0005	0.9765
-2	6.00	0.10	6.00	0.01	50	5	14.00	1	1	0	2.5646	1.0000	0.0000	1.0000
-2	14.00	0.01	6.00	0.10	150	5	4.00	1	0	1	3.2137	0.0000	0.0000	0.0000
-2	14.00	0.10	6.00	0.10	150	5	14.00	1	1	1	6.9816	0.0000	0.0000	0.0000

Dimension Adjust	Max Score Threshold	Bin Width SNR	PT SNR Threshold	Bin Width Identify	Smooth Iterations High	Smooth Iterations Low	Low SNR	Window Size	Threshold Both Sides	Clean Signal	Time to Completion	True Positive Fraction	False Positive Fraction	Total Fraction of Positives
-2	14.00	0.01	6.00	0.01	50	45	14.00	1	0	0	1.8728	0.9818	0.0177	0.6275
-2	14.00	0.06	6.00	0.10	50	5	14.00	1	0	1	4.7273	0.0000	0.0000	0.0000
-2	6.00	0.10	1.00	0.07	150	5	4.00	1	0	1	1.7776	1.0000	0.0000	1.0000
-2	14.00	0.10	1.00	0.10	150	5	14.00	1	0	1	128.5538	0.0000	0.0000	0.0000
-2	6.00	0.10	6.00	0.10	50	45	4.00	1	0	1	1.2963	1.0000	0.0000	1.0000
-2	11.33	0.10	1.00	0.10	50	5	4.00	1	0	1	2.227	0.0000	0.0000	0.0000
-2	6.00	0.10	6.00	0.01	150	5	10.67	1	0	1	3.5503	0.0000	0.0000	0.0000
-2	13.11	0.01	6.00	0.10	50	45	14.00	1	1	0	100.3038	0.8638	0.0019	0.7561
-2	6.00	0.01	1.00	0.08	50	14	14.00	5	0	1	3.6425	0.8598	0.0000	1.0000
-2	14.00	0.10	6.00	0.10	50	45	4.00	1	1	1	1.7261	1.0000	0.0000	1.0000
-2	6.00	0.01	1.00	0.10	150	9	14.00	9	1	1	2.1864	1.0000	0.0000	1.0000
-2	14.00	0.01	1.00	0.03	150	32	4.00	1	1	0	2.3232	0.7826	0.0056	0.4257
-2	6.00	0.01	1.00	0.01	50	45	14.00	9	1	0	2.2723	0.9801	0.0393	0.5054
-2	14.00	0.10	6.00	0.01	150	28	4.00	9	0	0	1.345	1.0000	0.0000	1.0000
-2	14.00	0.01	1.00	0.01	50	5	4.00	9	1	0	95.7794	0.9653	0.0538	0.1250

APPENDIX C. Blue Dart Submission Form

Blue Dart Submission Form

First Name: Christopher Last Name: Arendt

Rank (Military, AD, etc.): CAPT / AD Designator AFIT/GOR/ENS/09-01

Student's Involved in Research for Blue Dart: Christopher Arendt

Position/Title: Master's Student

Phone Number: ________________ E-mail: christopher.arendt@afit.edu

School/Organization: Air Force Institute of Technology

Status: [X] Student [ ] Faculty [ ] Staff [ ] Other

Optimal Media Outlet (optional): ________________________________

Optimal Time of Publication (optional): ____________________________

General Category / Classification:
[ ] core values [ ] command [ ] strategy
[ ] war on terror [ ] culture & language [ ] leadership & ethics
[ ] warfighting [ ] international security [ ] doctrine
[X] other (specify): Acquisitions

Suggested Headline: Efficient Remedy for Acquisition Ailments

Keywords: simulation based test & evaluation, trade-off analysis, efficiency

Blue Dart Text:

Many in Congress and the press have attacked the Department of Defense acquisition process as bloated, out-dated, wasteful and inefficient. These critics have singled out the Air Force for particularly harsh criticism due to the staggering price-tags associated with the acquisitions of the F-22 and the Joint Strike Fighter as well as its seeming unwillingness to embrace this new era of transition and change.

In addition, the Air Force embarrassed itself with its handling of the high profile Tanker Lease acquisition program. In fact, the Secretary of Defense was so unimpressed with the Air Force's attempts to find a new lead developer for the Tanker, that he decided the Air Force will no longer be the Acquisition Executive for its own future aircraft.

Clearly, the Air Force must adapt quickly and decisively to remedy its real and perceived failures in the Defense Acquisition process. Since the Defense Acquisition University estimates that nearly 80% of an acquisition's total costs have been built in to the program before it even enters Operational Test & Evaluation, the Air Force would do well to find a way to increase efficiency and lower costs very early in the acquisitions process.

In many cases, Air Force acquisition programs begin as research projects at the Air Force Research Laboratory, where immature designs progress through a series of tests and experiments. Since the physical testing and experimentation of new designs can be prohibitively expensive, the Air Force has pushed for simulation-based acquisitions in which engineers obtain physical results from a small sub-set of possible designs and use mathematical models and computer simulations to predict the performance of untested designs. An optimization technique that can handle the inherent randomness and complexities of simulation based testing could be used to eliminate inefficient designs through the analysis of simulated responses. This technique should select which efficient designs warrant further investigation and are worth the costs of physical testing.

Fortunately, researchers at the Air Force Institute of Technology's Department of Operational Sciences have developed dynamic optimization techniques that should prove effective in rehabilitating the early stages of many Air Force acquisition programs. Principal among these new methods is an adaptive process that can automatically and efficiently evaluate and display to a decision maker the benefits and trade-offs of a wide range of engineering design problems.

Whereas current optimization methods may only be able to find the design that is best in a single objective or one specific measure of performance, this new adaptive evaluation method can be used to find a wide variety of designs that represent efficient performance in multiple objectives, across a broad range of criteria, including life-cycle logistics and management costs.

Furthermore, this new optimization technique was developed specifically to work with simulation-based testing and to handle the many different kinds of variables needed to adequately describe real-world conditions. These qualities make the optimization method a natural and powerful tool for simulation-based acquisitions.

If implemented in the earliest research and design stages of Air Force acquisition programs, this new optimization method has the potential to eliminate the test and evaluation costs of inherently inferior designs. Further, this method is not limited to helping in the basic engineering phase of an acquisition program. It can easily be adapted to find and eliminate many kinds of programmatic inefficiencies in all stages of the acquisition process, including requirements or capabilities analysis, contract competition and program scheduling.

Adopting this efficiency-based optimization method in conjunction with a renewed emphasis on modeling and simulation in all phases of the acquisition process is a clear, direct and responsible approach to answering Air Force critics. The Air Force has invested a great deal of time, energy, and money to develop breakthrough technologies. Now is the perfect time for the Air Force to use one of these homegrown discoveries to take the lead and establish a new standard of excellence for acquisitions in the Department of Defense.

Disclaimer is placed on the bottom : ***The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government.***

Mar 09

Bibliography

1. Abramson, M.A. (2002). *Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems*. Ph.D. thesis, Rice University, Department of Computational and Applied Mathematics (CAAM). Also available as Rice CAAM Technical Report TR02-11.

2. Abramson, M.A. (2006). NOMADm version 4.02. Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

3. Abramson, M.A. , Audet, C., and Dennis, Jr., J.E. (to appear). Filter Pattern Search Algorithms for Mixed Variable Constrained Optimization Problems. *Pacific Journal of Optimization*. Also appears as Technical Report TR04-09, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 2004.

4. Andersson, J. and Krus, P. (2001). Multiobjective Optimization of Mixed Variable Design Problems. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, **1993**, 624-638.

5. Audet, C. and Dennis, Jr., J.E. (2000). Pattern Search Algorithms for Mixed Variable Programming. *SIAM Journal on Optimization*, **11**(3), 573-594.

6. Audet, C. and Dennis, Jr., J.E. (2004). A Pattern Search Filter Method for Nonlinear Programming without Derivatives. *SIAM Journal on Optimization*, **14**(4), 980-1010.

7. Audet, C. and Dennis, Jr., J.E. (2006). Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, **17**(1), 188-217.

8. Audet, C. and Orban, D. (2006). Finding Optimal Algorithmic Parameters Using Derivative-free Optimization. *SIAM Journal on Optimization*, **17**(3), 642-664.

9. Audet, C., Savard, G. and Zghal, W. (2008). Multiobjective Optimization Through a Series of Single-Objective Formulations. *SIAM Journal on Optimization*, **19**(1), 188-210.

10. Azarm, S., Tits, A.L., and Fan, M.K.H. (1989). Tradeoff-Driven Optimization-Based Design of Mechanical Systems. *4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, 551-558.

11. Bandyopadhyay, S., Saha, S., Maulik, U. and Deb, K. (2008). A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, **12**(3).

12. Baykasoglu, A. (2006). Applying Multiple Objective Tabu Search to Continuous Optimization Problems with a Simple Neighbourhood Strategy. *International Journal for Numerical Methods in Engineering*, **65**, 406-424.

13. Baykasoglu, A., Özbakir, L. and Sönmez, A.I. (2004). Using Multiple Objective Tabu Search and Grammars to Model and Solve Multi-Objective Flexible Job Shop scheduling problems. *Journal of Intelligent Manufacturing*, **15**, 777-785.

14. Bechhofer, R.E., Santner, T.J., Goldsman, D.M. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons.* John Wiley & Sons, New York, NY.

15. Bekki, J.M., Fowler, J.W., Mackulak, G.T., Nelson, B.L. (2007). Using Quantiles in Ranking and Selection Procedures. *Proceedings of the 2007 Winter Simulation Conference*, 1722-1728.

16. Cao, Y.J. and Wu, Q.H. (1999). Optimization of Control Parameters in Genetic Algorithms: A Stochastic Approach. *International Journal of Systems Science,* **30**(2), 551-559.

17. Chiba, K., Obayashi, S. and Nakahashi, K. (2006). Design Exploration of Aerodynamic Wing Shape for Reusable Launch Vehicle Flyback Booster. *Journal of Aircraft*, **43**(3), 832-836.

18. Chick, S.E., Inoue, K. (2001). New Two-Stage and Sequential Procedures for Selecting the Best Simulated System. *Operations Research,* **35**, 732-743.

19. Coit, D.W., Jin, T. and Wattanapongsakom, N. (2004). System Optimization with Component Reliability Estimation Uncertainty: A Multi-Criteria Approach. *IEEE Transactions on Reliability*, **53**(3).

20. Collette, Y. and Siarry, P. (2004). *Multiobjective Optimization Principles and Case Studies*. Springer-Verlag, Berlin, Germany.

21. Das, I. and Dennis, J.E. (1997). A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Structural Optimization*, **14**, 63-69.

22. Das, I. and Dennis, J.E. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, **8**(3), 631-657.

23. Davis, M. (2009). *Using Multiple Robust Parameter Design Techniques to Improve Hyperspectral Anomaly Detection Algorithm Performance*. Master's Thesis. Air Force Institute of Technology.

24. Dulikravich, G.S., Egorov, I.N., Sikka, V.K. and Muralidharan, G. (2003). Semi-Stochastic Multi-Objective Optimization of Chemical Composition of High Temperature Austenitic Steels for Desired Mechanical Properties. In *Proceedings of the Yazawa International Symposium, Metallurgical and Materials Processing: Principles and Technologies, Volume I: Materials Processing Fundamentals and New Technologies*.

25. Dulikravich, G.S., Sikka, V.K., Muralidharan, G. (2006). Development of Semi-Stochastic Algorithm for Optimizing Alloy Composition of High Temperature Austenitic Stainless Steels (H-Series) for Desired Mechanical and Corrosion Properties. *Final Technical Report, U.S. Department of Energy Industrial Technologies Program*.

26. Dulikravich, G.S., Egorov, I.N., Colaco, M.J. (2008). Optimizing Chemistry of Bulk Metallic Glasses for Improved Thermal Stability. *Modelling Simulation Material Science Engineering*, **8**.

27. Egorov, I.N. and Dulikravich, G.S. (2004). Optimization of Alloy Chemistry for Maximum Stress and Time-to-Rupture at High Temperature. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*.

28. Ehrgott, M. (2005). *Multicriteria Optimization*. Springer, Berlin, Germany.

29. Ehrgott, M. and Tenfelde-Podehl, D. (2003). Computation of Ideal and Nadir values and Implications for Their Use in MCDM methods. *European Journal of Operational Research*, **151**, 119-139.

30. Fonseca, C.M. and Fleming, P.J. (1996). On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. *Lecture Notes in Computer Science, 4th Conference on Parallel Problem Solving from Nature*.

31. Glover, F. (1989). Tabu Search – Part I. *ORSA Journal on Computing*, **1**, 190-206.

32. Glover, F. (1989). Tabu Search – Part II. *ORSA Journal on Computing*, **2**, 4-32.

33. Johnson, R. (2008). *Improved Feature Extraction, Feature Selection, and Identification Techniques that Create a Fast Unsupervised Hyperspectral Target Detection Algorithm*. Master's Thesis. Air Force Institute of Technology.

34. Kim, Y.I. and de Weck, O.L. (2004). Adaptive Weighted Sum Method for Multiobjective Optimization. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*.

35. Kulturel-Konak, S., Smith, A.E. and Norman, B.A. (2006). Multi-objective Tabu Search Using a Multinomial Probability Mass Function. *European Journal of Operational Research*, **169**, 918-931.

36. Kurapti, A. and Azarm, S. (2000). Immune Network Simulation with Mutliobjective Genetic Algorithms for Multidisciplinary Design Optimization. *Engineering Optimization*, **33**, 245-260.

37. Lewis, R.M. and Torczon, V. (2002). A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds. *SIAM Journal on Optimization*, **12**(4), 1075-1089.

38. Lian, Y. and Liou, M. (2005). Multiobjective Optimization Using Coupled Response Surface Model and Evolutionary Algorithm. *AIAA Journal*, **43**(6), 1316-1325.

39. Lim, D., Ong, Y.S. and Lee, B.S. (2005). Inverse Multi-Objective Robust Evolutionary Design Optimization in the Presence of Uncertainty. *GECCO '05*, 25-29.

40. Luc, D.T., Phong, T.Q. and Volle, M. (2005). Scalarizing Functions for Generating the Weakly Efficient Solution Set in Convex Multiobjective Problems. *SIAM Journal on Optimization*, **15**(4), 987-1001.

41. Marseguerra, M., Zio, E., Podofillini, L. (2004). Optimal Reliability/Availability of Uncertain Systems via Multi-Objective Genetic Algorithms. *IEEE Transactions on Reliability*, **53**(3).

42. Marti, K. (2005). *Stochastic Optimization Methods*. Springer, Berlin, Germany.

43. Mattson, C.A., Mullur, A.A. and Messac, A. (2002). Minimal Representation of Multiobjective Design Space Using a Smart Pareto Filter. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*.

44. Mattson, C.A. and Messac, A. (2005). Pareto Frontier Based Concept Selection under Uncertainty, with Visualization. *Optimization and Engineering*, **6**, 85-115.

45. Paciencia, T.J. (2008). *Multi-Objective Optimization of Mixed Variable, Stochastic Systems Using Single-Objective Formulations*. Master's Thesis. Air Force Institute of Technology.

46. Pitchitlamken, J., Nelson, B.L., Hong, L.J. (2006). A Sequential Procedure for Neighborhood Selection-of-the-Best in Optimization via Simulation. *European Journal of Operational Research*, **173**, 283-298.

47. Rahman, M.K. (2006). An Intelligent Moving Object Optimization Algorithm for Design Problems with Mixed Variables, Mixed Constraints and Multiple Objectives. *Structural Multidisciplinary Optimization*, **32**, 40-58.

48. Rao, S.S. and Xiong, Y. (2005). A Hybrid Genetic Algorithm for Mixed-Discrete Design Optimization. *Transactions of the ASME*, **127**, 1100-1112.

49. Royset, J.O. and Polak, E. (2007). Extensions of Stochastic Optimization Results to Problems with System Failure Probability Functions. *Journal of Optimization Theory and Applications*, **133**(1), 1-18.

50. Shukla, P.K. and Deb, K. (2007). On Finding Multiple Pareto-Optimal Solutions Using Classical and Evolutionary Generating Methods. *European Journal of Operational Research*, **181**, 1630-1652.

51. Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc., Hoboken, NJ.

52. Sriver, T.A., Chrissis, J.W. and Abramson, M.A. (2008). Pattern Search Ranking and Selection Algorithms for Mixed Variable Simulated-Based Optimization. *European Journal of Operational Research*, doi:10.1016/j.ejor.2008.10.020 **(IN PRESS)**.

53. Taboada, H.A., Espiritu, J.F. and Coit, D.W. (2008). MOMS-GA: A Multi-Objective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems. *IEEE Transactions on Reliability*, **57**(1).

54. Teleb, R. and Azadivar, F. (1994). A Methodology for Solving Multi-Objective Simulation-Optimization Problems. *European Journal of Operational Research*, **72**, 135-145.

55. Torczon, V. and Trosset, M. (1997). On the Convergence of Pattern Search Algorithms," *SIAM Journal on Optimization*, **7**(1), 1-25.

56. Walston, J. (1999). *Search Techniques for Multi-Objective Optimization of Mixed-Variable Systems Having Stochastic Responses*. PhD Dissertation. Air Force Institute of Technology.

57. Wu, J. and Azarm, S. (2001). Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set. *Transactions of the ASME*, **123**(1), 18-25.

58. Zabinsky, Z. (2003). *Stochastic Adaptive Search for Global Optimization*. Kluwer Academic Publishers, Boston, MA.

59. Zafiropoulos, E.P. and Dialynas, E.N. (2007). Methodology for the Optimal Component Selection of Electronic Devices under Reliability Cost Constraints. *Quality and Reliability Engineering International*, **23**, 885-897.

60. Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transaction on Evolutionary Computation*, **3**(4).

61. Zitzler, E., Deb, K. and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, **8**(3), 173-195.

62. Zitzler, E., Laumanns, M. and Bleuler, S. (2004). A Tutorial on Evolutionary Multiobjective Optimization. *Proceedings of the Workshop on Multiple Objective Metaheuristics*.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
26-03-2009	**Master's Thesis**	Jun 2008 – Mar 2009

4. TITLE AND SUBTITLE

ADAPTIVE PARETO SET ESTIMATION FOR STOCHASTIC MIXED VARIABLE DESIGN PROBLEMS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Arendt, Christopher, D., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Street, Building 642
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENS/09-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Many design problems require the optimization of competing objective functions that may be too complicated to solve analytically. These problems are often modeled in a simulation environment where static input may result in dynamic (stochastic) responses to the various objective functions. System reliability, alloy composition, algorithm parameter selection, and structural design optimization are classes of problems that often exhibit such complex and stochastic properties. Since the physical testing and experimentation of new designs can be prohibitively expensive, engineers need adequate predictions concerning the viability of various designs in order to minimize wasteful testing. Presumably, an appropriate stochastic multi-objective optimizer can be used to eliminate inefficient designs through the analysis of simulated responses. This research develops an adaptation of Walston's Stochastic Multi-Objective Mesh Adaptive Direct Search (SMOMADS) and Paciencia's NMADS based on Kim and de Weck's Adaptive Weighted Sum (AWS) procedure and standard distance to a reference point methods. The main contribution of this paper is a new implementation of MADS for Mixed Variable and Stochastic design problems that drastically reduces dependence on subjective decision maker interaction.

15. SUBJECT TERMS

Mutli-objective Optimization, Stochastic Optimization, Mixed Variable Optimization, Pareto front, Pareto set, Adaptive Constraint

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** James W. Chrissis, Civ, USAF (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	138	(937) 255-3636, ext 4606; e-mail: james.chrissis@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18